Configuring MPLS LSP Multipath Tree Trace

This chapter describes how to configure Multiprotocol Label Switching (MPLS) connectivity with the MPLS LSP Multipath Tree Trace feature.

This chapter includes the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information" chapter or the Feature History table below.

Information About MPLS LSP Multipath Tree Trace

.i.MPLS:tree trace;

The 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 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 LSP Multipath Tree Trace feature to discover all paths for an IPv4 LSP.

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

This section includes the following topics:

Overview of MPLS LSP Multipath Tree Trace

As the number of MPLS deployments increases, the number of traffic types that 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.

Before the release of the MPLS LSP Multipath Tree Trace feature, no automated way existed to discover all paths between provider edge (PE) routers, and troubleshooting forwarding problems between PEs was difficult.

The 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.

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

.i.load balancing;

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:

```
switch# traceroute 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 enter the **traceroute 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 many routes, you might need to use a larger bitmap size.

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

Table 35-1 describes the codes that the router processing a multipath LSP tree trace packet returns to the sender about the failure or success of the request.

Table 35-1 Echo Reply Return Codes

Output Code	Echo Return Code	Meaning
Period "."	—	A timeout occurred before the target router could 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.

Licensing Requirements for MPLS LSP Multipath Tree Trace

Product	License Requirement
Cisco NX-OS	The MPLS LSP Multipath Tree Trace feature requires an MPLS license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for MPLS LSP Multipath Tree Trace

The MPLS LSP Multipath Tree Trace feature has the following prerequisites:

- Before you can run MPLS ping and traceroute, ensure that the Intrusion Detection System (IDS) is disabled (specifically the option that drops packets if the IP address is in the reserved 127.x.x.x range).
- You must enable the MPLS LDP feature.

- You must understand the concepts and know how to use MPLS LSP ping or traceroute as described in the *MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV* document.
- The routers in your network must use an implementation based on IETF 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.

Guidelines and Limitations for MPLS LSP Multipath Tree Trace

The MPLS LSP Multipath Tree Trace feature has the following configuration guidelines and limitations:

- All restrictions that apply to the MPLS LSP ping and LSP traceroute features also apply to the MPLS LSP Multipath Tree Trace feature as follows:
  - You cannot use the MPLS LSP Multipath Tree Trace feature to trace the path taken by AToM packets. The MPLS LSP Multipath Tree Trace feature is not supported for AToM. (MPLS LSP ping is supported for AToM.) However, you can use the MPLS LSP Multipath Tree Trace feature to troubleshoot the Interior Gateway Protocol (IGP) LSP that is used by AToM.
  - You cannot use the MPLS LSP Multipath Tree Trace feature to validate or trace MPLS virtual private networks (VPNs). 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.

Configuring MPLS LSP Multipath Tree Trace

This section includes the following topics:

Customizing the Default Behavior of MPLS Echo Packets

.i.customized echo packets;

You can 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**, **traceroute mpls**, or **traceroute mpls multipath** command, you should ensure 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 can run 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

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

SUMMARY STEPS

1. **configure terminal**
2. **mpls oam**
3. **echo revision** {**3** | **4**}
4. [**no**] **echo vendor-extension**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal	Enters global configuration mode.
Step 2	**mpls oam** **Example:** switch(config)# mpls oam	Enters MPLS OAM configuration mode and customizes the default behavior of echo packets.
Step 3	**echo revision** {**3** | **4**} **Example:** switch(config-mpls)# echo revision 4	Customizes the default behavior of echo packets. • The **revision** keyword sets 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.
Step 4	[**no**] **echo vendor-extension** **Example:** switch(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**.

Configuring MPLS LSP Multipath Tree Trace

You can configure the MPLS multipath LSP tree trace traceroute. This task helps you to 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 can in some cases detect the formatting of the sender of an MPLS echo request. However, in certain cases an echo request or echo reply might not contain the Cisco extension TLV. To avoid complications in which 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. **configure terminal**
2. **mpls oam**
3. **echo revision 4**
4. (Optional) [**no**] **echo vendor-extension**
5. **traceroute mpls multipath ipv4** *destination-ip-address/destination-mask-length*

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**mpls oam** **Example:** `switch(config)# mpls oam`	Enters MPLS OAM configuration mode.
Step 3	**echo revision 4** **Example:** `switch(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 4	[no] **echo vendor-extension** **Example:** `switch(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 5	**traceroute mpls multipath ipv4** *destination-ip-address/destination-mask-length* **Example:** `switch# traceroute 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.

Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace

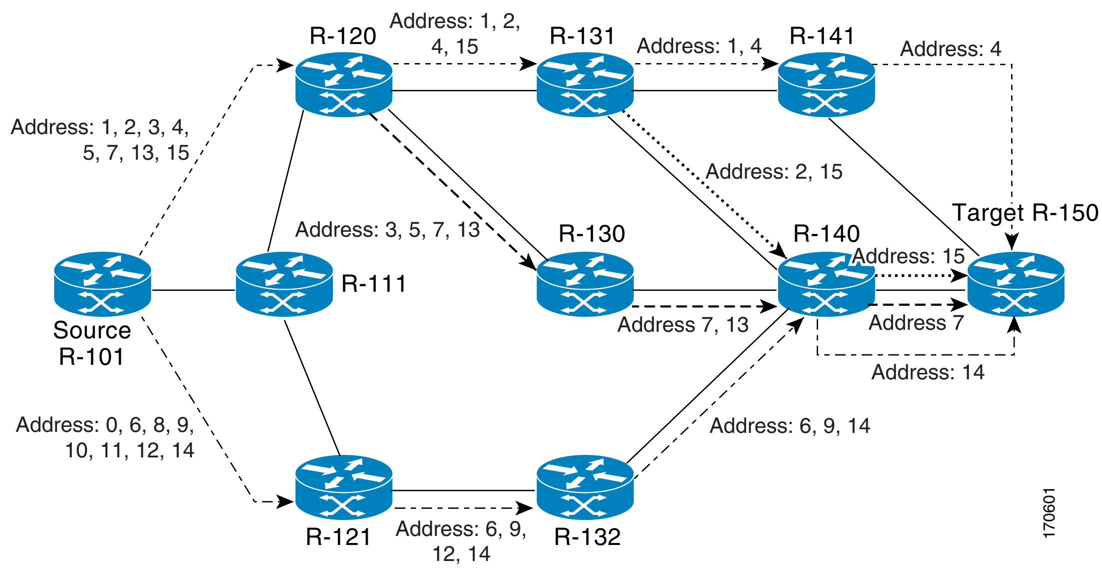
You can discover IPv4 load balancing paths using the MPLS LSP Multipath Tree Trace feature.

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 the MPLS LSP Multipath Tree Trace feature, you can use MPLS LSP traceroute to monitor these paths.

Figure 35-1 shows how the MPLS LSP Multipath Tree Trace feature discovers LSP paths in a sample network. In Figure 35-1, the bitmap size is 16 and the numbers 0 to 15 represent the bitmapped addresses that the MPLS LSP Multipath Tree Trace feature uses to discover all the paths from the source LSR R-101 to the target LSR R-150. Figure 35-1 illustrates how the **traceroute mpls multipath** command discovers all LSP paths in the sample network.

Figure 35-1 MPLS LSP Multipath Tree Trace Path Discovery in a Sample Network



SUMMARY STEPS

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

DETAILED STEPS

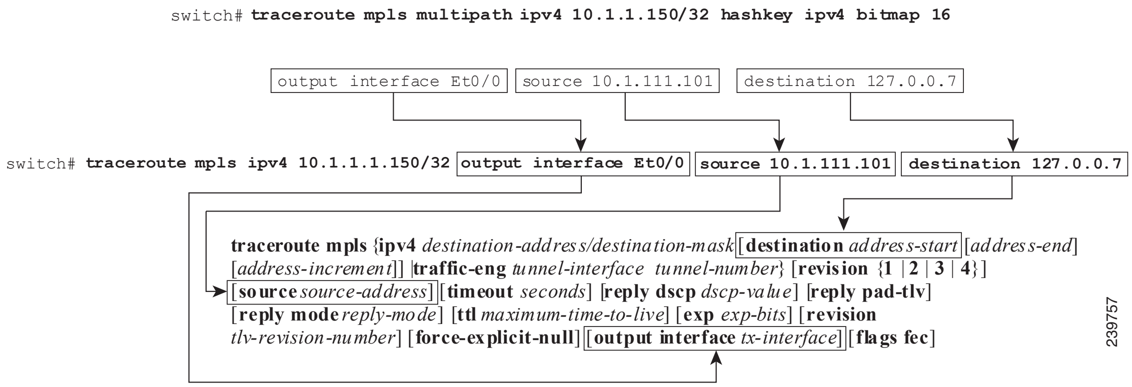
	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal	Enters global configuration mode.
Step 2	**mpls oam** **Example:** switch(config)# mpls oam	Enters MPLS OAM configuration mode and sets the echo packet attribute to Revision 4 (RFC 4379 compliant).
Step 3	**echo revision 4** **Example:** switch(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 4	**traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size* **Example:** switch# traceroute 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

You can monitor LSP paths that are discovered by the MPLS LSP Multipath Tree Trace feature using the MPLS LSP traceroute. You can take output directly from the **traceroute mpls multipath** command and add it to a **traceroute mpls** command periodically to verify that the path is still operating.

Figure 35-2 shows the mapping of the output of a **traceroute mpls multipath** command to a **traceroute mpls** command.

Figure 35-2 Mapping of traceroute mpls multipath Command Output to a traceroute mpls Command



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

SUMMARY STEPS

1. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*
2. **traceroute mpls ipv4** *destination-address***/***destination-mask-length* [**output interface** *tx-interface*] [**source** *source-address*] [**destination** *address-start*]

DETAILED STEPS

Step 1 Discover all MPLS LSPs from an egress router to an ingress router **by entering the traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size command.*

This example shows how to discover all MPLS LSPs from an egress router to an ingress router:

```
switch# traceroute 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 **traceroute mpls multipath** command in the example shows the result of path discovery with the MPLS LSP Multipath Tree Trace feature. In this example, the command sets the bitmap size to 16. Path discovery starts by the MPLS LSP Multipath Tree Trace feature 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 2 Verify that the paths discovered when you entered a **traceroute mpls multipath** command are still operating by entering the **traceroute mpls ipv4** *destination-address***/***destination-mask-length* [**output interface** *tx-interface*] [**source** *source-address*] [**destination** *address-start*] command.

For example, the output for Path 0 in the previous **traceroute mpls multipath** command in Step 1 is as follows:

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

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

```
switch# traceroute 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 **traceroute mpls multipath** command and add it to a **traceroute mpls** command periodically to verify that the path is still operating (see Figure 35-2).

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

Use the reply differentiated services code point (DSCP) option to 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.

SUMMARY STEPS

1. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**reply dscp** *dscp-value*]

DETAILED STEPS

	Command	Purpose
Step 1	**traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**reply dscp** *dscp-value*] **Example:** `switch# traceroute 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.

Controlling How a Responding Router Replies to an MPLS Echo Request

This section describes how to control how a responding router replies to an MPLS echo request.

This section includes the following topic:

- Reply Modes for an MPLS LSP Multipath Tree Trace Echo Request Response, page 35-44

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 **traceroute 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 **traceroute 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 supervisor process level for handling, which forces the supervisor of each intermediate router to 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 supervisor handling at each hop.

Table 35-2 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 35-2 Path Process Handling of IP and MPLS Router Alert Packets

Incoming Packet	Outgoing Packet	Software Switching Action
IP packet—Router alert option in IP header	IP packet—Router alert option in IP header	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	Removes the outermost router alert label and forwards the packet as an IP packet.

SUMMARY STEPS

1. **traceroute mpls multipath ipv4** *destination-address*/*destination-mask-length* **reply mode** {**ipv4** | **router-alert**}

DETAILED STEPS

	Command	Purpose
Step 1	**traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* **reply mode** {**ipv4** | **router-alert**} **Example:** `switch# traceroute 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.

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

You can 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 that are reachable through a given interface.

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. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**output interface** *tx-interface*]

DETAILED STEPS

	Command	Purpose
Step 1	**traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**output interface** *tx-interface*] **Example:** `switch# traceroute 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.

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

You can 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. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**interval** *milliseconds*]

DETAILED STEPS

	Command	Purpose
Step 1	**traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**interval** *milliseconds*] **Example:** `switch# traceroute 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.

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

You can enable the MPLS LSP Multipath Tree Trace feature 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.

For an MPLS LSP Multipath Tree Trace of LSPs that carry IPv4 FECs, you can force an explicit null label to be added to the MPLS label stack even though the label was unsolicited. This process allows MPLS LSP multipath tree trace to detect LSP breakages that are caused by an interface that is not configured for MPLS. The 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 **traceroute mpls multipath** command, you are looking for all MPLS LSP paths from an egress router to an ingress router. Failures 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. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* **force-explicit-null**

DETAILED STEP

	Command	Purpose
Step 1	**traceroute mpls multipath ipv4** *destination-address*/*destination-mask-length* **force-explicit-null** **Example:** `switch# traceroute 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.

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

You can request that a transit router validate the target Forwarding Equivalence Class (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 **traceroute mpls multipath** command. The default is that echo request packets are sent with the V flag set to 0.

SUMMARY STEPS

1. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**flags fec**] [**ttl** *maximum-time-to-live*]

DETAILED STEPS

	Command	Purpose
Step 1	**`traceroute mpls multipath ipv4`** *`destination-address`***`/`***`destination-mask-length`* [**`flags fec`**] [**`ttl`** *`maximum-time-to-live`*] **`Example:`** `switch# traceroute 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 request that the 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.

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

You can set the number of timeout attempts for the MPLS LSP Multipath Tree Trace feature.

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

SUMMARY STEPS

1. **traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**retry-count** *retry-count-value*]

DETAILED STEPS

	Command	Purpose
Step 1	**traceroute mpls multipath ipv4** *destination-address***/***destination-mask-length* [**retry-count** *retry-count-value*] **Example:** `switch# traceroute 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 retry-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.

Configuration Examples for MPLS LSP Multipath Tree Trace

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

- Example: Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace, page 35-59

Example: Customizing the Default Behavior of MPLS Echo Packets

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
```

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

Example: Configuring MPLS LSP Multipath Tree Trace

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
!
traceroute 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.

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

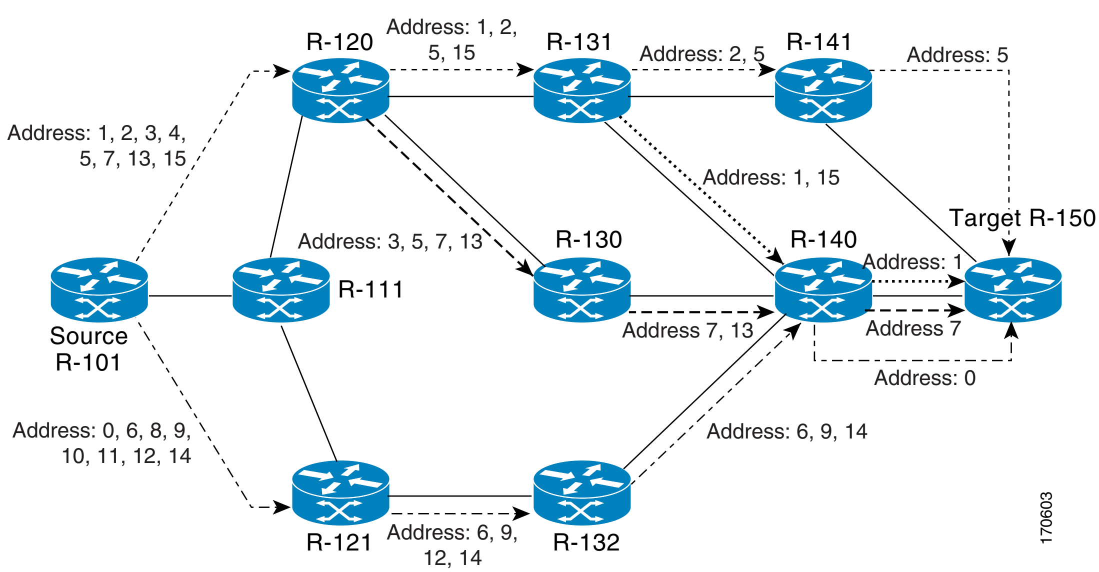
The following example shows how to use the MPLS LSP Multipath Tree Trace feature to discover IPv4 load-balancing paths. The example is based on the sample network shown in Figure 35-3. In this example, the bitmap size is set to 16. Therefore, path discovery starts by the MPLS LSP Multipath Tree Trace 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.

```
switch# traceroute 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 **traceroute mpls multipath** command in the example shows the result of path discovery with the MPLS LSP Multipath Tree Trace feature as shown in Figure 35-3.

Figure 35-3 MPLS LSP Multipath Tree Trace Path Discovery in a Sample Network



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

The following example shows how to use DSCP to request a specific Class of Service (CoS) in an echo reply:

```
switch# traceroute 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
```

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

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

```
switch# traceroute 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
```

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

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

```
switch# traceroute 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
```

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

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:

```
switch# traceroute 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
```

```
switch# traceroute 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.

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

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:

```
switch# traceroute 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:

```
switch# traceroute 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
```

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

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

```
switch# traceroute 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 **traceroute mpls multipath** command.

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

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

```
switch# traceroute 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 **traceroute mpls multipath** command that found one unexplored path, one successful path, and one broken path:

```
switch# traceroute 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 for MPLS LSP Multipath Tree Trace

For additional information related to the MPLS LSP Multipath Tree Trace feature, see the following sections:

- Related Documents, page 35-60
- MIBs, page 35-60

Related Documents

Related Topic	Document Title
Cisco NX-OS MPLS commands	*Cisco Nexus 7000 Series NX-OS MPLS Command Reference*

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

Feature History for MPLS LSP Multipath Tree Trace

Table 35-3 lists the release history for this feature.

Table 35-3 Feature History for MPLS LSP Multipath Tree Trace

Feature Name	Releases	Feature Information
MPLS LSP multipath tree trace	5.2(1)	This feature was introduced.