Network Working Group
Internet Draft
draft-ietf-mpls-lsp-ping-01.txt
Category: Standards Track
Expires: April 2003

K. Kompella (Juniper)
P. Pan (Ciena)
N. Sheth (Juniper)
D. Cooper (Global Crossing)
G. Swallow (Cisco)
S. Wadhwa (Juniper)
R. Bonica (WorldCom)

October 2002

Detecting MPLS Data Plane Liveness

*** DRAFT ***

Status of this Memo

Copyright Notice

Abstract

This document describes a simple and efficient mechanism that can be used to detect data plane failures in Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs). There are two parts to this document: information carried in an MPLS "echo request" and "echo reply" for the purposes of fault detection and isolation; and mechanisms for reliably sending the echo reply.

Sub-IP ID Summary

(This section to be removed before publication.)

(See Abstract above.)

RELATED DOCUMENTS

May be found in the "references" section.

WHERE DOES IT FIT IN THE PICTURE OF THE SUB-IP WORK

Fits in the MPLS box.

WHY IS IT TARGETED AT THIS WG

MPLS WG is currently looking at MPLS-specific error detection and recovery mechanisms. The mechanisms proposed here are for packet-based MPLS LSPs, which is why the MPLS WG is targeted.

JUSTIFICATION

The WG should consider this document, as it allows network operators to detect MPLS LSP data plane failures in the network. This type of failures have occurred, and are a source of concern to operators implementing MPLS networks.

1. Introduction

This document describes a simple and efficient mechanism that can be used to detect data plane failures in MPLS LSPs. There are two parts to this document: information carried in an MPLS "echo request" and "echo reply"; and mechanisms for transporting the echo reply. The first part aims at providing enough information to check correct operation of the data plane, as well as a mechanism to verify the data plane against the control plane, and thereby localize faults. The second part suggests two methods of reliable reply channels for the echo request message, for more robust fault isolation.

An important consideration in this design is that MPLS echo requests follow the same data path that normal MPLS packets would traverse. MPLS echo requests are meant primarily to validate the data plane, and secondarily to verify the data plane against the control plane. Mechanisms to check the control plane are valuable, but are not covered in this document.

To avoid potential Denial of Service attacks, it is recommended to regulate the MPLS ping traffic going to the control plane. A rate limiter should be applied to the well-known UDP port defined below.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [KEYWORDS].

1.2. Changes since last revision

(This section to be removed before publication.)

- Packet format changed; Version Number field added
- Reply modes: "don't reply" added
- Reply flags removed
- Return codes extended
- RSVP session formats modified
- VPN IPv4/v6 formats defined
- L2 VPN endpoint and L2 circuits defined
- Downstream mapping format changed
- Pad and Error Code TLVs introduced
- Aspects dealing with CR-LDP moved to non-normative appendix
- IPR notices and Full Copyright Statement (per 2026) added
- other nits to better conform to 2223bis

2. Motivation

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. There is a need to provide a tool that would enable users to detect such traffic "black holes" or misrouting within a reasonable period of time; and a mechanism to isolate faults.

In this document, we describe a mechanism that accomplishes these goals. This mechanism is modeled after the ping/traceroute philosophy: ping (ICMP echo request [ICMP]) is used for connectivity checks, and traceroute is used for hop-by-hop fault localization as well as path tracing. This document specifies a "ping mode" and a "traceroute" mode for testing MPLS LSPs.

The basic idea is to test that packets that belong to a particular Forwarding Equivalence Class (FEC) actually end their MPLS path on an LSR that is an egress for that FEC. This document proposes that this test be carried out by sending a packet (called an "MPLS echo request") along the same data path as other packets belonging to this FEC. An MPLS echo request also carries information about the FEC whose MPLS path is being verified. This echo request is forwarded just like any other packet belonging to that FEC. In "ping" mode (basic connectivity check), the packet should reach the end of the path, at which point it is sent to the control plane of the egress LSR, which then verifies that it is indeed an egress for the FEC. In "traceroute" mode (fault isolation), the packet is sent to the control plane of each transit LSR, which performs various checks that it is indeed a transit LSR for this path; this LSR also returns further information that helps check the control plane against the data plane, i.e., that forwarding matches what the routing protocols determined as the path.

One way these tools can be used is to periodically ping a FEC to ensure connectivity. If the ping fails, one can then initiate a traceroute to determine where the fault lies. One can also periodically traceroute FECs to verify that forwarding matches the control plane; however, this places a greater burden on transit LSRs and thus should be used with caution.

3. Packet Format

An MPLS echo request is a (possibly labelled) UDP packet; the contents of the UDP packet have the following format:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Version Number        |         Must Be Zero          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Message Type |   Reply mode  |  Return Code  | Return Subcode|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sender's Handle                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    TimeStamp Sent (seconds)                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  TimeStamp Sent (microseconds)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  TimeStamp Received (seconds)                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                TimeStamp Received (microseconds)              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            TLVs ...                           |
   .                                                               .
   .                                                               .
   .                                                               .
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Version Number is currently 1. (Note: the Version Number is to be incremented whenever a change is made that affects the ability of an implementation to correctly parse or process an MPLS echo request/reply. These changes include any syntactic or semantic changes made to any of the fixed fields, or to any TLV or sub-TLV assignment or format that is defined at a certain version number. The Version Number may not need to be changed if a TLV or sub-TLV is added.)

The Message Type is one of the following:

Value	Meaning
1	MPLS Echo Request
2	MPLS Echo Reply

The Reply Mode can take one of the following values:

Value	Meaning
1	Do not reply
2	Reply via an IPv4 UDP packet
3	Reply via an IPv4 UDP packet with Router Alert
4	Reply via the control plane

An MPLS echo request with "Do not reply" may be used for one-way connectivity tests; the receiving router may log gaps in the sequence numbers and/or maintain delay/jitter statistics. An MPLS echo request would normally have "Reply via an IPv4 UDP packet"; if the normal IPv4 return path is deemed unreliable, one may use "Reply via an IPv4 UDP packet with Router Alert" (note that this requires that all intermediate routers understand and know how to forward MPLS echo replies) or "Reply via the control plane" (this is currently only defined for control plane that uses RSVP).

The Return Code is set to zero by the sender. The receiver can set it to one of the following values:

Value	Meaning
0	The error code is contained in the Error Code TLV
1	Malformed echo request received
2	One or more of the TLVs was not understood
3	Replying router is an egress for the FEC
4	Replying router has no mapping for the FEC
5	Replying router is not one of the "Downstream Routers"
6	Replying router is one of the "Downstream Routers", and its mapping for this FEC on the received interface is the given label
7	Replying router is one of the "Downstream Routers", but its mapping for this FEC is not the given label

The Return Subcode is unused at present and SHOULD be set to zero.

The Sender's Handle is filled in by the sender, and returned unchanged by the receiver in the echo reply (if any). There are no semantics associated with this handle, although a sender may find this useful for matching up requests with replies.

The Sequence Number is assigned by the sender of the MPLS echo request, and can be (for example) used to detect missed replies.

The TimeStamp Sent is the time-of-day (in seconds and microseconds, wrt the sender's clock) when the MPLS echo request is sent. The TimeStamp Received in an echo reply is the time-of-day (wrt the receiver's clock) that the corresponding echo request was received.

TLVs (Type-Length-Value tuples) have the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
.                                                               .
.                                                               .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Types are defined below; Length is the length of the Value field in octets. The Value field depends on the Type; it is zero padded to align to a four-octet boundary.

Type #	Value Field
1	Target FEC Stack
2	Downstream Mapping
3	Pad
4	Error Code

3.1. Target FEC Stack

A Target FEC Stack is a list of sub-TLVs. The number of elements is determined by the looking at the sub-TLV length fields.

Sub-Type #	Length	Value Field
1	5	LDP IPv4 prefix
2	17	LDP IPv6 prefix
3	20	RSVP IPv4 Session Query
4	56	RSVP IPv6 Session Query
5		Reserved; see Appendix
6	13	VPN IPv4 prefix
7	25	VPN IPv6 prefix
8	14	L2 VPN endpoint
9	10	L2 circuit ID

Other FEC Types will be defined as needed.

Note that this TLV defines a stack of FECs, the first FEC element corresponding to the top of the label stack, etc.

An MPLS echo request MUST have a Target FEC Stack that describes the

FEC stack being tested. For example, if an LSR X has an LDP mapping for 192.168.1.1 (say label 1001), then to verify that label 1001 does indeed reach an egress LSR that announced this prefix via LDP, X can send an MPLS echo request with a FEC Stack TLV with one FEC in it, namely of type LDP IPv4 prefix, with prefix 192.168.1.1/32, and send the echo request with a label of 1001.

If LSR X wanted to verify that a label stack of <1001, 23456> is the right label stack to use to reach an IP VPN prefix of 10/8 in VPN foo on an egress LSR with loopback address 192.168.1.1 (learned via LDP), X has two choices. X can send an MPLS echo request with a FEC Stack TLV with a single FEC of type VPN IPv4 prefix with a prefix of 10/8 with the Route Distinguisher for VPN foo. Alternatively, X can send a FEC Stack TLV with two FECs, the first of type LDP IPv4 with a prefix of 192.168.1.1/32 and the second of type of IP VPN with a prefix 10/8 in VPN foo. In either case, the MPLS echo request would have a label stack of <1001, 23456>. (Note: in this example, 1001 is the "outer" label and 23456 is the "inner" label.)

3.1.1. IPv4 Prefix

The value consists of four octets of an IPv4 prefix followed by one octet of prefix length in bits. The IPv4 prefix is in network byte order. See [LDP] for an example of a Mapping for an IPv4 FEC.

3.1.2. IPv6 Prefix

The value consists of sixteen octets of an IPv6 prefix followed by one octet of prefix length in bits. The IPv6 prefix is in network byte order.

3.1.3. RSVP IPv4 Session

The value has the format below. The value fields are taken from [RFC3209, sections 4.6.1.1 and 4.6.2.1].

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 IPv4 tunnel end point address                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Must Be Zero         |      Tunnel ID                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Extended Tunnel ID                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                   IPv4 tunnel sender address                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Must Be Zero         |            LSP ID             |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.4. RSVP IPv6 Session

The value has the format below. The value fields are taken from [RFC3209, sections 4.6.1.2 and 4.6.2.2].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  IPv6 tunnel end point address                |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Must Be Zero         |          Tunnel ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Extended Tunnel ID                      |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   IPv6 tunnel sender address                  |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Must Be Zero         |            LSP ID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.5. VPN IPv4 Prefix

The value field consists of a Route Distinguisher, an IPv4 prefix and a prefix length, as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Route Distinguisher                      |
|                          (8 octets)                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv4 prefix                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |                 Must Be Zero                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.6. VPN IPv6 Prefix

The value field consists of a Route Distinguisher, an IPv6 prefix and a prefix length, as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Route Distinguisher                      |
   |                          (8 octets)                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         IPv6 prefix                           |
   |                                                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Prefix Length |                 Must Be Zero                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.7. L2 VPN Endpoint

The value field consists of a Route Distinguisher (8 octets), the sender (of the ping)'s CE ID (2 octets), the receiver's CE ID (2 octets), and an encapsulation type (2 octets), formatted as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Route Distinguisher                      |
   |                          (8 octets)                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Sender's CE ID        |       Receiver's CE ID        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      Encapsulation Type       |         Must Be Zero          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.8. L2 Circuit ID

The value field consists of a remote PE address (the address of the targetted LDP session), a VC ID and an encapsulation type, as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Remote PE Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             VC ID                             |
```

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      Encapsulation Type       |         Must Be Zero          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.2. Downstream Mapping

The Downstream Mapping is an optional TLV in an echo request. The Length is 12 + 4*N octets, where N is the number of Downstream Labels. The Value of a Downstream Mapping has the following format:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Downstream IPv4 Router ID                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |               MTU             | Address Type  |  Reserved     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Interface Address                           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |               Downstream Label                |    Protocol   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      .                                                               .
      .                                                               .
      .                                                               .
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |               Downstream Label                |    Protocol   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The MTU is the largest MPLS frame (including label stack) that fits on the interface to the Downstream LSR. The Address Type is one of:

Type #	Address Type
1	IPv4
2	Unnumbered

'Protocol' is taken from the following table:

Protocol #	Signaling Protocol
0	Unknown
1	Static
2	BGP
3	LDP
4	RSVP-TE
5	Reserved; see Appendix

The notion of "downstream router" should be explained. Consider an

LSR X. If a packet with outermost label L and TTL n>1 arrived at X on interface I, X must be able to compute which LSRs could receive the packet with TTL=n+1, and what label they would see. (It is outside the scope of this document to specify how this computation may be done.) The set of these LSRs are the downstream routers (and their corresponding labels) for X with respect to L.

The case where X is the LSR originating the echo request is a special case. X needs to figure out what LSRs would receive a labelled packet with TTL=1 when X tries to send a packet to the FEC Stack that is being pinged.

3.3. Pad TLV

The value part of the Pad TLV contains a variable number (>= 1) of octets. The first octet takes values from the following table; all the other octets (if any) are ignored. The receiver SHOULD verify that the TLV is received in its entirety, but otherwise ignores the contents of this TLV, apart from the first octet.

Value	Meaning
1	Drop Pad TLV from reply
2	Copy Pad TLV to reply
3-255	Reserved for future use

3.4. Error Code

The Error Code TLV is currently not defined; its purpose is to provide a mechanism for a more elaborate error reporting structure, should the reason arise.

4. Theory of Operation

4.1. Sending an MPLS Echo Request

An MPLS echo request is a (possibly) labelled UDP packet. The IP header is set as follows: the source IP address is a routable address of the sender; the destination IP address is a (randomly chosen) address from 127/8; the IP TTL is set to 1. The source UDP port is chosen by the sender; the destination UDP port is set to 3503 (assigned by IANA for MPLS echo requests). If the echo request is labelled, the MPLS TTL on all the labels except the outermost should be set to 1.

In "ping" mode (end-to-end connectivity check), the TTL in the outermost label is set to 255. In "traceroute" mode (fault isolation

mode), the TTL is set successively to 1, 2, ...

The sender chooses a Sender's Handle, and a Sequence Number. When sending subsequent MPLS echo requests, the sender SHOULD increment the sequence number by 1. However, a sender MAY choose to send a group of echo requests with the same sequence number to improve the chance of arrival of at least one packet with that sequence number.

The TimeStamp Sent is set to the time-of-day (in seconds and microseconds) that the echo request is sent. The TimeStamp Received is set to zero.

An MPLS echo request MUST have a FEC Stack TLV. Also, the Reply Mode must be set to the desired reply mode; the Return Code and Subcode are set to zero.

In the "traceroute" mode, the echo request SHOULD contain one or more Downstream Mapping TLVs. For TTL=1, all the downstream routers (and corresponding labels) for the sender with respect to the FEC Stack being pinged SHOULD be sent in the echo request. For n>1, the Downstream Mapping TLVs from the echo reply for TTL=(n-1) are copied to the echo request with TTL=n.

4.2. Receiving an MPLS Echo Request

An LSR L that receives an MPLS echo request first parses the packet to ensure that it is a well-formed packet, and that the TLVs are understood. If not, L SHOULD send an MPLS echo reply with the Return Code set to "Malformed echo request received" or "TLV not understood" (as appropriate), and the Subcode set to the appropriate value.

If the echo request is good, L then checks whether it is a valid transit or egress LSR for the FEC in the echo request. If not, L MAY log this fact.

If the echo request contains a Downstream Mapping TLV, L MUST further check whether its Router ID matches one of the Downstream IPv4 Router IDs; and if so, whether the given Downstream Label is in fact the label that L sent as its mapping for the FEC. For an RSVP FEC, the downstream label is the label that L sent in its Resv message. The result of the checks in the previous and this paragraph are captured in the Return Code/Subcode.

If the echo request has a Reply Mode that wants a reply, L uses the procedure in the next subsection to send the echo reply.

4.3. Sending an MPLS Echo Reply

An MPLS echo reply is a UDP packet. It MUST ONLY be sent in response to an MPLS echo request. The source IP address is a routable address of the replier; the source port is the well-known UDP port for MPLS ping. The destination IP address and UDP port are copied from the source IP address and UDP port of the echo request. The IP TTL is set to 255. If the Reply Mode in the echo request is "Reply via an IPv4 UDP packet with Router Alert", then the IP header MUST contain the Router Alert IP option.

The format of the echo reply is the same as the echo request. The Sender's Handle, the Sequence Number and TimeStamp Sent are copied from the echo request; the TimeStamp Received is set to the time-of-day that the echo request is received (note that this information is most useful if the time-of-day clocks on the requestor and the replier are synchronized). The FEC Stack TLV from the echo request MAY be copied to the reply.

The replier MUST fill in the Return Code and Subcode, as determined in the previous subsection.

If the echo request contains a Pad TLV, the replier MUST interpret the first octet for instructions regarding how to reply.

If the echo request contains a Downstream Mapping TLV, the replier SHOULD compute its downstream routers and corresponding labels for the incoming label, and add Downstream Mapping TLVs for each one to the echo reply it sends back.

4.4. Receiving an MPLS Echo Reply

An LSR X should only receive an MPLS Echo Reply in response to an MPLS Echo Request that it sent. Thus, on receipt of an MPLS Echo Reply, X should parse the packet to assure that it is well-formed, then attempt to match up the Echo Reply with an Echo Request that it had previously sent, using the destination UDP port and the Sender's Handle. If no match is found, then X jettisons the Echo Reply; otherwise, it checks the Sequence Number to see if it matches. Gaps in the Sequence Number MAY be logged and SHOULD be counted. Once an Echo Reply is received for a given Sequence Number (for a given UDP port and Handle), the Sequence Number for subsequent Echo Requests for that UDP port and Handle SHOULD be incremented.

If the Echo Reply contains Downstream Mappings, and X wishes to traceroute further, it SHOULD copy the Downstream Mappings into its next Echo Request (with TTL incremented by one).

4.5. Non-compliant Routers

If the egress for the FEC Stack being pinged does not support MPLS ping, then no reply will be sent, resulting in possible "false negatives". If in "traceroute" mode, a transit LSR does not support MPLS ping, then no reply will be forthcoming from that LSR for some TTL, say n. The LSR originating the echo request SHOULD try sending the echo request with TTL=n+1, n+2, ..., n+k in the hope that some transit LSR further downstream may support MPLS echo requests and reply. In such a case, the echo request for TTL>n MUST NOT have Downstream Mapping TLVs, until a reply is received with a Downstream Mapping.

5. Reliable Reply Path

One of the issues that are faced with MPLS ping is to distinguish between a failure in the forward path (the MPLS path being 'pinged') and a failure in the return path. Note that this problem exists with vanilla IP ping as well. In the case of MPLS ping, it is assumed that the IP control and data planes are reliable. However, it could be that the forwarding in the return path is via an MPLS LSP.

In this specification, we give two solutions for this problem. One is to set the Router Alert option in the MPLS echo reply. When a router sees this option, it MUST forward the packet as an IP packet. Note that this may not work if some transit LSR does not support MPLS ping.

Another option is to send the echo reply via the control plane. At present, this is defined only for RSVP-TE LSPs, and described below.

These options are controlled by the ingress LSR, using the Reply Mode in the MPLS echo request packet.

5.1. RSVP-TE Extension

To test an LSP's liveliness, an ingress LSR sends MPLS echo requests over the LSP being tested. When an egress LSR receives the message, it needs to acknowledge the ingress LSR by sending an LSP_ECHO object in a RSVP Resv message. The object has the following format:

```
      Class = LSP_ECHO  (use form 11bbbbbb for compatibility)

      C-Type = 1

    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      TimeStamp (seconds)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    TimeStamp (microseconds)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        UDP Source Port        |  Return Code  | Return Subcode|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Sequence Number is copied from the Sequence Number of the echo request. The TimeStamp is set to the time the echo request is received. The UDP Source Port is copied from the UDP source port of the MPLS echo request. The FEC is implied by the Session and the Sender Template Objects.

5.2. Operation

For the sake of brevity in the context of this document by "the control plane" we mean "the RSVP-TE component of the control plane".

Consider an LSP between an ingress LSR and an egress LSR spanning multiple LSR hops.

5.3. Procedures at the ingress LSR

One must ensure before setting the Reply Mode to "reply via the control plane" that the egress LSR supports this feature.

The ingress LSR, say X, builds an MPLS echo request as in section "Sending an MPLS Echo Request". The FEC Type must be an RVSP Session Query. X also sets the Reply Mode to "reply via the control plane".

If X does not receive an Resv message from the egress LSR that contains an LSP_ECHO object within some period of time, it declares the LSP as "down". At this point, the ingress LSR may apply the necessary procedures to fix the LSP. These may include generating a message to network management, tearing-down and re-building the LSP, and/or rerouting user traffic to a backup LSP.

To test an LSP that carries non-IP traffic, before injecting ICMP and MPLS ping messages into the LSP, the IPv4 Explicit NULL label should be prepended to such messages. The ingress and egress LSR's must follow the procedures defined in [LABEL-STACKING].

5.4. Procedures at the egress LSR

When the egress LSR receives an MPLS ping message, it follows the procedures given above. If the Reply Mode is set to "Reply via the control plane", the LSR can, based on the RSVP SESSION and SENDER_TEMPLATE objects carried in the MPLS ping message, find the corresponding LSP in its RSVP-TE database. The LSR then checks to see if the Resv message for this LSP contains an LSP_ECHO object with the same source UDP port value. If not, the LSR adds or updates the LSP_ECHO object and refreshes the Resv message.

5.5. Procedures for the intermediate LSR's

At intermediate LSRs, normal RSVP processing procedures will cause the LSP_ECHO object to be forwarded as RSVP messages are refreshed.

At the LSR's that support MPLS ping the Resv messages that carry the LSP_ECHO object MUST be delivered upstream immediately.

Note that an intermediate LSR using RSVP refresh reduction [RSVP-REFRESH], the new or changed LSP_ECHO object will cause the LSR to classify the RSVP message as a trigger message.

6. Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[LABEL-STACKING] Rosen, E., et al, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RSVP] Braden, R. (Editor), et al, "Resource ReSerVation protocol (RSVP) -- Version 1 Functional Specification," RFC 2205, September 1997.

[RSVP-REFRESH] Berger, L., et al, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[RSVP-TE] Awduche, D., et al, "RSVP-TE: Extensions to RSVP for LSP tunnels", RFC 3209, December 2001.

7. Informative References

[ICMP] Postel, J., "Internet Control Message Protocol", RFC 792.

[LDP] Andersson, L., et al, "LDP Specification", RFC 3036, January 2001.

8. Security Considerations

There are at least two approaches to attacking LSRs using the mechanisms defined here. One is a Denial of Service attack, by sending MPLS echo requests/replies to LSRs and thereby increasing their workload. The other is obfuscating the state of the MPLS data plane liveness by spoofing, hijacking, replaying or otherwise tampering with MPLS echo requests and replies.

Authentication will help reduce the number of seemingly valid MPLS echo requests, and thus cut down the Denial of Service attacks; beyond that, each LSR must protect itself.

Authentication sufficiently addresses spoofing, replay and most tampering attacks; one hopes to use some mechanism devised or suggested by the RPSec WG. It is not clear how to prevent hijacking (non-delivery) of echo requests or replies; however, if these messages are indeed hijacked, MPLS ping will report that the data plane isn't working as it should.

It doesn't seem vital (at this point) to secure the data carried in MPLS echo requests and replies, although knowledge of the state of the MPLS data plane may be considered confidential by some.

9. IANA Considerations

(To be filled in a later revision)

10. Acknowledgments

This document is the outcome of many discussions among many people, that include Manoj Leelanivas, Paul Traina, Yakov Rekhter, Der-Hwa Gan, Brook Bailey, Eric Rosen and Ina Minei.

11. Appendix

This appendix specifies non-normative aspects of detecting MPLS data plane liveness.

11.1. CR-LDP FEC

This section describes how a CR-LDP FEC can be included in an Echo Request using the following FEC subtype:

Sub-Type #	Length	Value Field
5	6	CR-LDP LSP ID

The value consists of the LSPID of the LSP being pinged. An LSPID is a four octet IPv4 address (a local address on the ingress LSR, for example, the Router ID) plus a two octet identifier that is unique per LSP on a given ingress LSR.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Ingress LSR Router ID                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Must Be Zero         |             LSP ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

11.2. Downstream Mapping for CR-LDP

If a label in a Downstream Mapping was learned via CR-LDP, the Protocol field in the Mapping TLV can use the following entry:

Protocol #	Signaling Protocol
5	CR-LDP

12. Authors' Addresses

Kireeti Kompella
Nischal Sheth
Juniper Networks
1194 N.Mathilda Ave
Sunnyvale, CA 94089
e-mail: kireeti@juniper.net
e-mail: nsheth@juniper.net

Ping Pan
Ciena
10480 Ridgeview Court
Cupertino, CA 95014
e-mail: ppan@ciena.com
phone: +1 408.366.4700

Dave Cooper
Global Crossing
960 Hamlin Court
Sunnyvale, CA 94089
email: dcooper@gblx.net
phone: +1 916.415.0437

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
e-mail: swallow@cisco.com
phone: +1 978.497.8143

Sanjay Wadhwa
Juniper Networks
10 Technology Park Drive
Westford, MA 01886-3146
email: swadhwa@unispherenetworks.com
phone: +1 978.589.0697

Ronald P. Bonica
WorldCom
22001 Loudoun County Pkwy
Ashburn, Virginia, 20147
email: ronald.p.bonica@wcom.com
phone: +1 703.886.1681

13. Intellectual Property Rights Notices

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.