LDP Results

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
Type	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR	FRR
Commit ID	3e71b5d	3d7746c	f731a65	f92f83b	c47b10c	fb13970	511684d	5cf0c43	2d67d5a
Commit Date	2017-04-02	2017-04-25	2017-05-24	2017-07-01	2017-08-09	2017-08-16	2017-08-24	2017-09-08	2017-09-14
ANVL-LDP-1.1 **MUST**	Setup Verification								
	Setup Verification Establish Hello Adjacency and check that DUT Transport Address matches configured value								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.2 **MUST**	Setup Verification								
	Setup Verification Establish LDP Session								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.3 **MUST**	Setup Verification								
	Setup Verification Request Label Mapping from DUT								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.4 **MUST**	Setup Verification								
	Setup Verification Establish 2 simultaneous LDP Sessions								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.5 **MUST**	Setup Verification								
	Setup Verification Establish 2 LDP Sessions, request Label Mapping								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.6 **MUST**	Setup Verification								
	Setup Verification Send Label Release for unsolicited Label Mapping								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.9 **MUST**	Setup Verification								
	Setup Verification Give Label Mapping to DUT								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.13 **MUST**	Setup Verification								
	Setup Verification Request Label Mapping from DUT for unknown FEC								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.14 **MUST**	Setup Verification								
	Setup Verification Establish LDP Session with ANVL as targeted peer								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-1.16 **MUST**	Setup Verification								

	`Setup Verification`
	`Send unsolicited Label Mapping to DUT using Liberal Label Retention`
	`and listen for Label Release.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.19 **MUST**	Setup Verification								
	`Setup Verification` `Send Address Message with Address List TLV`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.24 **MUST**	Setup Verification								
	`Setup Verification` `Send DUT labelled data which DUT should forward`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-1.25 **MUST**	Setup Verification								
	`Setup Verification` `Send DUT labelled data which DUT should not forward`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-2.3 **MUST**	RFC 3036, s1.2 p6 LDP Message Exchange								

	`LDP Message Exchange and Structure`
	`When an LSR chooses to establish a session with another LSR learned`
	`via the Hello message, it uses the LDP initialization procedure over`
	`TCP transport.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-2.4 **MAY**	RFC 3036, s1.2 p6 LDP Message Exchange								

	`LDP Message Exchange and Structure`
	`Upon successful completion of the initialization procedure, the two`
	`LSRs are LDP peers, and may exchange advertisement messages.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-2.6 **MUST**	RFC 3036, s1.2 p6 LDP Message Exchange								

	`LDP Message Exchange and Structure`
	`The LSR advertises a label mapping to a neighboring LSR when it wishes`
	`the neighbor to use a label.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-2.8 **MUST**	NEGATIVE RFC 3036, s1.2 p6 LDP Message Exchange								

	`LDP Message Exchange and Structure`
	`LDP uses the TCP transport for session, advertisement and notification`
	`messages; i.e., for everything but the UDP-based discovery mechanism.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-2.9 **MUST**	RFC 3036, s1.3 p7 LDP Message Structure								

	`LDP Message Exchange and Structure`
	`The Value part of a TLV-encoded object, or TLV for short, may itself`
	`contain one or more TLVs. (DUT Receiving TLV)`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-3.3 **MUST**	RFC 3036, s2.1 p8 FECs RFC 3036, s2.1 p8 FECs								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport We say that a particular address "matches" a particular address prefix if and only if that address begins with that prefix. We also say that a particular packet matches a particular LSP if and only if that LSP has an Address Prefix FEC element which matches the packet"s destination address.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-3.8 **MUST**	RFC 3036, s2.1 p9 FECs								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport If a packet matches multiple LSPs, it is mapped to the LSP whose matching prefix is the longest.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-3.9 **MUST**	RFC 3036, s2.1 p9 FECs								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport If there is no one LSP whose matching prefix is longest, the packet is mapped to one from the set of LSPs whose matching prefix is longer than the others.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-3.12 **MUST**	RFC 3036, s2.1 p9 FECs								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport A packet may match two LSPs, one with a Host Address FEC element and one with an Address Prefix FEC element; the packet is always assigned to the former.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
ANVL-LDP-3.16 **MUST**	RFC 3036, s2.2.2 p10 LDP Identifiers								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport The first four octets of the LDP Identifier octets identify the LSR and must be a globally unique value, such as a 32-bit router Id the LSR.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-3.18 **MUST**	RFC 3036, s2.2.2 p10 LDP Identifiers								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport The last two octets of LDP Identifiers for platform-wide label spaces are always both zero. (Note: this test is only valid for devices with platform-wide label spaces, and as such requires a LAN interface)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-3.21 **MUST**	RFC 3036, s2.2.4 p11 LDP Transport								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport LDP uses TCP as a reliable transport for sessions.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-3.23 **MUST**	NEGATIVE RFC 3036, s2.2.4 p11 LDP Transport								
	LDP Operation--FECs and Label Spaces, Identifiers, Sessions and Transport When multiple LDP sessions are required between two LSRs there is one TCP session for each LDP session.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.7 **MUST**	RFC 3036, s1.2 p6 LDP Message Exchange RFC 3036, s2.4.1 p12 Basic Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Discovery messages provide a mechanism whereby LSRs indicate their presence in a network by sending a Hello message periodically. To engage in LDP Basic Discovery on an interface an LSR periodically sends LDP Link Hellos out the interface.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.8 **MUST**	RFC 3036, s1.2 p6 LDP Message Exchange RFC 3036, s2.4.1 p12 Basic Discovery Mechanism RFC 3036, s3.10.1 p83 Well-known Numbers/UDP and TCP Ports								
	Basic and Extended Discovery Mechanisms This [Hello message] is transmitted as a UDP packet to the LDP port at the `all routers on this subnet" group multicast address. LDP Link Hellos are sent as UDP packets addressed to the well-known LDP discovery port for the "all routers on this subnet" group multicast address. The UDP port for LDP Hello messages is 646								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.10 **MUST**	RFC 3036, s2.4.1 p12 Basic Discovery Mechanism								
	Basic and Extended Discovery Mechanisms An LDP Link Hello sent by an LSR carries ... possibly additional information. (Receipt of Hello with Transport Address TLV)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.11 **MUST**	RFC 3036, s2.4.1 p12 Basic Discovery Mechanism								
	Basic and Extended Discovery Mechanisms An LDP Link Hello sent by an LSR carries ... possibly additional information. (Receipt of Hello with Configuration Sequence Number)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.12 **MUST**	NEGATIVE RFC 3036, s2.4.1 p12 Basic Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Receipt of an LDP Link Hello on an interface identifies a "Hello adjacency" with a potential LDP peer reachable at the link level on the interface as well as the label space the peer intends to use for the interface.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.14 **MUST**	RFC 3036, s1.2 p6 LDP Message Exchange RFC 3036, s2.4.2 p12 Extended Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Discovery messages provide a mechanism whereby LSRs indicate their presence in a network by sending a Hello message periodically. To engage in LDP Extended Discovery an LSR periodically sends LDP Targeted Hellos to a specific address.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-4.16 **MUST**	RFC 3036, s2.4.2 p12 Extended Discovery Mechanism								
	Basic and Extended Discovery Mechanisms An LDP Targeted Hello sent by an LSR carries the LDP Identifier for the label space the LSR intends to use and possibly additional optional information.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.19 **MUST**	NEGATIVE RFC 3036, s2.4.2 p12 Extended Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Extended Discovery differs from Basic Discovery in the following ways: One LSR initiates Extended Discovery with another targeted LSR, and the targeted LSR decides whether to respond to or ignore the Targeted Hello.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.20 **MUST**	RFC 3036, s2.4.2 p12 Extended Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Extended Discovery differs from Basic Discovery in the following ways: One LSR initiates Extended Discovery with another targeted LSR, and the targeted LSR decides whether to respond to or ignore the Targeted Hello.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.21 **MUST**	RFC 3036, s2.4.2 p12 Extended Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Extended Discovery differs from Basic Discovery in the following ways: A targeted LSR that chooses to respond does so by periodically sending Targeted Hellos to the initiating LSR.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-4.22 **MUST**	NEGATIVE RFC 3036, s2.4.2 p13 Extended Discovery Mechanism								
	Basic and Extended Discovery Mechanisms Receipt of an LDP Targeted Hello identifies a "Hello adjacency" with a potential LDP peer reachable at the network level and the label space the peer intends to use.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.1 **MUST**	RFC 3036, s2.5.1 p13 LDP Session Establishment								
	LDP Session Establishment and Transport Connection Establishment The exchange of LDP Discovery Hellos between two LSRs triggers LDP session establishment.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.5 **MUST**	RFC 3036, s2.5.2 p13 Transport Connection Establishment								
	LDP Session Establishment and Transport Connection Establishment LSR1 (DUT) determines the transport addresses to be used at its end (A1) and LSR2"s end (A2) of the LDP TCP connection.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-5.9 **MUST**	RFC 3036, s2.5.2 p13 Transport Connection Establishment LDP Session Establishment and Transport Connection Establishment If LSR2 (ANVL) uses the Transport Address optional object, A2 is the address LSR2 advertises via the optional object. (DUT is passive)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.10 **MUST**	RFC 3036, s2.5.2 p13 Transport Connection Establishment LDP Session Establishment and Transport Connection Establishment If LSR2 (ANVL) uses the Transport Address optional object, A2 is the address LSR2 advertises via the optional object. (DUT is active)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.12 **MUST**	RFC 3036, s2.5.2 p14 Transport Connection Establishment LDP Session Establishment and Transport Connection Establishment LSR1 (DUT) determines whether it will play the active or passive role in session establishment by comparing addresses A1 and A2 as unsigned integers. If A1 > A2, LSR1 plays the active role; otherwise it is passive.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.13 **MUST**	RFC 3036, s2.5.2 p14 Transport Connection Establishment LDP Session Establishment and Transport Connection Establishment If A1 and A2 are not in the same address family, they are incomparable, and no session can be established. (Basic Hello)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.19 **MUST**	RFC 3036, s2.5.2 p14 Transport Connection Establishment LDP Session Establishment and Transport Connection Establishment An LSR MUST advertise the same transport address in all Hellos that advertise the same label space.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-5.20 **MUST**	NEGATIVE RFC 3036, s2.5.2 p14 Transport Connection Establishment LDP Session Establishment and Transport Connection Establishment An LSR MUST advertise the same transport address in all Hellos that advertise the same label space.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.1 **MUST**	RFC 3036, s2.5.3 p14 Session Initialization Session Initialization After LSR1 and LSR2 establish a transport connection they negotiate session parameters by exchanging LDP Initialization messages.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.4 **MUST**	RFC 3036, s2.5.3 p15 Session Initialization Session Initialization The Initialization message carries both the LDP Identifier for the sender"s (active LSR"s) label space and the LDP Identifier for the receiver"s (passive LSR"s) label space.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-6.5 **MUST**	NEGATIVE RFC 3036, s2.5.3 p15 Session Initialization								
	Session Initialization The Initialization message carries both the LDP Identifier for the sender"s (active LSR"s) label space and the LDP Identifier for the receiver"s (passive LSR"s) label space.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.6 **MUST**	NEGATIVE RFC 3036, s2.5.3 p15 Session Initialization								
	Session Initialization The Initialization message carries both the LDP Identifier for the sender"s (active LSR"s) label space and the LDP Identifier for the receiver"s (passive LSR"s) label space.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.8 **MUST**	RFC 3036, s2.5.3 p15 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the passive role and receives an acceptable Initialization message, LSR1 replies with an Initialization message of its own to propose the parameters it wishes to use and a KeepAlive message to signal acceptance of LSR2s parameters.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.11 **MUST**	RFC 3036, s2.5.3 p15 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the passive role and if LSR1 cannot find a matching Hello adjacency it sends a Session Rejected/No Hello Error Notification message and closes the TCP connection.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.12 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the passive role and if LSR1 receives a KeepAlive in response to its Initialization message, the session is operational from LSR1"s point of view.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.13 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the passive role and if LSR1 receives an Error Notification message, LSR2 has rejected its proposed session and LSR1 closes the TCP connection.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.14 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the active role and if LSR1 receives an Error Notification message, LSR2 has rejected its proposed session and LSR1 closes the TCP connection.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-6.15 **MUST**	NEGATIVE RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the active role and if LSR1 does not receive an Initialization Message or a Keep Alive from the peer, LSR1 closes closes the TCP connection.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.16 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the active role and if LSR1 receives an acceptable Initialization message, it replies with a KeepAlive message.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.17 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization When LSR1 (DUT) plays the active role and if LSR1 receives a KeepAlive message, LSR2 has accepted its proposed session parameters.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.19 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization An LSR must throttle its session setup retry attempts with an exponential backoff in situations where Initialization messages are being NAK"d.								
	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-6.21 **MUST**	RFC 3036, s2.5.3 p16 Session Initialization								
	Session Initialization The session establishment setup attempt following a NAK"d Initialization message must be delayed no less than 15 seconds. [The specific session establishment action that must be delayed is the attempt to open the session transport connection by the LSR playing the active role.]								
	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.1 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	Initialization State Machine and Session Maintainance In state INITIALIZED, action is to transmit Initialization msg (Active Role).								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.2 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	Initialization State Machine and Session Maintainance In state INITIALIZED if LSR receives an acceptable Initialization msg (Passive Role), action is to transmit Initialization msg and KeepAlive msg.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.3 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	Initialization State Machine and Session Maintainance In state INITIALIZED if LSR receives any other LDP msg, action is to transmit Error Notification msg (NAK) and close transport connection.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-7.4 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPENREC if LSR receives a KeepAlive msg, the LSP is operational. (DUT is passive)`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.5 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPENREC if LSR receives a KeepAlive msg, the LSP is operational. (DUT is active)`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.6 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPENREC if LSR receives any other LDP msg, the action is to transmit Error Notification msg (NAK) and close transport connection. (DUT is passive)`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.7 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPENREC if LSR receives any other LDP msg, the action is to transmit Error Notification msg (NAK) and close transport connection. (DUT is active)`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.8 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPENSENT if LSR receives an acceptable Initialization msg, the action is to transmit KeepAlive msg.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.9 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPENSENT if LSR receives any other LDP msg, the action is to transmit Error Notification msg (NAK) and close transport connection.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.11 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPERATIONAL if LSR receives other LDP msgs, the session remains OPERATIONAL.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.12 **MUST**	RFC 3036, s2.5.4 p18 Initialization State Machine								
	`Initialization State Machine and Session Maintainance In state OPERATIONAL if a timeout occurs, the action is to transmit Shutdown msg and close transport connection.`								
	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2

ANVL-LDP-7.15
MUST
	RFC 3036, s2.5.5 p20 Maintaining Hello Adjacencies
	`Initialization State Machine and Session Maintainance`
	`An LSR maintains a hold timer with each Hello adjacency which it`
	`restarts when it receives a Hello that matches the adjacency.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-7.16
MUST
	RFC 3036, s2.5.5 p20 Maintaining Hello Adjacencies
	`Initialization State Machine and Session Maintainance`
	`If the timer expires without receipt of a matching Hello from the`
	`peer, LDP concludes that the peer no longer wishes to label switch`
	`using that label space for that link (or target, in the case of`
	`Targeted Hellos) or that the peer has failed.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-7.17
MUST
	RFC 3036, s2.5.5 p20 Maintaining Hello Adjacencies
	`Initialization State Machine and Session Maintainance`
	`When the last Hello adjacency for a LDP session is deleted, the LSR`
	`terminates the LDP session by sending a Notification message and`
	`closing the transport connection.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-7.18
MUST
	RFC 3036, s2.5.6 p20 Maintaining LDP Sessions
	`Initialization State Machine and Session Maintainance`
	`An LSR maintains a KeepAlive timer for each peer session which it`
	`resets whenever it receives an LDP PDU from the session peer.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-7.19
MUST
	RFC 3036, s2.5.6 p20 Maintaining LDP Sessions
	`Initialization State Machine and Session Maintainance`
	`If the KeepAlive timer expires without receipt of an LDP PDU from the`
	`peer the LSR concludes that the transport connection is bad or that`
	`the peer has failed, and it terminates the LDP session by closing the`
	`transport connection.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-7.21
MUST
	RFC 3036, s2.5.6 p20 Maintaining LDP Sessions
	RFC 3036, s3.5.4.1 p63 KeepAlive Message Procedures
	`Initialization State Machine and Session Maintainance`
	`After an LDP session has been established, an LSR must arrange that`
	`its peer receive an LDP PDU from it at least every KeepAlive time`
	`period to ensure the peer restarts the session KeepAlive timer.`
	`The LSR may send any protocol message to meet this requirement.`
	
	`The KeepAlive Timer mechanism described in Section "Maintaining LDP`
	`Sessions" resets a session KeepAlive timer every time an LDP PDU is`
	`received on the session TCP connection.  The KeepAlive Message is`
	`provided to allow reset of the KeepAlive Timer in circumstances where`
	`an LSR has no other information to communicate to an LDP peer.  An LSR`
	`must arrange that its peer receive an LDP Message from it at least`
	`every KeepAlive Time period.  Any LDP protocol message will do but, in`
	`circumstances where no other LDP protocol messages have been sent`
	`within the period, a KeepAlive message must be sent.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-7.22 **MUST**	RFC 3036, s2.5.6 p20 Maintaining LDP Sessions RFC 3036, s3.5.4.1 p63 KeepAlive Message Procedures								
	Initialization State Machine and Session Maintainance The LSR may send any protocol message to meet this requirement [KeepAlive requirement]. The KeepAlive Timer mechanism described in Section "Maintaining LDP Sessions" resets a session KeepAlive timer every time an LDP PDU is received on the session TCP connection. The KeepAlive Message is provided to allow reset of the KeepAlive Timer in circumstances where an LSR has no other information to communicate to an LDP peer. An LSR must arrange that its peer receive an LDP Message from it at least every KeepAlive Time period. Any LDP protocol message will do but, in circumstances where no other LDP protocol messages have been sent within the period, a KeepAlive message must be sent.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.23 **MUST**	RFC 3036, s2.5.6 p20 Maintaining LDP Sessions RFC 3036, s3.5.4.1 p63 KeepAlive Message Procedures								
	Initialization State Machine and Session Maintainance After an LDP session has been established, an LSR must arrange that its peer receive an LDP PDU from it at least every KeepAlive time period to ensure the peer restarts the session KeepAlive timer. In circumstances where an LSR has no other information to communicate to its peer, it sends a KeepAlive message. The KeepAlive Timer mechanism described in Section "Maintaining LDP Sessions" resets a session KeepAlive timer every time an LDP PDU is received on the session TCP connection. The KeepAlive Message is provided to allow reset of the KeepAlive Timer in circumstances where an LSR has no other information to communicate to an LDP peer. An LSR must arrange that its peer receive an LDP Message from it at least every KeepAlive Time period. Any LDP protocol message will do but, in circumstances where no other LDP protocol messages have been sent within the period, a KeepAlive message must be sent.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-7.25 **MAY**	RFC 3036, s2.5.6 p20 Maintaining LDP Sessions								
	Initialization State Machine and Session Maintainance An LSR may choose to terminate an LDP session with a peer at any time; should it choose to do so, it informs the peer with a Shutdown message.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-8.5 **MAY**	RFC 3036, s2.6.1.1 p21 Independent Label Distribution Control RFC 3036, s2.8.3 p28 Discussion								
	Label Distribution and Management When using independent LSP control, each LSR may advertise label mappings to its neighbors at any time it desires. In the case of independent label distribution, an LSR may originate a Label Mapping message for an FEC before receiving a Label Mapping message from its downstream peer for that FEC.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-8.6 **MUST**	RFC 3036, s2.6.1.1 p21 Independent Label Distribution Control								
	Label Distribution and Management When operating in independent Downstream Unsolicited mode, an LSR may advertise a label mapping for a FEC to its neighbors whenever it is prepared to label-switch that FEC.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-8.20 **MUST**	RFC 3036, s2.6.2.2 p22-23 Liberal Label Retention Mode								
	Label Distribution and Management When using liberal label retention, every label mapping received from a peer LSR is retained regardless of whether the LSR is the next hop for the advertised mapping. (Unknown FEC from valid next hop)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-8.21 **MUST**	RFC 3036, s2.6.2.2 p22-23 Liberal Label Retention Mode								
	Label Distribution and Management When using liberal label retention, every label mapping received from a peer LSR is retained regardless of whether the LSR is the next hop for the advertised mapping. (Known FEC from invalid next hop)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-9.3 **MUST**	RFC 3036, s2.7 p23 LDP Identifiers and Next Hop Addresses								
	LDP Identifiers and Next Hop Addresses When the next hop for a prefix changes the LSR must retrieve the label advertised by the new next hop from the LIB for use in forwarding.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass
ANVL-LDP-9.4 **MUST**	RFC 3036, s2.7 p23 LDP Identifiers and Next Hop Addresses								
	LDP Identifiers and Next Hop Addresses To retrieve the label the LSR must be able to map the next hop address for the prefix to an LDP Identifier.								
	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-9.5 **MUST**	RFC 3036, s2.7 p23 LDP Identifiers and Next Hop Addresses								
	LDP Identifiers and Next Hop Addresses Similarly, when the LSR learns a label for a prefix from an LDP peer, it must be able to determine whether that peer is currently a next hop for the prefix to determine whether it needs to start using the newly learned label when forwarding packets that match the prefix.								
	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: unpredict	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-9.8 **MUST**	RFC 3036, s2.7 p24 LDP Identifiers and Next Hop Addresses								
	LDP Identifiers and Next Hop Addresses An LSR sends an Address message to advertise its addresses to a peer.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-9.9 **MUST**	RFC 3036, s2.7 p24 LDP Identifiers and Next Hop Addresses								
	LDP Identifiers and Next Hop Addresses An LSR sends a Withdraw Address message to withdraw previously advertised addresses from a peer.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.2 **MUST**	RFC 3036, s3 p31 Protocol Specification								
	Protocol Specification--PDUs and FEC TLVs Each LDP PDU can carry one or more LDP messages.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-15.3 **MUST**	RFC 3036, s3 p31 Protocol Specification Protocol Specification--PDUs and FEC TLVs Note that the messages in an LDP PDU need not be related to one another.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.4 **MUST**	NEGATIVE RFC 3036, s3.1 p31 LDP PDUs Protocol Specification--PDUs and FEC TLVs Each LDP PDU is an LDP header followed by one or more LDP messages.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.5 **MUST**	RFC 3036, s3.1 p31-32 LDP PDUs Protocol Specification--PDUs and FEC TLVs Validate LDP Header from DUT. * Version: This version of the specification specifies LDP protocol version 1. * PDU Length: Two octet integer specifying the total length of this PDU in octets, excluding the Version and PDU Length fields. The maximum allowable PDU Length is negotiable when an LDP session is initialized. Prior to completion of the negotiation the maximum allowable length is 4096 bytes. * LDP Identifier: The first four octets identify the LSR and must be a globally unique value. It should be a 32-bit router Id assigned to the LSR and also used to identify it in loop detection Path Vectors. The last two octets identify a label space within the LSR. For a platform-wide label space, these should both be zero.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.7 **MUST**	RFC 3036, s3.3 p32-33 Type-Length-Value Encoding Protocol Specification--PDUs and FEC TLVs Validate LDP TLV encoding from DUT. An LDP TLV is encoded as a 2 octet field that uses 14 bits to specify a Type and 2 bits to specify behavior when an LSR doesn"t recognize the Type, followed by a 2 octet Length Field, followed by a variable length Value field.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.10 **MUST**	RFC 3036, s2.1 p8 FECs RFC 3036, s3.4.1 p34 FEC TLV RFC 3036, s3.4.1 p35 FEC TLV Protocol Specification--PDUs and FEC TLVs Each FEC is specified as a set of one or more FEC elements. A FEC is a list of one or more FEC elements. The FEC TLV encodes FEC items. Note that this version of LDP supports the use of multiple FEC Elements per FEC for the Label Mapping message only.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.11 **MUST**	RFC 3036, s3.4.1 p34-35 FEC TLV Protocol Specification--PDUs and FEC TLVs Validate FEC TLV Encoding from DUT.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-15.12 **MUST**	RFC 3036, s3.4.1 p35 FEC TLV Protocol Specification--PDUs and FEC TLVs A FEC Element value is encoded as a 1 octet field that specifies the element type, and a variable length field that is the type-dependent element value. The FEC Element value encoding is: FEC Element Type Value Type name Wildcard 0x01 No value; i.e., 0 value octets (see below) Prefix 0x02 See below. Host Address 0x03 Full host address; see below.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.15 **MUST**	NEGATIVE RFC 3036, s3.4.1 p35 FEC TLV Protocol Specification--PDUs and FEC TLVs Note that this version of LDP supports the use of multiple FEC Elements per FEC for the Label Mapping message only. The use of multiple FEC Elements in other [than Label Mapping] messages is not permitted in this version of LDP.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.16 **MUST**	NEGATIVE RFC 3036, s3.4.1 p35 FEC TLV Protocol Specification--PDUs and FEC TLVs The Wildcard FEC Element is to be used only in the Label Withdraw and Label Release Messages. (Label Request with Wildcard FEC)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.18 **MUST**	RFC 3036, s3.4.1 p35 FEC TLV RFC 3036, s3.5.10.1 p76 Label Withdraw Message Procedures Protocol Specification--PDUs and FEC TLVs The Wildcard FEC Element indicates the withdraw/release is to be applied to all FECs associated with the label within the following label TLV. The FEC TLV may contain the Wildcard FEC Element....if the Label Withdraw message contains an optional Label TLV, then the label is to be withdrawn from all FECs to which it is bound.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.19 **MUST**	NEGATIVE RFC 3036, s3.4.1 p35 FEC TLV RFC 3036, s3.5.10.1 p76 Label Withdraw Message Procedures Protocol Specification--PDUs and FEC TLVs The Wildcard FEC Element must be the only FEC Element in the FEC TLV. The FEC TLV may contain the Wildcard FEC Element; if so, it may contain no other FEC Elements.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-15.23 **SHOULD**	RFC 3036, s3.4.1.1 p37 FEC Procedures Protocol Specification--PDUs and FEC TLVs If in decoding a FEC TLV an LSR encounters a FEC Element with an Address Family it does not support, it should stop decoding the FEC TLV, abort processing the message containing the TLV, and send an "Unsupported Address Family" Notification message to its LDP peer signaling an error.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-15.24 **SHOULD**	RFC 3036, s3.4.1.1 p37 FEC Procedures Protocol Specification--PDUs and FEC TLVs If it encounters a FEC Element type it cannot decode, it should stop decoding the FEC TLV, abort processing the message containing the TLV, and send an "Unknown FEC" Notification message to its LDP peer signaling an error.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-16.2 **MUST**	RFC 3036, s3.4.2.1 p37 Generic Label TLV Protocol Specification--Label, Address, and Hop Count TLVs Validate Generic Label TLV encoding from DUT.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-16.14 **MUST**	NEGATIVE RFC 3036, s3.4.3 p40 Address List TLV Protocol Specification--Label, Address, and Hop Count TLVs The following address encodings are defined by this version of the protocol: Address Family Address Encoding IPv4 4 octet full IPv4 address IPv6 16 octet full IPv6 address								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-18.2 **SHOULD**	RFC 3036, s3.4.4.1 p40 Hop Count Procedures Hop Count Procedures During setup of an LSP an LSR R may receive a Label Mapping message for the LSP that contains the Hop Count TLV. If it does, it should record the hop count value and not release the mapping.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-20.1 **MUST**	NEGATIVE RFC 3036, s3.4.6 p43 Status TLV Status TLV Notification messages carry Status TLVs to specify events being signaled.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-20.2 **MUST**	RFC 3036, s3.4.6 p44 Status TLV Status TLV Validate Status TLV encoding from DUT.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-20.4	RFC 3036, s3.4.6 p44 Status TLV								
MUST	Status TLV F bit should be the same as the setting of the F-bit in the Status Code field.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-20.8	RFC 3036, s3.4.6 p44 Status TLV								
SHOULD	Status TLV Forward bit (F-Bit)....If clear (=0), the notification should not be forwarded.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-20.12	RFC 3036, s3.4.6 p45 Status TLV								
MUST	Status TLV A message other than a Notification message may carry a Status TLV as an Optional Parameter.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.1	RFC 3036, s3.5 p45 LDP Messages								
MUST	LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Upon receipt of an unknown [LDP] message, if Unknown Message bit (U) is clear (=0), a notification is returned to the message originator.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.2	RFC 3036, s3.5 p45 LDP Messages								
MUST	LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Upon receipt of an unknown [LDP] message, if Unknown Message bit (U) ...is set (=1), the unknown message is silently ignored.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.5	RFC 3036, s3.5.1 p45 Notification Message								
MUST	LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Validate Notification Message TLV encoding from DUT								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.11	RFC 3036, s3.5.4 p63 KeepAlive Message								
MUST	LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Validate KeepAlive Messages from DUT								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.13	RFC 3036, s3.5.5 p64 Address Message								
MUST	LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Validate Address Message format from DUT.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.14	RFC 3036, s3.5.5.1 p65 Address Message Procedures								
SHOULD	LDP Messages, Notification Messages, KeepAlive Messages, Address Messages When a new LDP session is initialized and before sending Label Mapping or Label Request messages an LSR should advertise its interface addresses with one or more Address messages.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-21.15 **SHOULD**	RFC 3036, s3.5.5.1 p65 Address Message Procedures								
	`LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Whenever an LSR "activates" a new interface address, it should advertise the new address with an Address message.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.16 **SHOULD**	RFC 3036, s3.5.5.1 p65 Address Message Procedures								
	`LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Whenever an LSR "de-activates" a previously advertised address, it should withdraw the address with an Address Withdraw message; see Section "Address Withdraw Message".`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.17 **MUST**	RFC 3036, s3.5.5.1 p65 Address Message Procedures								
	`LDP Messages, Notification Messages, KeepAlive Messages, Address Messages If an LSR does not support the Address Family specified in the Address List TLV, it should send an "Unsupported Address Family" Notification to its LDP signalling an error and abort processing the message.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-21.18 **MUST**	RFC 3036, s3.5.6 p65 Address Withdraw Message								
	`LDP Messages, Notification Messages, KeepAlive Messages, Address Messages Validate Address Withdraw Message format from DUT.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.1 **MUST**	RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	`Events Signaled by Notification Messages Malformed LDP PDUs or Messages that are part of the LDP Discovery mechanism are handled by silently discarding them.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.2 **MUST**	RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	`Events Signaled by Notification Messages Malformed LDP PDUs or Messages that are part of the LDP Discovery mechanism are handled by silently discarding them. (Targeted Hello)`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.3 **MUST**	RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	`Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if (1) The LDP Identifier in the PDU header is unknown to the receiver....This is a fatal error signaled by the Bad LDP Identifier Status Code.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.4 **MUST**	RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	`Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if (1) The LDP Identifier in the PDU header is...known but is not the LDP Identifier associated by the receiver with the LDP peer for this LDP session. This is a fatal error signaled by the Bad LDP Identifier Status Code.`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-22.5 **MUST**	RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (2) The LDP protocol version is not supported by the receiver....This is a fatal error signaled by the Bad Protocol Version Status Code. (DUT takes passive role)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
ANVL-LDP-22.6 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (2) The LDP protocol version is not supported by the receiver, or it is supported but is not the version negotiated for the session during session establishment. This is a fatal error signaled by the Bad Protocol Version Status Code.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.8 **MUST**	RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (2) The LDP protocol version is not supported by the receiver....This is a fatal error signaled by the Bad Protocol Version Status Code. (DUT takes active role)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
ANVL-LDP-22.9 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (3) The PDU Length field is too small (14).... This is a fatal error signaled by the Bad PDU Length Status Code.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.10 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (3) The PDU Length field is...too large (> maximum PDU length). This is a fatal error signaled by the Bad PDU Length Status Code. (PDU contains random data)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.11 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (3) The PDU Length field is...too large (> maximum PDU length). This is a fatal error signaled by the Bad PDU Length Status Code. (PDU contains Label Mapping messages)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-22.12 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP PDU received on a TCP connection for an LDP session is malformed if: (3) The PDU Length field is...too large (> maximum PDU length). This is a fatal error signaled by the Bad PDU Length Status Code. (PDU contains Label Request messages)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.13 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP Message is malformed if: (1) The Message Type is unknown. If the Message Type is 0x8000 (high order bit = 0) it is an error signaled by the Unknown Message Type Status Code. If the Message Type is >= 0x8000 (high order bit = 1) it is silently discarded.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.15 **MUST**	NEGATIVE RFC 3036, 3.5.1.2.1 p49 Malformed PDU or Message								
	Events Signaled by Notification Messages An LDP Message is malformed if: (3) The message is missing one or more Mandatory Parameters. This is a non-fatal error signalled by the Missing Message Parameters Status Code.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.16 **MUST**	RFC 3036, 3.5.1.2.2 p50 Unknown or Malformed TLV								
	Events Signaled by Notification Messages Malformed TLVs contained in LDP messages that are part of the LDP Discovery mechanism are handled by silently discarding the containing message.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.17 **MUST**	RFC 3036, 3.5.1.2.2 p50 Unknown or Malformed TLV								
	Events Signaled by Notification Messages A TLV contained in an LDP message received on a TCP connection of an LDP is malformed if: (1) The TLV Length is too large, that is, indicates that the TLV extends beyond the end of the containing message. This is a fatal error signaled by the Bad TLV Length Status Code.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.18 **MUST**	RFC 3036, 3.5.1.2.2 p50 Unknown or Malformed TLV								
	Events Signaled by Notification Messages A TLV contained in an LDP message received on a TCP connection of an LDP is malformed if: (2) The TLV type is unknown. If the TLV type is 0x8000 (high order bit 0) it is an error signaled by the Unknown TLV Status Code. If the TLV type is >= 0x8000 (high order bit 1) the TLV is silently dropped.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-22.19 **MUST**	RFC 3036, 3.5.1.2.2 p50 Unknown or Malformed TLV Events Signaled by Notification Messages A TLV contained in an LDP message received on a TCP connection of an LDP is malformed if: (3) The TLV Value is malformed. This occurs when the receiver handles the TLV but cannot decode the TLV Value. This is interpreted as indicative of a bug in either the sending or receiving LSR. It is a fatal error signaled by the Malformed TLV Value Status Code.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
ANVL-LDP-22.20 **MUST**	RFC 3036, s3.5.1.2.3 p48 Session KeepAlive Timer Expiration Events Signaled by Notification Messages Timer expiration is a fatal error signaled by the KeepAlive Timer Expired Status Code.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.21 **MUST**	RFC 3036, s3.5.1.2.4 p51 Unilateral Session Shutdown Events Signaled by Notification Messages This is a fatal event signaled by the Shutdown Status Code. The Notification Message may optionally include an Extended Status TLV to provide a reason for the Shutdown. The sending LSR terminates the session immediately after sending the Notification.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-22.23 **MUST**	RFC 3036, s3.5.1.2.7 p51 Internal Errors Events Signaled by Notification Messages An LDP implementation may be capable of detecting problem conditions specific to its implementation. When such a condition prevents an implementation from interacting correctly with a peer, the implementation should, when capable of doing so, use the Internal Error Status Code to signal the peer. This is a fatal error.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-23.1 **MUST**	RFC 3036, s3.5.2 p52 Hello Messages Hello Messages Validate Hello Messages encoding from DUT								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-23.3 **MUST**	RFC 3036, s3.5.2 p52 Hello Messages Hello Messages Hold Time: A value of 0 means use the default, which is 15 seconds for Link Hellos. A value of 0xffff means infinite.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-23.4 **MUST**	RFC 3036, s3.5.2 p52 Hello Messages Hello Messages Hold Time: A value of 0 means use the default, which is 45 seconds for Targeted Hellos.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
ANVL-LDP-23.8 **MUST**	RFC 3036, s3.5.2 p53 Hello Messages Hello Messages Reserved - This field is reserved. It must be set to zero on transmission and ignored on receipt.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-23.10 **MAY**	RFC 3036, s3.5.2 p52 Hello Messages								
	Hello Messages Optional TLV Configuration Sequence Number - Specifies a 4 octet unsigned configuration sequence number that identifies the configuration state of the sending LSR. Used by the receiving LSR to detect configuration changes on the sending LSR.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-23.13 **MUST**	RFC 3036, s3.5.2.1 p54 Hello Message Procedures								
	Hello Messages We recommend that the interval between Hello transmissions be at most one third of the Hello hold time.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
ANVL-LDP-23.14 **MUST**	NEGATIVE RFC 3036, s3.5.2.1 p54 Hello Message Procedures								
	Hello Messages Received LDP Hello Message Step 2: If the Hello is not acceptable, the LSR ignores it.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-23.16 **MUST**	NEGATIVE RFC 3036, s3.5.2.1 p54 Hello Message Procedures								
	Hello Messages A Link Hello is acceptable if the interface on which it was received has been configured for label switching.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.1 **MUST**	RFC 3036, s3.5.3 p55 Initialization Message								
	Initialization Messages Validate Initialization Messages encoding from DUT								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.3 **MUST**	RFC 3036, s3.5.3 p56 Initialization Messages								
	Initialization Messages A, Label Advertisement Discipline - Indicates the type of Label advertisement. A value of 0 means Downstream Unsolicited advertisement.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.8 **MUST**	RFC 3036, s3.5.3 p57 Initialization Messages								
	Initialization Messages D, Loop Detection - Indicates whether loop detection based on path vectors is enabled. A value of 0 means loop detection is disabled.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.10 **MUST**	RFC 3036, s3.5.3 p57 Initialization Messages								
	Initialization Messages PVLim, Path Vector Limit - The configured maximum path vector length. Must be 0 if loop detection is disabled (D = 0).								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-24.14 **MUST**	RFC 3036, s3.5.3 p57 Initialization Messages								
	Initialization Messages Reserved - This field is reserved. It must be set to zero on transmission and ignored on receipt.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.15 **MUST**	RFC 3036, s3.5.3 p57 Initialization Messages								
	Initialization Messages Max PDU Length - Two octet unsigned integer that proposes the maximum allowable length for LDP PDUs for the session. A value of 255 or less specifies the default maximum length of 4096 octets.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.19 **MUST**	RFC 3036, s3.5.3 p57 Initialization Messages								
	Initialization Messages Receiver LDP Identifier - If there is no matching Hello adjacency, the LSR must send a Session Rejected/No Hello Notification message in response to the Initialization message and not establish the session. (Receiver LDP ID: incorrect LSR Id, correct label space)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-24.20 **MUST**	RFC 3036, s3.5.3 p57 Initialization Messages								
	Initialization Messages Receiver LDP Identifier - If there is no matching Hello adjacency, the LSR must send a Session Rejected/No Hello Notification message in response to the Initialization message and not establish the session. (Receiver LDP ID: correct LSR Id, incorrect label space)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-26.7 **MUST**	RFC 3036, s3.5.7.1 p67 Label Mapping Message Procedures								
	Label Mapping Messages An LSR receiving a Label Mapping message from a downstream LSR for a Prefix or Host Address FEC Element should not use the label for forwarding unless its routing table contains an entry that exactly matches the FEC Element.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-26.8 **MUST**	RFC 3036, s3.5.7.1.1 p67 Independent Control Mapping								
	Label Mapping Messages An LSR configured for Independent Control and Downstream Unsolicited mode sends a mapping message when the LSR recognizes a new FEC via the forwarding table.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
ANVL-LDP-26.11 **MUST**	RFC 3036, s3.5.7.1.1 p67 Independent Control Mapping								
	Label Mapping Messages An LSR configured for Independent Control sends a mapping message when when the attributes of a mapping change.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2

ANVL-LDP-26.12
MUST
	RFC 3036, s3.5.7.1.1 p67 Independent Control Mapping
	`Label Mapping Messages`
	`An LSR configured for Independent Control sends a mapping message when`
	`receiving a mapping from the downstream next hop and no upstream`
	`mapping has been created.`								

	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-27.7
SHOULD
	RFC 3036, s3.5.8.1 p71 Label Request Message Procedures
	`Label Request Messages`
	`The receiving LSR should respond to a Label Request message with a`
	`Label Mapping for the requested label or with a Notification message`
	`indicating why it cannot satisfy the request.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-27.8
MUST
	RFC 3036, s3.5.8.1 p71 Label Request Message Procedures
	RFC 3036, s3.5.8.1 p71 Label Request Message Procedures
	`Label Request Messages`
	`When the FEC for which a label is requested is a Prefix FEC Element or`
	`a Host Address FEC Element, the receiving LSR uses its routing table`
	`to determine its response.  Unless its routing table includes an entry`
	`that exactly matches the requested Prefix or Host Address, the LSR`
	`must respond with a No Route Notification message.`
	
	`A Notification message that signals a request cannot be satisfied`
	`contains one of the following Status Codes: (1) No Route.`								

	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL

ANVL-LDP-28.12
MUST
	RFC 3036, s3.5.10 p74 Label Withdraw Message
	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`Validate the Label Withdraw Message encoding from DUT`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-28.15
MUST
	RFC 3036, s3.5.10.1 p75 Label Withdraw Message Procedures
	RFC 3036, Appendix A.1.14 p120 LSR decides to no longer label switch a FEC
	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`An LSR transmits a Label Withdraw message under the following`
	`conditions: (1) The LSR no longer recognizes a previously known FEC`
	`for which it has advertised a label; (2) The LSR has decided`
	`unilaterally (e.g., via configuration) to no longer label switch a FEC`
	`(or FECs) with the label mapping being withdrawn.`
	
	`When LSR unilaterally decides (or is re-configured) to no longer label`
	`switch a particular FEC, Execute procedure Send_Label_Withdraw (Peer,`
	`FEC, PrevAdvLabel)`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

ANVL-LDP-28.19
MUST
	RFC 3036, s3.5.10.1 p76 Label Withdraw Message Procedures
	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`The FEC TLV may contain the Wildcard FEC Element; if so, it may`
	`contain no other FEC Elements.  In this case, if...there is not an`
	`optional Label TLV in the Label Withdraw message, then the sending`
	`LSR is withdrawing all label mappings previously advertised to the`
	`receiving LSR.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-28.21 **MUST**	RFC 3036, s3.5.11 p76 Label Release Message								
	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages` `Validate Label Release Message encoding from DUT`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-28.22 **MUST**	RFC 3036, s3.5.11 p77 Label Release Message								

	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`Validate optional Label TLV encoding from DUT in Label Release`
	`Message`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-28.23 **MUST**	RFC 3036, s3.5.11.1 p77 Label Release Message Procedures								

	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`An LSR must transmit a Label Release message under any of the`
	`following conditions: (3) The LSR receives a Label Withdraw`
	`message.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-28.26 **MUST**	RFC 3036, s3.5.11.1 p77 Label Release Message Procedures								

	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`Note that if an LSR is configured for "liberal mode", a Release`
	`message will never be transmitted in the case of condition (1)`
	`as specified above.  In this case [LSR which sent the label`
	`mapping is no longer the next hop for the mapped FEC], the upstream`
	`LSR keeps each unused label, so that it can immediately be used later`
	`if the downstream peer becomes the next hop for the FEC.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-28.27 **MUST**	RFC 3036, s3.5.11.1 p77 Label Release Message Procedures								

	`Label Abort Request Messages, Label Withdraw Messages, Label Release Messages`
	`Note that if an LSR is configured for "liberal mode", a Release`
	`message will never be transmitted in the case of condition (2)`
	`as specified above.  In this case [LSR receives a label mapping`
	`from an LSR which is not the next hop for the FEC], the upstream LSR`
	`keeps each unused label, so that it can immediately be used later if`
	`the downstream peer becomes the next hop for the FEC.`								

	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-31.1 **MUST**	NEGATIVE RFC 3036, s3.10.1 p83 Well-known Numbers/UDP and TCP Ports								
	`Well-known Numbers, Name Spaces` `The UDP port for LDP Hello messages is 646`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-31.2 **MUST**	RFC 3036, s3.10.1 p83 Well-known Numbers/UDP and TCP Ports								
	`Well-known Numbers, Name Spaces` `The TCP port for establishing LDP session connections is 646`								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-32.1 **MUST**	NEGATIVE RFC 3036, s5.1 p86 Spoofing RFC 3036, s5.3 p87 Denial of Service								
	Security Considerations An LSR can reduce the threat of spoofed Basic Hellos by accepting Basic Hellos only on interfaces to which LSRs that can be trusted are directly connected. LDP provides two potential targets for denial of service (DoS) attacks: (1) Well known UDP Port for LDP Discovery. An LSR administrator can address the threat of DoS attacks via Basic Hellos by ensuring that the LSR is directly connected only to peers which can be trusted to not initiate such an attack.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-32.4 **MUST**	NEGATIVE RFC 3036, s5.1 p86 Spoofing								
	Security Considerations An LSR can reduce the threat of spoofed Extended Hellos by filtering them and accepting only those originating at sources permitted by an access list. (DUT is passive for session establishment)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-32.5 **MUST**	RFC 3036, s5.1 p86 Spoofing								
	Security Considerations An LSR can reduce the threat of spoofed Extended Hellos by filtering them and accepting only those originating at sources permitted by an access list. (DUT is active for session establishment)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-32.6 **MUST**	RFC 3036, s5.1 p86 Spoofing								
	Security Considerations An LSR can reduce the threat of spoofed Extended Hellos by filtering them and accepting only those originating at sources permitted by an access list.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
ANVL-LDP-32.7 **MUST**	NEGATIVE RFC 3036, s5.1 p86 Spoofing								
	Security Considerations An LSR can reduce the threat of spoofed Extended Hellos by filtering them and accepting only those originating at sources permitted by an access list.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-32.10 **MUST**	NEGATIVE RFC 3036, s5.1 p86 Spoofing								
	Security Considerations An LSR can reduce the threat of spoofed Basic Hellos by ignoring Basic Hellos not addressed to the All Routers on this Subnet multicast group.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-33.4 **MUST**	RFC 3036, Appendix A.1.1 p97 Receive Label Request								
	Receive Label Request If there is no Next Hop, Execute procedure Send_Notification (MsgSource, No Route)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-34.2 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR does not have a previously received label mapping for FEC from MsgSource for the LSP in question, and the MsgSource is not the Next Hop for the FEC, and LSR is using liberal label retention, record label mapping for FEC with label and received attributes from MsgSource. (LMp.1->3->9->11->12->13->33)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-34.3 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR does not have a previously received label mapping for FEC from MsgSource for the LSP in question, and the MsgSource is the Next Hop for the FEC, and LSR is not ingress for FEC, and for each peer that LSR has previously sent a label mapping for FEC for the LSP in question, and for each peer that received attributes in the received label mapping are not consistent with those previously sent, and for each peer that LSR does not have any pending label requests for FEC, record label mapping for FEC with label and received attributes from MsgSource, and send a label mapping to peer and update record of label mapping for FEC previously sent to peer to include the new attributes sent, and perform LSR Label Use procedure. (LMp.1->3->9->11->12->14->16->17->18->22->23->24->25->26->27->28->30->31->33)								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-34.5 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR does not have a previously received label mapping for FEC from MsgSource for the LSP in question, and the MsgSource is the Next Hop for the FEC, and LSR is not ingress for FEC, and for each peer that LSR has not previously sent a label mapping for FEC for the LSP in question, and if DU ordered control is not in use by LSR, and LSR has no label requests for FEC from peer marked as pending, record label mapping for FEC with label and received attributes from MsgSource, and perform LSR Label Use procedure. (LMp.1->3->9->11->12->14->16->17->18->19->28->30->31->33)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-34.11 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR has a previously received label mapping for FEC from MsgSource for the LSP in question, and the label previously received from MsgSource does not match label received in message, execute procedure Send_Message(MsgSource, Label Release, FEC, Label). (LMp.1->3->9->10->32->33)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-34.13 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR does have a previously received label mapping for FEC from MsgSource for the LSP in question, and the label previously received from MsgSource matches label received in the message, and the MsgSource is not the Next Hop for the FEC, and LSR is using liberal label retention, record label mapping for FEC with label and received attributes from MsgSource. (LMp.1->3->9->10->11->12->13->33)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-34.14 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR has a previously received label mapping for FEC from MsgSource for the LSP in question, and the label previously received from MsgSource matches label received in the message, and the MsgSource is the Next Hop for the FEC, and LSR is not ingress for FEC, and for each peer that LSR has previously sent a label mapping for FEC for the LSP in question, and for each peer that received attributes in the received label mapping are not consistent with those previously sent, and for each peer that LSR does not have any pending label requests for FEC, record label mapping for FEC with label and received attributes from MsgSource, and send a label mapping to peer and update record of label mapping for FEC previously sent to peer to include the new attributes sent, and perform LSR Label Use procedure. (LMp.1->3->9->10->11->12->14->16->17->18->22->23->24->25->26->27->28->30->31->33)								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
ANVL-LDP-34.16 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping does not match an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR has a previously received label mapping for FEC from MsgSource for the LSP in question, and the label previously received from MsgSource matches label received in the message, and the MsgSource is the Next Hop for the FEC, and LSR is not ingress for FEC, and for each peer that LSR has not previously sent a label mapping for FEC for the LSP in question, and if DU ordered control is not in use by LSR, and LSR has no label requests for FEC from peer marked as pending, record label mapping for FEC with label and received attributes from MsgSource, and perform LSR Label Use procedure. (LMp.1->3->9->10->11->12->14->16->17->18->19->28->30->31->33)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-34.23 **MUST**	RFC 3036, Appendix A.1.2 p99 Receive Label Mapping								
	Receive Label Mapping Part One If the received label mapping matches an outstanding label request for FEC previously sent to MsgSource, and no loop detected, and LSR does not have a previously received label mapping for FEC from MsgSource for the LSP in question, and the MsgSource is the Next Hop for the FEC, and LSR is not ingress for FEC, and for each peer that LSR has previously sent a label mapping for FEC for the LSP in question, and for each peer that received attributes in the received label mapping are not consistent with those previously sent, and for each peer that LSR does not have any pending label requests for FEC, delete record of outstanding FEC label request, record label mapping for FEC with label and received attributes from MsgSource, and send a label mapping to peer and update record of label mapping for FEC previously sent to peer to include the new attributes sent, and perform LSR Label Use procedure. (LMp.1->2->3->9->11->12->14->16->17->18->22->23->24->25->26->27->28->30->31->33)								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL
ANVL-LDP-35.18 **MUST**	NEGATIVE RFC 3036 Appendix A - A.1.2 p104 Receive Label Mapping								
	Receive Label Mapping Part Two Note 4: An unsolicited mapping with a different label from the same peer would be an attempt to establish multipath label switching, which is not supported in this version of LDP.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-37.4 **MUST**	RFC 3036, Appendix A.1.4 p107 Receive Label Release								
	Receive Label Release, Receive Label Withdraw If LSR receives a Label Release (that does not match any outstanding Label Withdraws) and LSR is the egress and is not merging, then Remove Label from forwarding/switching use for traffic from MsgSource and if any peers do not still hold the label, free the label. LRl.1->2->4->6->10->11->12->13								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-37.6 **MUST**	RFC 3036, Appendix A.1.4 p107 Receive Label Release								
	Receive Label Release, Receive Label Withdraw If LSR receives a Label Release (that does not match any outstanding Label Withdraws) and LSR is not the egress and is not merging, and the LSR is not configured to propagate releases, then Remove Label from forwarding/switching use for traffic from MsgSource and if any peers do not still hold the label, free the label. LRl.1->2->4->6->7->8->10->11->12->13								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-37.10 **MUST**	RFC 3036 Appendix A - A.1.4 p108 Receive Label Release								
	Receive Label Release, Receive Label Withdraw Note 1: If LSR is using Downstream Unsolicited label distribution, it should not re-advertise a label mapping for FEC to MsgSource until MsgSource requests it.								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-37.13 **MUST**	RFC 3036, s3.5.10.1 p76 Label Withdraw Message Procedures RFC 3036, Appendix A.1.5 p110 Receive Label Withdraw								
	Receive Label Release, Receive Label Withdraw An LSR that receives a Label Withdraw message must respond with a Label Release message. When receiving a Label Withdraw, remove Label from forwarding/ switching use and Execute procedure Send_Message (MsgSource, Label Release, FEC, Label)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-38.2 **MUST**	RFC 3036, Appendix A.1.6 p111 Recognize New FEC								
	Recognize New FEC When learning a new FEC while configured for Downstream Unsolicited Independent Control, if LSR does not have previously retained label mapping from the Next Hop for FEC, and Next Hop is not a peer, repeat LSR Label Distribution procedure (FEC.1) for each Peer. (FEC.1->2->3->6)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-38.3 **MUST**	RFC 3036, Appendix A.1.6 p111 Recognize New FEC RFC 3036, Appendix A.1.6 p113 Recognize New FEC								
	Recognize New FEC When learning a new FEC while configured for Downstream Unsolicited Independent Control, if LSR has previously retained label mapping from the Next Hop for FEC, repeat LSR Label Distribution procedure (FEC.1) for each Peer and generate Received Label Mapping Event. (FEC.1->2->5->6) Note 3: If the LSR has a label for the FEC from the Next Hop, it should behave as if it had just received the label from the Next Hop. This occurs in the case of Liberal label retention mode.								
	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-42.3 **MUST**	RFC 3036, Appendix A.2.1 p121 Send_Label								
	Send Label, Send Label Request, Check Received Attributes If the LSR has a label to allocate, allocate label and bind it to the FEC, install label for forwarding/switching use, execute procedure Send_Message(Peer, Label Mapping, FEC, Label, Attributes), record label mapping for FEC with label and attributes has been sent to peer, and if LSR does not have a record of a FEC label request from peer marked as pending, return success. (SL.1->2->3->4->5->6->8)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: FAIL	Ubuntu 16.04: pass
ANVL-LDP-42.11 **MUST**	RFC 3036, Appendix A.2.6 p126 Check_Received_Attributes								
	Send Label, Send Label Request, Check Received Attributes If received attributes do not include Hop Count, return No Loop Detected. (CRa.1->5)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass
ANVL-LDP-42.13 **MUST**	RFC 3036, Appendix A.2.6 p126 Check_Received_Attributes								
	Send Label, Send Label Request, Check Received Attributes If received attributes include Hop Count and Hop Count does not exceed Max allowable hop count, and received attributes do not include Path Vector, return No Loop Detected. (CRa.1->2->3->5)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass

	Release 2.0	3.0-dev 2017-04-25	3.0-dev 2017-05-24	3.0-dev 2017-06-30	Release 3.0-rc1	Master 2017-08-16	Master 2017-08-24	Master 2017-09-08	Release 3.0-rc2
ANVL-LDP-42.15	RFC 3036, Appendix A.2.6 p126 Check_Received_Attributes								
MUST	Send Label, Send Label Request, Check Received Attributes If received attributes include Hop Count and Hop Count does not exceed Max allowable hop count, and received attributes include Path Vector, and the Path Vector does not include LSR Id, and length of Path Vector does not exceed Max allowable length, return No Loop Detected. (CRa.1->2->3->4->5)								
	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass	Ubuntu 16.04: pass