        MPLS Transport Profile Lock Instruct and Loopback Functions

Abstract

   Two useful Operations, Administration, and Maintenance (OAM)
   functions in a transport network are "lock" and "loopback". The lock
   function enables an operator to lock a transport path such that it
   does not carry client traffic, but can continue to carry OAM messages
   and may carry test traffic. The loopback function allows an operator
   to set a specific node on the transport path into loopback mode such
   that it returns all received data.

   This document specifies the lock function for MPLS networks and
   describes how the loopback function operates in MPLS networks.

   This document updates Sections 7.1.1 and 7.1.2 of RFC 6371.

Copyright Notice

1. Introduction

   Two useful Operations, Administration, and Maintenance (OAM)
   functions in a transport network are "lock" and "loopback". This
   document discusses these functions in the context of MPLS networks.

   - The lock function enables an operator to lock a transport path
      such that it does not carry client traffic. As per RFC 5860 [1],
      lock is an administrative state in which it is expected that no
      client traffic may be carried. However, test traffic and OAM
      messages can still be mapped onto the locked transport path. The
      lock function may be applied to the Label Switched Paths (LSPs),
      Pseudowires (PWs) (including multi-segment Pseudowires) (MS-PWs),
      and bidirectional MPLS Sections as defined in RFC 5960 [9]).

   - The loopback function allows an operator to set a specific node on
      a transport path into loopback mode such that it returns all
      received data. Loopback can be applied at a Maintenance Entity
      Group End Point (MEP) or a Maintenance Entity Group Intermediate
      Point (MIP) on a co-routed bidirectional LSP, on a PW, or on a
      bidirectional MPLS Section. It can also be applied at a MEP on an
      associated bidirectional LSP.

      Loopback is used to test the integrity of the transport path to
      and from the node that is performing loopback. It requires that
      the transport path be locked and that a MEP on the transport path
      send test data that it also validates on receipt.

   This document specifies the lock function for MPLS networks and
   describes how the loopback function operates in MPLS networks.

1.1. Updates RFC 6371

This document updates Sections 7.1.1 and 7.1.2 of RFC 6371 [6].

The framework in RFC 6371 makes the assumption that the Lock Instruct message is used to independently enable locking and requires a response message.

The mechanism defined in this document requires that when a lock instruction is sent by management to both ends of the locked transport path, the Lock Instruct message does not require a response.

2. Terminology and Conventions

2.1. 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 RFC 2119 [2].

2.2. Acronyms and Terms

ACH: Associated Channel Header

LI: Lock Instruct

MEG: Maintenance Entity Group

MEP: Maintenance Entity Group End Point

MIP: Maintenance Entity Group Intermediate Point

MPLS-TP: MPLS Transport Profile

NMS: Network Management System

TLV: Type Length Value

Transport path: MPLS-TP LSP or PW

TTL: Time To Live

3. Lock Function

Lock is used to request that a MEP take a transport path out of service for administrative reasons. For example, Lock can be used to allow some form of maintenance to be done for a transport path. Lock

is also a prerequisite of the loopback function described in Section
4. The NMS or a management process initiates a Lock by sending a
Lock command to a MEP. The MEP takes the transport path out of
service, that is, it stops injecting or forwarding traffic onto the
transport path.

To properly lock a transport path (for example, to ensure that a
loopback test can be performed), both directions of the transport
path must be taken out of service; therefore, a Lock command is sent
to the MEPs at both ends of the path. This ensures that no traffic
is sent in either direction. Thus, the lock function can be realized
entirely using the management plane.

However, dispatch of messages in the management plane to the two MEPs
may present coordination challenges. It is desirable that the lock
be achieved in a coordinated way within a tight window, and this may
be difficult with a busy management plane. In order to provide
additional coordination, an LI OAM message can also be sent. A MEP
locks a transport path when it receives a command from a management
process or when it receives an LI message as described in Section 6.

This document defines an LI message for MPLS OAM. The LI message is
based on a new ACH Type as well as an existing TLV. This is a common
mechanism applicable to lock LSPs, PWs, and bidirectional MPLS
Sections.

4. Loopback Function

This section provides a description of the loopback function within
an MPLS network. This function is achieved through management
commands, so there is no protocol specification necessary. However,
the loopback function is dependent on the lock function, so it is
appropriate to describe it in this document.

The loopback function is used to test the integrity of a transport
path from a MEP up any other node in the same MEG. This is achieved
by setting the target node into loopback mode, and transmitting a
pattern of test data from the MEP. The target node loops all
received data back toward the originator, and the MEP extracts the
test data and compares it with what it sent.

Loopback is a function that enables a receiving MEP or MIP to return
traffic to the sending MEP when in the loopback state. This state
corresponds to the situation where, at a given node, a forwarding
plane loop is configured, and the incoming direction of a transport
path is cross-connected to the outgoing reverse direction.
Therefore, except in the case of early TTL expiry, traffic sent by
the source will be received by that source.

Data-plane loopback is an out-of-service function, as required in
Section 2.2.5 of RFC 5860 [1]. This function loops back all traffic
(including user data and OAM). The traffic can be originated from
one internal point at the ingress of a transport path within an
interface or inserted from an input port of an interface using
external test equipment. The traffic is looped back unmodified
(other than normal per-hop processing such as TTL decrement) in the
direction of the point of origin by an interface at either an
intermediate node or a terminating node.

It should be noted that the data-plane loopback function itself is
applied to data-plane loopback points residing on different
interfaces from MIPs/MEPs. All traffic (including both payload and
OAM) received on the looped back interface is sent on the reverse
direction of the transport path.

For data-plane loopback at an intermediate point in a transport path,
the loopback needs to be configured to occur at either the ingress or
egress interface. This is done using management.

The management plane can be used to configure the loopback function.
The management plane must ensure that the two MEPs are locked before
it requests setting MEP or MIP in the loopback state.

The nature of test data and the use of loopback traffic to measure
packet loss, delay, and delay variation are outside the scope of this
document.

4.1. Operational Prerequisites

Obviously, for the loopback function to operate, there are several
prerequisites:

- There must be a return path, so the transport path under test must
  be bidirectional.

- The node in loopback mode must be on both the forward and return
  paths. This is possible for all MEPs and MIPs on a co-routed
  bidirectional LSP, on a PW, or on a bidirectional MPLS Section,
  but it is only possible for MEPs on associated bidirectional LSPs.

- The transport path cannot deliver client data when one of its
  nodes is in loopback mode, so it is important that the transport
  path be locked before loopback is enabled.

- Management-plane coordination between the node in loopback mode and the MEP sending test data is required. The MEP must not send test data until loopback has been properly configured because this would result in the test data continuing toward the destination.

- The TTL of the test packets must be set sufficiently large to account for both directions of the transport path under test; otherwise, the packets will not be returned to the originating MEP.

- OAM messages intended for delivery to nodes along the transport path under test can be delivered by correct TTL expiry. However, OAM messages cannot be delivered to points beyond the loopback node until the loopback condition is lifted.

5. Lock Instruct Message

5.1. Message Identification

The Lock Instruct message is carried in the Generic ACH described in [4]. It is identified by a new PW ACH Type of 0x0026.

5.2. LI Message Format

The format of an LI message is shown below.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Vers  | Reserved                              | Refresh Timer |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    MEP Source ID TLV                          |
~                                                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
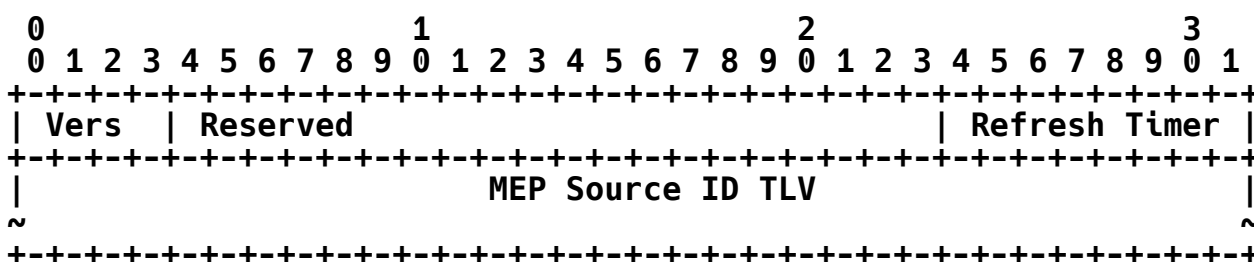
Figure 1: MPLS Lock Instruct Message Format

Version: 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 the message. These changes include any syntactic or semantic changes made to any of the fixed fields, or to any Type-Length-Value (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 an optional TLV or sub-TLV is added.)

Reserved: The Reserved field MUST be set to zero on transmission and SHOULD be ignored on receipt.

Refresh Timer: The Refresh Timer is the maximum time between
successive LI messages specified in seconds. The default value is 1.
The value 0 is not permitted. When a lock is applied, a refresh
timer is chosen. This value MUST NOT be changed for the duration of
that lock. A node receiving an LI message with a changed refresh
timer MAY ignore the new value and continue to apply the old value.

MEP Source ID TLV: This is one of the three MEP Source ID TLVs
defined in [3] and identifies the MEP that originated the LI message.

6. Operation of the Lock Function

6.1. Locking a Transport Path

When a MEP receives a Lock command from an NMS or through some other
management process, it MUST take the transport path out of service.
That is, it MUST stop injecting or forwarding traffic onto the LSP,
PW, or bidirectional Section that has been locked.

If rapid coordination of lock state is to be achieved (as described
in Section 3) then as soon as the transport path has been locked, the
MEP MUST send an LI message targeting the MEP at the other end of the
locked transport path. In this case, the source MEP MUST set the
Refresh Timer value in the LI message and MUST retransmit the LI
message at the frequency indicated by the value set.

When locking a transport path, the NMS or management process is
required to send a Lock command to both ends of the transport path.
Thus, a MEP may receive either the management command or an LI
message first. A MEP MUST take the transport path out of service
immediately in either case, but sends LI messages itself after it has
received a management Lock command. Thus, a MEP is locked if either
Lock was requested by management (and, as a result, the MEP is
sending LI messages) or it is receiving LI messages from the remote
MEP.

Note that a MEP that receives an LI message MUST identify the correct
transport path and validate the message. The label stack on the
received message is used to identify the transport path to be locked:

- If no matching label binding exists, then there is no
  corresponding transport path and the received LI message is in
  error.

- If the transport path can be identified, but there is no return
  path (for example, the transport path was unidirectional) then
  (obviously) the receiving MEP cannot apply a lock to the return
  path.

   - If the transport path is suitable for locking but the Source MEP-
      ID identifies an unexpected MEP for the MEG to which the receiving
      MEP belongs, the received LI message is in error.

   When an errored LI message is received, the receiving MEP MUST NOT
   apply a lock. A MEP receiving errored LI messages SHOULD perform
   local diagnostic actions (such as counting the messages) and MAY log
   the messages.

   A MEP keeps a transport path locked as long as it is either receiving
   the periodic LI messages or has an in-force Lock command from
   management (see Section 6.2 for an explanation of unlocking a MEP).
   Note that in some scenarios (such as the use of loopback as described
   in Section 4), LI messages will not continue to be delivered on a
   locked transport path. This is why a transport path is considered
   locked while there is an in-force Lock command from a management
   process regardless of whether LI messages are being received.

6.2. Unlocking a Transport Path

   Unlock is used to request that a MEP bring the previously locked
   transport path back in service.

   When a MEP receives an Unlock command from a management process, it
   MUST cease sending LI messages. However, as described in Section
   6.1, if the MEP is still receiving LI messages, the transport path
   MUST remain out of service. Thus, to unlock a transport path, the
   management process has to send an Unlock command to the MEPs at both
   ends.

   When a MEP has been unlocked and has not received an LI message for a
   multiple of 3.5 times the Refresh Timer on the LI message (or has
   never received an LI message), the MEP unlocks the transport path and
   puts it back into service.

7. Security Considerations

   MPLS-TP is a subset of MPLS and builds upon many of the aspects of
   the security model of MPLS. MPLS networks make the assumption that
   it is very hard to inject traffic into a network, and it is equally
   hard to cause traffic to be directed outside the network. For more
   information on the generic aspects of MPLS security, see [7].

   This document describes a protocol carried in the G-ACh [4], so it is
   dependent on the security of the G-ACh, itself. The G-ACh is a
   generalization of the Pseudowire Associated Channel defined in [8].
   Thus, this document relies heavily on the security mechanisms
   provided for the Associated Channel as described in [4] and [8].

A specific concern for the G-ACh is that is can be used to provide a covert channel. This problem is wider than the scope of this document and does not need to be addressed here, but it should be noted that the channel provides end-to-end connectivity and SHOULD NOT be policed by transit nodes. Thus, there is no simple way to prevent traffic from being carried in the G-ACh between consenting nodes.

A good discussion of the data-plane security of an associated channel may be found in [5]. That document also describes some mitigation techniques.

It should be noted that the G-ACh is essentially connection-oriented, so injection or modification of control messages specified in this document requires the subversion of a transit node. Such subversion is generally considered hard in MPLS networks, and impossible to protect against at the protocol level. Management-level techniques are more appropriate.

8. IANA Considerations

8.1. Pseudowire Associated Channel Type

LI OAM requires a unique Associated Channel Type that has been assigned by IANA in the "Pseudowire Associated Channel Types" registry.

Registry:

Value	Description	TLV Follows	Reference
0x0026	LI	No	[RFC6435]

9. Acknowledgements

The authors would like to thank Loa Andersson, Yoshinori Koike, Alessandro D'Alessandro Gerardo, Shahram Davari, Greg Mirsky, Yaacov Weingarten, Liu Guoman, Matthew Bocci, Adrian Farrel, and Jia He for their valuable comments.

10. References

10.1. Normative References

[1] Vigoureux, M., Ed., Ward, D., Ed., and M. Betts, Ed.,
    "Requirements for Operations, Administration, and Maintenance
    (OAM) in MPLS Transport Networks", RFC 5860, May 2010.

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

[3] Allan, D., Ed., Swallow, G., Ed., and J. Drake, Ed., "Proactive
    Connectivity Verification, Continuity Check, and Remote Defect
    Indication for the MPLS Transport Profile", RFC 6428, November
    2011.

[4] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS
    Generic Associated Channel", RFC 5586, June 2009.

[5] Nadeau, T., Ed., and C. Pignataro, Ed., "Pseudowire Virtual
    Circuit Connectivity Verification (VCCV): A Control Channel for
    Pseudowires", RFC 5085, December 2007.

[6] Busi, I., Ed., and D. Allan, Ed., "Operations, Administration,
    and Maintenance Framework for MPLS-Based Transport Networks", RFC
    6371, September 2011.

10.2. Informative References

[7] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks",
    RFC 5920, July 2010.

[8] Bryant, S., Swallow, G., Martini, L., and D. McPherson,
    "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use
    over an MPLS PSN", RFC 4385, February 2006.

[9] Frost, D., Ed., Bryant, S., Ed., and M. Bocci, Ed., "MPLS
    Transport Profile Data Plane Architecture", RFC 5960, August
    2010.

Contributing Authors

   George Swallow
   Cisco Systems, Inc.
   EMail: swallow@cisco.com

   David Ward
   Juniper Networks.
   EMail: dward@juniper.net

   Stewart Bryant
   Cisco Systems, Inc.
   EMail: stbryant@cisco.com

   Carlos Pignataro
   Cisco Systems, Inc.
   EMail: cpignata@cisco.com

   Eric Osborne
   Cisco Systems, Inc.
   EMail: eosborne@cisco.com

   Nabil Bitar
   Verizon.
   EMail: nabil.bitar@verizon.com

   Italo Busi
   Alcatel-Lucent.
   EMail: italo.busi@alcatel-lucent.com

   Lieven Levrau
   Alcatel-Lucent.
   EMail: lieven.levrau@alcatel-lucent.com

   Laurent Ciavaglia
   Alcatel-Lucent.
   EMail: laurent.ciavaglia@alcatel-lucent.com

   Bo Wu
   ZTE Corporation.
   EMail: wu.bo@zte.com.cn

   Jian Yang
   ZTE Corporation.
   EMail: yang_jian@zte.com.cn

Editors' Addresses

   Sami Boutros
   Cisco Systems, Inc.
   EMail: sboutros@cisco.com

   Siva Sivabalan
   Cisco Systems, Inc.
   EMail: msiva@cisco.com

   Rahul Aggarwal
   Arktan, Inc
   EMail: raggarwa_1@yahoo.com

   Martin Vigoureux
   Alcatel-Lucent.
   EMail: martin.vigoureux@alcatel-lucent.com

   Xuehui Dai
   ZTE Corporation.
   EMail: dai.xuehui@zte.com.cn