MPLS EM—MPLS LSP Multipath Tree Trace

First Published: December 4, 2006
Last Updated: February 19, 2007

The MPLS EM—MPLS LSP Multipath Tree Trace feature provides the means to discover all possible equal-cost multipath (ECMP) routing paths of a label switched path (LSP) between an egress and ingress router. Once discovered, these paths can be retested on a periodic basis using Multiprotocol Label Switching (MPLS) LSP ping or traceroute. This feature is an extension to the MPLS LSP traceroute functionality for the tracing of IPv4 LSPs.

You can use the MPLS Embedded Management (MPLS EM)—MPLS LSP Multipath Tree Trace feature to discover all paths for an IPv4 LSP.

This implementation of the MPLS EM—MPLS LSP Multipath Tree Trace feature is based on RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*.

For information on the use of MPLS LSP ping and traceroute, see the *MPLS Embedded Management—LSP Ping for LDP* feature module.

Cisco IOS MPLS Embedded Management (EM) is a set of standards and value-added services that facilitate the deployment, operation, administration, and management of MPLS-based networks in line with the fault, configuration, accounting, performance, and security (FCAPS) model.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for MPLS EM—MPLS LSP Multipath Tree Trace" section on page 50.

Finding Support Information for Platforms and Cisco IOS Software Images

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.

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

Contents

Prerequisites for MPLS EM—MPLS LSP Multipath Tree Trace

The following are prerequisites for using the MPLS EM—MPLS LSP Multipath Tree Trace feature:

- You must understand the concepts and know how to use MPLS LSP ping or traceroute as described in the *MPLS Embedded Management—LSP Ping for LDP* document.
- The routers in your network must be using an implementation based on RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*.
- You should know the following about your MPLS network:
  - The topology
  - The number of links in your network
  - The expected number of LSPs, and how many LSPs
- Understand label switching, forwarding, and load balancing.

Restrictions for MPLS EM—MPLS LSP Multipath Tree Trace

- All restrictions that apply to the MPLS Embedded Management—MPLS LSP Ping and LSP Traceroute feature also apply to the MPLS EM—MPLS LSP Multipath Tree Trace feature.
- Multiple LSP paths are not discovered unless all routers in the MPLS core support an RFC 4379 implementation of *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures.*
- MPLS LSP multipath tree trace is not expected to operate in networks that support time-to-live (TTL) hiding.

Information About MPLS EM—MPLS LSP Multipath Tree Trace

Before using the MPLS EM—MPLS LSP Multipath Tree Trace feature, you need an understanding of the following concepts:

Overview of MPLS LSP Multipath Tree Trace

As the number of MPLS deployments increases, the number of traffic types the MPLS networks carry could increase. In addition, load balancing on label switch routers (LSRs) in the MPLS network provides alternate paths for carrying MPLS traffic to a target router. The ability of service providers to monitor LSPs and quickly isolate MPLS forwarding problems is critical to their ability to offer services.

Prior to the release of the MPLS EM—MPLS LSP Multipath Tree Trace feature no automated way existed to discover all paths between provider edge (PE) routers. Troubleshooting forwarding problems between PEs was cumbersome.

The release of the MPLS EM—MPLS LSP Multipath Tree Trace feature provides an automated way to discover all paths from the ingress PE router to the egress PE router in multivendor networks that use IPv4 load balancing at the transit routers. Once the PE-to-PE paths are discovered, use MPLS LSP ping and MPLS LSP traceroute to periodically test them.

The MPLS EM—MPLS LSP Multipath Tree Trace feature requires the Cisco RFC-compliant implementation which is based on RFC 4379. If you do not have a Cisco IOS release that supports RFC 4379, MPLS LSP multipath tree trace does not operate to discover all PE-to-PE paths.

Discovery of IPv4 Load Balancing Paths by MPLS LSP Multipath Tree Trace

IPv4 load balancing at a transit router is based on the incoming label stack and the source and destination addresses in the IP header. The outgoing label stack and IP header source address remain constant for each branch being traced.

When you execute MPLS LSP multipath tree trace on the source LSR, the router needs to find the set of IP header destination addresses to use all possible output paths. The source LSR starts path discovery by sending a transit router a bitmap in an MPLS echo request. The transit router returns information in an MPLS echo request that contains subsets of the bitmap in a downstream map (DS Map) in an echo reply. The source router can then use the information in the echo reply to interrogate the next router. The source router interrogates each successive router until it finds one bitmap setting that is common to all routers along the path. The router uses TTL expiry to interrogate the routers to find the common bits.

For example, you could start path discovery by entering the following command at the source router:

```
Source_LSR# trace mpls multipath ipv4 10.131.101.129/32 hashkey ipv4 bitmap 16
```

This command sets the IP address of the target router as 10/131.101.192 255.255.255.255 and configures:

- The default hash key type to 8, which requests that an IPv4 address prefix and bit mask address set be returned in the DS Map in the echo reply.

- The bitmap size to 16. This means that MPLS LSP multipath tree trace uses 16 addresses (starting with 127.0.0.1) in the discovery of all paths of an LSP between the source router and the target router.

If you entered the **trace mpls multipath ipv4 10.131.101.129/32** command, MPLS LSP multipath tree trace uses the default hash type of 8 or IPv4 and a default bitmap size of 32. Your choice of a bitmap size depends on the number of routes in your network. If you have a large number of routes, you might need to use a larger bitmap size.

Echo Reply Return Codes Sent by the Router Processing Multipath LSP Tree Trace

Table 1 describes the characters that the router processing a multipath LSP tree trace packet returns to the sender about the failure or success of the request.

Table 1 Echo Reply Return Codes

Output Code	Echo Return Code	Meaning
Period "."	—	A timeout occurred before the target router can reply.
x	0	No return code.
M	1	Malformed request.
m	2	Unsupported type, length, values (TLVs).
!	3	Success.
F	4	No Forwarding Equivalence Class (FEC) mapping.
D	5	DS Map mismatch.
R	6	Downstream router but not target.
U	7	Reserved.
L	8	Labeled output interface.
B	9	Unlabeled output interface.
f	10	FEC mismatch.
N	11	No label entry.
P	12	No receive interface label protocol.
p	13	Premature termination of the LSP.
X	unknown	Undefined return code.

How to Configure MPLS EM—MPLS LSP Multipath Tree Trace

This section contains the following tasks:

- Customizing the Default Behavior of MPLS Echo Packets, page 5 (optional)
- Configuring MPLS LSP Multipath Tree Trace, page 7 (required)
- Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace, page 9 (required)

- Monitoring LSP Paths Discovered by MPLS LSP Multipath Tree Trace Using MPLS LSP Traceroute, page 10 (optional)
- Using DSCP to Request a Specific Class of Service in an Echo Reply, page 13 (optional)
- Controlling How a Responding Router Replies to an MPLS Echo Request, page 14 (optional)
- Specifying the Output Interface for Echo Packets Leaving a Router for MPLS LSP Multipath Tree Trace, page 16 (optional)
- Setting the Pace of MPLS Echo Request Packet Transmission for MPLS LSP Multipath Tree Trace, page 17 (optional)
- Enabling MPLS LSP Multipath Tree Trace to Detect LSP Breakages Caused by an Interface that Lacks an MPLS Configuration, page 18 (optional)
- Requesting that a Transit Router Validate the Target FEC Stack for MPLS LSP Multipath Tree Trace, page 19 (optional)
- Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace, page 21 (optional)

Customizing the Default Behavior of MPLS Echo Packets

Perform the following task to customize the default behavior of MPLS echo packets. You might need to customize the default echo packet encoding and decoding behavior to allow later implementations of the *Detecting MPLS Data Plane Failures* (RFC 4379) to be deployed in networks running earlier versions of the draft.

MPLS Embedded Management Configuration

Before using the **ping mpls**, **trace mpls**, or **trace mpls multipath** command, you should consider ensuring that the router is configured to encode and decode MPLS echo packets in a format that all receiving routers in the network can understand.

LSP ping drafts after Version 3 (draft-ietf-mpls-ping-03) have undergone numerous TLV format changes, but the implementations based on different drafts might not interoperate properly.

To allow later Cisco implementations to interoperate with draft Version 3 Cisco and non-Cisco implementations, a global configuration mode (MPLS OAM configuration) allows you to encode and decode echo packets in formats specified by draft Version 3 implementations.

Unless configured otherwise, a Cisco implementation encodes and decodes echo requests assuming the version on which the Internet Engineering Task Force (IETF) implementation is based.

To allow for seamless interoperability with earlier Revision 1 and 3 images, you can use MPLS Operation, Administration, and Maintenance (OAM) configuration mode parameters to force the default behavior of the Revision 4 images to be compliant or compatible in networks with Revision 1 or Revision 3 images.

To prevent failures reported by the replying router due to TLV version issues, you should configure all routers in the core. Encode and decode MPLS echo packets in the same draft version. For example, if the network is running RFC 4379 (Cisco Revision 4) implementations but one router is capable of only Version 3 (Cisco Revision 3), configure all routers in the network to operate in Revision 3 mode.

Cisco Revision 4 is the default version. The default version is the latest LSP Ping version supported by the image on the router.

Prerequisites

MPLS LSP Multipath Tree Trace requires RFC 4379 (Revision 4).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls oam**
4. **echo revision** {**3** | **4**}
5. [**no**] **echo vendor-extension**
6. **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	**mpls oam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode and customizes the default behavior of echo packets.
Step 4	**echo revision** {**3** | **4**} **Example:** `Router(config-mpls)# echo revision 4`	Customizes the default behavior of echo packets. • The **revision** keyword set echo packet attributes to one of the following: – **3** = draft-ietf-mpls-ping-03 (Revision 2) – **4** = RFC 4379 compliant (default) **Note** The MPLS LSP Multipath Tree Trace feature requires Revision 4.

	Command or Action	Purpose

Step 5	`[no] echo vendor-extension`
	
	Example:
	`Router(config-mpls)# echo vendor-extension`
		Customizes the default behavior of echo packets.
		• The **vendor-extension** keyword sends the Cisco-specific extension of TLVs with the echo packets.
		• The **no** form of the command allows you to disable a Cisco vendor's extension TLVs that another vendor's noncompliant implementations may not support.
		The router default is **echo vendor-extension**.

Step 6	`end`
	
	Example:
	`Router(config-mpls)# end`
		Exits to privileged EXEC mode.

Configuring MPLS LSP Multipath Tree Trace

Perform the following task to configure MPLS multipath LSP traceroute. This task helps discover all LSPs from an egress router to an ingress router.

Prerequisites

Cisco LSP ping or traceroute implementations based on draft-ietf-mpls-lsp-ping-11 are capable in some cases of detecting the formatting of the sender of an MPLS echo request. However, certain cases exist in which an echo request or echo reply might not contain the Cisco extension TLV. To avoid complications due to certain cases where the echo packets are decoded assuming the wrong TLV formats, configure all routers in the network to operate in the same mode.

For an MPLS LSP multipath tree trace to be successful, the implementation in your routers must support RFC 4379 on all core routers.

If all routers in the network support FRC-4379 and another vendor's implementation exists that is not capable of properly handling Cisco's vendor TLV, the routers supporting the RFC-compliant or later configuration must include commands to disable the Cisco vendor TLV extensions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls oam**
4. **echo revision 4**
5. [**no**] **echo vendor-extension**
6. **end**
7. **trace mpls multipath ipv4** *destination-ip-address/destination-mask-length*
8. **debug mpls lspv multipath**

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	**mpls oam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode.
Step 4	**echo revision 4** **Example:** `Router(config-mpls)# echo revision 4`	Customizes the default behavior of echo packets. • The **revision 4** keywords set echo packet attributes to the default Revision 4 (RFC 4379 compliant). **Note** The MPLS LSP Multipath Tree Trace feature requires Revision 4.
Step 5	[**no**] **echo vendor-extension** **Example:** `Router(config-mpls) echo vendor-extension`	(Optional) Customizes the default behavior of echo packets. • The **vendor-extension** keyword sends the Cisco-specific extension of TLVs with the echo packets. • The **no** form of the command allows you to disable a Cisco vendor's extension TLVs that another vendor's noncompliant implementations may not support. The router default is **echo vendor-extension**.
Step 6	**end** **Example:** `Router(config-mpls)# end`	Exits to privileged EXEC mode.
Step 7	**trace mpls multipath ipv4** *destination-ip-address***/***destination-mask-length* **Example:** `Router# trace mpls multipath ipv4 10.131.161.251/32`	Discovers all LSPs from an egress router to an ingress router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-ip-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required.
Step 8	**debug mpls lspv multipath** **Example:** `Router# debug mpls lspv multipath`	Displays multipath information related to the MPLS LSP Multipath Tree Trace feature.

Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace

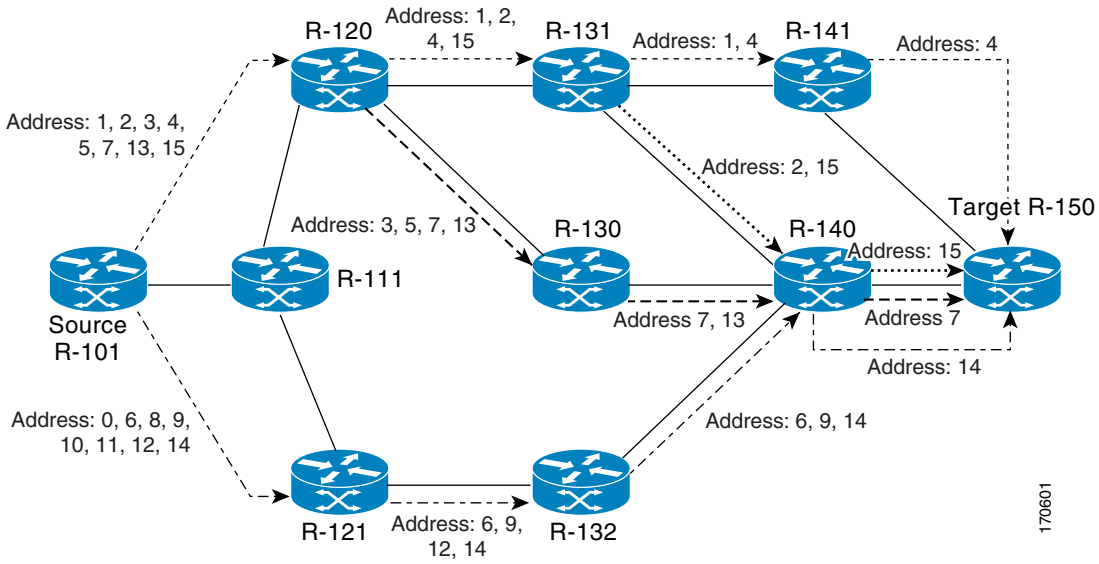
Perform the following task to discover IPv4 load balancing paths using MPLS LSP multipath tree trace.

MPLS Multipath LSP Traceroute Path Discovery

A Cisco router load balances MPLS packets based on the incoming label stack and the source and destination addresses in the IP header. The outgoing label stack and IP header source address remain constant for each path being traced. The router needs to find the set of IP header destination addresses to use all possible output paths. This might require exhaustive searching of the 127.*x.y.z*/8 address space. Once you discover all paths from the source LSR to the target or destination LSR with MPLS LSP multipath tree trace, you can use MPLS LSP traceroute to monitor these paths.

Figure 1 shows how MPLS LSP multipath tree trace discovers LSP paths in a sample network. In Figure 1, the bitmap size is 16 and the numbers 0 to 15 represent the bitmapped addresses that MPLS LSP multipath tree trace uses to discover all the paths from the source LSR R-101 to the target LSR R-150. Figure 1 illustrates how the **trace mpls multipath** command discovers all LSP paths in the sample network.

Figure 1 MPLS LSP Multipath Tree Trace Path Discovery in a Sample Network



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls oam**
4. **echo revision 4**
5. **end**
6. **trace mpls multipath ipv4** *destination-ip-address/destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*

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	**mpls oam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode and sets the echo packet attribute to Revision 4 (RFC 4379 compliant).
Step 4	**echo revision 4** **Example:** `Router(config-mpls)# echo revision 4`	Customizes the default behavior of echo packets. • The **revision 4** keywords set echo packet attributes to the default Revision 4 (RFC 4379 compliant). **Note** The MPLS LSP Multipath Tree Trace feature requires Revision 4.
Step 5	**end** **Example:** `Router(config-mpls)# end`	Exits to privileged EXEC mode.
Step 6	**trace mpls multipath ipv4** *destination-address/destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size* **Example:** `Router# trace mpls multipath ipv4 10.131.161.251/32 hashkey ipv4 bitmap 16`	Discovers all MPLS LSPs from an egress router to an ingress router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **hashkey ipv4** keywords set the hashkey type to IPv4 addresses. • The **bitmap** *bitmap-size* keyword and arguments set the bitmap size for multipath discovery.

Monitoring LSP Paths Discovered by MPLS LSP Multipath Tree Trace Using MPLS LSP Traceroute

Perform the following task to monitor LSP paths discovered by MPLS LSP multipath tree trace using MPLS LSP traceroute. You can take output directly from the **trace mpls multipath** command and add it to a **trace mpls** command periodically to verify that the path is still operating.

Figure 2 shows the mapping of the output of a **trace mpls multipath** command to a **trace mpls** command.

Figure 2 Mapping of trace mpls multipath Command Output to a trace mpls Command

Each path you discover with MPLS LSP Multipath Tree Trace can be tested in this manner periodically to monitor the LSP paths in your network.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*
3. **trace mpls ipv4** *destination-address/destination-mask-length* [**output interface** *tx-interface*] [**source** *source-address*] [**destination** *address-start*]
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

```
Router> enable
Router#
```

Step 2 **trace mpls multipath ipv4** *destination-address/destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*

Use this command to discover all MPLS LSPs from an egress router to an ingress router. For example:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 hashkey ipv4 bitmap 16

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
```

```
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 468 ms
```

The output of the **trace mpls multipath** command in the example shows the result of path discovery with MPLS LSP multipath tree trace. In this example, the command sets the bitmap size to 16. Path discovery starts by MPLS LSP multipath tree trace using 16 bitmapped addresses as it locates LSP paths from the source router to the target router with prefix and mask 10.1.1.150/32. MPLS LSP multipath tree trace starts using the 127.*x.y.z*/8 address space with 127.0.0.1.

Step 3 **trace mpls ipv4** *destination-address/destination-mask-length* [**output interface** *tx-interface*] [**source** *source-address*] [**destination** *address-start*]

Use this command to verify that the paths discovered when you entered a **trace mpls multipath** command are still operating. For example, the output for Path 0 in the previous **trace mpls multipath** command in Step 2 is:

```
output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
```

If you put the output for path 0 in the **trace mpls** command, you see the following results:

```
Router# trace mpls ipv4 10.1.1.150/32 output interface Et0/0 source 10.1.111.101
destination 127.0.0.0

Tracing MPLS Label Switched Path to 10.1.1.150/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 10.1.111.101 MRU 1500 [Labels: 33 Exp: 0]
L 1 10.1.111.111 MRU 1500 [Labels: 34 Exp: 0] 40 ms
L 2 10.2.121.121 MRU 1500 [Labels: 34 Exp: 0] 32 ms
L 3 10.3.132.132 MRU 1500 [Labels: 32 Exp: 0] 16 ms
L 4 10.4.140.240 MRU 1504 [Labels: implicit-null Exp: 0] 20 ms
! 5 10.5.150.50 20 ms
```

You can take output directly from the **trace mpls multipath** command and add it to a **trace mpls** command periodically to verify that the path is still operating (see Figure 2).

Step 4 **exit**

Use this command to exit to user EXEC mode. for example:

```
Router# exit
Router>
```

Using DSCP to Request a Specific Class of Service in an Echo Reply

For Cisco IOS Release 12.2(27)SXE, Cisco added a reply differentiated services code point (DSCP) option that lets you request a specific class of service (CoS) in an echo reply.

The reply DSCP option is supported in the experimental mode for IETF draft-ietf-mpls-lsp-ping-03.txt. Cisco implemented a vendor-specific extension for the reply DSCP option rather than using a Reply TOS TLV. A Reply TOS TLV serves the same purpose as the **reply dscp** command in IETF draft-ietf-mpls-lsp-ping-11.txt. This draft provides a standardized method of controlling the reply DSCP.

Note Before RFC 4379, Cisco implemented the Reply DSCP option as an experimental capability using a Cisco vendor extension TLV. If a router is configured to encode MPLS echo packets for draft Version 3 implementations, a Cisco vendor extension TLV is used instead of the = Reply TOS TLV that was defined in draft Version 8.

To use DSCP to request a specific CoS in an echo reply, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**reply dscp** *dscp-value*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address*/*destination-mask-length* [**reply dscp** *dscp-value*] **Example:** `Router# trace mpls multipath ipv4 10.131.191.252/32 reply dscp 50`	Discovers all MPLS LSPs from an ingress router to an egress router and controls the DSCP value of an echo reply. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **reply dscp** *dscp-value* keywords and argument are the DSCP value of an echo reply. A Reply TOS TLV serves the same purpose as the **reply dscp** command in IETF draft-ietf-mpls-lsp-ping-11.txt. **Note** To specify a DSCP value, you must enter the **reply dscp** *dscp-value* keywords and argument.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Controlling How a Responding Router Replies to an MPLS Echo Request

This section contains information about and instructions for controlling how a responding router replies to an MPLS echo request. You should understand the following information before you configure a reply mode for the echo request response:

• Reply Modes for an MPLS LSP Multipath Tree Trace Echo Request Response, page 14

Reply Modes for an MPLS LSP Multipath Tree Trace Echo Request Response

The reply mode controls how a responding router replies to an MPLS echo request sent by a **trace mpls multipath** command. There are two reply modes for an echo request packet:

- ipv4—Reply with an IPv4 User Datagram Protocol (UDP) packet (default)
- router-alert—Reply with an IPv4 UDP packet with router alert

Note Use the ipv4 and router-alert reply modes with each other to prevent false negatives. If you do not receive a reply via the ipv4 mode, send a test with the router-alert reply mode. If both fail, something is wrong in the return path. The problem might be due to an incorrect ToS setting.

IPv4 UDP Reply Mode

The IPv4 UDP reply mode is the most common reply mode used with a **trace mpls multipath** command when you want to periodically poll the integrity of an LSP. With this option, you do not have explicit control over whether the packet traverses IP or MPLS hops to reach the originator of the MPLS echo request. If the originating (headend) router fails to receive a reply to an MPLS echo request when you use the **reply mode ipv4** keywords, use the **reply mode router-alert** keywords.

Router-alert Reply Mode

The router-alert reply mode adds the router alert option to the IP header. When an IP packet that contains an IP router alert option in its IP header or an MPLS packet with a router alert label as its outermost label arrives at a router, the router punts (redirects) the packet to the Route Processor (RP) process level for handling. This forces the RP of each intermediate router to specifically handle the packet at each intermediate hop as it moves back to the destination. Hardware and line-card forwarding inconsistencies are thus bypassed. Router-alert reply mode is slower than IPv4 mode because the reply requires process-level RP handling at each hop.

Table 2 describes how an incoming IP packet with an IP router alert is handled by the router switching path processes when the outgoing packet is an IP packet or an MPLS packet. It also describes how an MPLS packet with a router alert option is handled by the router switching path processes when the outgoing packet is an IP packet or an MPLS packet.

Table 2 Path Process Handling of IP and MPLS Router Alert Packets

Incoming Packet	Outgoing Packet	Normal Switching Action	Process Switching Action
IP packet—Router alert option in IP header	IP packet—Router alert option in IP header	Router alert option in IP header causes the packet to be punted to the process switching path.	Forwards the packet as is
	MPLS packet		Forwards the packet as is
MPLS packet—Outermost label contains a router alert	IP packet—Router alert option in IP header	If the router alert label is the outermost label, it causes the packet to be punted to the process switching path.	Removes the outermost router alert label and forwards the packet as an IP packet
	MPLS packet—Outermost label contains a router alert		Preserves the outermost router alert label and forwards the MPLS packet

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* **reply mode** {**ipv4** | **router-alert**}
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address*/*destination-mask-length* **reply mode** {**ipv4** | **router-alert**} **Example:** `Router# trace mpls multipath ipv4 10.131.191.252/32 reply mode router-alert`	Discovers all MPLS LSPs from an ingress router to an egress router and specifies the reply mode. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **reply mode** keyword requires that you enter one of the following keywords to specify the reply mode: – The **ipv4** keyword—Reply with an IPv4 UDP packet (default). – The **router-alert** keyword—Reply with an IPv4 UDP packet with router alert. **Note** To specify the reply mode, you must enter the **reply mode** keyword with the **ipv4** or **router-alert** keyword.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Specifying the Output Interface for Echo Packets Leaving a Router for MPLS LSP Multipath Tree Trace

Perform the following task to specify the output interface for echo packets leaving a router for the MPLS LSP Multipath Tree Trace feature. You can use this task to test the LSPs reachable through a given interface.

Echo Request Output Interface Control

You can control the interface through which packets leave a router. Path output information is used as input to LSP ping and traceroute.

The echo request output interface control feature allows you to force echo packets through the paths that perform detailed debugging or characterizing of the LSP. This feature is useful if a PE router connects to an MPLS cloud and there are broken links. You can direct traffic through a certain link. The feature also is helpful for troubleshooting network problems.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**output interface** *tx-interface*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**output interface** *tx-interface*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.251/32 output interface ethernet0/0`	Discovers all MPLS LSPs from an ingress router to an egress router and specifies the interface through which echo packets leave a router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **output interface** *tx-interface* keywords and argument specify the output interface for the MPLS echo request. **Note** You must specify the **output interface** keywords.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Setting the Pace of MPLS Echo Request Packet Transmission for MPLS LSP Multipath Tree Trace

Perform the following task to set the pace of MPLS echo request packet transmission for the MPLS LSP Multipath Tree Trace feature. Echo request traffic pacing allows you to set the pace of the transmission of packets so that the receiving router does not drop packets. If you have a large amount of traffic on your network you might increase the size of the interval to help ensure that the receiving router does not drop packets.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**interval** *milliseconds*]

3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address*/*destination-mask-length* [**interval** *milliseconds*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.251/32 interval 100`	Discovers all MPLS LSPs from an egress router to an ingress router and sets the time in milliseconds between successive MPLS echo requests. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **interval** *milliseconds* keyword and argument set the time between successive MPLS echo requests in milliseconds. The default is 0 milliseconds. **Note** To pace the transmission of packets, you must specify the **interval** keyword.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Enabling MPLS LSP Multipath Tree Trace to Detect LSP Breakages Caused by an Interface that Lacks an MPLS Configuration

Perform the following task to enable MPLS LSP multipath tree trace to detect LSP breakages caused by an interface that lacks an MPLS configuration. If an interface is not configured for MPLS, then it cannot forward MPLS packets.

Explicit Null Label Shimming Tests LSP Ability to Carry MPLS Traffic

For an MPLS LSP multipath tree trace of LSPs carrying IPv4 FECs, you can force an explicit null label to be added to the MPLS label stack even though the label was unsolicited. This allows MPLS LSP multipath tree trace to detect LSP breakages caused by an interface that is not configured for MPLS. MPLS LSP multipath tree trace does not report that an LSP is functioning when it is unable to send MPLS traffic.

An explicit null label is added to an MPLS label stack if MPLS echo request packets are forwarded from an interface not configured for MPLS that is directly connected to the destination of the MPLS LSP multipath tree trace or if the IP TTL value for the MPLS echo request packets is set to 1.

When you enter a **trace mpls multipath** command, you are looking for all MPLS LSP paths from an egress router to an ingress router. Failure at output interfaces that are not configured for MPLS at the penultimate hop are not detected. Explicit-null shimming allows you to test an LSP's ability to carry MPLS traffic.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* **force-explicit-null**
3. **exit**

DETAILED STEP

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* **force-explicit-null** **Example:** `Router# trace mpls multipath ipv4 10.131.191.252/32 force-explicit-null`	Discovers all MPLS LSPs from an egress router to an ingress router and forces an explicit null label to be added to the MPLS label stack. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **force-explicit-null** keyword forces an explicit null label to be added to the MPLS label stack even though the label was unsolicited. **Note** You must enter the **force-explicit-null** keyword to enable MPLS LSP multipath tree trace to detect LSP breakages caused by an interface that is not configured for MPLS.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Requesting that a Transit Router Validate the Target FEC Stack for MPLS LSP Multipath Tree Trace

Perform the following task to request that a transit router validate the target FEC stack for the MPLS LSP Multipath Tree Trace feature.

An MPLS echo request tests a particular LSP. The LSP to be tested is identified by the FEC stack.

During an MPLS LSP multipath tree trace, the echo packet validation rules do not require that a transit router validate the target FEC stack TLV. A downstream map TLV containing the correct received labels must be present in the echo request for target FEC stack checking to be performed.

To request that a transit router validate the target FEC stack, set the V flag from the source router by entering the **flags fec** keywords in the **trace mpls multipath** command. The default is that echo request packets are sent with the V flag set to 0.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**flags fec**] [**ttl** *maximum-time-to-live*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**flags fec**] [**ttl** *maximum-time-to-live*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.252/32 flags fec ttl 5`	Discovers all MPLS LSPs from an egress router to an ingress router and requests validation of the target FEC stack by a transit router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **flags fec** keywords requests that target FEC stack validation be done at a transit router. • The **ttl** *maximum-time-to-live* keyword and argument pair specify a maximum hop count. **Note** For a transit router to validate the target FEC stack, you must enter the **flags fec** and **ttl** keywords.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace

Perform the following task to set the number of timeout attempts for the MPLS LSP Multipath Tree Trace feature.

A retry is attempted if an outstanding echo request times out waiting for the corresponding echo reply.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**retry-count** *retry-count-value*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**retry-count** *retry-count-value*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.252/32 retry-count 4`	Sets the number of retry attempts during an MPLS LSP multipath tree trace. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The **/** keyword before this argument is required. • The **retry-count** *retry-count-value* keyword and argument sets the number of retry attempts after a timeout occurs. A rertry-count value of “0” means infinite retries. A retry-count value from 0 to10 is suggested. You might want to increase the retry value to greater than 10, if 10 is too small a value. The default retry-count value is 3. **Note** To set the number of retries after a timeout, you must enter the **retry-count** keyword.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Configuration Examples for MPLS EM—MPLS LSP Multipath Tree Trace

This section includes the following configuration examples for the MPLS EM—MPLS LSP Multipath Tree Trace feature:

Customizing the Default Behavior of MPLS Echo Packets: Example

The following example shows how to customize the behavior of MPLS echo packets so that the MPLS LSP Multipath Tree Trace feature interoperates with a vendor implementation that does not interpret RFC 4379 as Cisco does:

```
configure terminal
!
mpls oam
 echo revision 4
 no echo vendor-extension
 end
```

The **echo revision** command is included in this example for completeness. The default echo revision number is 4, which corresponds to RFC 4379.

Configuring MPLS LSP Multipath Tree Trace: Example

The following example shows how to configure the MPLS LSP Multipath Tree Trace feature to interoperate with a vendor implementation that does not interpret RFC 4379 as Cisco does:

```
configure terminal
!
mpls oam
 echo revision 4
 no echo vendor-extension
 end
!
trace mpls multipath ipv4 10.131.161.151/32
```

The **echo revision** command is included in this example for completeness. The default echo revision number is 4, which corresponds to the RFC 4379.

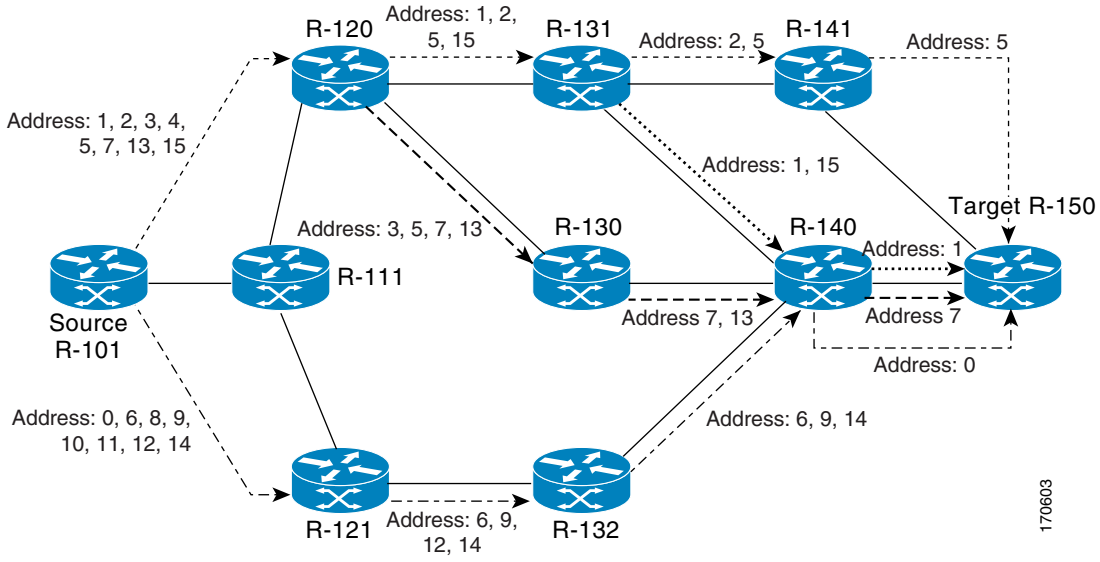
Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace: Example

The following example shows how to use the MPLS LSP Multipath Tree Trace feature to discover IPv4 load balancing paths. The example is based on the sample network shown in Figure 3. In this example, the bitmap size is set to 16. Therefore, path discovery starts by the MPLS LSP Multipath Tree Trace feature using 16 bitmapped addresses as it locates LSP paths from the source router R-101 to the target router R-150 with prefix and mask 10.1.1.150/32. The MPLS LSP Multipath Tree Trace feature starts using the 127.*x.y.z*/8 address space with 127.0.0.0.

```
Router# trace mpls multipath ipv4 10.1.1.150/32 hashkey ipv4 bitmap 16

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 468 ms
```

The output of the **trace mpls multipath** command in the example shows the result of path discovery with the MPLS LSP Multipath Tree Trace feature as shown in Figure 3.

Figure 3 MPLS LSP Multipath Tree Trace Path Discovery in a Sample Network



Using DSCP to Request a Specific Class of Service in an Echo Reply: Example

The following example shows how to use DSCP to request a specific CoS in an echo reply:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 reply dscp 50

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 448 ms
```

Controlling How a Responding Router Replies to an MPLS Echo Request: Example

The following example shows how to control how a responding router replies to an MPLS echo request:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 reply mode router-alert

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0
 Total Time Elapsed 708 ms
```

Specifying the Output Interface for Echo Packets Leaving a Router for MPLS LSP Multipath Tree Trace: Example

The following example shows how to specify the output interface for echo packets leaving a router for the MPLS LSP Multipath Tree Trace feature:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 output interface ethernet0/0

Tracing MPLS Label Switched Path to 10.1.1.150/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 10.1.111.101 MRU 1500 [Labels: 33 Exp: 0]
L
  1 10.1.111.111 MRU 1500 [Labels: 33 Exp: 0] 40 ms
L
```

```
  2 10.2.120.120 MRU 1500 [Labels: 33 Exp: 0] 20 ms
L
  3 10.3.131.131 MRU 1500 [Labels: 34 Exp: 0] 20 ms
L
  4 10.4.141.141 MRU 1504 [Labels: implicit-null Exp: 0] 20 ms !
  5 10.5.150.150 16 ms
```

Setting the Pace of MPLS Echo Request Packet Transmission for MPLS LSP Multipath Tree Trace: Example

The following examples show how set the pace of MPLS echo request packet transmission for the MPLS LSP Multipath Tree Trace feature. The time between successive MPLS echo requests is set to 300 milliseconds in the first example and 400 milliseconds in the second example:

```
Router# trace mpls multipath ipv4 10.131.159.252/32 interval 300

Starting LSP Multipath Traceroute for 10.131.159.252/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LL!
Path 0 found,
 output interface Et1/0 source 10.2.3.2 destination 127.0.0.0

Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (3/0)
 Echo Reply (received/timeout) (3/0)
 Total Time Elapsed 1604 ms

Router# trace mpls multipath ipv4 10.131.159.252/32 interval 400

Starting LSP Multipath Traceroute for 10.131.159.252/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LL!
Path 0 found,
 output interface Et1/0 source 10.2.3.2 destination 127.0.0.0

Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (3/0)
 Echo Reply (received/timeout) (3/0)
 Total Time Elapsed 1856 ms
```

Notice that the elapsed time increases as you increase the interval size.

Enabling MPLS LSP Multipath Tree Trace to Detect LSP Breakages Caused by an Interface that Lacks an MPLS Configuration: Example

The following examples shows how to enable the MPLS LSP Multipath Tree Trace feature to detect LSP breakages caused by an interface that lacks an MPLS configuration:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 force-explicit-null

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 460 ms
```

This example shows the additional information provided when you add the **verbose** keyword to the command:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 force-explicit-null verbose

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
 'L' - labeled output interface, 'B' - unlabeled output interface,
 'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
 'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
 'P' - no rx intf label prot, 'p' - premature termination of LSP,
 'R' - transit router, 'I' - unknown upstream index,
 'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
  1 10.1.111.111 10.2.121.121 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
```

```
  2 10.2.121.121 10.3.132.132 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 1
L
  3 10.3.132.132 10.4.140.240 MRU 1500 [Labels: 32/explicit-null Exp: 0/0] ret code 8
multipaths 1
L
  4 10.4.140.240 10.5.150.50 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
1 !
  5 10.5.150.50, ret code 3 multipaths 0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
  1 10.1.111.111 10.2.120.120 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  2 10.2.120.120 10.3.131.131 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  3 10.3.131.131 10.4.141.141 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  4 10.4.141.141 10.5.150.150 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8
multipaths 1
!
5 10.5.150.150, ret code 3 multipaths 0
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
 1 10.1.111.111 10.2.120.120 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
 2 10.2.120.120 10.3.131.131 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
 3 10.3.131.131 10.4.140.140 MRU 1500 [Labels: 32/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
 4 10.4.140.140 10.5.150.50 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
1 ! 5 10.5.150.50, ret code 3 multipaths 0
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
  1 10.1.111.111 10.2.120.120 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  2 10.2.120.120 10.3.130.130 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  3 10.3.130.130 10.4.140.40 MRU 1500 [Labels: 32/explicit-null Exp: 0/0] ret code 8
multipaths 1
L
  4 10.4.140.40 10.5.150.50 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
1
!
  5 10.5.150.50, ret code 3 multipaths 0

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
```

```
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 492 ms
```

Requesting that a Transit Router Validate the Target FEC Stack for MPLS LSP Multipath Tree Trace: Example

The following example shows how to request that a transit router validate the target FEC stack for the MPLS LSP Multipath Tree Trace feature:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 flags fec ttl 5

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 464 ms
```

Target FEC stack validation is always done at the egress router when the **flags fec** keywords are specified in the **trace mpls multipath** command.

Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace: Example

The following example sets the number of timeout attempts for the MPLS LSP Multipath Tree Trace feature to four:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 retry-count 4

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
```

```
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 460 ms
```

The following output shows a **trace mpls multipath** command that found one unexplored path, one successful path, and one broken path:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 retry-count 4

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLL....
Path 0 Unexplorable,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1 B
Path 2 Broken,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7

Paths (found/broken/unexplored) (1/1/1)
 Echo Request (sent/fail) (12/0)
 Echo Reply (received/timeout) (8/4)
 Total Time Elapsed 7868 ms
```

Additional References

The following sections provide references related to the MPLS EM—MPLS LSP Multipath Tree Trace feature.

Related Documents

Related Topic	Document Title
Concepts and configuration tasks for MPLS LSP ping or traceroute	*MPLS Embedded Management—LSP Ping for LDP*, Cisco IOS 12.4(6)T feature module
Concepts and configuration for MPLS and other MPLS applications	*Cisco IOS Multiprotocol Label Switching Configuration Guide,* Release 12.4
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference,* Release 12.4
Troubleshooting procedures for MPLS	*Cisco—MPLS Troubleshooting*

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
RFC 2113	*IP Router Alert Option*
RFC 3443	*Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*
RFC 4377	*Operations and Management (OAM) Requirements for Multi-Protocol Label Switched (MPLS) Networks*
RFC 4378	*A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)*
RFC 4379	*Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*

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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Command Reference

This section documents new and modified commands only.

- **debug mpls lspv**
- **echo**
- **mpls oam**
- **trace mpls**
- **trace mpls multipath**

debug mpls lspv

To display information related to the MPLS LSP Ping/Traceroute feature, use the **debug mpls lspv** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lspv [**tlv**] [**error**] [**event**] [**ipc**] [**packet** [**data** | **error**]] [**path-discovery**] [**multipath**] [**all**]

no debug mpls lspv

Syntax Description

tlv	(Optional) Displays Multiprotocol Label Switching (MPLS) echo packet type, length, values (TLVs) information as it is being coded and decoded.
error	(Optional) Displays error conditions encountered during MPLS echo request and echo reply encoding and decoding. See Table 3.
event	(Optional) Displays MPLS echo request and reply send and receive event information.
ipc	(Optional) Interprocess communication. Displays debug information regarding communication between the Route Processor and line cards.
packet data	(Optional) Displays detailed debug information for the MPLS echo packets sent and received. This output is seen only on the originating router and the router generating the reply.
packet error	(Optional) Displays packet errors for MPLS echo request and reply. No output is expected for this command.
path-discovery	(Optional) Provides information regarding LSP traceroute path discovery operations.
multipath	(Optional) Displays multipath information.
all	(Optional) Enables all the command keywords.

Command Default

MPLS LSP debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(27)S	This command was introduced.
12.4(6)T	The following keywords were added: **ipc**, **path-discovery**, **multipath**, and **all**.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.28(SB) and implemented on the Cisco 10000 series router.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

Release	Modification
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

Use this command to monitor activity associated with the **ping mpls** and the **trace mpls** commands.

Table 3 lists the messages displayed by the **debug mpls lspv error** command and the reason for each error message.

Table 3 Messages Displayed by the debug mpls lspv error Command

Message	Reason Why Message Is Displayed
Echo reply discarded because not routable	An echo reply message is sent because the IP header indicates that the packet has the Router Alert set and the packet is not routable.
UDP checksum error, packet discarded	A packet is received on the port being used by Label Switched Path Verification (LSPV) and there is a checksum error on the packet.
Invalid echo message type	An MPLS echo packet with an invalid echo message type (neither a request nor a reply) is received.
Illegal Action	The state machine that drives the LSPV software detects an invalid condition.

Examples

The following is sample output from the **ping mpls** command when LSPV event debugging is enabled:

```
Router# debug mpls lspv event

LSPV event debugging is on

Router# ping mpls ipv4 10.131.159.252/32 repeat 1

Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:


Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target


Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 48/48/48 ms
Router#
*Dec 31 19:31:15.366: LSPV:
waiting for 2 seconds
*Dec 31 19:31:15.366: LSPV: sender_handle: 2000002D, Event Echo Requests Start,
[Idle->Waiting for Echo Reply]
*Dec 31 19:31:15.414: LSPV: sender_handle: 2000002D, Event Echo Reply Received,
[Waiting for Echo Reply->Waiting for Interval]
*Dec 31 19:31:15.466: LSPV: sender_handle: 2000002D, Event Echo Requests Cancel,
[Waiting for Interval->Idle]

Router# undebug all
```

```
All possible debugging has been turned off
```

The following is sample output from the **ping mpls** command when LSPV TLV debugging is enabled:

```
Router# debug mpls lspv tlv

LSPV tlv debugging is on

Router# ping mpls ipv4 10.131.159.252/32 repeat 1

Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:


Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target

Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
Router#
*Dec 31 19:32:32.566: LSPV: Echo Hdr encode: version 1, msg type 1, reply mode 2
, return_code 0, return_subcode 0, sender handle 9400002E, sequence number 1,
timestamp sent 14:32:32 EST Wed Dec 31 2003, timestamp rcvd 19:00:00 EST Thu Dec 31 1899
*Dec 31 19:32:32.566: LSPV: IPV4 FEC encode: destaddr 10.131.159.252/32
*Dec 31 19:32:32.566: LSPV: Pad TLV encode: type 1, size 18, pattern 0xABCD
*Dec 31 19:32:32.606: LSPV: Echo Hdr decode: version 1, msg type 2, reply mode 2,
return_code 3, return_subcode 0, sender handle 9400002E, sequence number 1,
timestamp sent 14:32:32 EST Wed Dec 31 2003, timestamp rcvd 14:32:32 EST Wed Dec 31 2003

Router# undebug all

All possible debugging has been turned off
```

The following is sample output from the **trace mpls multipath** command when LSPV multipath debugging is on:

```
Router# debug mpls lspv multipath

multipath information debugging is on

Router# trace mpls multipath ipv4 10.5.5.5/32

Starting LSP Multipath Traceroute for 10.5.5.5/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LL
*Aug 30 20:39:03.719: LSPV: configuring bitmask multipath, base 0x7F000000, bitmapsize 32,
start 0x7F000000, numbits 32
*Aug 30 20:39:03.719: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
8, start 127.0.0.0, base 127.0.0.0, numbits 32
*Aug 30 20:39:03.719: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
8, start 127.0.0.0, base 127.0.0.0, numbits 32
*Aug 30 20:39:03.719: LSPV: getnext bit_cursor 0, index 0, mask 0x80000000
```

```
*Aug 30 20:39:03.719: LSPV: next addr 127.0.0.0
*Aug 30 20:39:03.719: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:03.719: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.007: LSPV: multipath info: !
Path 0 found,
 output interface Et1/0 source 10.2.3.2 destination 127.0.0.0

Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (3/0)
 Echo Reply (received/timeout) (3/0)
 Total Time Elapsed 924 ms
Router#
*Aug 30 20:39:04.007: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.007: LSPV: ds map convert: rtr_id A030404, mtu 1500 intf_addr 10.3.4.4
hashkey 8, multipath length 8, info 2130706432
*Aug 30 20:39:04.007: LSPV: multipath info: hashkey type 8, base 0x7F000000, bitmapsize
32, info0 0xFFFFFFFF
*Aug 30 20:39:04.007: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
8, start 127.0.0.0, base 127.0.0.0, numbits 32
*Aug 30 20:39:04.007: LSPV: getnext bit_cursor 0, index 0, mask 0x80000000
*Aug 30 20:39:04.007: LSPV: next addr 127.0.0.0
*Aug 30 20:39:04.007: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:04.007: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.299: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:04.299: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.299: LSPV: ds map convert: rtr_id A040505, mtu 1504 intf_addr 10.4.5.5
hashkey 8, multipath length 8, info 2130706432
*Aug 30 20:39:04.299: LSPV: multipath info: hashkey type 8, base 0x7F000000, bitmapsize
32, info0 0xFFFFFFFF
*Aug 30 20:39:04.299: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
8, start 127.0.0.0, base 127.0.0.0, numbits 32
*Aug 30 20:39:04.299: LSPV: getnext bit_cursor 0, index 0, mask 0x80000000
*Aug 30 20:39:04.299: LSPV: next addr 127.0.0.0
*Aug 30 20:39:04.299: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:04.299: 7F 00 00 00 FF FF FF FF

Router# undebug all

multipath information debugging is off
```

Related Commands

Command	Description
ping mpls	Checks MPLS LSP connectivity.
trace mpls	Discovers MPLS LSP routes that packets will actually take when traveling to their destinations.

echo

To customize the default behavior of echo packets, use the **echo** command in MPLS OAM configuration mode. To set the echo packet's behavior to its default value, use the **no** form of this command.

echo {**revision** {**3** | **4**} | **vendor-extension**}

no echo {**revision** {**3** | **4**} | **vendor-extension**}

Syntax Description

revision	Specifies the revision number of the echo packet's default values. Valid values are: • **3**—draft-ietf-mpls-lsp-ping-03 (Revision 2) • **4**—RFC 4379 Compliant (Default)
vendor-extension	Sends Cisco-specific extension of type, length, values (TLVs) with echo packets.

Command Default

Cisco-specific extension TLVs are sent with the echo packet. Revision 4 is the router's default.

Command Modes

MPLS OAM configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

Before you can enter the **echo** command, you must first enter the **mpls oam** command to enter MPLS OAM configuration mode.

Specify the **revision** keyword only if one of the following conditions exists:

- You want to change the revision number from the default of revision **4** to revision **3**.
- You previously entered the **mpls oam** command and changed the revision number to **3** and now you want to change the revision back to **4**.

To prevent failures reported by the replying router due to TLV version issues, you can use the **echo revision** command to configure all routers in the core for the same version of the Internet Engineering Task Force (IEFT) label switched paths (LSP) ping draft. For example, if the network is running draft RFC 4379 implementations, but one router is capable of only Version 3 (Cisco Revision 3), configure all routers in the network to operate in Revision 3 mode. Revision 3 mode applies only to Multiprotocol Label Switching (MPLS) LSP ping or traceroute. Revision 3 mode does not support MPLS multipath LSP traceroute.

The **vendor-extension** keyword is enabled by default in the router. If your network includes routers that are not Cisco routers, you may want to disable Cisco extended TLVs. To disable Cisco extended TLVs, specify the **no echo vendor-extension** command in MPLS OAM configuration mode. To enable Cisco extended TLVs again, respecify the **echo** command with the **vendor-extension** keyword.

Examples

The following example uses Revision 3 of the echo packets and sends the vendor's extension TLV with the echo packet:

```
mpls oam
 echo revision 3
 echo vendor-extension
 exit
```

Related Commands

Command	Description
mpls oam	Enters MPLS OAM configuration mode for customizing the default behavior of echo packets.

mpls oam

To enter MPLS OAM configuration mode for customizing the default behavior of echo packets, use the **mpls oam** command in global configuration mode. To disable MPLS OAM functionality, use the **no** format of this command.

mpls oam

no mpls oam

Syntax Description

This command has no arguments or keywords.

Command Default

Customizing the default behavior of echo packets is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.
12.4(11)T	The **no** and **default** keywords were removed.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

After you enter the **mpls oam** command, you can enter the **echo** command in MPLS OAM configuration mode to specify the revision number of the echo packet's default values or to send the vendor's extension type, length, values (TLVs) with the echo packet.

Examples

The following example enters MPLS OAM configuration mode for customizing the default behavior of echo packets:

```
mpls oam
```

Related Commands

Command	Description
echo	Customizes the default behavior of echo packets.
ping mpls	Checks MPLS LSP connectivity.
trace mpls	Discovers MPLS LSP routes that packets will actually take when traveling to their destinations.

trace mpls

To discover Multiprotocol Label Switching (MPLS) label switched path (LSP) routes that packets actually take when traveling to their destinations, use the **trace mpls** command in privileged EXEC mode.

trace mpls
{**ipv4** *destination-address*/*destination-mask* | **traffic-eng Tunnel** *tunnel-number*}
[**timeout** *seconds*]
[**destination** *address-start* [*address-end* | *address-increment*]]
[**revision** {**1** | **2** | **3** | **4**}]
[**source** *source-address*]
[**exp** *exp-bits*]
[**ttl** *maximum-time-to-live*]
[**reply** {**dscp** *dscp-bits* | **mode** *reply-mode* {**ipv4** | **no-reply** | **router-alert**} | **pad-tlv**}]
[**force-explicit-null**]
[**output interface** *tx-interface* [**nexthop** *ip-address*]]
[**flags fec**]
[**revision** *tlv-revision-number*]

Syntax Description

ipv4	Specifies the destination type as a Label Distribution Protocol (LDP) IPv4 address.
destination-address	Address prefix of the target to be tested.
/destination-mask	Number of bits in the network mask of the target address. The slash is required.
traffic-eng Tunnel *tunnel-number*	Specifies the destination type as an MPLS traffic engineering (TE) tunnel.
timeout *seconds*	(Optional) Specifies the timeout interval in seconds. The range is from 0 to 3600. The default is 2 seconds.
destination	(Optional) Specifies a network 127 address.
address-start	(Optional) The beginning network 127 address.
address-end	(Optional) The ending network 127 address.
address-increment	(Optional) Number by which to increment the network 127 address.
revision {**1** | **2** | **3** | **4**}	(Optional) Selects the type, length, values (TLVs) version of the implementation. Use the **revision 4** default unless attempting to interoperate with devices running Cisco IOS Release 12.0(27)S1 or 12.0(27)S2. If you do not select a **revision** keyword, the software uses the latest version. See Table 4 in the"Revision Keyword Usage" section of the "Usage Guidelines" section for information on when to select the **1**, **2**, **3**, and **4** keywords.
source *source-address*	(Optional) Specifies the source address or name. The default address is loopback0. This address is used as the destination address in the MPLS echo response.
exp *exp-bits*	(Optional) Specifies the MPLS experimental field value in the MPLS header for an MPLS echo reply. Valid values are from 0 to 7. Default is 0.
ttl *maximum-time-to-live*	(Optional) Specifies a maximum hop count.

reply dscp *dscp-bits*	(Optional) Provides the capability to request a specific class of service (CoS) in an echo reply by providing a differentiated services code point (DSCP) value. The echo reply is returned with the IP header ToS byte set to the value specified in the **reply dscp** keyword.
reply mode *reply-mode*	(Optional) Specifies the reply mode for the echo request packet. The *reply-mode* is one of the following: • **ipv4**—Reply with an IPv4 User Datagram Protocol (UDP) packet (default). • **no-reply**—Do not send an echo request packet in response. • **router-alert**—Reply with an IPv4 UDP packet with router alert.
reply pad-tlv	(Optional) Tests the ability of the sender of an echo reply to support the copy pad TLV to echo reply.
force-explicit-null	(Optional) Forces an explicit null label to be added to the MPLS label stack even though the label was unsolicited.
output interface *tx-interface*	(Optional) Specifies the output interface for echo requests.
nexthop *ip-address*	(Optional) Causes packets to go through the specified next-hop address.
flags fec	(Optional) Requests that target Forwarding Equivalence Class (FEC) stack validation be done at the egress router. A downstream map TLV containing the correct received labels must be present in the echo request for target FEC stack checking to be performed. **Note** Be sure to use this keyword in conjunction with the **ttl** keyword.
revision *tlv-revision-number*	(Optional) Cisco TLV revision number.

Defaults

revision = 4
timeout = 2 seconds
reply mode = ipv4 via UDP (2)
Maximum time-to-live = 30 hops
Experimental bits in MPLS header = 0

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(27)S	This command was introduced.
12.2(18)SXE	The **reply dscp** and **reply pad-tlv** keywords were added.
12.4(6)T	The following keywords were added: **force-explicit-null**, **output interface**, **flags fec**, and **revision**.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.

Release	Modification
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2. The **nexthop** keyword was added.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

Use the **trace mpls** command to validate, test, or troubleshoot IPv4 LDP LSPs and IPv4 Resource Reservation Protocol (RSVP) TE tunnels.

UDP Destination Address Usage

The destination address is a valid 127/8 address. You can specify a single address or a range of numbers from 0.0.0 to *x.y.z*, where *x*, *y*, and *z* are numbers from 0 to 255 and correspond to the 127.*x.y.z* destination address.

The MPLS echo request destination address in the UDP packet is not used to forward the MPLS packet to the destination router. The label stack that is used to forward the echo request routes the MPLS packet to the destination router. The 127/8 address guarantees that the packets are routed to the localhost (the default loopback address of the router processing the address) if the UDP packet destination address is used for forwarding.

In addition, the destination address is used to adjust load balancing when the destination address of the IP payload is used for load balancing.

Time-to-Live Keyword Usage

The time-to-live value indicates the maximum number of hops a packet should take to reach its destination. The value in the TTL field in a packet is decremented by 1 each time the packet travels through a router.

For MPLS LSP ping, the TTL is a value after which the packet is discarded and an MPLS echo reply is sent back to the originating router.

For MPLS Multipath LSP Traceroute, the TTL is a maximum time-to-live value and is used to discover the number of downstream hops to the destination router. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4, ...) to accomplish this.

Revision Keyword Usage

The **revision** keyword allows you to issue a **trace mpls ipv4** or **trace mpls traffic-eng** command based on the format of the TLV. Table 4 lists the revision option and usage guidelines for each option.

Table 4 Revision Options and Option Usage Guidelines

Revision Option	Option Usage Guidelines
1[1]	Not supported in Cisco IOS Release 12.4(11)T or later releases. Version 1 (draft-ietf-mpls-ping-03) For a device running Cisco IOS Release 12.0(27)S3 or a later release, you must use the **revision 1** keyword when you send LSP ping or LSP traceroute commands to devices running Cisco IOS Release 12.0(27)S1 or 12.0(27)S2.
2	Version 2 functionality was replaced by Version 3 functionality before any images were shipped.

Table 4 Revision Options and Option Usage Guidelines (continued)

Revision Option	Option Usage Guidelines
3	Version 3 (draft-ietf-mpls-ping-03). • For a device implementing Version 3 (Cisco IOS Release 12.0(27)S3 or a later release), you must use the **revision 1** keyword when you send the LSP **ping** or LSP **traceroute** command to a device implementing Version 1 (that is, either Cisco IOS Release 12.0(27)S1 or Release 12.0(27)S2). • A **ping mpls pseudowire** command does not work with devices running Cisco IOS Release 12.0(27)S1 or Release 12.0(27)S2.
4	• Version 8 (draft-ietf-mpls-ping-08)—Applicable before Cisco IOS Release 12.4(11)T. All echo packet's TLVs are formatted as specified in Version 8. • RFC 4379 compliant—Applicable after Cisco IOS Release 12.4(11)T. All echo packet's TLVs are formatted as specified in RFC 4379. This is the recommended version.

1. If you do not specify the **revision** keyword, the software uses the latest version.

Examples

The following example shows how to trace packets through an MPLS LDP LSP:

```
Router# trace mpls ipv4 10.131.191.252/32
```

Alternatively, you can use the interactive mode:

```
Protocol [ip]: mpls
Target IPv4, pseudowire or traffic-eng [ipv4]: <ipv4 |pseudowire |tunnel> ipv4
Target IPv4 address: 10.131.191.252
Target mask: /32
Repeat [1]:
Packet size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Destination start address:
Destination end address:
Source address:
EXP bits in mpls header [0]:
TimeToLive [255]:
Reply mode (2-ipv4 via udp, 3-ipv4 via udp with router alert) [2]:
Reply ip header DSCP bits [0]:

Tracing MPLS Label Switched Path to 10.131.191.252/32, timeout is 2 seconds

Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0


Type escape sequence to abort.

  0 10.131.159.245 mtu 1500 []
! 1 10.131.191.252 100 ms
```

The following example shows how to trace packets through an MPLS TE tunnel:

```
Router# trace mpls traffic-eng Tunnel 0

Tracing MPLS TE Label Switched Path on Tunnel0, timeout is 2 seconds

Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 10.131.159.230 mtu 1500 [Labels: 22 Exp: 0]
R 1 10.131.159.225 mtu 1500 [Labels: 22 Exp: 6] 72 ms
R 2 10.131.191.229 mtu 1504 [implicit-null] 72 ms
! 3 10.131.191.252 92 ms
```

Alternatively, you can use the interactive mode:

```
Router# traceroute
Protocol [ip]: mpls
Target IPv4 or tunnel [ipv4]: traffic-eng
Tunnel number [0]:
Repeat [1]:
Timeout in seconds [2]:
Extended commands? [no]:
Tracing MPLS TE Label Switched Path on Tunnel0, timeout is 2 seconds

Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 10.131.159.230 mtu 1500 [Labels: 22 Exp: 0]
R 1 10.131.159.225 mtu 1500 [Labels: 22 Exp: 6] 72 ms
R 2 10.131.191.229 mtu 1504 [implicit-null] 72 ms
! 3 10.131.191.252 92 ms
```

Use the **show running-config** command to verify the configuration of Tunnel 0 (shown in bold):

```
Router# show running-config interface tunnel 0

Building configuration...

Current configuration : 210 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.131.191.252       <---- Tunnel destination IP address.
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng path-option 5 explicit name as1pe-long-path
end

Router# show mpls traffic-eng tunnels tunnel 0 brief
```

```
Signalling Summary:
    LSP Tunnels Process:            running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 1369 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    disabled
TUNNEL NAME               DESTINATION      UP IF     DOWN IF   STATE/PROT
PE_t0                     10.131.191.252   -         Et0/0     up/up

Router# show ip cef 10.131.191.252

10.131.191.252/32, version 37, epoch 0, cached adjacency 10.131.159.246
0 packets, 0 bytes
  tag information set, all rewrites owned
    local tag: 21
  via 10.131.159.246, Ethernet1/0, 0 dependencies
    next hop 10.131.159.246, Ethernet1/0
    valid cached adjacency
    tag rewrite with Et1/0, 10.131.159.246, tags imposed {}
```

The tunnel destination has the same IP address as the one in the earlier trace IPv4 example, but the trace takes a different path, even though tunnel 0 is not configured to forward traffic by means of autoroute or static routing. The **trace mpls traffic-eng** command is powerful; it enables you to test the tunnels to verify that they work before you map traffic onto them.

Related Commands

Command	Description
ping mpls	Checks MPLS LSP connectivity.

trace mpls multipath

To discover all Multiprotocol Label Switching (MPLS) label switched paths (LSPs) from an egress router to an ingress router, use the **trace mpls multipath** command in privileged EXEC mode.

trace mpls multipath ipv4 *destination-address/destination-mask-length*
[**timeout** *seconds*]
[**interval** *milliseconds*]
[**destination** *address-start address-end*]
[**source** *source-address*]
[**exp** *exp-bits*]
[**ttl** *maximum-time-to-live*]
[**reply mode** {**ipv4** | **router-alert**}]
[**reply dscp** *dscp-value*]
[**retry-count** *retry-count-value*]
[**force-explicit-null**]
[**output interface** *tx-interface* [**nexthop** *ip-address*]]
[**hashkey ipv4 bitmap** *bitmap-size*]
[**flags fec**]
[**verbose**]

Syntax Description

ipv4	Specifies the destination type as a Label Distribution Protocol (LDP) IPv4 address.
destination-address	Address prefix of the target to be tested.
/destination-mask-length	Number of bits in the network mask of the target address. The slash is required.
timeout *seconds*	(Optional) Specifies the timeout interval in seconds. The range is from 0 to 3600. The default is 2 seconds.
interval *milliseconds*	(Optional) Sets the time between successive MPLS echo requests in milliseconds. This allows you to pace the transmission of packets so that the receiving router does not drop packets. The default is 0 milliseconds. Valid values are from 0 to 3500000 milliseconds.
destination	(Optional) Specifies a network 127 address.
address-start	(Optional) The beginning network 127 address.
address-end	(Optional) The ending network 127 address.
source	(Optional) Specifies the source address or name.
source-address	(Optional) Source address or name.
exp *exp-bits*	(Optional) Specifies the MPLS experimental field value in the MPLS header for an MPLS echo reply. Valid values are from 0 to 7. Default is 0.
ttl *maximum-time-to-live*	(Optional) Specifies a maximum hop count.
reply mode {**ipv4** | **router-alert**}	(Optional) Specifies the reply mode for the echo request packet. The reply mode is one of the following: • **ipv4** = Reply with an IPv4 User Datagram Protocol (UDP) packet (default). • **router-alert** = Reply with an IPv4 UDP packet with router alert.

reply dscp *dscp-value*	(Optional) Controls the differentiated services codepoint (DSCP) value of an echo reply. Allows the support of a class of service (CoS) in an echo reply.
retry-count re*try-count-value*	(Optional) Sets the number of timeout retry attempts during a multipath LSP trace. A retry is attempted if an outstanding echo request times out waiting for the corresponding echo reply. A re*try-count-value* of 0 means infinite retries. Valid values are from 0 to10.
force-explicit-null	(Optional) Forces an explicit null label to be added to the MPLS label stack even though the label was unsolicited.
output interface *tx-interface*	(Optional) Specifies the output interface for MPLS echo requests.
nexthop *ip-address*	(Optional) Causes packets to go through the specified next-hop address.
hashkey ipv4 bitmap *bitmap-size*	(Optional) Allows you to control the hash key and multipath settings. • **ipv4**—Indicates an IPv4 address, which is the only hashkey type valid for multipath (type 8). • **bitmap** *bitmap-size*—Size of the bitmap IPv4 addresses.
flags fec	(Optional) Requests that target Forwarding Equivalence Class (FEC) stack validation of a transit router be done at the egress router. **Note** Be sure to use the **flags fec** keywords in conjunction with the **ttl** keyword.
verbose	(Optional) Displays the MPLS echo reply sender address of the packet and displays return codes.

Command Default

timeout = 2 seconds
interval = 0 milliseconds
reply mode = IPv4 via UDP (2)
Maximum time-to-live = 30 hops
Experimental bits in MPLS header = 0

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(31)SB2	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

Use the **trace mpls multipath** command to discover all possible paths between an egress and ingress router in multivendor networks that use IPv4 load balancing at the transit routers.

Use the **destination** *address-start address-end* keyword and arguments to specify a valid 127/8 address. You have the option to specify a single *x.y.z-address* or a range of numbers from 0.0.0 to *x.y.z*, where *x*, *y*, and *z* are numbers from 0 to 255 and correspond to the 127.*x.y.z* destination address. The MPLS echo request destination address in the UDP packet is not used to forward the MPLS packet to the destination

router. The label stack that is used to forward the echo request routes the MPLS packet to the destination router. The 127/8 address guarantees that the packets are routed to the localhost (the default loopback address of the router processing the address) if the UDP packet destination address is used for forwarding. In addition, the destination address is used to adjust load balancing when the destination address of the IP payload is used for load balancing.

Examples

The following example shows how to discover all IPv4 LSPs to a router whose IP address is 10.1.1.150:

```
Router# trace mpls multipath ipv4 10.1.1.150/32

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0 LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1 L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5 LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
  Echo Request (sent/fail) (14/0)
  Echo Reply (received/timeout) (14/0)
  Total Time Elapsed 472 ms
```

The following example shows how to set the number of timeout retry attempts to 4 during a multipath LSP trace:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 retry-count 4

Starting LSP Multipath Traceroute for 10.1.1.150/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.0 LLL!
Path 1 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.1 L!
Path 2 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.5 LL!
Path 3 found,
 output interface Et0/0 source 10.1.111.101 destination 127.0.0.7
```

```
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 460 ms
```

The following example shows that outgoing MPLS Operation, Administration, and Management (OAM) echo request packets will go through the interface e0/0 and will be restricted to the path with the next hop address of 10.0.0.3:

```
Router# trace multipath ipv4 10.4.4.4/32 output interface e0/0 nexthop 10.0.0.3

Starting LSP Multipath Traceroute for 10.4.4.4/32

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
L!
Path 0 found,
 output interface Et0/0 nexthop 10.0.0.3
 source 10.0.0.1 destination 127.0.0.0

Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (2/0)
 Echo Reply (received/timeout) (2/0)
 Total Time Elapsed 728 ms
```

Related Commands

Command	Description
echo	Customizes the default behavior of echo packets.
mpls oam	Enters MPLS OAM configuration mode for customizing the default behavior of echo packet.
ping mpls	Checks MPLS LSP connectivity.
trace mpls	Discovers MPLS LSP routes that packets will actually take when traveling to their destinations.

Feature Information for MPLS EM—MPLS LSP Multipath Tree Trace

Table 5 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 5 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release. Unless noted otherwise, subsequent releases of that Cisco IOS software release also support that feature.

Table 5 Feature Information for MPLS EM—MPLS LSP Multipath Tree Trace

Feature Name	Releases	Feature Information
MPLS EM—MPLS LSP Multipath Tree Trace	12.2(31)SB2 12.2(33)SRB	The MPLS EM—MPLS LSP Multipath Tree Trace feature provides the means to discover all the possible paths of a label switched path (LSP) between an egress and ingress router. Once discovered, these paths can be retested on a periodic basis using Multiprotocol Label Switching (MPLS) LSP ping or traceroute. This feature is an extension to the MPLS LSP traceroute functionality for the tracing of IPv4 LSPs. Cisco IOS MPLS Embedded Management (EM) is a set of standards and value-added services that facilitate the deployment, operation, administration, and management of MPLS-based networks in line with the fault, configuration, accounting, performance, and security (FCAPS) model. In Cisco IOS Release 12.2(31)SB2, this feature was introduced. In Cisco IOS Release 12.2(33)SRB, support was added for a Cisco IOS 12.2SR release. The following sections provide information about this feature: • Overview of MPLS LSP Multipath Tree Trace, page 3 • Discovery of IPv4 Load Balancing Paths by MPLS LSP Multipath Tree Trace, page 3 • Echo Reply Return Codes Sent by the Router Processing Multipath LSP Tree Trace, page 4 • Customizing the Default Behavior of MPLS Echo Packets, page 5

Table 5 Feature Information for MPLS EM—MPLS LSP Multipath Tree Trace (continued)

Feature Name	Releases	Feature Information
		• Configuring MPLS LSP Multipath Tree Trace, page 7 • Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace, page 9Monitoring LSP Paths Discovered by MPLS LSP Multipath Tree Trace Using MPLS LSP Traceroute, page 10 • Using DSCP to Request a Specific Class of Service in an Echo Reply, page 13 • Controlling How a Responding Router Replies to an MPLS Echo Request, page 14 • Specifying the Output Interface for Echo Packets Leaving a Router for MPLS LSP Multipath Tree Trace, page 16 • Setting the Pace of MPLS Echo Request Packet Transmission for MPLS LSP Multipath Tree Trace, page 17 • Enabling MPLS LSP Multipath Tree Trace to Detect LSP Breakages Caused by an Interface that Lacks an MPLS Configuration, page 18 • Requesting that a Transit Router Validate the Target FEC Stack for MPLS LSP Multipath Tree Trace, page 19 • Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace, page 21 The following commands were modified by this feature: **debug mpls lspv**, **echo**, **mpls oam**, **trace mpls**, and **trace mpls multipath**.

Glossary

ECMP—equal-cost multipath. Multiple routing paths of equal cost that may be used for packet forwarding.

FEC—Forwarding Equivalence Class. A set of packets that can be handled equivalently for forwarding purposes and are thus suitable for binding to a single label. Examples include the set of packets destined for one address prefix and the packets in any flow.

flow—A set of packets traveling between a pair of hosts, or between a pair of transport protocol ports on a pair of hosts. For example, packets with the same source address, source port, destination address, and destination port might be considered a flow.

A flow is also a stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

localhost—A name that represents the host router (device). The localhost uses the reserved loopback IP address 127.0.0.1.

LSP—label switched path. A connection between two routers in which MPLS forwards the packets.

LSPV—Label Switched Path Verification. An LSP ping subprocess. It encodes and decodes MPLS echo requests and replies, and it interfaces with IP, MPLS, and AToM switching for sending and receiving MPLS echo requests and replies. At the MPLS echo request originator router, LSPV maintains a database of outstanding echo requests for which echo responses have not been received.

MPLS router alert label—An MPLS label of 1. An MPLS packet with a router alert label is redirected by the router to the Route Processor (PR) processing level for handling. This allows these packets to bypass any forwarding failures in hardware routing tables.

OAM—Operation, Administration, and Management.

punt—Redirect packets with a router alert from the line card or interface to Route Processor (RP) level processing for handling.

RP—Route Processor. The processor module in a Cisco 7000 series router that contains the CPU, system software, and most of the memory components that are used in the router. It is sometimes called a supervisory processor.

TTL—time-to-live. A parameter you can set that indicates the maximum number of hops a packet should take to reach its destination.

TLV—type, length, values. A block of information included in a Cisco Discovery Protocol address.

UDP—User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, so error processing and retransmission must be handled by other protocols. UDP is defined in RFC 768.

XDR—eXternal Data Representation. Standard for machine-independent data structures developed by Sun Microsystems. Used to transport messages between the Route Processor (RP) and the line card.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.