Network Working Group
Internet-Draft
Updates: RFC4379
(if approved)
Intended status: Standards Track
Expires: May 18, 2008

N. Bahadur, Ed.
K. Kompella, Ed.
Juniper Networks, Inc.
G. Swallow, Ed.
Cisco Systems
November 15, 2007

Mechanism for performing LSP-Ping over MPLS tunnels
draft-nitinb-lsp-ping-over-mpls-tunnel-01

Status of this Memo

Copyright Notice

Abstract

This document describes methods for performing lsp-ping traceroute over mpls tunnels. The techniques outlined in RFC 4379 fail to perform correct traceroute validation and path discovery for a LSP that goes over other mpls tunnels. This document describes new procedures that can be used in conjunction with the standard procedures described in RFC 4379 to trace such LSPs.

Table of Contents

1. Introduction

This documents describes methods for performing lsp-ping traceroute over mpls tunnels. The techniques outlined in [1] outline a traceroute mechanism that includes FEC validation and ECMP path discovery. Those mechanisms are insufficient and do not provide details in case the FEC being traced traverses one or more mpls tunnels. This document uses the existing definitions of [1] to define a mechanism using which a traceroute request can correctly traverse mpls tunnels with proper FEC and label validations.

1.1. Conventions used in this document

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 [2].

2. Motivation

A LSP-Ping traceroute may cross multiple mpls tunnels en-route the destination. Let us consider a simple case.

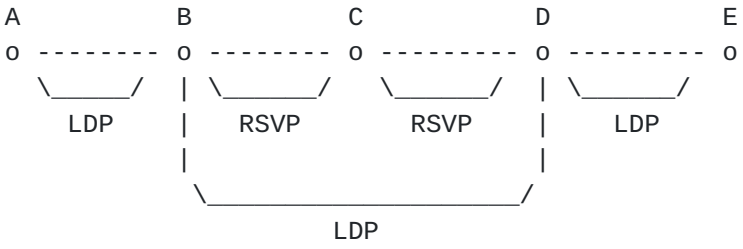

Figure 1: LDP over RSVP tunnel

When a traceroute is initiated from router A, router B returns downstream mapping information for node C in the echo-response. The next echo request reaches router C with a LDP FEC. Node C is a pure RSVP node and does not run LDP. Node C will receive the packet with 2 labels but only 1 FEC in the Target FEC stack. Consequently, node C will be unable to perform FEC complete validation. It will let the trace continue by just providing next-hop information based on incoming label, and by looking up the forwarding state associated with that label. However, ignoring FEC validation defeats the purpose of control plane validatations. The echo request should contain sufficient information to allow node C to perform FEC validations to catch any misrouted echo-requests.

The above problem can be extended for a generic case of tunnel over tunnel or multiple tunnels (e.g. B-C can be a separate RSVP tunnel and C-D can be a separate RSVP tunnel). The problem of FEC validation for tunnels can be solved if the transit routers (router B in the above example) provide some hint or information to the ingress regarding the start of a new tunnel.

Stitched LSPs involve 2 or more LSP segments stitched together. The LSP segments can be signaled using the same or different signaling protocols. In order to perform an end-to-end trace of a stitched LSP, the ingress needs to know FEC information regarding each of the stitched LSP segments. For example, conside the figure below.

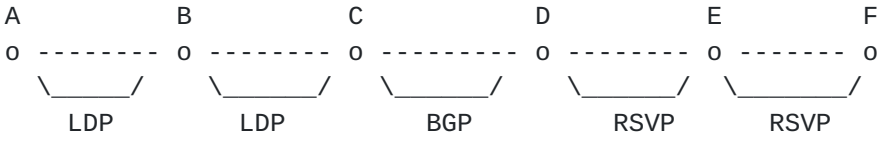

Figure 2: Stitched LSP

Consider ingress (A) tracing end-to-end LSP A--F. When an echo request reaches router C, there is a FEC change happening at router C. With current lsp-ping mechanisms, there is no way to convey this information to A. Consequently, when the next echo request reaches router D, router D will know nothing about the LDP FEC that A is trying to trace.

Thus, the procedures outlined [1] do not make it possible for the ingress node to:

1. Know that tunneling has occured
2. Trace the path of the tunnel
3. Trace the path of stitched LSPs

3. Packet format

3.1. Introduction

In many cases there has been a need to associate additional data in the lsping echo response. In most cases, the additional data needs to be associated on a per downstream neighbor basis. Currently, the echo response contains 1 downstream map TLV (DSMAP) per downstream neighbor. But the DSMAP format is not extensible and hence it's not possible to associate more information with a downstream neighbor. This draft defines a new extensible format for the DSMAP and provides

mechanisms for solving the tunneled lsp-ping problem using the new format. In summary, the draft makes the following tlv changes:

- o Addition of new Downstream Detailed Mapping TLV (DDMAP).
- o Deprecation of existing Downstream Mapping TLV.
- o Addition of Downstream FEC Stack Change Sub-TLV to DDMAP.

3.2. Downstream Detailed Mapping TLV

A new TLV has been added to the mandatory range of TLVs. The TLV type is pending IANA allocation.

Type #	Value Field
TBD	Downstream detailed mapping

Figure 3

The Downstream Detailed Mapping object is a TLV that MAY be included in an echo request message. Only one Downstream Detailed Mapping object may appear in an echo request. The presence of a Downstream Mapping object is a request that Downstream Detailed Mapping objects be included in the echo reply. If the replying router is the destination of the FEC, then a Downstream Detailed Mapping TLV SHOULD NOT be included in the echo reply. Otherwise the replying router SHOULD include a Downstream Detailed Mapping object for each interface over which this FEC could be forwarded.

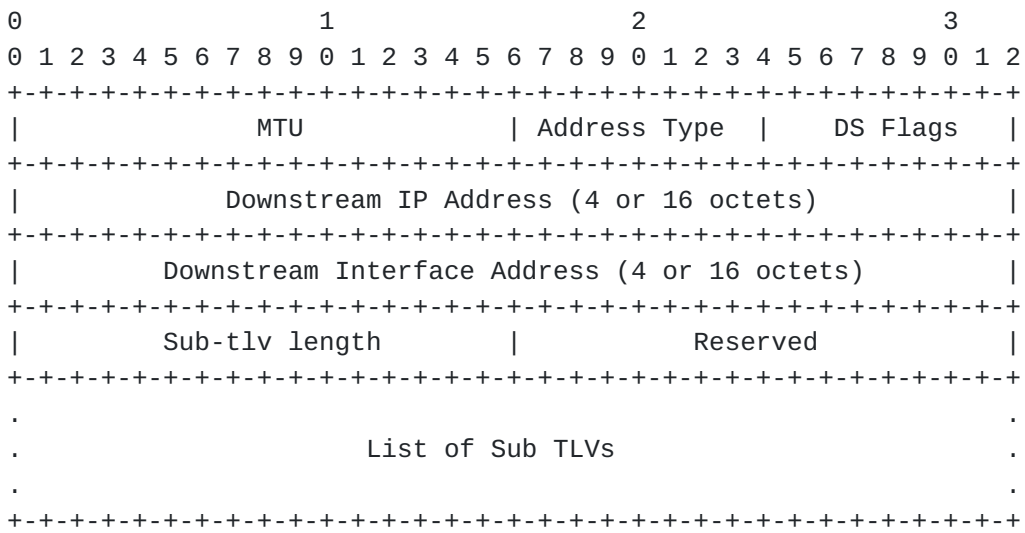

Figure 4: Downstream Detailed Mapping TLV

The Downstream Detailed Mapping TLV format is derived from the Downstream Mapping TLV format. The key change is that variable length and optional fields have been coverted into sub-TLVs. The fields have the same use and meaning as in [1]. The newly added sub-TLVs and their fields are as described below.

Sub-tlv length
Total length in bytes of the sub-tlvs associated with this TLV.

Sub-Type	Value Field
TBD	Multipath data
TBD	Label stack
TBD	FEC Stack change

Figure 5: Downstream Detailed Mapping Sub-TLV List

3.2.1. Multipath data sub-TLV

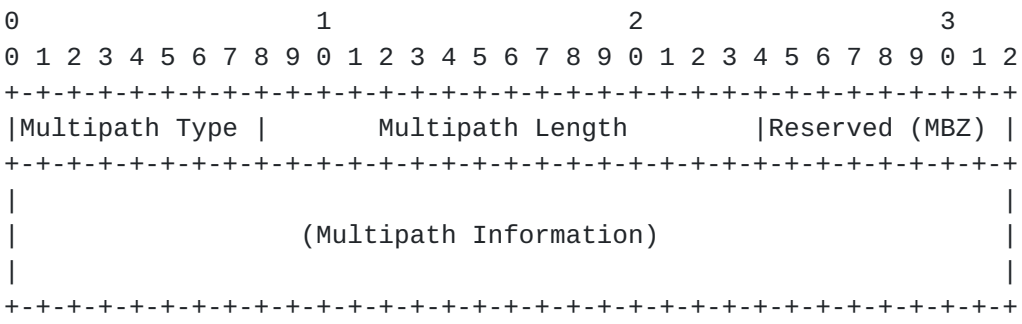

Figure 6: Multipath Sub-TLV

The multipath data sub-TLV includes information multipath information. The TLV fields and their usage is as defined in [1].

3.2.2. Label stack sub-TLV

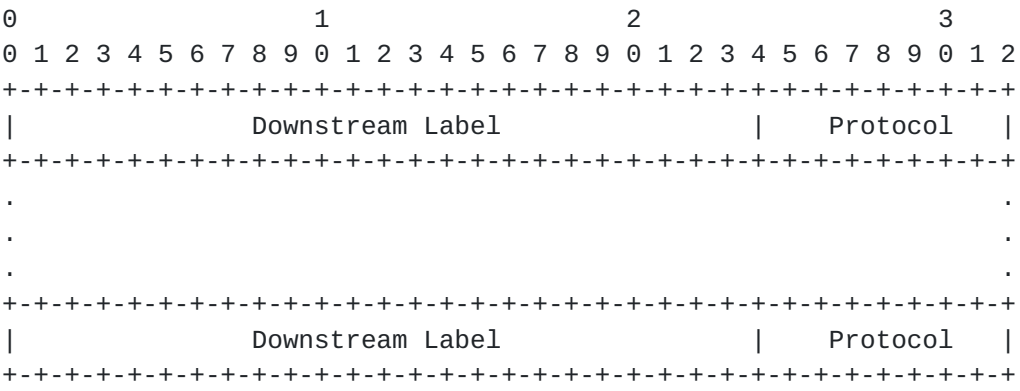

Figure 7: Label Stack Sub-TLV

The Label stack sub-TLV contains the set of labels in the label stack as it would have appeared if this router were forwarding the packet through this interface. Any Implicit Null labels are explicitly included. The number of labels present in the sub-TLV is determined based on the sub-TLV data length. Labels are treated as numbers, i.e., they are right justified in the field. The label format and protocol type are as defined in [1]. When the Detailed Downstream Mapping TLV in sent in the echo response, this sub-TLV MUST be included.

3.2.3. Stack change sub-TLV

A router SHOULD include the the FEC Stack change sub-TLV when the downstream node in the echo response has a different FEC stack than the FEC stack received in the echo request. One ore more FEC Stack change sub-TLVs MAY be present in the Downstream Detailed Mapping TLV. The format is as below.

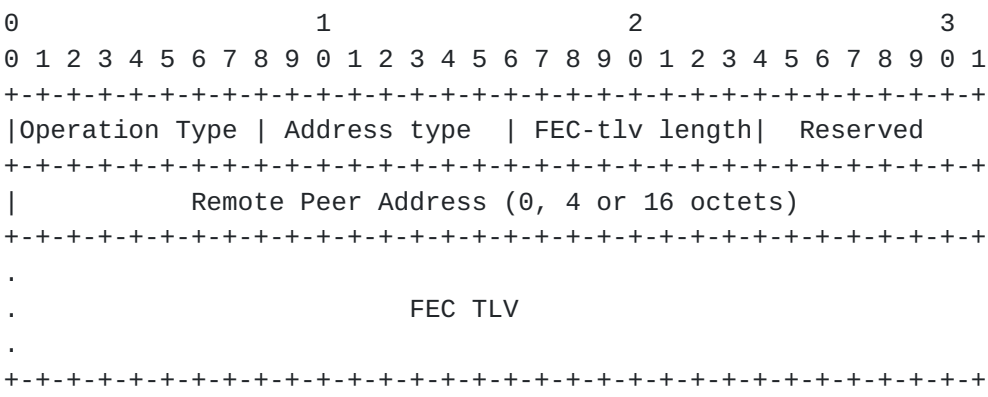

Figure 8: Stack Change Sub-TLV

Operation Type

The operation type specifies the action associated with the FEC change. The following operation types are defined.

Type #	Operation
1	Push
2	Pop

Operation Type Values

A FEC Stack change sub-TLV containing a PUSH operation MUST NOT be followed by a FEC Stack change sub-TLV containing a POP operation. One or more POP operations MAY be followed by one or more PUSH operations. One FEC Stack change sub-TLV MUST be included per FEC change. For example, if 2 labels are going to be pushed, then 1 FEC change sub-TLV MUST be included for each FEC. A FEC Swap operation is to be simulated by including a POP type FEC change sub-TLV followed by a PUSH type FEC change sub-TLV.
A Downstream detailed mapping TLV containing only 1 FEC change sub-TLV with Pop operation is equivalent to EGRESS_OK for the outermost FEC in the FEC stack. The ingress router performing the lsp trace MUST treat such a case as an EGRESS_OK for the outermost FEC.

FEC tlv Length

Length in bytes of the FEC TLV.

Address Type

The Address Type indicates the remote peer's address type. The Address Type is set to one of the following values. The peer address length is determined based on the address type. The address type MAY be different from the address type included in the Downstream Detailed Mapping TLV. This can happen in case the LSP goes over a tunnel of a different address family. The address type MAY be set to Unspecified if the peer-address is either unavailable or the transit router does not wish it provide it for security or administrative reasons.

Type #	Address Type	Address length
0	Unspecified	0
1	IPv4	4
2	IPv6	16

Figure 10: Remote peer address type

Remote peer address

The remote peer address specifies the remote peer which is the next-hop for the FEC being currently traced. E.g. In the LDP over RSVP case Figure 1, router B would respond back with the address of router D as the remote peer address for the LDP FEC being traced. This allows the ingress node to provide helpful information regarding FEC peers. If the operation type is PUSH, the remote peer address is the address of the peer from which the FEC was learned. If the operation type is POP, the remote peer address MAY not be set to Unspecified. For upstream assigned labels [3], an operation type of POP will have a remote peer address (the upstream node that assigned the label) and this SHOULD be included in the FEC change sub-TLV.

FEC TLV

The FEC TLV is present only when FEC-tlv length field is non-zero. The FEC TLV specifies the FEC associated with the FEC stack change operation. This TLV MAY be included when the operation type is POP. It SHOULD be included when the operation type is PUSH. The FEC TLV contains exactly 1 FEC from the list of FECs specified in [1]. A NIL FEC MAY be associated with a PUSH operation if the responding router wishes to hide the details of the FEC being pushed.

3.3. Deprecation of Downstream Mapping TLV

The Downstream Mapping TLV has been deprecated. LSP-ping procedures should now use the Downstream Detailed Mapping TLV. Detailed procedures regarding interoperability between the deprecatd TLV and the new tlv are specified in Section 4.3.

4. Performing lsp-ping traceroute on tunnels

This section describes the procedures to be followed by an ingress node and transit nodes when performing lsp-ping traceroute over mpls tunnels.

4.1. Transit node procedure

4.1.1. Addition of a new tunnel

A transit node (Figure 1) knows when that the FEC being traced is going to enter a tunnel at that node. Thus, it knows about the new outer FEC. All transit nodes that are the origination point of a new tunnel SHOULD add the a FEC Stack change sub-TLV (Section 3.2.3) to the Downstream Detailed Mapping TLV Figure 4 in the echo-response. The transit node SHOULD add 1 FEC Stack change sub-TLV of operation type PUSH, per new tunnel being originated at the transit node.

A transit node that sends a Downstream FEC Stack change sub-TLV in the echo response SHOULD fill the address of the remote peer; which is the peer of the current LSP being traced. If the transit node does not know the address of the remote peer, it MAY leave it as unspecified.

If the transit node wishes to hide the nature of the tunnel from the ingress of the echo-request, then it MAY not want to send details about the new tunnel FEC to the ingress. In such a case, the transit node SHOULD use the NIL FEC. The echo response would then contain a FEC Stack change sub-TLV with operation type PUSH and a NIL FEC. The value of the label in the NIL FEC MUST be set to zero. The remote peer address length MUST be set to 0 and the remote peer address type MUST be set to Unspecified. The transit node SHOULD add 1 FEC Stack change sub-TLVs of operation type PUSH, per new tunnel being originated at the transit node.

4.1.2. Transition between tunnels

Figure 11: Stitched LSPs

In the above figure, we have 3 seperate LSP segments stitched at C and D. Node C SHOULD include 2 FEC Stack change sub-TLVs. One with a POP operation for the LDP FEC and one for the PUSH operation for the BGP FEC. Similarly, node D SHOULD include 2 FEC Stack change sub-TLVs, one with a POP operation for the BGP FEC and one with a PUSH operation for the RSVP FEC.

If node C wishes to perform FEC hiding, it SHOULD respond back with 2 FEC Stack change sub-TLVs. One POP followed by 1 PUSH. The POP operation MAY either not include the FEC TLV (by setting FEC-tlv length to 0) or set the FEC TLV to contain the LDP FEC. The PUSH operation SHOULD have the FEC TLV contain the NIL FEC.

If node C performs FEC hiding and node D also performs FEC hiding, then node D MAY choose to not send any FEC change sub-TLVs in the echo response since the number of labels has not changed (for the downstream of node D) and the FEC type also has not changed (NIL FEC). If node D performs FEC hiding, then node F will respond as EGRESS_OK for the NIL FEC. The ingress (node A) will know that EGRESS_OK corresponds to the end-to-end LSP.

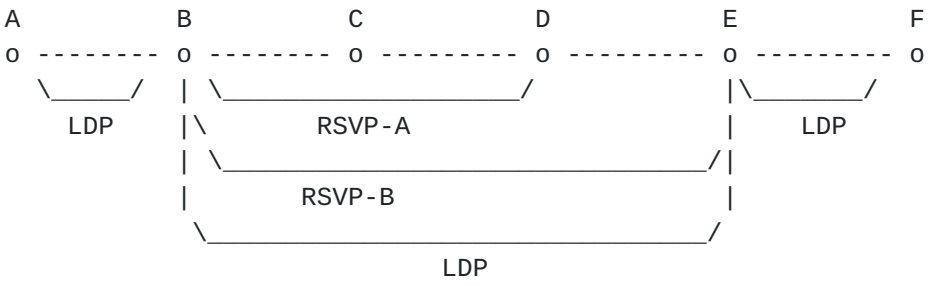

Figure 12: Hierarchical LSPs

In the above figure, the following sequence of FEC change sub-TLVs will be performed

Node B:

Respond with 2 FEC change sub-TLVs: Push RSVP-B, Push RSVP-A.

Node D:

Respond with EGRESS_OK when RSVP-A is top of FEC stack. Downstream information for node E when echo request contains RSVP-B as top of FEC stack.

If node B is performing tunnel hiding, then:

Node B:

Respond with 2 FEC change sub-TLVs: PUSH NIL-FEC, PUSH NIL-FEC.

Node D:

Respond with either EGRESS_OK (if D can co-relate that the NIL-FEC corresponds to RSVP-A which is terminating at D) or respond with FEC change sub-TLV: POP (since D knows that number of labels towards next-hop is decreasing).

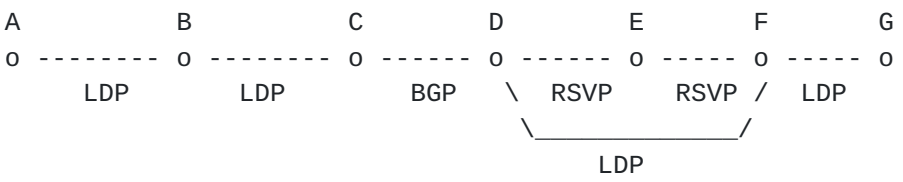

Figure 13: Stitched hierarchical LSPs

In the above case, node D will send 3 FEC change sub-TLVs. One POP (for the BGP FEC) followed by 2 PUSHes (one for LDP and one for RSVP).

4.2. Ingress node procedure

It is the responsibility of an ingress node to understand tunnel within tunnel semantics and lsp stitching semantics when performing a lsp traceroute. This section describes the ingress node procedure based on the kind of response an ingress node receives from a transit node.

4.2.1. Processing Downstream Detailed Mapping TLV

Downstream Detailed Mapping TLV should be processed in procedures similar to those of Downstream Mapping TLV, defined in Section 4.4 of [1]

4.2.1.1. Stack Change sub-TLV not present

This would be the default behavior as described in [1]. The ingress node MUST perform echo response processing as per the procedures in [1].

4.2.1.2. Stack Change sub-TLV(s) present

If one or more FEC Stack change sub-TLVs (Section 3.2.3) are received in the echo response, the ingress node SHOULD process them and perform some validation.

The FEC stack changes are associated with a downstream neighbor and along a particular path of the LSP. Consequently, the ingress will need to maintain a FEC-stack per path being traced (in case of multipath). All changes to the FEC stack resulting from the processing of FEC Stack change sub-TLV(s) should be applied only for the path along a given downstream neighbor. The following algorithm should be followed for processing FEC Stack change sub-TLVs.

```
    push_seen = FALSE
    fec_stack_depth = current-depth-of-fec-stack-being-traced
    saved_fec_stack = current_fec_stack

    while (sub-tlv = get_next_sub_tlv(downstream_detailed_map_tlv))

        if (sub-tlv == NULL) break

        if (sub-tlv.type == FEC-Stack-Change) {

            if (sub-tlv.operation == POP) {
                if (push_seen) {
                    Drop the echo response
                    current_fec_stack = saved_fec_stack
                    return
                }

                if (fec_stack_depth == 0) {
                    Drop the echo response
                    current_fec_stack = saved_fec_stack
                    return
                }

                Pop FEC from FEC stack being traced
                fec_stack_depth--;
            }

            if (sub-tlv.operation == PUSH) {
                push_seen = 1
                Push FEC on FEC stack being traced
                fec_stack_depth++;
            }
        }
     }


     if (fec_stack_depth == 0) {
         Drop the echo response
         current_fec_stack = saved_fec_stack
         return
     }
```

Figure 14: FEC Stack Change Sub-TLV Processing Guideline

The next echo request along the same path should use the modified FEC stack obtained after processing the FEC Stack change sub-TLVs. A non-NIL FEC guarantees that the next echo request along the same path

will have the Downstream Detailed Mapping TLV validated for IP address, Interface address and label stack mismatches.

If the top of the FEC stack is a NIL FEC and the echo response does not contain any FEC Stack change sub-TLV, then it does not necessarily mean that the LSP has not started traversing a different tunnel. It could be that the LSP associated with the NIL FEC terminated at a transit node and at the same time a new LSP started at the same transit node. The NIL FEC would now be associated with the new LSP (and the ingress has no way of knowing this). Thus, it is not possible to build an accurate hierarchical LSP topology if a traceroute contains NIL FECs.

4.2.2. Modifications to handling to EGRESS_OK responses.

The procedures above allow the addition of new FECs to the original FEC being traced. Consequently, the EGRESS_OK response from a downstream node may not necessarily be for the FEC being traced. It could be for one of the new FECs that was added. On receipt of an EGRESS_OK response, the ingress should check if the Target FEC sent to the node that just responded was the base FEC that was being traced. If it was not, then it should pop the an entry from the Target FEC stack and resend the request with the same TTL (as previously sent). The process of popping a FEC is to be repeated until either the ingress receives a non-EGRESS_OK response or until all the additional FECs added to the FEC stack have already been popped. Using EGRESS_OK responses, an ingress can build a map of the hierarchical LSP structure traversed by a given FEC.

4.3. Handling deprecated Downstream Mapping TLV

The Downstream Mapping TLV has been deprecated. Applications should now use the Downstream Detailed Mapping TLV. The following procedures SHOULD be used for backward compatibility with routers that do not support the Downstream Detailed Mapping TLV.

- The Downstream Mapping TLV and the Downstream Detailed Mapping TLV MUST never be sent together in the same echo request or in the same echo response.
- If the echo request contains a Downstream Detailed Mapping TLV and the corresponding echo response contains an error code of 2 (one or more of the TLVs was not understood), then the sender of the echo request MAY resend the echo request with the Downstream Mapping TLV (instead of the Downstream Detailed Mapping TLV). In cases where the a detailed response is needed, the sender can choose to ignore the router that does not support the Downstream Detailed Mapping TLV.

- o If the echo request contains a Downstream Mapping TLV, then a Downstream Detailed Mapping TLV MUST NOT be sent in the echo response. This is to handle the case that the sender of the echo request does not support the new tlv.
- o If echo request forwarding is in use; such that the echo request is processed at an intermediate router and then forwarded on; then the intermediate router is responsible for making sure that the TLVs being used among the ingress, intermediate and destination are consistent. The intermediate router MUST NOT forward an echo request or an echo response containing a Downstream Detailed Mapping TLV if it itself does not support that TLV.

5. Security Considerations

Tracing inside a tunnel might have some security implications. There are different ways to prevent tracing tunnel details.

1. If one wants to prevent tracing inside a tunnel, one can hide the outer MPLS tunnel by not propagating the MPLS TTL into the outer tunnel (at the start of the outer tunnel). By doing this, lsp-ping packets will not expire in the outer tunnel and the outer tunnel will not get traced. TTL hiding can be imposed on a per LSP basis as need be.
2. If one doesn't wish to expose the details of the new outer LSP, then the NIL FEC can be used to hide those details. Using the NIL FEC ensures that the trace progresses without false negatives and all transit nodes (of the new outer tunnel) perform some minimal validations on the received echo requests.

In inter-AS (autonomous system) scenarios, information regarding the LSP FEC change(s) SHOULD not be passed across domains. A NIL FEC MAY be used to make the trace go through without false positives. An ASBR (autonomous system border router) may choose to intercept all echo requests and echo responses and change them to hide FEC information from other domains. Detailed operation regarding the same is outside the scope of this document. Passing of FEC change information between domains MAY be done if the two AS domains belong to the same provider/organization.

Other security considerations, as discussed in [1] are also applicable to this document.

6. IANA Considerations

This document introduces a new Downstream Detailed Mapping TLV. It is requested that IANA assign a TLV type in the range of 0-32767 from

the TLV type registry created in [1].

It is requested that IANA create a new registry for the Sub-Type field of Downstream Detailed Mapping TLV. The valid range for this is 0-65535. Assignments in the range 0-32767 are made via Standards Action as defined in ; assignments in the range 32768-64511 are made via Expert Review (see below); values in the range 64512-65535 are for Vendor Private Use, and MUST NOT be allocated. If a sub-TLV has a Type that falls in the range for Vendor Private Use, the Length MUST be at least 4, and the first four octets MUST be that vendor's SMI Enterprise Code, in network octet order. The rest of the Value field is private to the vendor.

It is requested that IANA assign a sub-TLV types from the 0-32767 range for the sub-TLVs defined in Figure 5.

7. Acknowledgements

The authors would like to thank Yakov Rekhter and Adrian Farrel for their suggestions on the draft.

8. References

8.1. Normative References

[1] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

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

8.2. Informative References

[3] Aggarwal, R., "MPLS Upstream Label Assignment and Context-Specific Label Space", draft-ietf-mpls-upstream-label-03 (work in progress), November 2007.

Authors' Addresses

Nitin Bahadur (editor)
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
US

Phone: +1 408 745 2000
Email: nitinb@juniper.net
URI: www.juniper.net

Kireeti Kompella (editor)
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
US

Phone: +1 408 745 2000
Email: kireeti@juniper.net
URI: www.juniper.net

George Swallow (editor)
Cisco Systems
1414 Massachusetts Ave
Boxborough, MA 01719
US

Email: swallow@cisco.com
URI: www.cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).