

**MPLS Forum**

**LDP Conformance**

**Implementation Agreement**

**MPLS Forum 3.0**

**MPLS Forum Technical Committee**  
**December 4, 2002**

**Note:** The user's attention is called to the possibility that implementation of the MPLS implementation agreement contained herein may require the use of inventions covered by patent rights held by third parties. By publication of this MPLS implementation agreement the MPLS Forum makes no representation that the implementation of the specification will not infringe on any third party rights. The MPLS Forum take no position with respect to any claim that has been or may be asserted by any third party, the validity of any patent rights related to any such claims, or the extent to which a license to use any such rights may not be available.

**Editor:**

**Ananda Sen Gupta**  
**Agilent Technologies**

**For more information contact:**

**The MPLS Forum**  
Suite 307  
39355 California Street  
Fremont, CA 94538 USA

Phone: +1 (510) 608-3997

FAX: +1 (510) 608-5917

E-Mail: [info@mplsforum.org](mailto:info@mplsforum.org)

WWW: <http://www.mplsforum.org/>

**Full Notice**

Copyright © 2002 MPLS Forum.  
All rights reserved.

This document and translations of it may be copied and furnished to others, and works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the MPLS Forum, except as needed for the purpose of developing MPLS implementation agreements (in which case the procedures copyrights defined by the MPLS Forum must be followed), or as required to translate it into languages other than English

This document and the information contained herein is provided on an "AS IS" basis and THE MPLS FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	LDP CONFORMANCE TEST PHILOSOPHY .....	1
1.2	DEFINITIONS .....	1
<b>2</b>	<b>MODES OF OPERATION OF AN LSR .....</b>	<b>2</b>
<b>3</b>	<b>REQUIRED TOPOLOGIES .....</b>	<b>7</b>
3.1	TEST CONFIGURATION .....	7
<b>4</b>	<b>DETAILED TEST COVERAGE.....</b>	<b>8</b>

## Revision History

Version	Change	Date
MPLS 3.0	Initial version	December 2002

This page intentionally left blank.

# 1 Introduction

Before a Service Provider (SP) thinks of deploying MPLS (we will only refer to LDP in this document) enabled devices into the field, they have to go through functional and conformance testing to ensure the basic MPLS functionality works. If the SP plans to use devices from more than one vendor, then they have to test for Interoperability between these different vendor devices.

However, Conformance tests have to be run first on these devices. Conformance testing is to pay a lot of attention to the protocol specification details to ensure that the device can handle positive (correct) as well as negative situations (malformed packets, out-of-state or out-of-sequence messages). It is important to have positive tests to test whether the device will work in normal conditions, but it is extremely important to also have negative tests, to ensure robust operation in the field. Typically, it is quite a lot of effort to create such test scenarios (as normal devices are not expected to behave incorrectly), and this document will attempt to list many such scenarios.

This document deals with the recommended test coverage for the LDP protocol implementation. Other documents will need to be created based on this document which will be detailed Abstract Test Cases (ATC) which will be part of an LDP Abstract Test Suite (ATS).

The LDP Conformance Test Coverage Document aims at providing comprehensive test coverage scenarios based on various MPLS LDP RFCs / Internet drafts. The RFCs/drafts used for references and coverage in this document are:

Function	Standard
Multi protocol Label Switching Architecture	RFC-3031
LDP State Machine	Draft-ietf-mpls-ldp-state-03.txt
LDP Specification	RFC-3036
MPLS using ATM VC Switching	RFC-3035

## 1.1 LDP Conformance Test Philosophy

The tests can be broken into 2 broad categories:

1. Positive Testing
  - a) Coverage of the state machine and message formats
2. Negative Testing
  - a) Test for robust operation with error values in conformance tests
  - b) Tests for robust operation in case of failure of a node

The devices will have to support the LDP Signaling protocol, as well as at least one Internal Gateway Protocol amongst OSPF and IS-IS. The overall test requirements will be the same for any set of the protocols.

## 1.2 Definitions

This list will be a growing one, which will discuss all the abbreviations used in this document.

LDP	Label Distribution Protocol
LSR	Label Switch Router (which supports LDP Signaling)
FEC	Forwarding Equivalence Class

## 2 Modes of operation of an LSR

Before launching into a detailed discussion on testing, it is important to understand that there are various 'modes of operation' of a Label Switch Router, which supports LDP. In the rest of the document, whenever we use the term LSR, it will refer to such a router.

An LSR can support the following modes of operation. These modes of operation have been taken from Multiprotocol Label Switching Architecture (RFC 3031). In each test, the modes of operation for which the test is valid are stated.

**1.Mode\_1:** Mode\_1 can be the mode of functionality of a non-VC-Merge ATM LSR without Loop Detection. An LSR operating in this mode supports the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Ordered
Loop Detection	Disabled
Merge/Non-Merge	Non-Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledConditional
Request Procedure	RequestOnRequest
NotAvailable Procedure	RequestRetry
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseImmediate

**2.Mode\_2:** Mode\_2 can be the mode of functionality of a non-VC-Merge ATM LSR with Loop Detection. An LSR operating in this mode supports the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Ordered
Loop Detection	Enabled
Merge/Non-Merge	Non-Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledConditional
Request Procedure	RequestOnRequest
NotAvailable Procedure	RequestRetry
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

**3.Mode\_3:** Mode\_3 can be the mode of functionality of a non-VC-Merge ATM LSR without Loop Detection. An LSR operating in this mode supports the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Independent
Loop Detection	Disabled
Merge/Non-Merge	Non-Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledConditional
Request Procedure	RequestOnRequest
NotAvailable Procedure	N/A
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseImmediate

**4.Mode\_4:** Mode\_4 can be the mode of functionality of a non-VC-Merge ATM LSR with Loop Detection. An LSR operating in this mode supports the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Independent
Loop Detection	Enabled
Merge/Non-Merge	Non-Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledConditional
Request Procedure	RequestOnRequest
NotAvailable Procedure	N/A
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

**5.Mode\_5:** Mode\_5 can be the mode of operation of a VC-Merge ATM LSR without Loop Detection. An LSR operating in this mode supports the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Ordered
Loop Detection	Disabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledConditional
Request Procedure	RequestOnRequest
NotAvailable Procedure	RequestRetry
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseImmediate

**6.Mode\_6:** Mode\_6 operation can be the mode of operation of a VC-Merge ATM LSR with Loop Detection. An LSR operating in this mode supports the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Ordered
Loop Detection	Enabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledConditional
Request Procedure	RequestWhenNeeded
NotAvailable Procedure	RequestRetry
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

**7.Mode\_7:** An LSR operating in this mode can be a VC-Merge ATM LSR that does not support Loop Detection. An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-Unsolicited
Label Retention Mode	Conservative
Label Distribution Control Mode	Ordered
Loop Detection	Disabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PushedConditional
Request Procedure	RequestWhenNeeded
NotAvailable Procedure	RequestNoRetry
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseImmediate

**8.Mode\_8:** An LSR operating in this mode can be a VC-Merge ATM LSR that supports Loop Detection. An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-Unsolicited
Label Retention Mode	Conservative
Label Distribution Control Mode	Ordered
Loop Detection	Enabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PushedConditional
Request Procedure	RequestWhenNeeded
NotAvailable Procedure	RequestNoRetry
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

**9.Mode\_9:** An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-Unsolicited
Label Retention Mode	Liberal
Label Distribution Control Mode	Independent
Loop Detection	Disabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PushedUnConditional
Request Procedure	RequestNever
NotAvailable Procedure	N/A
Release Procedure	NoReleaseOnChange
LabelUse Procedure	UseImmediate



**10. Mode\_10:** An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-Unsolicited
Label Retention Mode	Liberal
Label Distribution Control Mode	Independent
Loop Detection	Enabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PushedUnConditional
Request Procedure	RequestNever
NotAvailable Procedure	N/A
Release Procedure	NoReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

**11. Mode\_11:** An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-Unsolicited
Label Retention Mode	Liberal
Label Distribution Control Mode	Ordered
Loop Detection	Disabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PushedConditional
Request Procedure	RequestNever
NotAvailable Procedure	N/A
Release Procedure	NoReleaseOnChange
LabelUse Procedure	UseImmediate

**12. Mode\_12:** An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-Unsolicited
Label Retention Mode	Liberal
Label Distribution Control Mode	Ordered
Loop Detection	Enabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PushedConditional
Request Procedure	RequestNever
NotAvailable Procedure	N/A
Release Procedure	NoReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

**13: Mode\_13:** Mode\_13 can be the mode of operation of a VC-Merge ATM LSR without Loop Detection  
An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Independent
Loop Detection	Disabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledUnConditional
Request Procedure	RequestWhenNeeded
NotAvailable Procedure	N/A
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseImmediate

**14: Mode\_14:** Mode\_14 can be the mode of operation of a VC-Merge ATM LSR with Loop Detection  
An LSR operating in this mode has the following features:

Label Distribution Mode	Downstream-on-Demand
Label Retention Mode	Conservative
Label Distribution Control Mode	Independent
Loop Detection	Enabled
Merge/Non-Merge	Merge

Various procedures which have to be performed for this mode of operation:

Distribution Procedure	PulledUnConditional
Request Procedure	RequestWhenNeeded
NotAvailable Procedure	N/A
Release Procedure	ReleaseOnChange
LabelUse Procedure	UseIfLoopNotDetected

### 3 Required Topologies

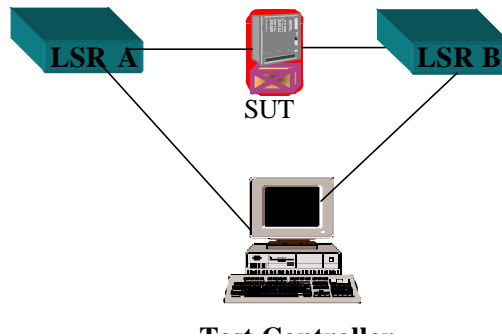
Any Conformance Test implementation will have to assume a “Generic Test Platform”, as ‘negative’ scenarios are not expected to be created in a real device.

Typically, the test platform will represent one or more simulated LSRs, of which some will be physically connected to the LSR under test (DUT). The physical interface of a Simulated LSR No. *i* will be referred to as SimInterface\_*i*, and it will be connected to DUTInterface\_*i* by Link\_*i*.

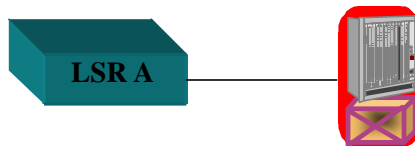
The following diagrams represent some typical topologies that will be necessary to support the tests discussed.

#### 3.1 Test Configuration

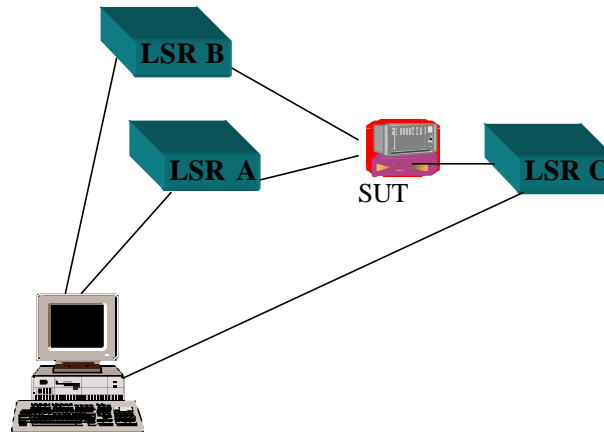
The configurations for testing a label switch router (LSR) implementing LDP are shown in the Figure 1.a. Two interfaces of the DUT are connected to two simulated LSRs on the tester. The DUT may act as an intermediate LSR for some FECs and egress for others. For some FECs for which the DUT is an intermediate LSR simulated LSR 1 should be the downstream peer and simulated LSR 2 the upstream peer. For another set of FECs for which the DUT is an intermediate LSR it should be vice-versa. In other words the LSR should recognize some sets of FECs for which simulated LSR A and simulated LSR B are the next hops. In testing an egress LSR, the tester emulates an upstream LSR only; and in testing an ingress LSR, the tester emulates a downstream LSR only, as shown in Figure 1.b. In Figure 1.c, three interfaces of the DUT are connected to two simulated LSR’s. As can be seen below the DUT is connected to simulated LSR via two links. The DUT should act as an intermediate LSR for some FECs and egress for others.



**Figure 1.a LSR Test Configuration**



**Figure 1.b Ingress or Egress LER Test Configuration**



**Figure 1.c LSR Test Configuration for Merge Scenarios**

For the above configurations, it may be necessary to specify the following information for testing.

1. Routing Protocol used (may be static too)
2. Physical Interface Type
3. Label Range to be used
4. Label Type (i.e. General, ATM, Frame Relay, etc.)

## 4 Detailed Test Coverage

### Discovery & Session Establishment

-----

Basic & extended LDP Discovery: 1.1.1 - 1.1.6  
 Transport Connection Establishment: 1.2.1 - 1.2.8  
 Session Initialization: 1.3.1 - 1.3.10  
 Session FSM: 1.4.1 - 1.4.6

### Loop Detection

-----

Label Request (Hop Count): 2.1.1 - 2.1.5  
 Label Request (Path Vector): 2.2.1 - 2.2.5  
 Label Mapping (Hop Count): 2.3.1 - 2.3.6  
 Label Mapping (Path Vector): 2.4.1 - 2.4.13

### Notification procedures

-----

Malformed PDU & message: 3.1.1 - 3.1.49  
 Unknown & Malformed TLV: 3.2.1 - 3.2.40  
 Generic (Include action to fatal notifications): 3.3.1 - 3.3.4

Hello message: 4.1.1 - 4.1.6  
 Initialization message: 5.1.1 - 5.1.16  
 Address message: 6.1.1 - 6.1.2  
 Address Withdraw message: 7.1.1  
 Label Mapping: 8.1.1 - 8.1.19

Label Request: 9.1.1 - 9.1.17  
 Label Request Abort message: 10.1.1 - 10.1.10  
 Label Withdraw: 11.1.1 - 11.1.7  
 Label Release: 12.1.1 12.1.18

The table following the heading has the following columns:

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
--------------	----------------------	--------------	-------------------	----------

1. Test Reference Number - A simple reference number to allow the reader to refer to a specific test.
2. RFC & Section Number - The Section Number in the LDP Specification (RFC 3036)
3. Test Purpose - The description of Test purpose for the respective section.
4. Mode of Operation - The mode of operation for which the test is valid. These modes of operation have been taken from Multiprotocol Label Switching Architecture (RFC 3031).
5. Category – This column explains what the recommended support should be as per the RFC, as interpreted by the authors.

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
1.1.1	RFC 3036, Section 1.2	Verify if the SUT is able to exchange link Hello's over a link	All	MUST
1.1.2	RFC 3036, Section 2.4.1, 3.5.1.2.1	Verify that the SUT will not respond after it receives a Hello message with TLV length that is too large (i.e. the TLV extends beyond the end of the containing message)	All	MUST
1.1.3	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will not respond after it receives a Hello message with unsupported LDP protocol version	All	MUST
1.1.4	RFC 3036, Section 2.4.2	Verify that the SUT will respond with targeted Hello's if <ol style="list-style-type: none"> <li>a) It receives a targeted Hello's on one of its interfaces</li> <li>b) It is configured to respond to targeted Hello's from the peer in consideration</li> </ol>	All	MUST
1.1.5	RFC 3036, Section 2.4.2	Verify that the SUT will send targeted Hello messages to a peer if it is configured to do so	All	MUST
1.1.6	RFC 3036, Section 2.4.2, 3.5.1.2.1	Verify that the SUT will not respond after it receives targeted Hello messages with TLV length that is too large	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		(i.e. the TLV extends beyond the end of the containing message)		
1.2.1	RFC 3036, Section 2.5.2,	Verify that the SUT will set up the TCP connection if a) The received Hello messages do not have a Transport Address TLV b) It plays the passive role (i.e. transport IP address is smaller than the transport IP address of the peer)	All	MUST
1.2.2	RFC 3036, Section 2.5.2	Verify that the SUT will set up the TCP connection if a) The received Hello messages do not have a Transport Address TLV b) It plays the active role (i.e. transport IP address is larger than the transport IP address of the peer)	All	MUST
1.2.3	RFC 3036, Section 2.5.2	Verify that the SUT will set up the TCP connection if a) The received Hello messages have a Transport Address TLV b) It plays the passive role (i.e. transport IP address is smaller than the transport IP address of the peer)	All	MUST
1.2.4	RFC 3036, Section 2.5.2	Verify that the SUT will set up the TCP connection if a) The received Hello messages have a Transport Address TLV b) It plays the active role (i.e. transport IP address is larger than the transport IP address of the peer)	All	MUST
1.2.5	RFC 3036, Section 2.5.2	Verify that the LSR will set up the TCP connection if a) Extended discovery mode is used b) The received targeted Hello messages do not	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		<p>have a Transport Address TLV</p> <p>c) It plays the passive role (i.e. transport IP address is smaller than the transport IP address of the peer)</p>		
1.2.6	RFC 3036, Section 2.5.2	<p>Verify that the LSR will set up the TCP connection if</p> <p>a) Extended discovery mode is used</p> <p>b) The received targeted Hello messages do not have a Transport Address TLV</p> <p>c) It plays the active role (i.e. transport IP address is larger than the transport IP address of the peer)</p>	All	MUST
1.2.7	RFC 3036, Section 2.5.2	<p>Verify that the LSR will set up the TCP connection if</p> <p>d) Extended discovery mode is used</p> <p>e) The received targeted Hello messages have a Transport Address TLV</p> <p>f) It plays the passive role (i.e. transport IP address is smaller than the transport IP address of the peer)</p>	All	MUST
1.2.8	RFC 3036, Section 2.5.2	<p>Verify that the LSR will set up the TCP connection if</p> <p>a) Extended discovery mode is used</p> <p>b) The received targeted Hello messages have a Transport Address TLV</p> <p>c) It plays the active role (i.e. transport IP address is larger than the transport IP address of the peer)</p>	All	MUST
1.3.1	RFC 3036, Section 2.5.3	<p>Verify that the SUT will set up the LDP session if</p> <p>a) The SUT plays the active role</p>	All	MUST
1.3.2	RFC 3036,	Verify that the SUT will set	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 2.5.3	up the LDP session if b) The SUT plays the passive role		
1.3.3	RFC 3036, Section 2.5.3	Verify that the SUT will respond with a Notification (Session Rejected/Bad Protocol Version) message and close the TCP connection if a) It receives an Initialization message with incompatible protocol version in the Common Session Parameters TLV b) The SUT plays the passive role	All	MUST
1.3.4	RFC 3036, Section 2.5.3	Verify that the SUT will respond with a Notification (Session Rejected/No Hello) message and close the TCP connection if a) It receives an Initialization message with a LDP ID in the Common Session Parameters TLV which has an unmatched Hello adjacency b) The SUT plays the passive role	All	MUST
1.3.5	RFC 3036, Section 2.5.3	Verify that the SUT will respond with a Notification (Session Rejected/Max PDU Length) and close the TCP connection if a) It receives an Initialization message with incompatible Max PDU Length (larger than it supports) in the Common Session Parameters TLV b) The SUT plays the passive role	All	MUST
1.3.6	RFC 3036, Section 2.5.3	Verify that the SUT will respond with Notification (Session Rejected/Parameter Label Range) and closes the TCP connection if	ATM SUT only	MUST



TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		<ul style="list-style-type: none"> <li>a) Only VCI values are being used for the labels. The tester proposes max and min VPI values as 0</li> <li>b) <math>Vci\_min(SUT) &gt; Vci\_min(Tester)</math></li> <li>c) <math>Vci\_min(SUT) &gt; Vci\_max(Tester)</math></li> </ul>		
1.3.7	RFC 3036, Section 2.5.3	<p>Verify that the SUT will respond with Notification (Session Rejected/Parameter Label Range) and closes the TCP connection if</p> <ul style="list-style-type: none"> <li>a) Only VCI values are being used for the labels. The tester proposes max and min VPI values as 0</li> <li>b) <math>Vci\_max(SUT) &lt; Vci\_min(Tester)</math></li> <li>c) <math>Vci\_max(SUT) &lt; Vci\_max(Tester)</math></li> </ul>	ATM SUT only	MUST
1.3.8	RFC 3036, Section 2.5.3	<p>Verify that the SUT will respond with Notification (Session Rejected/Parameter Advertisement Mode) and closes the TCP connection if</p> <ul style="list-style-type: none"> <li>a) An ATM link exists between the SUT and the tester</li> <li>b) The SUT proposes DOD as the advertisement mode</li> <li>c) The tester proposes DU as the advertisement mode</li> </ul>	ATM SUT only	MAY
1.3.9	RFC 3036, Section 2.5.3	<p>Verify that the SUT will respond with Notification (Session Rejected/Parameter Advertisement Mode) and closes the TCP connection if</p> <ul style="list-style-type: none"> <li>d) The session is not for a ATM or a Frame Relay link</li> <li>e) The SUT proposes DU as the advertisement mode</li> <li>f) The tester proposes DOD as the advertisement mode</li> </ul>	non-ATM SUT only	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
1.3.10	RFC 3036, Section 2.5.3	Verify that the SUT will throttle the session setup retry (First retry should be delayed by at least 15 sec., subsequent at least by 120 sec) and later restores it if a) It plays the active role and its Initialization messages are NAK'd successively b) It later receives a Hello message indicating reconfiguration	All	MUST
1.4.1	RFC 3036, Section 2.5.4	Verify that the SUT will send a Notification message and close the TCP connection if a) It receives any other message except the Initialization message when it is in the Initialized state	All	MUST
1.4.2	RFC 3036, Section 2.5.4	Verify that the SUT will send a Notification message and close the TCP connection if a) It receives any other message except the Keepalive message when it is in the OpenRec state	All	MUST
1.4.3	RFC 3036, Section 2.5.4	Verify that the SUT will send a Notification message and close the TCP connection if a) It receives any other message except the Initialization message when it is in the Opensest state	All	MUST
1.4.4	RFC 3036, Section 2.5.4	Verify that the SUT will transmit a Shutdown message and close the TCP connection if a) The Session between the SUT and the tester is in Operational state b) It receives a Notification (Shutdown) message	All	MUST
1.4.5	RFC 3036, Section 2.5.4	Verify that the SUT will transmit a Notification (HoldTime Expired) message and closes the TCP connection if a) The SUT is in Operational state b) HoldTime expires	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		because Hello messages are not received		
1.4.6	RFC 3036, Section 2.5.4	Verify that the SUT will transmit a Notification (KeepAlive Timer Expired) message and closes the TCP connection if a) The SUT is in Operational state b) KeepAlive timer expires because no LDP message is received	All	MUST
2.1.1	RFC 3036, Section 2.8.1, A.1.1	Verify that the SUT will respond with a Notification (Loop Detected ) message if a) It detects a loop (max hop count) in the Label Request message it receives	Mode_1 (ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_12, Mode_13 (ATM only), Mode_14	MUST
2.1.2	RFC 3036, Section 2.8.1, A.1.1	Verify that the SUT will not propagate the Label Request if a) It detects a loop in the Label Request it receives	Mode_1 (ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_13 (ATM only), and Mode_14	MUST
2.1.3	RFC 3036, Section 2.8.1, A.2.7	Verify that the SUT will include the Hop Count TLV and increment the hop count in the Label Request that it propagates if a) The Label Request it received from upstream has the Hop Count TLV	Mode_1, Mode_2, Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_7, Mode_8, Mode_13, and Mode_14	MUST
2.1.4	RFC 3036, Section 2.8.1, A.2.7	Verify that the SUT will include a Hop Count TLV and set the hop count to 1 in the Label Request message it transmits if a) It is the ingress for the FEC	Mode_1 (ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_12, Mode_13 (ATM only), and Mode_14	MUST
2.1.5	RFC 3036, Section 2.8.1, A.2.7	Verify that the SUT will include a Hop Count TLV and set the hop count to 0 in the Label Request it propagates if a) The received Label Request that is being propagated does not	Mode_2 (non ATM only), Mode_4 (non ATM only), Mode_6 (non ATM only), Mode_8(non ATM only), Mode_12 (non ATM only), and Mode_14 (non ATM	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		have a Hop Count TLV	only)	
2.2.1	RFC 3036, Section 2.8.1, A.1.1	Verify that the SUT will respond with a Notification (Loop Detected ) message if b) It detects a loop ( path vector) in the Label Request message it receives	Mode_2, Mode_4, Mode_6, Mode_8, Mode_12, Mode_14	MUST
2.2.2	RFC 3036, Section 2.8.1, A.1.1	Verify that the SUT will respond with a Notification (Loop Detected ) message if c) It detects a loop ( path vector limit) in the Label Request message it receives	Mode_2, Mode_4, Mode_6, Mode_8, Mode_12, Mode_14	MUST
2.2.3	RFC 3036, Section 2.8.1, A.2.7	Verify that the SUT will include a Path Vector TLV of length 1 in the Label Request if a) It is the ingress for the FEC	Mode_2, Mode_4	MUST
2.2.4	RFC 3036, Section 2.8.1, A.2.7	Verify that the SUT will add its own ID in the Path Vector TLV in the Label Request that it propagates if a) The Label Request it received from upstream has a Path Vector TLV	Mode_2, Mode_4, Mode_6, Mode_8, Mode_14	MUST
2.2.5	RFC 3036, Section 2.8.1, A.2.7	Verify that the SUT will include a Path Vector TLV of length 1 with its own ID in the Label Request it propagates if a) The Label Request it received from upstream does not have a Path Vector b) It does not have merge capabilities	Mode_2, Mode_4	MUST
2.3.1	RFC 3036, Section 2.8.2, A.1.2	Verify that the SUT will send a Label Release message with a Status TLV (Loop Detected) if a) It detects a loop (Max Hop Count) in the Label Mapping message	Mode_1 (ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_10, Mode_12, Mode_13 (ATM only), and Mode_14	MUST
2.3.2	RFC 3036, Section 2.8.2,	Verify that the SUT will not propagate a Label Mapping if	Mode_1 (ATM only), Mode_2, Mode_3	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	A.1.2	a) It detects a loop in the Label Mapping message	(ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_10, Mode_12, Mode_13 (ATM only), and Mode_14	
2.3.3	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will increment the hop count before propagating a Label Mapping if a) It is not a member of the edge set of LSR's domain whose LSR's do not perform TTL-decrement b) The Label Mapping that it received contains a known hop count	All	MUST
2.3.4	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include the Hop Count TLV and set the hop count to 1 in the Label Mapping if a) It is the egress for the FEC	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_13, Mode_14	MUST
2.3.5	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include the Hop Count TLV and set the hop count to 0 in the Label Mapping it is propagating if a) The Label Mapping received from downstream does not have a Hop Count TLV	Mode_2 (non ATM only), Mode_4 (non ATM only), Mode_6 (non ATM only), Mode_8(non ATM only), Mode_10, Mode_12, Mode_14 (non ATM only)	MUST
2.3.6	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include the Hop Count TLV and set the hop count to 0 in the Label Mapping if a) It is not propagating the Label Mapping	Mode_3, Mode_4, Mode_10, Mode_13, Mode_14	MUST
2.4.1	RFC 3036, Section 2.8.2, A.1.2	Verify that the SUT will send a Label Release message with a Status TLV (Loop Detected) if a) It detects a loop (Path Vector) in the Label Mapping message	Mode_2, Mode_4, Mode_6, Mode_8, Mode_14	MUST
2.4.2	RFC 3036, Section 2.8.2, A.1.2	Verify that the SUT will send a Label Release message with a Status TLV (Loop Detected) if	Mode_2, Mode_4, Mode_6, Mode_8, Mode_14	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		a) It detects a loop (Path Vector Limit) in the Label Mapping message		
2.4.3	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will add its own ID in the Path Vector TLV in the Label Mapping that it is propagating if a) The Label Mapping received from downstream has a Path Vector TLV	Mode_2, Mode_4, Mode_6, Mode_8, Mode_10, Mode_12, Mode_14	MUST
2.4.4	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping if a) It is not propagating the Label Mapping	Mode_4, Mode_10, Mode_14	MUST
2.4.5	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping it propagates if a) The Label Mapping received from downstream does not have Path Vector TLV b) It does not have merge capabilities c) The Label Mapping received from downstream has a Hop Count TLV with Hop Count as 0	Mode_2	MUST
2.4.6	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping it propagates if a) The Label Mapping received from downstream does not have Path Vector TLV b) It does not have Merge capabilities c) The Label Mapping received from downstream has a Hop Count TLV with a known Hop Count d) The SUT has sent a Label Mapping to its upstream peer for the FEC and PrevHopCount < Hop count in	Mode_2, Mode_4	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		RAttributes		
2.4.6	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own LSR ID in the Label Mapping it propagates if <ul style="list-style-type: none"> <li>a) The Label Mapping received from downstream does not have Path Vector TLV</li> <li>b) It does not have merge capabilities</li> <li>c) The Label Mapping received from downstream has a Hop Count TLV with a known Hop Count</li> <li>d) The SUT has sent a Label Mapping to its upstream peer for the FEC and PrevHopCount == 0</li> </ul>	Mode_2, Mode_4	MUST
2.4.8	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping it propagates if <ul style="list-style-type: none"> <li>a) The Label Mapping received from downstream does not have Path Vector TLV</li> <li>b) It is merge capable</li> <li>c) The SUT has not sent a Label Mapping to its upstream peer for the FEC</li> </ul>	Mode_6, Mode_8, Mode_12	MUST
2.4.9	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping it propagates if <ul style="list-style-type: none"> <li>a) The Label Mapping received from downstream does not have Path Vector TLV</li> <li>b) It is merge capable</li> <li>c) The SUT has sent a Label Mapping to its upstream peer for the FEC</li> <li>d) The Label Mapping received from downstream has a Hop Count TLV with an</li> </ul>	Mode_6, Mode_8, Mode_10, Mode_12, Mode_14	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		unknown hop count		
2.4.10	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping it propagates if <ul style="list-style-type: none"> <li>a) The Label Mapping received from downstream does not have Path Vector TLV</li> <li>b) It is merge capable</li> <li>c) The LSR has sent a Label Mapping to its upstream peer for the FEC</li> <li>e) The Label Mapping received from downstream has a Hop Count TLV with a known Hop Count and PrevHopCount &lt; Hop Count in Rattributes</li> </ul>	Mode_6, Mode_8, Mode_10, Mode_12 Mode_14	MUST
2.4.11	RFC 3036, Section 2.8.2, A.2.8	Verify that the SUT will include a Path Vector TLV with its own ID in the Label Mapping it propagates if <ul style="list-style-type: none"> <li>a) The Label Mapping received from downstream does not have Path Vector TLV</li> <li>b) It is merge capable</li> <li>c) The SUT has sent a Label Mapping to its upstream peer for the FEC</li> <li>d) The Label Mapping received from downstream has a Hop Count TLV with a known Hop Count and PrevHopCount=0(unknown)</li> </ul>	Mode_6, Mode_8, Mode_10, Mode_12 Mode_14	MUST
2.4.12	RFC 3036, Section A.1.1	Verify that the SUT will respond with a Notification (Loop Detected) message if <ul style="list-style-type: none"> <li>a) It receives a Label Request for a FEC for which the message source is the next hop</li> </ul>	All	MUST
3.1.1	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) message if	All	MUST



TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		a) It receives an Initialization message with an unmatched LDP identifier b) The SUT plays the passive role		
3.1.2	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if a) It receives an Initialization message with an unmatched LDP Protocol Version b) The SUT plays the passive role	All	MUST
3.1.3	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if a) It receives an Initialization message with a PDU length field exceeding the maximum PDU length b) The SUT plays the passive role	All	MUST
3.1.4	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type) message with an 'Unknown Message Type' if a) It receives an Initialization message with unknown message type (<0x8000 and high order bit=0) b) The SUT plays the passive role	All	MUST
3.1.5	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Message Length) message if a) It receives an Initialization message with a bad message length b) The SUT plays the passive role	All	MUST
3.1.6	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Parameters) message if a) It receives an Initialization message with the mandatory parameter missing	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		b) The SUT plays the passive role		
3.1.7	RFC 3036, Section 3.5.1.2.1	Verify that the SUT returns a Notification message and closes the TCP connection if a) It receives an Initialization message with unknown message type ( $\geq 0x8000$ and high order bit=1) from the upstream LSR A b) The SUT plays the passive role	All	MUST
3.1.8	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) message if a) It receives a Label Request message with an unmatched LDP identifier	All	MUST
3.1.9	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if a) It receives a Label Request message with an unmatched LDP Protocol Version	All	MUST
3.1.10	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if a) It receives a Label Request message with a PDU length field exceeding the maximum PDU length.	All	MUST
3.1.11	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type) message if c) It receives a Label Request message with unknown message type ( $< 0x8000$ and high order bit=0)	All	MUST
3.1.12	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives a Label Request message with unknown message type ( $\geq 0x8000$ and high order bit=1) from the upstream LSR A	All	MUST
3.1.13	RFC 3036,	Verify that the SUT will	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.1.2.1	return a Notification (Bad Message Length) message if a) It receives a Label Request message with a bad message length		
3.1.14	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) message if a) It receives a Label Request message with the mandatory parameter missing	All	MUST
3.1.15	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) message if a) It receives a Label Mapping message with an unmatched LDP identifier	All	MUST
3.1.16	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad protocol Version) message if a) It receives a Label Mapping message with an unmatched LDP Protocol Version.	All	MUST
3.1.17	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if a) It receives a Label Mapping message with a PDU length field exceeding the maximum PDU length	All	MUST
3.1.18	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type) message if a) It receives a Label Mapping message with unknown message type (<0x8000 and high order bit=0)	All	MUST
3.1.19	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives a Label Mapping message with unknown message type (>=0x8000 and high order bit=1) from the downstream LSR B.	All	MUST
3.1.20	RFC 3036,	Verify that the SUT will	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.1.2.1	return a Notification (Bad Message Length) message if a) It receives a Label Mapping message with a bad message length		
3.1.21	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) if a) It receives a Label Mapping message with the mandatory parameter FEC TLV missing.	All	MUST
3.1.22	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) if a) It receives a Label Mapping message with the mandatory parameter Label TLV missing.	All	MUST
3.1.23	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification ( Bad LDP Identifier) message if a) It receives a Label Withdraw message with an unmatched LDP identifier	All	MUST
3.1.24	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if a) It receives a Label Withdraw message with an unmatched LDP Protocol Version	All	MUST
3.1.25	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if a) It receives a Label Withdraw message with the PDU length field exceeding the maximum PDU length	All	MUST
3.1.26	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type) message if a) It receives a Label Withdraw message with unknown message type (<0x8000 and high order bit=0)	All	MUST
3.1.27	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives a Label	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		Withdraw message with unknown message type (>=0x8000 and high order bit=1)		
3.1.28	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (bad Message Length) message if a) It receives a Label Withdraw message with a bad message length.	All	MUST
3.1.29	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) message if a) It receives a Label Withdraw message with the mandatory parameter missing.	All	MUST
3.1.30	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) if a) It receives a Label Release message with an unmatched LDP identifier	All	MUST
3.1.31	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if a) It receives a Label Release message with an unmatched LDP Protocol Version	All	MUST
3.1.32	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if a) It receives a Label Release message with a PDU length field exceeding the maximum PDU length.	All	MUST
3.1.33	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type) message if a) It receives a Label Release message with unknown message type (<0x8000 and high order bit=0)	All	MUST
3.1.34	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives a Label Release message with	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		unknown message type (>=0x8000 and high order bit=1)		
3.1.35	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Message Length) message if a) It receives a Label Release message with a bad message length	All	MUST
3.1.36	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing message Params) if a) It receives a Label Release message with the mandatory parameter missing	All	MUST
3.1.37	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing message Params) message if b) It receives a Label Release message with the mandatory parameter missing	All	MUST
3.1.38	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) message if a) It receives a Label Abort Request message with an unmatched LDP identifier	All	MUST
3.1.39	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if a) It receives a Label Abort Request message with an unmatched LDP Protocol Version	All	MUST
3.1.40	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Message Length) message if a) It receives a Label Abort Request message with a PDU length field exceeding the maximum PDU length	All	MUST
3.1.41	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type ) message if a) It receives a Label Abort Request message with unknown message type	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		(<0x8000 and high order bit=0)		
3.1.42	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives a Label Abort Request message with unknown message type (>=0x8000 and high order bit=1)	All	MUST
3.1.43	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Message Length) message if a) It receives a Label Abort Request message with a bad message length	All	MUST
3.1.44	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) message if a) It receives a Label Abort Request message with the mandatory parameter FEC TLV missing	All	MUST
3.1.45	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) message if a) It receives a Label Abort Request message with the mandatory parameter Label Request Msg ID TLV missing	All	MUST
3.1.46	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) message if a) It receives a Address message with unmatched LDP identifier	All	MUST
3.1.47	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type) message if a) It receives an Address message with unknown message type (<0x8000 and high order bit=0)	All	MUST
3.1.48	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives an Address message with unknown message type (>=0x8000 and high order bit=1)	All	MUST
3.1.49	RFC 3036,	Verify that the SUT will	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.1.2.1	return a Notification (Bad Message Length) message if b) It receives an Address message with a bad message length		
3.1.50	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) message if b) It receives an Address message with the mandatory parameter Address TLV missing	All	MUST
3.1.51	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if a) It receives an Address message with an unmatched LDP Protocol Version	All	MUST
3.1.52	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if a) It receives an Address message with a PDU length field exceeding the maximum PDU length	All	MUST
3.1.53	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad LDP Identifier) message if a) It receives a Address Withdraw message with unmatched LDP identifier	All	MUST
3.1.54	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Protocol Version) message if b) It receives an Address Withdraw message with an unmatched LDP Protocol Version	All	MUST
3.1.55	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad PDU Length) message if b) It receives an Address Withdraw message with a PDU length field exceeding the maximum PDU length	All	MUST
3.1.56	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Unknown Message Type)	All	MUST



TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		message if a) It receives an Address Withdraw message with unknown message type (<0x8000 and high order bit=0)		
3.1.57	RFC 3036, Section 3.5.1.2.1	Verify that the SUT does not respond if a) It receives an Address Withdraw message with unknown message type (>=0x8000 and high order bit=1)	All	MUST
3.1.58	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Bad Message Length) message if a) It receives an Address Withdraw message with a bad message length	All	MUST
3.1.59	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will return a Notification (Missing Message Params) message if a) It receives an Address Withdraw message with the mandatory parameter Address TLV missing	All	MUST
3.1.60	RFC 3036, Section 3.5.1.2.1	Verify that the SUT will transmit a Notification (Missing Message Params) if a) It receives a Notification message with the mandatory parameter Message ID missing	All	MUST
3.2.1	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV Length) message if a) It receives an Initialization message with TLV length that is too large, when the SUT plays the passive role.	All	MUST
3.2.2	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives an Initialization message with unknown TLV type (<0x8000 and high order bit=0) b) The SUT plays the passive role	All	MUST
3.2.3	RFC 3036,	Verify that the SUT does not	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.1.2.2	respond if a) It receives an Initialization message with unknown TLV type ( $\geq 0x8000$ and high order bit=1) b) The SUT plays the passive role		
3.2.4	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (malformed TLV Value) message if a) It receives an Initialization message with wrong TLV value b) The SUT plays the passive role	All	MUST
3.2.5	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV Length) message if a) It receives a Label Request message with TLV length that is too large	All	MUST
3.2.5A	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives a Label Request message with unknown TLV type ( $< 0x8000$ and high order bit=0)	All	MUST
3.2.6	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives a Label Request message with unknown TLV type ( $\geq 0x8000$ and high order bit=1)	All	MUST
3.2.7	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV value) message if a) It receives a Label Request message with wrong TLV value	All	MUST
3.2.8	RFC 3036, Section 3.5.1.2.2, 3.4.1	Verify that the SUT will return a Notification (Unknown FEC) message if a) It receives a Label Request message with the wrong FEC element type	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
3.2.9	RFC 3036, Section 3.4.1	Verify that the SUT will respond with a Notification (Unsupported Address Family) message and will not propagate the Label Request if a) It receives a Label Request with a FEC TLV that has an invalid address family	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	MUST
3.2.10	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV Length) if a) It receives a Label Mapping message with TLV length that is too large	All	MUST
3.2.11	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives a Label Mapping message with unknown TLV type (<0x8000 and high order bit=0)	All	MUST
3.2.12	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives a Label Mapping message with unknown TLV type (>=0x8000 and high order bit=1)	All	MUST
3.2.13	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) if a) It receives a Label Mapping message with the wrong FEC TLV value	All	MUST
3.2.14	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown FEC) message if a) It receives a Label Mapping message with the wrong FEC element type	All	MUST
3.2.15	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) if a) It receives a Label Mapping message with wrong Label TLV value	All	MUST
3.2.16	RFC 3036,	Verify that the SUT will	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.1.2.2	return a Notification (Bad TLV Length) if a) It receives a Label Withdraw message with TLV length that is too large		
3.2.17	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives a Label Withdraw message with unknown TLV type (<0x8000 and high order bit=0)	All	MUST
3.2.18	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives a Label Withdraw message with unknown TLV type (>=0x8000 and high order bit=1)	All	MUST
3.2.19	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) message if a) It receives a Label Withdraw message with wrong TLV value	All	MUST
3.2.20	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown FEC) message if a) It receives a Label Withdraw message with the wrong FEC element type.	All	MUST
3.2.21	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV Length) message if a) It receives a Label Release message with TLV length that is too large	All	MUST
3.2.22	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives a Label Release message with unknown TLV type (<0x8000 and high order bit=0)	All	MUST
3.2.23	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives a Label	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		Release message with unknown TLV type (>=0x8000 and high order bit=1)		
3.2.24	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) message if a) It receives a Label Release message with wrong TLV value	All	MUST
3.2.25	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown FEC) message if a) It receives a Label Release message with the wrong FEC element type	All	MUST
3.2.26	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV length) message if a) It receives a Label Abort Request message with TLV length that is too large	All	MUST
3.2.27	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives a Label Abort Request message with unknown TLV type (<0x8000 and high order bit=0)	All	MUST
3.2.28	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives a Label Abort Request message with unknown TLV type (>=0x8000 and high order bit=1)	All	MUST
3.2.29	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) message if a) It receives a Label Abort Request message with wrong TLV value	All	MUST
3.2.30	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown FEC) message if a) It receives a Label Abort Request message with the wrong FEC element	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		type		
3.2.31	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV Length) message if a) It receives a Address message with TLV length that is too large	All	MUST
3.2.32	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives an Address message with an unknown TLV type (<0x8000 and high order bit=0)	All	MUST
3.2.33	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives an Address message with an unknown TLV type (>=0x8000 and high order bit=1)	All	MUST
3.2.34	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) if a) It receives an Address message with wrong TLV value	All	MUST
3.2.35	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Bad TLV Length) if a) It receives a Address Withdraw message with TLV length that is too large.	All	MUST
3.2.36	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Unknown TLV) message if a) It receives an Address Withdraw message with unknown TLV type (<0x8000 and high order bit=0)	All	MUST
3.2.37	RFC 3036, Section 3.5.1.2.2	Verify that the SUT does not respond if a) It receives an Address Withdraw message with unknown TLV type (>=0x8000 and high order bit=1)	All	MUST
3.2.38	RFC 3036, Section 3.5.1.2.2	Verify that the SUT will return a Notification (Malformed TLV Value) if	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		a) It receives an Address Withdraw message with wrong TLV value		
3.3.1	RFC 3036, Section 3.4.6	Verify that the SUT will forward the Notification message if a) It receives a Notification message from the downstream LSR with the 'F' bit set to 1	All	MUST
3.3.2	RFC 3036, Section 3.4.6	Verify that the SUT will not respond if a) It receives a Notification message from the downstream LSR B with the 'F' bit set to 0	All	MUST
3.3.3	RFC 3036, Section A.1.13	Verify that the SUT will send a Notification (Resources Available) if it detects the resources have become available	ATM SUT only	MUST
3.3.4	RFC 3036, Section 3.5.1.1	Verify that the SUT will remove all label(s) learned through a session if a) The TCP connection pertaining to the session is broken	All	MUST
4.1.1	RFC 3036, Section 3.5.2	Verify that the SUT will adjust its HoldTime if a) It receives a Hello message with a Hold Time smaller than what it proposes	All	MUST
4.1.2	RFC 3036, Section 3.5.2	Verify that the SUT will adjust its HoldTime if a) It receives a targeted Hello with a smaller Hold Time than what it proposes	All	MUST
4.1.3	RFC 3036, Section 3.5.2	Verify that the SUT will interpret the Hold Time as 15 seconds (default) if a) It receives a link Hello message with a Hold Time set to 0	All	MUST
4.1.4	RFC 3036, Section 3.5.2	Verify that the SUT will interpret the Hold Time as 45 seconds (default) if a) It receives a targeted Hello message with a Hold Time of 0	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
4.1.5	RFC 3036, Section 3.5.2	Verify that the SUT will ignore the contents of the Reserved Field in the Common Hello Parameters TLV if a) It receives a Hello message with the Reserved Field in the Common Hello parameters TLV set to a non-zero value	All	MUST
4.1.6	RFC 3036, Section 3.5.2	Verify that the SUT will set the reserved field in the Common Hello Parameters TLV in the Hello message to zero	All	MUST
5.1.1	RFC 3036, Section 3.5.3	Verify that the SUT will adjust its KeepAlive timer if a) The KeepAlive time proposed to it is less than the value it proposes	All	MUST
5.1.2	RFC 3036, Section 3.5.3	Verify that the SUT will respond with a Notification (Missing Message Params) message and closes the TCP connection if a) It receives a message that an Initialization message without the ATM Session Parameters on an ATM link	ATM SUT only	MUST
5.1.3	RFC 3036, Section 3.5.3	Verify that the SUT will ignore the contents of the reserved field in the Common Session Parameters TLV if a) It receives a Initialization message with the Reserved Field in Common Session parameters TLV set to a non-zero value	All	MUST
5.1.4	RFC 3036, Section 3.5.3	Verify that the SUT will ignore the reserved field in the ATM Session Parameter if a) It receives an Initialization Message with Reserved Field in the ATM Session Parameters set to a non-	All	MUST



TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		zero value		
5.1.5	RFC 3036, Section 3.5.3	Verify that the SUT will set the reserved field in the Common Session Parameters to zero in the Initialization message	All	MUST
5.1.6	RFC 3036, Section 3.5.3	Verify that the SUT will set the reserved field in the ATM Session Parameters to zero in the Initialization message	All	MUST
5.1.7	RFC 3036, Section 3.5.3	Verify that the SUT will respond with a Notification (Malformed TLV Value) message and close the TCP connection if a) It receives a Initialization message with N (field in ATM Session Parameter) not equal to the number of label ranges in the ATM Session Parameter TLV	ATM SUT only	MUST
5.1.8	RFC 3036, Section 3.5.3	Verify that the SUT will use even values only for the labels it assigns if a) It receives an Initialization message with D bit in the ATM Session Parameters set to 1 b) It has a smaller LDP ID	ATM SUT only	MUST
5.1.9	RFC 3036, Section 3.5.3	Verify that the SUT will use odd values only for the labels it assigns if a) It receives an Initialization message with D bit in the ATM Session Parameters set to 1 b) It has a larger LDP ID	ATM SUT only	MUST
5.1.10	RFC 3036, Section 3.5.3	Verify that the SUT will calculate the intersection between the proposed ATM Label Range and the supported ATM Label Range if a) Only VCI values are being used for the labels	ATM SUT only	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		b) $Vci\_min(SUT) < Vci\_min(Tester)$ c) $Vci\_max(SUT) > Vci\_max(Tester)$		
5.1.11	RFC 3036, Section 3.5.3	Verify that the SUT will calculate the intersection between the proposed ATM Label Range and the supported ATM Label Range if a) Only VCI values are being used for the labels b) $Vci\_min(SUT) > Vci\_min(Tester)$ c) $Vci\_max(SUT) < Vci\_max(Tester)$	ATM SUT only	MUST
5.1.12	RFC 3036, Section 3.5.3	Verify that the SUT will calculate the intersection between the proposed ATM Label Range and the supported ATM Label Range if a) Only VCI values are being used for the labels b) $Vci\_min(SUT) > Vci\_min(Tester)$ c) $Vci\_max(SUT) > Vci\_max(Tester)$	ATM SUT only	MUST
5.1.13	RFC 3036, Section 3.5.3	Verify that the SUT will calculate the intersection between the proposed ATM Label Range and the supported ATM Label Range if a) Only VCI values are being used for the labels b) $Vci\_min(SUT) < Vci\_min(Tester)$ c) $Vci\_max(SUT) < Vci\_max(Tester)$	ATM SUT only	MUST
5.1.14	RFC 3036, Section 3.5.3	Verify that the SUT will ignore the reserved field in the ATM Label Range component if a) It receives an Initialization message with a non-zero Reserved Field in the ATM Label Range component	ATM SUT only	MUST
5.1.15	RFC 3036, Section 3.5.3	Verify that the SUT will set the reserved field in the ATM Label Range component to zero in the	ATM SUT only	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		Initialization message		
5.1.16	RFC 3036, Section 3.5.3	Verify that the SUT will send a Notification or a Label Release message and not propagate the Label Mapping message if a) The value of the label received in the Label Mapping does not fall in the label range negotiated during Initialization	ATM SUT only	MAY
6.1.1	RFC 3036, Section 3.5.5.1, 2.7	Verify that the SUT will send an Address message before any Label Mapping or Label Request message	All	SHOULD
6.1.2	RFC 3036, Section 3.5.5.1	Verify that the SUT will send an Notification ( Unsupported Address Family) message if a) It receives an Address message with an unsupported address family specified in the address list TLV	All	SHOULD
7.1.1	RFC 3036, Section 3.5.5.1, 2.7	Verify that the SUT will send an Address Withdraw message if a) One of its interfaces is de-activated	All	SHOULD
8.1.1	RFC 3036, Section 3.5.7	Verify that the SUT will include a Request Message ID TLV in the Label Mapping if a) The Label Mapping is in response to a Label Request message	Mode_1, Mode_2, Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	MUST
8.1.2	RFC 3036, Section 3.5.7.1.1, A.1.1	Verify that the SUT will respond with a Label Mapping message if a) A Label Request is received for a FEC for which it has already provided a mapping b) The Label Request is not a duplicate request	Mode_1, Mode_2, Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_13, Mode_14	MAY
8.1.3	RFC 3036,	Verify that the SUT will	Mode_1, Mode_2,	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.7.1.1	respond with a Label Mapping message if a) A Label Request is received for a FEC for which it is the egress	Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	
8.1.4	RFC 3036, Section 3.5.7.1.1, A.1.1	Verify that the SUT will respond with a Label Mapping if a) It is operating in Independent mode b) A Label Request is received for a FEC c) It does not have a mapping from its downstream peer	Mode_3 , Mode_4, Mode_13, Mode_14	MAY
8.1.5	RFC 3036, Section 3.5.7.1.1, A.1.2	Verify that the SUT will propagate the Label Mapping message if a) It already has a mapping and the attributes of the new mapping are inconsistent (different hop count) with those previously received	Mode_1 (ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_10, Mode_12, Mode_13 (ATM only), Mode_14	SHOULD
8.1.6	RFC 3036, Section 3.5.7.1.1, A.1.2	Verify that the SUT will propagate the Label Mapping message if a) It already has a mapping and the attributes of the new mapping are inconsistent (different path vector) with those previously received	Mode_2, Mode_4, Mode_6, Mode_8, Mode_10, Mode_12, Mode_14	SHOULD
8.1.7	RFC 3036, Section 3.5.7.1.1, A.1.6	Verify that the SUT will send a Label Mapping for a newly recognized FEC if a) Its mode of operation is Downstream Unsolicited , Independent b) It is not the egress for the FEC	Mode_9, Mode_10,	MAY
8.1.8	RFC 3036, Section 3.5.7.1.1, A.1.6	Verify that the SUT will send a Label Mapping for a newly recognized FEC if a) Its mode of operation is Downstream Unsolicited b) It is the egress for the FEC	Mode_7, Mode_8, Mode_9, Mode_10, Mode_11, Mode_12	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
8.1.9	RFC 3036, Section 3.5.7.1.1, A.1.6	Verify that the SUT will generate an event "Receive Label Mapping" if a) Its mode of operation is Downstream Unsolicited, Ordered b) It recognizes a new FEC c) It already has a mapping from its next hop as it is operating in Liberal label retention mode	Mode_11, Mode_12	MAY
8.1.10	RFC 3036, Section 3.5.7.1.1	Verify that the SUT will send a Label Mapping if a) Its mode of operation is Downstream Unsolicited b) It is the egress for the FEC in consideration	Mode_7, Mode_8, Mode_9, Mode_10, Mode_11, Mode_12	MAY
8.1.11	RFC 3036, Section 3.5.7.1.2, A.1.1	Verify that the SUT will respond with a Label Mapping message if a) A Label Request is received for a FEC for which it is the egress	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	MAY
8.1.12	RFC 3036, Section 3.5.7.1.2, A.1.1	Verify that the SUT will respond with a Label Mapping message if a) It is operating in Ordered mode b) There is a pending Label Request for a FEC c) It has a mapping from its downstream peer	Mode_1, Mode_2, Mode_5, Mode_6, Mode_7, Mode_8	MAY
8.1.13	RFC 3036, Section 3.5.7.1.2	Verify that the SUT will send a Label Mapping if a) It recognizes a new FEC b) It already has a mapping from its next hop as a result of Liberal label retention mode c) Its Mode of operation is Downstream Unsolicited, Ordered	Mode_11, Mode_12	MAY
8.1.14	RFC 3036, Section A.1.2	Verify that the SUT will not propagate a Label Mapping message if a) It already has a mapping and the attributes of the mapping are consistent (same hop count) with those previously	Mode_1 (ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7(ATM only), Mode_8, Mode_10, Mode_12, Mode_13 (ATM only),	SHOULD

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		received.	and Mode_14	
8.1.15	RFC 3036, Section A.1.2	Verify that the SUT will not propagate a Label Mapping message if a) It already has a mapping and the attributes of the mapping are consistent (Same path vector) with those previously received (assuming that the LSR stores the path vector to perform the consistency check)	Mode_2, Mode_4, Mode_6, Mode_8, Mode_12, Mode_14	SHOULD
8.1.16	RFC 3036, Section A.1.4	Verify that the SUT will not re-advertise labels to a peer if a) Its mode of operation is Downstream Unsolicited b) The peer has earlier released the label pertaining to the FEC c) The peer has not explicitly requested for a mapping for the FEC	Mode_7(ATM only), Mode_8, Mode_10, Mode_12	SHOULD
8.1.17	RFC 3036, Section 2.6.1.2	Verify that the SUT will not respond with a Label Mapping message if a) It is operating in Ordered Mode b) It does not have a label binding from its downstream LSR for the FEC in consideration	Mode_7(ATM only), Mode_8, Mode_10, Mode_12	MUST
8.1.18	RFC 3036, Section 3.4.1	Verify that the SUT will respond with a Notification or a Label Release message and will not propagate the Label Mapping if a) It receives a Label Mapping with a Wildcard FEC element	All	MUST
8.1.19	RFC 3036, Section 3.4.2.2	Verify that the SUT will ignore the contents of the reserved field present in the ATM Label TLV if a) It receives a Label Mapping with the reserved field in the ATM Label TLV having a non-zero value	ATM SUT only	MUST
8.1.20	RFC 3036, Section 3.4.2.2	Verify that the SUT will set the reserved field in the ATM label TLV to zero if a) It sends a Label	ATM SUT only	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		Mapping message to its peer		
9.1.1	RFC 3036, Section 3.5.7.1.4, 2.6.2.1	Verify that the SUT will send a Label Request to its new next hop if a) It detects a change in next hop for the FEC b) It does not have a mapping from its new next hop	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
9.1.2	RFC 3036, Section 3.5.8.1, A.1.1	Verify that the SUT will propagate a Label Request message if a) It does not have a mapping for the FEC in consideration from its downstream peer	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
9.1.3	RFC 3036, Section 3.5.8.1, A.1.1	Verify that the SUT will respond with a Notification (No Route) message if a) It receives a Label Request for a FEC for which it does not have a route in its forwarding table	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MUST
9.1.4	RFC 3036, Section 3.5.8.1, A.1.6	Verify that the SUT will request for a label mapping if a) It recognizes a new FEC b) It does not have a mapping from its next hop c) Its request procedure is not RequestNever	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
9.1.5	RFC 3036, Section A.1.6	Verify that the SUT will not request for a mapping if a) It recognizes a new FEC b) It does not have a mapping from its next hop c) Its request procedure is RequestNever	Mode_9, Mode_10, Mode_11, Mode_12	SHOULD
9.1.6	RFC 3036, Section 3.5.8.1, A.1.9, A.2.2	Verify that the SUT will not send any more Label Request(s) to a peer if a) It receives a Notification (No Resources) message in response to one of the its Label Request	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	SHOULD
9.1.7	RFC 3036, Section 3.5.8.1, A.1.13	Verify that the SUT will respond with a Notification (No Resources) message if	ATM SUT only	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		<ul style="list-style-type: none"> <li>a) The mapping to be provided is in response to a Label Request message</li> <li>b) It detects that it does not have any more label resources</li> </ul>		
9.1.8	RFC 3036, Section 3.5.8.1, A.1.1	<p>Verify that the SUT will propagate multiple Label Requests for the same FEC if</p> <ul style="list-style-type: none"> <li>a) It does not have merge capabilities</li> <li>b) The Label Request's received are not duplicate request's</li> </ul>	Mode_1, Mode_2, Mode_3, Mode_4	MAY
9.1.9	RFC 3036, Section 3.5.8.1, A.1.1	<p>Verify that the SUT will not propagate multiple Label Requests for the same FEC if</p> <ul style="list-style-type: none"> <li>a) It is merge capable</li> <li>b) The Label Request's are not duplicate request's</li> </ul>	Mode_5, Mode_6, Mode_13, Mode_14	MAY
9.1.10	RFC 3036, Section 3.5.8.1	<p>Verify that the SUT will send a Label Request if</p> <ul style="list-style-type: none"> <li>a) Its mode of operation is Downstream on Demand</li> <li>b) It is the ingress for the FEC in consideration</li> </ul>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_13, Mode_14	MAY
9.1.11	RFC 3036, Section 3.5.8.1, A.1.1	<p>Verify that the SUT will not propagate multiple requests for the same FEC and respond with a Label Mapping if</p> <ul style="list-style-type: none"> <li>a) It is merge capable</li> <li>b) It is operating in Ordered Mode</li> <li>c) The received Label Request's messages are not duplicate request's</li> <li>d) It already has a label mapping from its downstream for the FEC in consideration</li> </ul>	Mode_5, Mode_6	MAY
9.1.12	RFC 3036, Section A.1.1	<p>Verify that the SUT will silently discard a duplicate Label Request message</p>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
9.1.13	RFC 3036, Section A.1.10	<p>Verify that the SUT will send a Label Request to the peer if</p> <ul style="list-style-type: none"> <li>a) If the Request procedure being used is RequestRetry</li> </ul>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	SHOULD



TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		<p>b) It had received a Notification (No Route) message earlier for the FEC in consideration in response to one of its requests</p> <p>c) Request Retry timer has expired</p>		
9.1.14	RFC 3036, Section A.1.11	<p>Verify that the SUT will send a Label Request to the peer if</p> <p>c) If the Request procedure being used is RequestRetry</p> <p>d) It had received a Notification (Loop Detected) message earlier for the FEC in consideration in response to one of its requests</p> <p>d) Request Retry timer has expired</p>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_13, Mode_14	SHOULD
9.1.15	RFC 3036, Section A.1.12	<p>Verify that the SUT will send a pending Label Request(s) after it receives a Notification (Resources Available) message if</p> <p>a) It had received a Notification (No Resources) message earlier from the peer in consideration</p>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
9.1.16	RFC 3036, Section A.1.12	<p>Verify that the SUT will propagate Label Request(s) if</p> <p>a) It receives a Notification (Resources Available) from its peer</p> <p>b) It had received a Notification (No Resources) message earlier from the peer in consideration</p>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
9.1.17	RFC 3036, Section 3.4.1	<p>Verify that the SUT will responds with a Notification message and will not propagate the Label Request if</p> <p>a) It receives a Label Request with a Wildcard FEC element</p>	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	MAY
10.1.1	RFC 3036,	Verify that the SUT will	Mode_1, Mode_2,	SHOULD

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
	Section 3.5.9.1, A.1.3	silently discard a Label Request Abort message if a) It has sent a label mapping to its peer for the FEC in consideration thereby satisfying the request	Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_7, Mode_8, Mode_11, Mode_12, Mode_13, Mode_14	
10.1.2	RFC 3036, Section 3.5.9.1, A.1.3	Verify that the SUT will not respond with a Notification (Label request Aborted) message if a) A Label Request Abort message is received and the msg ID in Request Message ID TLV does not match the msg ID of a pending label request	Mode_1, Mode_2, Mode_5 , Mode_6, Mode_7, Mode_8	MUST
10.1.3	RFC 3036, Section 3.5.9.1, A.1.3	Verify that the SUT will respond with a Notification (Label Request Aborted) message if a) It has not sent a Label Mapping to its peer for the FEC in consideration in response to a request	Mode_1, Mode_2, Mode_5 , Mode_6, Mode_7, Mode_8	MUST
10.1.4	RFC 3036, Section 3.5.9.1, A.1.3	Verify that the SUT will propagate a Label Request Abort message if a) It has not sent a Label Mapping to its upstream peer for the FEC in consideration b) It has no other requests from upstream pending for the FEC in consideration c) It has sent a Label Request to its next hop	Mode_1, Mode_2, Mode_5 , Mode_6, Mode_7, Mode_8	MAY
10.1.5	RFC 3036, Section 3.5.9.1, A.1.7	Verify that the SUT will abort a request made for a mapping from its previous next hop if a) It detects a change in next hop for the FEC b) It is operating in Conservative retention mode	Mode_1, Mode_2, Mode_3, Mode_4, Mode_5 , Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MAY
10.1.6	RFC 3036, Section 3.5.9.1, A.1.3	Verify that the SUT will not propagate the Label Request Abort message if a) The msg ID in Request Message ID TLV does not match the msg ID of a pending label request	Mode_1, Mode_2, Mode_5 , Mode_6, Mode_7, Mode_8	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
10.1.7	RFC 3036, Section 3.5.9.1	Verify that the SUT will abort multiple label request's if a) It has sent multiple Label Request messages for the FEC as it does not have merge capabilities b) It detects a change in next hop for the FEC. c) It is operating in Conservative mode	Mode_1, Mode_2	MAY
10.1.8	RFC 3036, Section 3.5.9.1	Verify that the SUT will include a Request Message ID TLV with the correct Request Message ID in the Notification (Label Request Aborted) message when a request is successfully aborted	Mode_1, Mode_2, Mode_5, Mode_6, Mode_7, Mode_8	MUST
10.1.9	RFC 3036, Section 3.5.9.1	Verify that the SUT will not try and "order" the Label Request Abort procedure	Mode_1, Mode_2, Mode_5, Mode_6, Mode_7, Mode_8	MUST
10.1.10	RFC 3036, Section A.1.3	Verify that the SUT will not propagate a Label Request Abort message if a) It is merge capable b) It has other requests from upstream pending for the FEC in consideration c) It has sent a Label Request downstream	Mode_5, Mode_6, Mode_7, Mode_8	MUST
11.1.1	RFC 3036, Section 3.5.10.1	Verify that the SUT will withdraw all labels sent upstream through the session if a) It receives a Label Withdraw for a Wildcard FEC element from downstream b) It is operating in Ordered Mode	Mode_1, Mode_2, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12	MAY
11.1.2	RFC 3036, Section 3.5.10.1	Verify that the SUT will withdraw the corresponding labels for a FEC sent upstream through the session if a) It receives a Label Withdraw for a Wildcard FEC element from	Mode_1, Mode_2, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		<p>downstream</p> <p>b) It receives a Label TLV in the Label Withdraw message</p> <p>c) It is operating in Ordered Mode</p>		
11.1.3	RFC 3036, Section 3.5.10.1, A.1.5	<p>Verify that the SUT will withdraw the label sent upstream for a FEC learned through the session if</p> <p>a) It receives a Label Withdraw message for the FEC with a Label TLV from downstream</p> <p>b) It is operating in Ordered Mode</p>	Mode_1, Mode_2, Mode_5, Mode_6, Mode_7, Mode_8, Mode_11, Mode_12	MAY
11.1.4	RFC 3036, Section 3.5.10.1, A.1.7	<p>Verify that the SUT will withdraw all advertised labels for the FEC if</p> <p>a) It detects a change in next hop for the FEC and it does not have a next hop for the FEC</p> <p>b) It had distributed labels to its peers for the FEC</p>	All	MAY
11.1.5	RFC 3036, Section 3.5.10.1	<p>Verify that the SUT will withdraw the label bound to a FEC learned through a session if</p> <p>a) The SUT does not have merge capabilities</p> <p>b) Multiple LSPs had been set up for the same FEC</p> <p>c) It receives a Label Withdraw message for the FEC with a Label TLV</p> <p>d) It is operating in Ordered mode</p>	Mode_1, Mode_2	MAY
11.1.6	RFC 3036, Section 3.5.10.1	<p>Verify that the SUT will withdraw label(s) bound to a FEC learned through session if</p> <p>a) Multiple LSPs had been set up for the same FEC</p> <p>b) It receives a Label Withdraw message for the FEC with no Label TLV</p> <p>c) It is operating in Ordered Mode</p> <p>d) It does not have merge</p>	Mode_1, Mode_2	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		capabilities		
11.1.7	RFC 3036, Section A.1.5	Verify that the SUT will generate the event "Recognize New FEC" for a FEC if a) It receives a Label Withdraw for the FEC in consideration b) Its mode of operation is Downstream on Demand, Independent	Mode_3, Mode_4, Mode_13, Mode_14	MAY
12.1.1	RFC 3036, Section 3.5.11.1, A.1.2	Verify that the SUT will send a Label Release message if a) It detects a loop (Max hop count) in the Label Mapping message b) It already has a mapping for the FEC in consideration but the label received in this mapping does not match the earlier one	Mode_1(ATM only), Mode_2, Mode_3 (ATM only), Mode_4, Mode_5 (ATM only), Mode_6, Mode_7 (ATM only), Mode_8, Mode_12, Mode_13 (ATM only), Mode_14	MUST
12.1.2	RFC 3036, Section 3.5.11.1, A.1.2	Verify that the SUT will send a Label Release message if a) It detects a loop (Path vector) in the Label Mapping message b) It already has a mapping for the FEC in consideration but the label received in this mapping does not match the earlier one	Mode_2, Mode_4, Mode_6, Mode_8, Mode_12, Mode_14	MUST
12.1.3	RFC 3036, Section 3.5.11.1, A.1.2	Verify that the SUT will send a Label Release message if a) It receives a Label Mapping for the FEC in consideration b) It already has a mapping for the FEC in consideration but the label received in this mapping does not match the earlier one	Mode_2, Mode_4, Mode_6, Mode_8, Mode_12, Mode_14	MUST
12.1.4	RFC 3036, Section 3.5.11.1, A.1.4	Verify that the SUT will release the label for a FEC if a) It is configured to propagate releases b) No upstream peer holds a label pertaining to the FEC advertised by the	All	MUST

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		SUT		
12.1.5	RFC 3036, Section 3.5.11.1, A.1.5	Verify that the SUT will release the label for a FEC if a) It receives a Label Withdraw message for the FEC in consideration	All	MUST
12.1.6	RFC 3036, Section 3.5.11.1, A.1.7	Verify that the SUT will release the label advertised by its previous next hop for a FEC if a) It detects a change in next hop for the FEC b) It is operating in Conservative retention mode	Mode_1, Mode_2, Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_7, Mode_8, Mode_13, Mode_14	MUST
12.1.7	RFC 3036, Section 3.5.11.1, A.1.7	Verify that the SUT will not release the label for a FEC if a) It detects a change in next hop for the FEC b) It is operating in Liberal retention mode	Mode_9, Mode_10, Mode_11 , Mode_12	MUST
12.1.8	RFC 3036, Section 3.5.11.1	Verify that the SUT will release all the labels learned through session if a) It receives a Label Withdraw with a Wildcard FEC element	All	MUST
12.1.9	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label bound to FEC(s) learned through session if a) It receives a Label Withdraw for a Wildcard FEC element b) The Label Withdraw has a Label TLV	All	MUST
12.1.10	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label(s) bound to different FECs learned through session if a) It receives a Label Release for a Wildcard FEC element b) It is configured to propagate a release	All	MAY
12.1.11	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label bound to a FEC learned through session if a) It receives a Label Release for a Wildcard FEC element	All	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		b) The Label Release has a Label TLV c) It is configured to propagate releases		
12.1.12	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label bound to a FEC learned through session if a) It receives a Label Release for the FEC in consideration that has a Label TLV b) It is configured to propagate releases	All	MAY
12.1.13	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label(s) bound to a FEC learned through session if a) It does not have merge capabilities b) Multiple LSPs are setup for the same FEC c) It receives a Label Release with no Label TLV in it d) It is configured to propagate releases	Mode_1, Mode_2, Mode_3, Mode_4	MAY
12.1.14	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label(s) bound to a FEC learned through session if a) It does not have merge capabilities b) Multiple LSPs are setup for the same FEC c) It receives a Label Withdraw with no Label TLV	Mode_1, Mode_2, Mode_3, Mode_4	MUST
12.1.15	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label bound to a FEC learned through session if a) It does not have merge capabilities b) Multiple LSPs are setup for the same FEC It receives a Label Withdraw with a Label TLV	Mode_1, Mode_2, Mode_3, Mode_4	MUST
12.1.16	RFC 3036, Section 3.5.11.1	Verify that the SUT will release the label(s) bound to a FEC learned through session if a) It does not have merge capabilities	Mode_1, Mode_2, Mode_3, Mode_4	MAY

TEST CASE ID	RFC & SECTION NUMBER	TEST PURPOSE	MODE OF OPERATION	CATEGORY
		<ul style="list-style-type: none"> <li>b) Multiple LSPs are setup for the same FEC</li> <li>c) It receives a Label Release with a Label TLV</li> <li>d) It is configured to propagate releases</li> </ul>		
12.1.17	RFC 3036, Section 3.5.11.1, 2.6.2.1	<p>Verify that the SUT will release the label bound to a FEC learned through session if</p> <ul style="list-style-type: none"> <li>a) It is operating in Conservative retention mode</li> <li>b) The mapping received is not from the next hop for the FEC in consideration</li> </ul>	Mode_7, Mode_8	MUST
12.1.18	RFC 3036, Section A.1.4	<p>Verify that the SUT will not release the label for a FEC if</p> <ul style="list-style-type: none"> <li>a) One upstream peer releases the label associated with the FEC</li> <li>b) Other upstream peers still hold a label pertaining to the FEC</li> </ul>	Mode_5 , Mode_6, Mode_7, Mode_8, Mode_9, Mode_10, Mode_11, Mode_12, Mode_13, Mode_14	SHOULD
12.1.19	RFC 3036, Section 3.5.7, A.1.2	<p>Verify that the SUT will release the label if</p> <ul style="list-style-type: none"> <li>b) It receives a Label Mapping which is in response to a Label Request but the Label Mapping does not have a Request Message ID TLV</li> </ul>	Mode_1, Mode_2, Mode_3 , Mode_4, Mode_5 , Mode_6, Mode_13, Mode_14	SHOULD

## END OF DOCUMENT