

Network Working Group  
Request for Comments: 3215  
Category: Informational

C. Boscher  
P. Cheval  
Alcatel  
L. Wu  
Cisco  
E. Gray  
Sandburst  
January 2002

## LDP State Machine

### Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

### Abstract

This document provides state machine tables for ATM (Asynchronous Transfer Mode) switch LSRs. In the current LDP specification, there is no state machine specified for processing LDP messages. We think that defining a common state machine is very important for interoperability between different LDP and CR-LDP implementations.

We begin in section 1 by defining a list of terminologies. Then in section 2, we propose two sets of state machine tables for ATM switch LSRs that use downstream-on-demand mode, one method can be used for non-vc merge capable ATM LSRs, while the other one can be used for the vc-merge capable ATM LSRs. In section 3, we provides a state machine for downstream unsolicited mode ATM LSRs.

We focus on the LDP state machines and the associated control blocks used for establishing and maintaining LSPs. We do not describe state machines for the "LDP controller" that is in charge of LDP session initialization, address mapping messages management, routing interface, etc. that is defined in the LDP specification.

Even though the state machines in this document are specific for ATM-LSR, they can be easily adapted for other types of LSRs.

## Table Of Contents

1. Terminology .....	4
2. State Machine for Downstream-on-Demand Mode .....	4
2.1 An LSR's Behavior in the Case of a Next Hop Change .....	4
2.2. ATM Switch LSR with No VC-merge Capability .....	5
2.2.1 LSP Control Block .....	5
2.2.2 States .....	7
2.2.3 Events .....	8
2.2.4 State Transitions .....	9
2.2.5 State Machine .....	10
2.2.5.1 State -- "IDLE" .....	10
2.2.5.2 State -- "RESPONSE_AWAITED" .....	13
2.2.5.3 State -- "ESTABLISHED" .....	16
2.2.5.4 State -- "RELEASE_AWAITED" .....	21
2.2.6 Handling the Next Hop Change .....	23
2.2.6.1 Next Hop Trigger Control Block .....	24
2.2.6.2 States .....	24
2.2.6.3 Events .....	24
2.2.6.4 State Transition for next hop change .....	25
2.2.6.5 State Machine .....	25
2.2.6.5.1 State -- "IDLE" .....	25
2.2.6.5.2 State -- "NEW_NH_RETRY" .....	27
2.2.6.5.3 State -- "NEW_NH_RESPONSE_AWAITED" .....	28
2.2.7 LDP Related Message Handling .....	29
2.3. ATM Switch LSR with VC-merge .....	31
2.3.1 VC-merge .....	31
2.3.2 Control Block .....	31
2.3.3 State Machines for Downstream-on-demand Mode .....	34
2.3.3.1 State of the Upstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	34
2.3.3.2 Events of the Upstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	35
2.3.3.3 State Transitions of the Upstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	36
2.3.3.4 Upstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	37
2.3.3.4.1 State -- "IDLE" .....	37
2.3.3.4.2 State -- "RESPONSE_AWAITED" .....	39
2.3.3.4.3 State -- "ESTABLISHED" .....	42
2.3.3.4.4 State -- "RELEASE_AWAITED" .....	45
2.3.3.5 State of the Downstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	47
2.3.3.6 Events of the Downstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	47
2.3.3.7 State Transitions of the Downstream LSP Control Block's State Machine for Downstream-on-demand mode .....	48

2.3.3.8 Downstream LSP Control Block's State Machine for Downstream-on-demand Mode .....	49
2.3.3.8.1 State -- "IDLE" .....	48
2.3.3.8.2 State -- "RESPONSE_AWAITED" .....	50
2.3.3.8.3 State -- "ESTABLISHED" .....	52
2.3.3.9 State of the Next_Hop_Trigger_Control_Block's State Machine for Downstream-on-demand Mode .....	53
2.3.3.10 Events of the Next_Hop_Trigger_Control_Block's State Machine for Downstream-on-demand Mode .....	53
2.3.3.11 State Transitions of the Next_Hop_Trigger_Control_Block's State Machine for Downstream-on-demand Mode .....	55
2.3.3.12 State Machine .....	55
2.3.3.12.1 State -- "IDLE" .....	55
2.3.3.12.2 State -- "NEW_NH_RETRY" .....	57
2.3.3.12.3 State -- "NEW_NH_RESPONSE_AWAITED" .....	58
2.3.4 LDP Related Message Processing .....	59
3. State Machine for Downstream Unsolicited .....	61
3.1 Control Block .....	61
3.2 States of the Upstream LSP Control Block's State Machine for Downstream Mode .....	62
3.3 Events of the Upstream LSP Control Block's State Machine for Downstream Mode .....	62
3.4 State Transitions of Upstream LSP Control Block's State Machine for Downstream Mode .....	64
3.5 Upstream LSP Control Block's State Machine for Downstream Mode .....	65
3.5.1 : State -- "IDLE" .....	65
3.5.2 : State -- "ESTABLISHED" .....	66
3.5.3 : State -- "RELEASE_AWAITED" .....	67
3.5.4 : State -- "RESOURCE_AWAITED" .....	69
3.6 State of the Downstream LSP Control Block's State Machine for Downstream Mode .....	70
3.7 Events of the Downstream LSP Control Block's State Machine for Downstream Mode .....	70
3.8 State Transitions of Downstream LSP Control Block's State Machine for Downstream Mode .....	71
3.9 Downstream LSP Control Block's State Machine for Downstream Mode .....	71
3.9.1 : State -- "IDLE" .....	71
3.9.2 : State -- "ESTABLISHED" .....	73
3.10 LDP Related Message Processing for downstream mode .....	74
4. Security Considerations .....	75
5. Acknowledgements .....	76
6. References .....	76
7. Authors' Address .....	77
8. Full Copyright Statement .....	78

## 1. Terminology

- LDP-REQUEST: LDP Label Request message
- LDP-MAPPING: LDP Label Mapping message
- LDP-WITHDRAW: LDP Label Withdraw message
- LDP-RELEASE: LDP Label Release message
- LDP-ABORT: LDP Abort message used to abort a LSP setup.
- LDP-NAK: LDP Notification message used to reject an LDP message.

## 2. State Machine for Downstream-on-Demand Mode

In this document, we provide two sets of state machines: one for the ATM LSR that does not have VC-merge capability, and the other for the ATM LSR that does have VC-merge capability.

State machine descriptions are given in terms of control blocks, states, events, response actions and state transitions. Control blocks contain the information that is required to support handling of events. A control block may also contain any additional information that is required either of any specific implementation or in support of other required functions. In every case, additional information required to support the procedures defined in the LDP specification [4] or management objects defined in the LDP MIB [3] would be stored in a specific LDP implementation - either as part of the control block structure or in some other way.

The state machines cover both independent LSP control and ordered LSP control.

Loop detection and loop prevention messages will be processed as specified in [4]. The impact of loop detection and loop prevention messages on state transitions is left for further study.

### 2.1 An LSR's Behavior in the Case of a Next Hop Change

When there is a topology change and an LSR detects a new better next hop for an LSP, it may behave in 2 different ways:

- 1) It tries to do a "local repair". This means that it extends the LSP through the new next hop, releases the old path from this LSR forward and then splices into this newly extended LSP.

- 2) If the LSP is created with the "pinned" option (CR-LDP[5]), the LSR ignores the new next hop change, and the LSP stays unchanged. The LSR may decide to send an LDP-MAPPING containing attributes for this New Next Hop (NH) that have changed.

## 2.2. ATM Switch LSR with No VC-merge Capability

In an MPLS domain where some ATM LSRs do not have VC-merge capability, downstream-on-demand mode is required for these ATM LSRs [1]. Also, "conservative label retention mode" is required in this case [1].

For each LSP, there are 2 kinds of state machines involved:

- 1) the LSP Control Block and its state machine that can be used to handle normal LSP setup. It is created when the LSR receives a new LDP Request and it is deleted when the LSP of this request is torn down.
- 2) the Next Hop Trigger Control Block and its state machine that is used to handle switching over to a better LSP through a different next hop. It is created when the LSR decides to switch over to a better next hop and it is deleted when the LSR finishes switching over to the better next hop. This state machine uses a timer (and has corresponding states) to ensure that switch over occurs in a timely fashion after a routing transient has had time to stabilize.

### 2.2.1 LSP Control Block

For each LSP, an LSP Control Block is defined that may contain the following information:

- Upstream Label Request ID (assigned by the upstream LSR), that is the 'Message Id' in the Label Request Message received from the upstream LSR.
- Downstream Label Request ID (assigned by this LSR itself), that is 'Message Id' in the Label Request Message sent to the downstream LSR.
- Upstream LDP Identifier
- Downstream LDP Identifier
- State
- FEC

- Upstream Label (assigned by this LSR)
- Downstream Label (assigned by the downstream LSR)
- Trigger Control Block Pointer, (Only used at the ingress LSR of a LSP) that points to the control block that triggers setting up this LSP or tearing down this LSP.
- Next Hop Control Block Pointer, that points to the control block that is used for switching over to a better LSP.

The following index combinations can be used to locate a unique LSP Control Block:

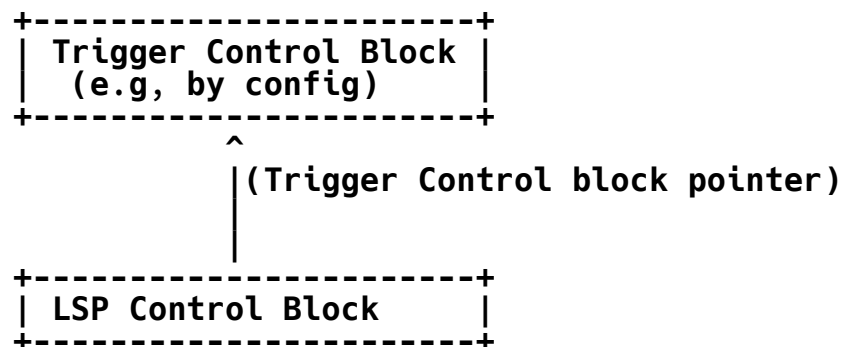
- Downstream Label and Downstream LDP Identifier, or
- Upstream Label and Upstream LDP Identifier, or
- Downstream Label Request ID and Downstream LDP Identifier
- Upstream Label Request ID and Upstream LDP Identifier

Here is the relationship between different control blocks, the detail definition of Next Hop Trigger Control Block is described in section 2.2.6.

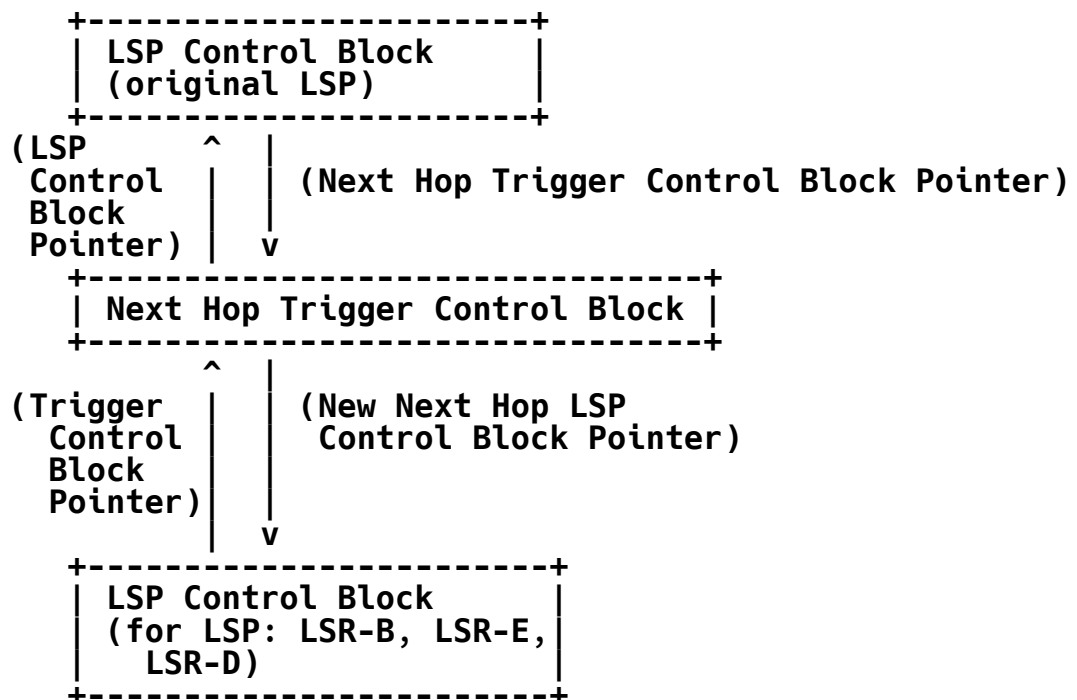
For example, an LSP that transits through (LSR-A, LSR-B, LSR-C, LSR-D):

LSR-A ----> LSR-B ---> LSR-C ---> LSR-D

The control blocks in LSR-A are:



When LSR-B detects a better next hop to LSR-D through LSR-E, and it decides to switch over to it, so control blocks in LSR-B are:



### 2.2.2 States

This section describes the various states that are used in the state machine for the ATM non VC-merge LSR.

#### -- IDLE

This is the initial LSP state, when the LSP Control Block is created.

#### -- RESPONSE\_AWAITED

This state means that the LSR has received and processed an LDP-REQUEST from an upstream LSR, or it has received an internal set up request. It has sent a new LDP-REQUEST towards a downstream LSR. The LSR is waiting for the LDP-MAPPING from the downstream LSR.

#### -- ESTABLISHED

This state means that the LSR has received the LDP-MAPPING from the downstream LSR and the LSP is up and operational.

#### -- RELEASE\_AWAITED

This state means that the LSR has sent a LDP-WITHDRAW upstream and is waiting for the LDP-RELEASE before freeing up the label resource.

### 2.2.3 Events

#### -- LDP Request

The LSR receives an LDP-REQUEST from an upstream LSR.

#### -- LDP Mapping

The LSR receives an LDP-MAPPING from a downstream LSR.

#### -- LDP Release

The LSR receives an LDP-RELEASE from an upstream LSR.

#### -- LDP Withdraw

The LSR receives an LDP-WITHDRAW from a downstream LSR.

#### -- LDP Upstream Abort

The LSR receives an LDP-ABORT from an upstream LSR.

-- LDP Downstream NAK The LSR receives an LDP-NAK (notification) from an downstream LSR.

#### -- Upstream Lost

The LSR loses its LDP session with an upstream LDP peer.

#### -- Downstream Lost

The LSR loses its LDP session with a downstream LDP peer.

#### -- Internal SetUp

For some reason, e.g. a configuration request of a traffic engineering tunnel, or recognizing a new FEC could trigger an Internal SetUp event to set up a new LSP from this node.

#### -- Internal Destroy

The LSR send an Internal Destroy event to tear down an LSP.

#### -- Internal Cross-Connect

The LSR send an Internal Cross-Connect to splice two LSPs into one LSP. This happens when a LSR switches over to a better next hop.

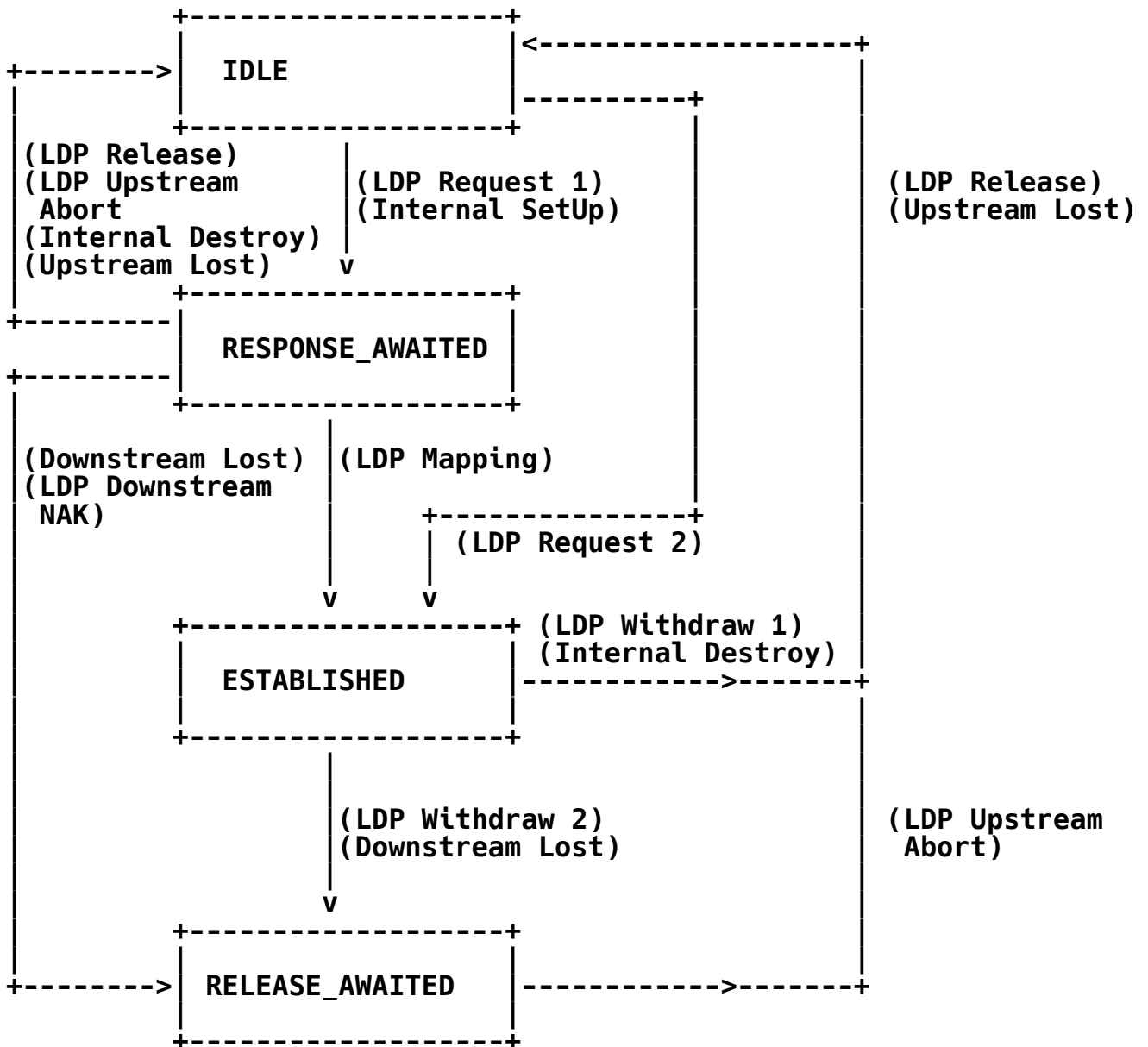


## -- Internal New NH

The LSR decides to switch over the better next hop.

### 2.2.4 State Transitions

The following diagram describes briefly the state transitions.



## 2.2.5 State Machine

### 2.2.5.1 State -- "IDLE"

State: IDLE

Event: LDP Request

New State: Depends upon the action routine.

Actions:

If this LSR is the LSP Egress or Proxy Egress [2]

Then:

Choose an upstream label, connect this upstream label to the local IP forwarding module, allocate the resources, send the LDP-MAPPING upstream with the upstream label, and go to the new state 'ESTABLISHED'.

else

Obtain a next hop (or interface) with the FEC specified in the LDP-REQUEST, propagate the LDP-REQUEST, with newly assigned Message ID by this LSR, towards the obtained next hop, and go to the new state 'RESPONSE\_AWAITED'.

If the LSR uses the independent control mode [2], choose an upstream label, connect this upstream label to the local IP forwarding module, go to the ESTABLISHED state and send an LDP-MAPPING upstream with the upstream label.

If unable to process the request for any reason, issue an LDP-NAK to the sender with the appropriate error code, go to IDLE and delete the LSP Control Block.

State: IDLE

Event: LDP Mapping

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: LDP Release

**New State: IDLE**

**Actions:**

Ignore the event. It is an internal implementation error.

**State: IDLE**

**Event: LDP Withdraw**

**New State: IDLE**

**Actions:**

Ignore the event. It is an internal implementation error.

**State: IDLE**

**Event: LDP Upstream Abort**

**New State: IDLE**

**Actions:**

Ignore the event. It is an internal implementation error.

**State: IDLE**

**Event: LDP Downstream NAK**

**New State: IDLE**

**Actions:**

Ignore the event. It is an internal implementation error.

**State: IDLE**

**Event: Upstream Lost**

**New State: IDLE**

**Actions:**

Ignore the event. It is an internal implementation error.

**State: IDLE**

Event: Downstream Lost

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Internal SetUp

New State: RESPONSE\_AWAITED

Actions:

Set up the Trigger Control Block pointer,

Obtain a next hop (or interface) with the FEC specified in the Internal SetUp message, send a LDP-REQUEST towards the obtained next hop, and go to the new state 'RESPONSE\_AWAITED'.

State: IDLE

Event: Internal Destroy

New State: IDLE

Actions:

Ignore. It is an internal implementation error.

State: IDLE

Event: Internal Cross-Connect

New State: IDLE

Actions:

Ignore. It is an internal implementation error.

State: IDLE

Event: Internal New NH

New State: IDLE

**Actions:**

Ignore. It is an internal implementation error.

**2.2.5.2 State -- "RESPONSE\_AWAITED"**

**State:** RESPONSE\_AWAITED

**Event:** LDP Request

**New State:** RESPONSE\_AWAITED

**Actions:**

Ignore the event. It is an internal implementation error. A non VC merge ATM LSR must create a new LSP control block for a new LDP request.

**State:** RESPONSE\_AWAITED

**Event:** LDP Mapping

**New State:** ESTABLISHED

**Actions:**

- 1) If the LSP is triggered by the local router (Trigger Control Block Pointer is not zero), send event 'Internal LSP UP' to the Trigger control block.
- 2) Else If the LSR uses the ordered control mode, choose an upstream label.
- 3) Connect the upstream label to the downstream label. Allocate the resources. Propagate the LDP-MAPPING upstream with the upstream label.

If unable to process the message, disconnect the upstream label from the downstream label, free the upstream label, release the resources, send an LDP-RELEASE downstream and an LDP-NAK upstream with status (No Label Resources [4]), go to IDLE and delete the LSP Control Block.

**State:** RESPONSE\_AWAITED

**Event:** LDP Release

**New State:** IDLE

**Actions:**

If the LSR uses the independent control mode, free the upstream label.

Send an LDP-ABORT downstream, go to IDLE and delete the LSP Control Block.

Note: This should only occur if the LSR uses the independent control mode. In the ordered control mode, no upstream label mapping will have been sent corresponding to this LSP while waiting for a label mapping from downstream.

**State:** RESPONSE\_AWAITED

**Event:** LDP Withdraw

**New State:** RESPONSE\_AWAITED

**Actions:**

Ignore the event. It's a protocol error from the downstream LSR.

**State:** RESPONSE\_AWAITED

**Event:** LDP Upstream Abort

**New State:** IDLE

**Actions:**

If the LSR uses the independent control mode, free the upstream label.

Send an LDP-ABORT downstream.

Delete the LSP Control Block.

**State:** RESPONSE\_AWAITED

**Event:** LDP Downstream NAK

**New State:** Depends on the action routine.

**Actions:**

1. If the LSP is triggered by the local router (Trigger Control Block Pointer is not zero), send event 'Internal LSP DOWN' to the Trigger control block, go to IDLE and delete the LSP Control Block.
2. Else If the LSR uses the independent control mode, send an LDP-WITHDRAW upstream and go to state 'RELEASE\_AWAITED'.

If the LSR uses the ordered control mode, send an LDP-NAK upstream, go to IDLE and delete the LSP Control Block.

**State:** RESPONSE\_AWAITED

**Event:** Upstream Lost

**New State:** IDLE

**Actions:**

If the LSR uses the independent control mode, free the upstream label.

Send an LDP-ABORT downstream, go to IDLE and delete the LSP Control Block.

**State:** RESPONSE\_AWAITED

**Event:** Downstream Lost

**New State:** Depends on the action routine.

**Actions:**

1. If the LSP is triggered by the local router (Trigger Control Block Pointer is not zero), send event 'Internal LSP DOWN' to the trigger control block, go to IDLE and delete the LSP Control Block.
2. Else, If the LSR uses the independent control mode, free the upstream label and send an LDP-WITHDRAW upstream and go to state 'RELEASE\_AWAITED'.

If the LSR uses the ordered control mode, send an LDP-NAK upstream (with a status 'No Route' [4]), go to IDLE and delete the LSP Control Block.

State: RESPONSE\_AWAITED

Event: Internal SetUp

New State: RESPONSE\_AWAITED

Actions:

Ignore, it is an internal implementation error.

State: RESPONSE\_AWAITED

Event: Internal Destroy

New State: IDLE

Actions:

Send an LDP-ABORT downstream, go to IDLE and delete the LSP Control Block.

State: RESPONSE\_AWAITED

Event: Internal Cross-Connect

New State: RESPONSE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RESPONSE\_AWAITED

Event: Internal New NH

New State: RESPONSE\_AWAITED

Actions:

Send LDP-ABORT to the old downstream, and send LDP-REQUEST to the new next hop.

#### 2.2.5.3 State -- "ESTABLISHED"

State: ESTABLISHED

Event: LDP Request



New State: ESTABLISHED

Actions:

Ignore the event. It's an internal implementation error. For non VC-merge ATM LSR, a new LSP control block is always created for each LDP request.

State: ESTABLISHED

Event: LDP Mapping

New State: ESTABLISHED

Actions:

Process the LDP-MAPPING, that may contain the new attributes of the label mapping and then propagate the LDP-MAPPING upstream.

State: ESTABLISHED

Event: LDP Release

New State: IDLE

Actions:

Disconnect the upstream label from the downstream label.

Free the upstream label.

Free the resources.

Send event 'Internal Destroy' to the Next Hop Trigger Control Block if it was in the middle of switching over to the better next hop.

Propagate the LDP-RELEASE downstream if the LSR is not the egress for the LSP, go to IDLE and delete the LSP Control Block.

State: ESTABLISHED

Event: LDP Withdraw

New State: Depends on the action routine.

**Actions:**

- 1) Free the resources and send LDP-RELEASE downstream.
- 2) If it is independent control mode, set the state to `IDLE', create a internal LDP Request with the information in the LSP Control Block, and pass event `LDP Request' to its own state machine.
- 3) Else for the ordered control mode
  - 3.1) If the LSP is triggered to be setup by itself (e.g it is the ingress LSR of this LSP), send event `Internal LSP Down' to the trigger control block, go to IDLE and delete the LSP Control Block.
  - 3.2) Else, if it is triggered by the incoming LDP Request, Disconnect the upstream label from the downstream label. Propagate the LDP-WITHDRAW upstream and go to state `RELEASE\_AWAITED'.
  - 3.3) If the LSP is in the middle of switching over to a better LSP, send event `Internal Destroy' to the state machine of its New Next Hop LSP Control Block, go to IDLE and delete the LSP Control Block.

**State:** ESTABLISHED

**Event:** LDP Upstream Abort

**New State:** ESTABLISHED

**Actions:**

Ignore the event.

Note: This scenario can occur if the upstream LSR sends a LDP-ABORT at about the same time as the local LSR sends a LDP-MAPPING. In this situation, it should be up to exactly one of the two LSRs as to whether or not the label that was sent remains valid. The LDP specification [4] procedures leave the choice to the upstream LSR that must send an LDP-RELEASE if it will not use the label provided.

**State:** ESTABLISHED

**Event:** LDP Downstream NAK

**New State:** ESTABLISHED

**Actions:**

Ignore the event. It is a protocol error from the downstream LSR. The downstream LSR should always LSP-WITHDRAW to tear down the LSP when the LSP is established.

**State:** ESTABLISHED

**Event:** Upstream Lost

**New State:** IDLE

**Actions:**

Disconnect the upstream label from the downstream label.

Free the upstream label.

Send event 'Internal Destroy' to the Next Hop Trigger Control Block if it was in the middle of switching over to the better next hop.

Free the resources.

Propagate an LDP-RELEASE downstream, go to IDLE and delete the LSP Control Block.

**State:** ESTABLISHED

**Event:** Downstream Lost

**New State:** Depends on the action routine.

**Actions:**

- 1) If the LSP is triggered by the local router (Trigger Control Block Pointer is not zero), send event 'Internal LSP NAK' to the Trigger control block, go to IDLE and delete the LSP Control Block.
- 2) Else, disconnect the upstream label from the downstream label. Propagate an LDP-WITHDRAW upstream and go to 'RELEASE\_AWAITED' state.

- 3) Send event 'Internal Destroy' to the Next Hop Trigger Control Block if it was in the middle of switching over to the better next hop.

State: ESTABLISHED

Event: Internal Setup

New State: ESTABLISHED

Actions:

Ignore, it is an internal implementation error.

State: ESTABLISHED

Event: Internal Destroy

New State: IDLE

Actions:

Disconnect the upstream label from the downstream label if it is not the ingress of the LSP.

Free the resources.

Send an LDP-RELEASE downstream, go to IDLE and delete the LSP Control Block.

State: ESTABLISHED

Event: Internal Cross-Connect

New State: ESTABLISHED

Actions:

Connect the upstream label to the downstream label

May need to send a new LDP-MAPPING upstream with the attributes from the new next hop.

Reset Trigger Control Block Pointer to zero.

State: ESTABLISHED

Event: Internal New NH

New State: ESTABLISHED

Actions:

- 1) If the LSR was in the middle of switching over to a better next hop (Next Hop Trigger Control Block Pointer is not zero), it send 'Internal New NH' to that control block.
- 2) Else, create a new Next Hop Trigger Control Block, set Next Hop Trigger Control Block pointer to point to this control block, and pass 'Internal New NH' to this control block.

#### 2.2.5.4 State -- "RELEASE\_AWAITED"

State: RELEASE\_AWAITED

Event: LDP Request

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RELEASE\_AWAITED

Event: LDP Mapping

New State: RELEASE\_AWAITED

Actions:

It is a protocol error from the downstream LDP peer, but anyway send a LDP-RELEASE downstream.

State: RELEASE\_AWAITED

Event: LDP Release

New State: IDLE

Actions:

- 1) Free the upstream label
- 2) Delete the control block.

State: RELEASE\_AWAITED

**Event:** LDP Withdraw

**New State:** RELEASE\_AWAITED

**Actions:**

It is a protocol error from the downstream LDP peer, but send a LDP- RELEASE anyway.

**State:** RELEASE\_AWAITED

**Event:** LDP Upstream Abort

**New State:** IDLE

**Actions:**

1) Free the upstream label

2) Delete the control block.

**State:** RELEASE\_AWAITED

**Event:** LDP Downstream NAK

**New State:** RELEASE\_AWAITED

**Actions:**

Ignore the event. Continue waiting for the LDP-RELEASE from upstream.

**State:** RELEASE\_AWAITED

**Event:** Upstream Lost

**New State:** IDLE

**Actions:**

1) Free the upstream label

2) Delete the control block.

**State:** RELEASE\_AWAITED

**Event:** Downstream Lost

New State: RELEASE\_AWAITED

Actions:

Ignore the event. Continue waiting for the LDP-RELEASE from upstream.

State: RELEASE\_AWAITED

Event: Internal SetUp

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RELEASE\_AWAITED

Event: Internal Destroy

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RELEASE\_AWAITED

Event: Internal Cross-Connect

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

## 2.2.6 Handling the Next Hop Change

When an LSR detects a better next hop, it may decide to establish a new LSP through this next hop. For example, an LSR is configured as "local repair", or the LSR is configured as "global repair" and it is the ingress end of a LSP. It can then create a Next Hop Trigger Control Block and use the state machine of Next Hop Trigger Control Block to establish a new LSP through the better next hop.

#### 2.2.6.1 Next Hop Trigger Control Block

-- State

-- LSP Control Block Pointer, that points to the original LSP control block.

-- New Next Hop LSP Control Block Pointer, that points to the LSP Control Block that is setting up an LSP through the new next hop.

#### 2.2.6.2 States

-- IDLE

This is the initial LSP state, when the Trigger\_Control\_Block is created.

-- NEW\_NH\_RETRY

This is the state where an LSR waits for a retry timer to expire and then tries to establish an LSP through a new next hop.

-- NEW\_NH\_RESPONSE\_AWAITED

This is the state where an LSR is in the middle of establishing a new LSP through a new next hop. It has triggered a LSP control block to send an LDP-REQUEST towards the new next hop and is waiting for the LDP-MAPPING.

#### 2.2.6.3 Events

-- Internal New NH

The LSR detects there is a new next hop for a FEC.

-- Internal Retry Timeout

The LSP retry timer expires.

-- Internal LSP UP

The LSP to the new Next Hop is UP

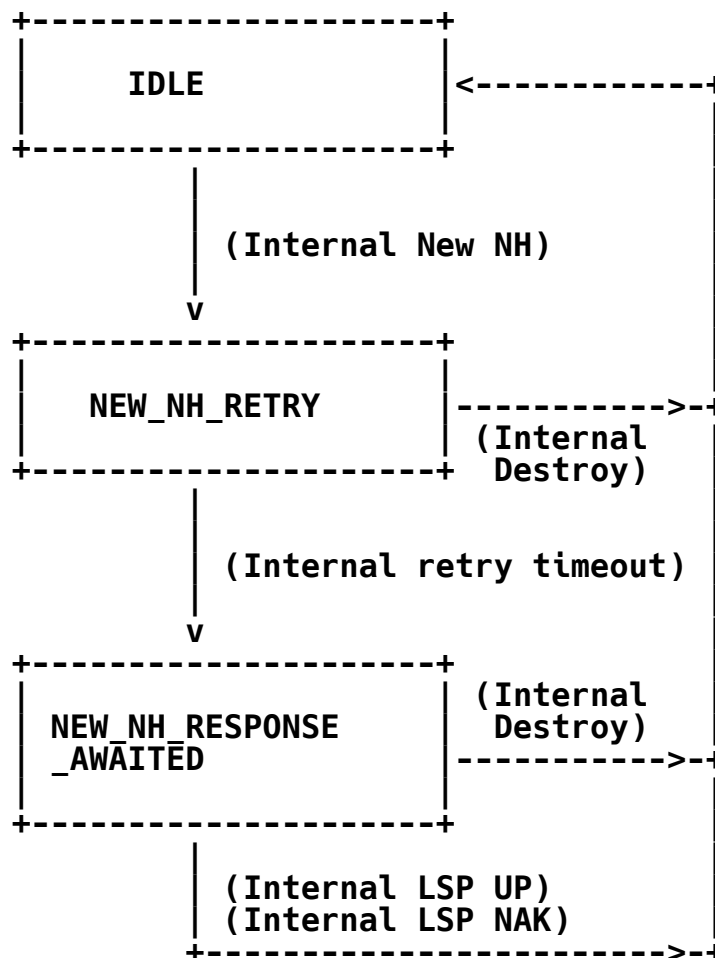
-- Internal LSP NAK

The LSP through the new next hop could not get set up



**-- Internal Destroy**

This event is triggered when the LSR lost the LDP session with its upstream neighbor.

**2.2.6.4 State Transition for next hop change****2.2.6.5 State Machine****2.2.6.5.1 State -- "IDLE"**

State: IDLE  
 Event: Internal New NH  
 New State: NEW\_NH\_RETRY

**Actions:**

Start the LSP retry timer and go to the `NEW\_NH\_RETRY' state.

**State:** IDLE

**Event:** Internal retry timeout

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

**State:** IDLE

**Event:** Internal LSP UP

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

**State:** IDLE

**Event:** Internal LSP NAK

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

**State:** IDLE

**Event:** Internal destroy

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

#### 2.2.6.5.2 State -- "NEW\_NH\_RETRY"

State: NEW\_NH\_RETRY  
Event: Internal New NH  
New State: NEW\_NH\_RETRY  
Actions:

Restart the LSP retry timer.

State: NEW\_NH\_RETRY  
Event: Internal retry timeout  
New State: Depends on action routine.  
Actions:

If the new next hop is the same one as the old next hop, go to IDLE and delete the control block.

Otherwise, create an LSP control block that will try to establish a new LSP through the new next hop, send event 'Internal Setup' to its state machine and go to NEW\_NH\_RESPONSE\_AWAITED.

State: NEW\_NH\_RETRY  
Event: Internal LSP UP  
New State: NEW\_NH\_RETRY  
Actions:

Ignore. It is an internal implementation error.

State: NEW\_NH\_RETRY  
Event: Internal LSP NAK  
New State: NEW\_NH\_RETRY  
Actions:

Ignore. It is an internal implementation error.

State: NEW\_NH\_RETRY

Event: Internal destroy

New State: IDLE

Actions:

Stop the timer, go to IDLE and delete the control block.

#### 2.2.6.5.3 State -- "NEW\_NH\_RESPONSE\_AWAITED"

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal New NH

New State: NEW\_NH\_RETRY

Actions:

Restart the LSP retry timer, send 'Internal destroy' to the control block of the LSP for the new next hop and go to the 'NEW\_NH\_RETRY' state.

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal retry timeout

New State: NEW\_NH\_RESPONSE\_AWAITED

Actions:

Ignore. It is an internal implementation error.

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal LSP UP

New State: IDLE

Actions:

Send event 'Internal cross-connect' event to the LSP control block of the new next hop.

Send event 'Internal destroy' event to the original LSP control block.

Then go to IDLE and delete the control block.

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal LSP NAK

New State: IDLE

Actions:

Delete the control block.

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal destroy

New State: IDLE

Actions:

Send event 'Internal destroy' the control block for the new LSP through the new next hop.

#### 2.2.7 LDP Related Message Handling

- If an LSR receives an LDP-REQUEST from an upstream LSR:
  - a) If this is a duplicate request, discard the message. A duplicate request means that there is a LSP Control Block that has the same FEC, Upstream Label Request ID and Upstream Label Request ID and same Upstream LDP Session Identifier.
  - b) Otherwise, create a new LSP Control Block, store the relevant information from the message into the control block, then pass the event 'LDP Request' to its state machine.
- If an LSR receives an LDP-MAPPING from a downstream LSR:
  - a) Extract the 'Label Request Message ID' field and from the LDP-MAPPING.
  - b) Find an LSP Control Block that has the same Downstream Label Request ID and the same Downstream LDP Session Identifier.
  - c) If an LSP Control Block is found, pass the event 'LDP Mapping' to its state machine.
  - d) If there is no matching LSP Control Block found, then try to find an LSP Control Block that has the same Downstream Label and the same Downstream LDP Session Identifier.

- e) If an LSP Control Block is found, pass the event `LDP Mapping' to its state machine.
- f) Otherwise, ignore the LDP-MAPPING and send a LDP-RELEASE downstream.
- If an LSR receives an LDP-RELEASE from an upstream LSR:
  - a) Find an LSP Control Block that has the same Upstream Label and the same Upstream LDP Session Identifier.
  - b) If an LSP Control Block is found, pass the event `LDP Release' to its state machine.
  - c) Otherwise, ignore the message.
- If an LSR receives an LDP-WITHDRAW from a downstream LSR:
  - a) Find an LSP Control Block that has the same Downstream Label and the same Downstream LDP Session Identifier.
  - b) If an LSP Control Block is found, pass the event `LDP Withdraw' to its state machine.
  - c) Otherwise, ignore the LDP-WITHDRAW and send a LDP-RELEASE downstream.
- If an upstream LDP peer is lost:
  - a) Find all the LSP Control Blocks whose upstream LDP peer is that LSR.
  - b) Then pass the event `Upstream Lost' to their state machines.
- If a downstream LDP peer is lost:
  - a) Find all the LSP Control Blocks whose downstream LDP peer is that LSR.
  - b) Then pass the event `Downstream Lost' to their state machines.
- If the LSR detects a new next hop for an FEC:

For each LSP that needs "local repair", or it needs "global repair" and it is the ingress of the LSP, pass event "Internal New NH" to its state machine.

- If an LSR receives an LDP-Abort from an upstream LSR:
  - a) Extract the LDP Request ID value from the LDP-Abort message.
  - b) Find an LSP Control Block that has the same Upstream Label Request ID and the same Upstream LDP Session Identifier.
  - c) If an LSP Control Block is found, pass the event 'LDP Upstream Abort' to its state machine.
  - d) Otherwise, ignore the message.
- If the LSR receives an LDP-NAK from a downstream LSR:
  - a) Extract the LDP Request ID value from the LDP-NAK.
  - b) Find an LSP Control Block that has the same Downstream Label Request ID and the same Downstream LDP Session Identifier.
  - c) If an LSP Control Block is found, pass the event 'LDP Downstream NAK' to its state machine.
  - d) Otherwise, ignore the message.

### 2.3. ATM Switch LSR with VC-merge

#### 2.3.1 VC-merge

A VC-merge capable LSR can map multiple incoming labels (VPI/VCI) into one outgoing label. It is possible that this LSR can only merge a limited number of incoming labels into a single outgoing label. As described in [2], suppose, for example, that due to some hardware limitation a node is capable of merging four incoming labels into a single outgoing label. Suppose however, that this particular node has six incoming labels arriving at it for a particular FEC. In this case, this node may merge these into two outgoing labels.

When an upstream LSR has a limited merging capability, it is difficult for a downstream LSR to know how many labels should be assigned to each FEC. In this case, downstream-on-demand is recommended.

#### 2.3.2 Control Block

There are 3 types of control blocks involved: Upstream LSP Control Block, Downstream LSP Control Block, and Next Hop Trigger Control Block.

There is one Upstream LSP Control Block for each LDP-REQUEST received.

There is one Downstream LSP Control Block for each unique LDP-REQUEST sent to a downstream LSR. There can be multiple Downstream LSP Control Blocks per FEC in an LSR. This can be the result of an upstream LSR asking for a label for an FEC. This LSR must assign a unique upstream label and it can not merge this upstream label into an existing downstream label for this FEC.

There is one Next Hop Trigger Control Block for each FEC for which a better next hop has been detected and the LSR has decided to switch to this better next hop. It could be the result of "local repair" or "global repair" if the LSR is the ingress LSR of the LSP.

A Downstream LSP Control Block contains a list of pointers to Upstream LSP Control Blocks or the Next Hop Trigger Control Block. This means that this LSR has decided to map the multiple labels listed in the Upstream LSP Control Blocks and the Next Hop Trigger Control Block into a single label listed in the Downstream LSP Control Block.

An Upstream LSP Control Block may contain the following information:

- Upstream LDP Session Identifier
- State
- Upstream Label (assigned by this LSR)
- Downstream LSP Control Block pointer
- Upstream LDP Request ID (assigned by the upstream LSR in downstream-on-demand mode)
- Next\_Hop\_Trigger\_Block pointer

Upstream Label and Upstream LDP Session Identifier can be used to locate a unique Upstream LSP Control Block.

If an LSR is using downstream-on-demand mode, it can use the Upstream LDP Request ID and the Upstream LDP Session Identifier to locate a unique Upstream LSP Control Block.



An `Next_Hop_Trigger` LSP Control Block may contain the following information:

- Upstream LSP Control Block pointer, that points to the one that is needed to switch over to the better next hop
- State
- Downstream LSP Control Block pointer

A Downstream LSP Control Block may contain the following information:

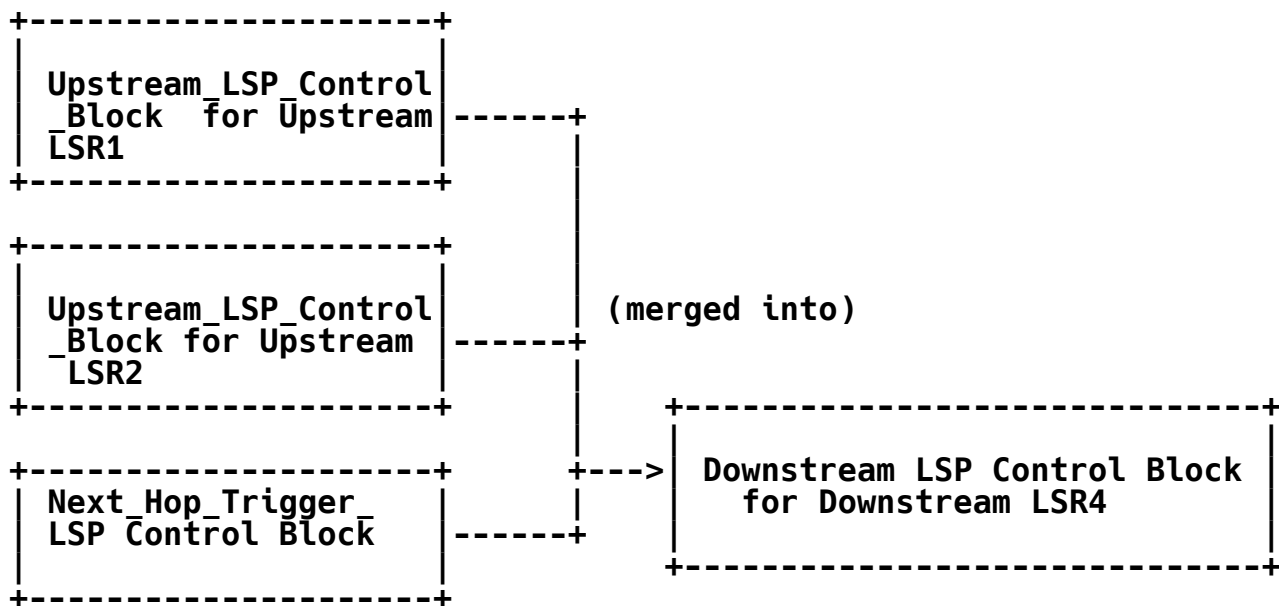
- FEC
- State
- Downstream LDP Session Identifier
- list of pointers to the Upstream LSP Control Blocks or the `Trigger_Control_Blocks` that are merged at this LSR for this FEC
- Downstream Label (assigned by the downstream LSR)
- Downstream Label Request ID (assigned by the LSR itself if it is using downstream-on-demand mode)

Downstream Label, Downstream LDP Session Identifier can be used to locate a unique Downstream LSP Control Block.

If an LSR is using downstream-on-demand mode, it can also use the Downstream Label Request ID and the Downstream LDP Session Identifier to locate a unique Downstream LSP Control Block.

The following diagram details the relationship between these 2 types of control blocks:

For example, the LSR has decided to merge 3 LDP-REQUESTs of a FEC from upstream LSR1, LSR2, LSR3 into one LDP-REQUEST and sent it to a downstream LSR4.



### 2.3.3 State Machines for Downstream-on-demand Mode

The following sections describe the state machines used in downstream-on-demand mode.

#### 2.3.3.1 State of the Upstream LSP Control Block's State Machine for Downstream-on-demand Mode

##### -- IDLE

This is the initial LSP state.

##### -- RESPONSE\_AWAITED

This state means that the LSR has received and processed an LDP-REQUEST from an upstream LSR, and has sent a new LDP-REQUEST towards a downstream LSR. The LSR is waiting for the LDP-MAPPING from the downstream LSR.

##### -- ESTABLISHED

This state means that the LSR has received the LDP-MAPPING from the downstream LSR and the LSP is up and operational.

##### -- RELEASE\_AWAITED

This state means that the LSR has sent a LDP-WITHDRAW upstream and is waiting for the LDP-RELEASE before freeing up the label resource.

### 2.3.3.2 Events of the Upstream LSP Control Block's State Machine for Downstream-on-demand Mode

#### -- LDP Request

The LSR receives an LDP-REQUEST from an upstream LSR.

#### -- Internal Downstream Mapping

This event is sent by one Downstream LSP Control Block's state machine. This Downstream LSP Control Block is the merged Downstream LSP Control Block of this Upstream LSP Control Block. The event is the result of receiving an LDP-MAPPING by the Downstream LSP Control Block's state machine.

#### -- LDP Release

The LSR receives an LDP-RELEASE from an upstream LSR.

#### -- Internal Downstream Withdraw

This event is sent by one Downstream LSP Control Block's state machine. This Downstream LSP Control Block is the merged Downstream LSP Control Block of this Upstream LSP Control Block. The event is the result of receiving an LDP-WITHDRAW by the Downstream LSP Control Block's state machine.

#### -- LDP Upstream Abort

The LSR receives an LDP-ABORT from an upstream LSR.

#### -- Internal Downstream NAK

This event is sent by one Downstream LSP Control Block's state machine. This Downstream LSP Control Block is the merged Downstream LSP Control Block of this Upstream LSP Control Block. The event is the result of receiving an LDP-NAK by the Downstream LSP Control Block's state machine, or it detects an error.

#### -- Upstream Lost

The LSR loses the LDP session with its upstream LDP peer.

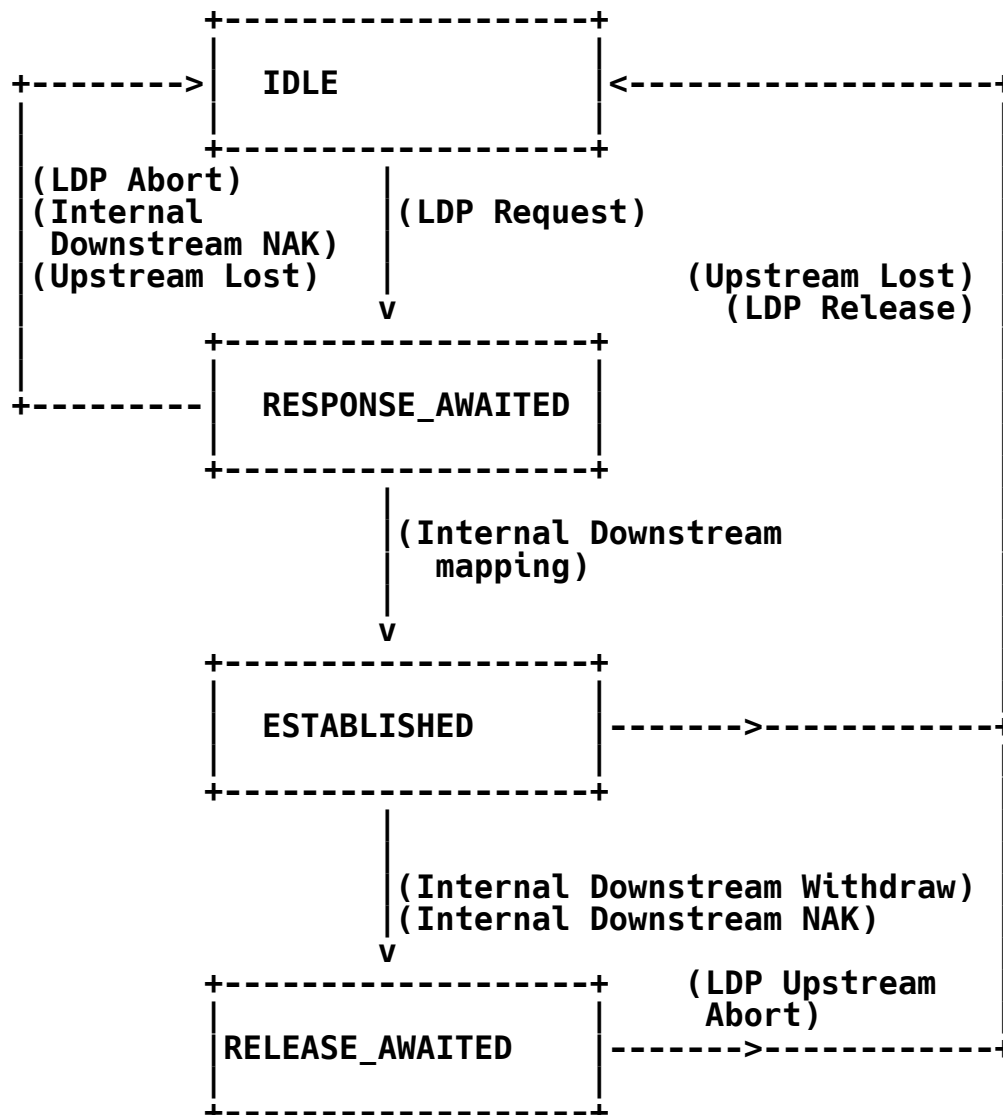
#### -- Internal New NH

The LSR detects there is better next hop and decides to establish the lsp through this better next hop.

### -- Internal Re-Cross-Connect

This event is used to trigger splicing into a different downstream LSP. This can happen when it is switched over to a better LSP through the new next hop.

#### 2.3.3.3 State Transitions of the Upstream LSP Control Block's State Machine for Downstream-on-demand Mode



#### 2.3.3.4 Upstream LSP Control Block's State Machine for Downstream-on-demand Mode

##### 2.3.3.4.1 State -- "IDLE"

State: IDLE

Event: LDP Request

New State: Depends upon the action routine.

Actions:

If this LSR is the LSP Egress or Proxy Egress [2],

Then:

choose an upstream label, allocate the resources, connect this upstream label to the local IP forwarding module, send an LDP-MAPPING upstream with the upstream label and go to the state `ESTABLISHED'.

else

Obtain a next hop (or interface). Find a Downstream LSP Control Block that has the same FEC and the same next hop and also is able to merge more input labels. If not found, create a new Downstream LSP Control Block with the state `IDLE'.

If the state of the Downstream LSP Control Block is `ESTABLISHED', choose an upstream label, connect the upstream label with the downstream label and send an LDP-MAPPING upstream with the upstream label, and go to the state `ESTABLISHED'.

If the state of Downstream LSP Control Block is not `ESTABLISHED', set the state of the Upstream LSP Control Block to `RESPONSE\_AWAITED'. If the LSR use the independent control mode [2], choose an upstream label, and send an LDP-MAPPING upstream.

Pass the event `Internal AddUpstream' to the Downstream LSP Control Block's state machine.

If unable to process the request for any reason, issue an LDP-NAK to the sender with the appropriate error code, go to IDLE and delete the control block.

State: IDLE

Event: Internal Downstream Mapping

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: LDP Release

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Internal Downstream Withdraw

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: LDP Upstream Abort

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Internal Downstream NAK

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Upstream Lost

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Internal Re-Cross-Connect

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Internal New NH

New State: IDLE

Actions:

Ignore the event. It is an internal implementation error.

#### 2.3.3.4.2 State -- "RESPONSE\_AWAITED"

State: RESPONSE\_AWAITED

Event: LDP Request

New State: RESPONSE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RESPONSE\_AWAITED

Event: Internal Downstream Mapping

New State: Depends on the action routine.

**Actions:**

If the LSR uses the ordered control mode, assign an upstream label, connect the upstream label to the downstream label and allocate the resources, send an LDP-MAPPING upstream with the upstream label and go to `ESTABLISHED`.

If unable to process the message for any reason, issue an LDP-NAK upstream and an LDP-RELEASE downstream, go to IDLE and delete the control block.

**State:** RESPONSE\_AWAITED

**Event:** LDP Release

**New State:** RESPONSE\_AWAITED

**Actions**

Ignore the event. It is a protocol error from the upstream peer.

**State:** RESPONSE\_AWAITED

**Event:** Internal Downstream Withdraw

**New State:** RESPONSE\_AWAITED

**Actions**

Ignore the event. It is an internal implementation error.

**State:** RESPONSE\_AWAITED

**Event:** LDP Upstream Abort

**New State:** IDLE

**Actions**

If the LSR uses the independent control mode, free the upstream label and the resources.

Send the event `Internal DeleteUpstream` to its Downstream LSP Control Block's state machine.

Delete the control block.

**State:** RESPONSE\_AWAITED



Event: Internal Downstream NAK

New State: IDLE

Actions:

If the LSR uses the independent control mode, free the upstream label and the resources. Then, send an LDP-WITHDRAW upstream.

If the LSR uses the ordered control mode, propagate the LDP-NAK upstream.

Delete the control block.

State: RESPONSE\_AWAITED

Event: Upstream Lost

New State: IDLE

Actions

If the LSR uses the independent control mode, free the upstream label and the resources.

Send the event `Internal DeleteUpstream' to its Downstream LSP Control Block's state machine.

Delete the control block.

State: RESPONSE\_AWAITED

Event: Internal Re-Cross-Connect

New State: RESPONSE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RESPONSE\_AWAITED

Event: Internal New NH

New State: depends on the actions

**Actions:**

Send event `Internal DeleteUpstream' to its old downstream control block.

Find a Downstream LSP Control Block that has the same FEC and the same next hop and also is able to merge more input labels. If not found, create a new Downstream LSP Control Block with the state `IDLE'.

If the state of the Downstream LSP Control Block is `ESTABLISHED', choose an upstream label, connect the upstream label with the downstream label and send an LDP-MAPPING upstream with the upstream label, and go to the state `ESTABLISHED'.

If the state of Downstream LSP Control Block is not `ESTABLISHED', set the state of the Upstream LSP Control Block to `RESPONSE\_AWAITED'.

Pass the event `Internal AddUpstream' to the new Downstream LSP Control Block's state machine.

**2.3.3.4.3 State -- "ESTABLISHED"**

State: ESTABLISHED

Event: LDP Request

New State: ESTABLISHED

**Actions**

Ignore the event. It is an internal implementation error.

State: ESTABLISHED

Event: Internal Downstream Mapping

New State: ESTABLISHED

**Actions**

Process the new attributes of the mapping and then propagate the LDP-MAPPING upstream.

State: ESTABLISHED

Event: LDP Release

New State: IDLE

Actions

Disconnect the upstream label from the downstream label, free the upstream label and resources.

Send the event `Internal DeleteUpstream' to its Downstream LSP Control Block's state machine.

Send the event `Internal Destroy' to the Next\_Hop\_Trigger\_Block's state machine if the LSR was in the middle of switching over to the better next hop.

Delete the control block.

State: ESTABLISHED

Event: Internal Downstream Withdraw

New State: Depends on the action routine.

Actions

If it uses independent mode, set its state to `IDLE' and create a internal `LDP Request' and send to its own state machine.

Else

Disconnect the upstream label from the downstream label.

Propagate the LDP-WITHDRAW upstream and go to state `RELEASE\_AWAITED'.

Send the event `Internal Destroy' to the Next\_Hop\_Trigger\_Block's state machine if the LSR was in the middle of switching over to the better next hop.

State: ESTABLISHED

Event: LDP Upstream Abort

New State: ESTABLISHED

Actions

Ignore the event.

Note: This scenario can occur if the upstream LSR sends a LDP-ABORT at about the same time as the local LSR sends a LDP-MAPPING. In this situation, it should be up to exactly one of the two LSRs as to whether or not the label that was sent remains valid. The LDP specification [4] procedures leave the choice to the upstream LSR that must send an LDP-RELEASE if it will not use the label provided.

State: ESTABLISHED  
Event: Internal Downstream NAK  
New State: Depends on the action routine.  
Actions:

If it uses independent mode, set its state to 'IDLE' and create a internal 'LDP Request' and send to its own state machine.

Else  
Disconnect the upstream label from the downstream label  
Send an LDP-WITHDRAW upstream and go to state  
'RELEASE\_AWAITED'.

Send the event 'Internal Destroy' to the Next\_Hop\_Trigger\_Block's state machine if the LSR was in the middle of switching over to the better next hop.

State: ESTABLISHED  
Event: Upstream Lost  
New State: IDLE  
Actions:

Disconnect the upstream label from the downstream label, free the upstream label and the resources.

Send the event 'Internal DeleteUpstream' to its Downstream LSP Control Block's state machine.

Send the event 'Internal Destroy' to the Next\_Hop\_Trigger\_Block's state machine if the LSR was in the middle of switching over to the better next hop.

Delete the control block.

State: ESTABLISH

Event: Internal Re-Cross-Connect

New State: ESTABLISH

Actions:

Reconnect the upstream label to the new downstream label.

Send the event 'Internal DeleteUpstream' to its old Downstream LSP Control Block's state machine.

State: ESTABLISH

Event: Internal New NH

New State: ESTABLISH

Actions:

Create a new Next\_Hop\_Trigger\_Control\_Block and pass event 'Internal New NH' to its state machine.

#### 2.3.3.4.4 State -- "RELEASE\_AWAITED"

State: RELEASE\_AWAITED

Event: LDP Request

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is a protocol error from the upstream LSR.

State: RELEASE\_AWAITED

Event: Internal Downstream Mapping

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

State: RELEASE\_AWAITED

**Event:** LDP Release

**New State:** IDLE

**Actions:**

Free the upstream label resource and delete the control block.

**State:** RELEASE\_AWAITED

**Event:** Internal Downstream Withdraw

**New State:** RELEASE\_AWAITED

**Actions:**

Ignore the event. It is a protocol error from the downstream LSR.

**State:** RELEASE\_AWAITED

**Event:** LDP Upstream Abort

**New State:** IDLE

**Actions:**

Free the upstream label resource and delete the control block.

**State:** RELEASE\_AWAITED

**Event:** Internal Downstream NAK

**New State:** RELEASE\_AWAITED

**Actions:**

Ignore the event. And continue waiting for the LDP-RELEASE.

**State:** RELEASE\_AWAITED

**Event:** Upstream Lost

**New State:** IDLE

**Actions:**

Free the upstream label resource and delete the control block.

State: RELEASE\_AWAITED

Event: Internal New NH

New State: RELEASE\_AWAITED

Actions:

Ignore the event. And continue waiting for the LDP-RELEASE.

State: RELEASE\_AWAITED

Event: Internal Re-Cross-Connect

New State: RELEASE\_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

#### 2.3.3.5 State of the Downstream LSP Control Block's State Machine for Downstream-on-demand Mode

-- IDLE

This is the initial LSP state.

-- RESPONSE\_AWAITED

This state means that the LSR has received an LDP-REQUEST from an upstream LSR, has processed the LDP-REQUEST, and has sent a new LDP-REQUEST towards a downstream LSR. The LSR is waiting for the LDP-MAPPING from the downstream LSR.

-- ESTABLISHED

This state means that the LSR has received the LDP-MAPPING from the downstream LSR and the LSP is up and operational.

#### 2.3.3.6 Events of the Downstream LSP Control Block's State Machine for Downstream-on-demand Mode

-- Internal AddUpstream

This event is sent by an Upstream LSP Control Block's state machine when it is created.

**-- Internal DeleteUpstream**

This event is sent by an Upstream LSP Control Block's state machine when it is deleted.

**-- LDP Mapping**

The LSR receives an LDP-MAPPING from a downstream LSR.

**-- LDP Withdraw**

The LSR receives an LDP-WITHDRAW from a downstream LSR.

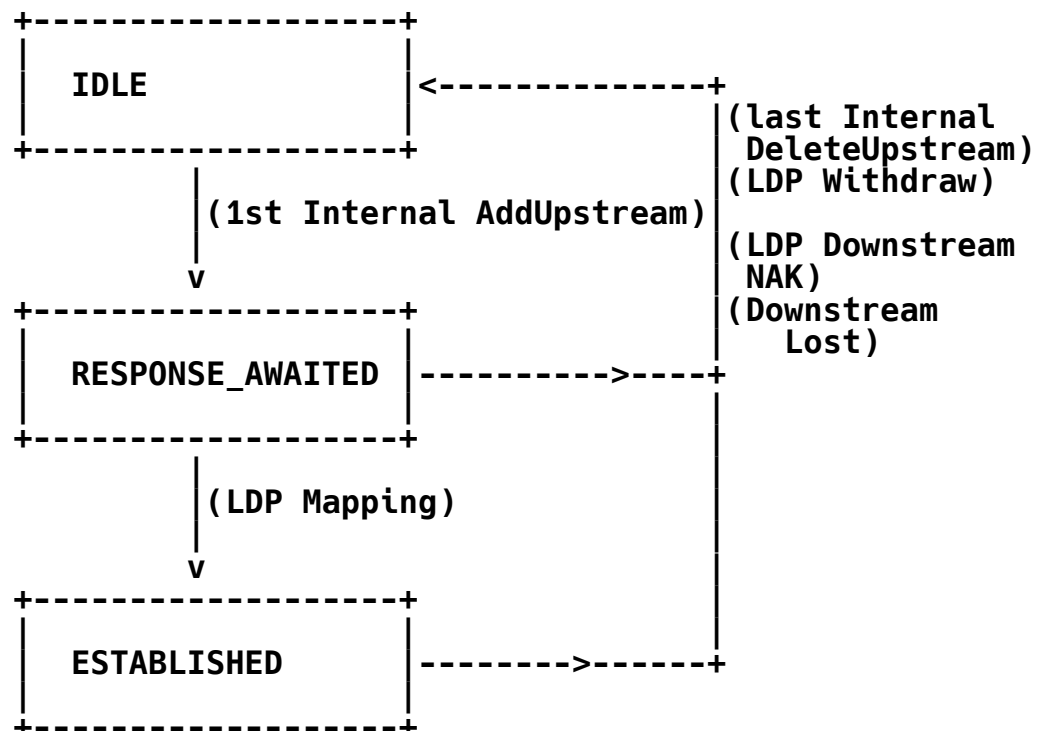
**-- LDP Downstream NAK**

The LSR receives an LDP-NAK from a downstream LSR.

**-- Downstream Lost**

The LSR loses the LDP session with its downstream LSR.

### 2.3.3.7 State Transitions of the Downstream LSP Control Block's State Machine for Downstream-on-demand mode





### 2.3.3.8 Downstream LSP Control Block's State Machine for Downstream-on-demand Mode.

#### 2.3.3.8.1 State -- "IDLE"

State: IDLE

Event: Internal AddUpstream

New State: RESPONSE\_AWAITED

##### Actions

Initialize the list of pointers in the Upstream LSP Control Block to contain the newly added upstream pointer.

Send a new LDP-REQUEST downstream and go to the state 'RESPONSE\_AWAITED'.

State: IDLE

Event: Internal DeleteUpstream

New State: IDLE

##### Actions

Ignore the event. It is an internal implementation error.

State: IDLE

Event: LDP Mapping

New State: IDLE

##### Actions

Ignore the event. It is an internal implementation error.

State: IDLE

Event: LDP Withdraw

New State: IDLE

##### Actions

Ignore the event. It is an internal implementation error.

State: IDLE

Event: LDP Downstream NAK

New State: IDLE

Actions

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Downstream Lost

New State: IDLE

Actions

Ignore the event. It is an internal implementation error.

2.3.3.8.2 State -- "RESPONSE\_AWAITED"

State: RESPONSE\_AWAITED

Event: Internal AddUpstream

New State: RESPONSE\_AWAITED

Actions

Add the pointer to new Upstream LSP Control Block to the Upstream LSP Control Blocks pointer list.

State: RESPONSE\_AWAITED

Event: Internal DeleteUpstream

New State: Depend on the action routine

Actions

Delete the Upstream LSP Control Block pointer from the Upstream LSP Control Block pointers list.

If the list becomes empty, release the resources, send an LDP-Abort downstream, go to IDLE and then delete the control block.

State: RESPONSE\_AWAITED

Event: LDP Mapping

New State: ESTABLISHED

**Actions**

For each Upstream LSP Control Block in the Upstream LSP Control Block pointers list, pass the event 'Internal Downstream Mapping' to its state machine.

State: RESPONSE\_AWAITED

Event: LDP Withdraw

New State: RESPONSE\_AWAITED

**Actions**

It is a protocol error from the downstream LDP peer; send a LDP-RELEASE downstream

State: RESPONSE\_AWAITED

Event: LDP Downstream NAK

New State: IDLE

**Actions**

For each Upstream LSP Control Block in the Upstream LSP Control Block pointers list, pass the event 'Internal Downstream NAK' to its state machine.

Release the resources, and delete the control block.

State: RESPONSE\_AWAITED

Event: Downstream Lost

New State: IDLE

**Actions**

For each Upstream LSP Control Block in the Upstream LSP Control Block pointers list, pass the event 'Internal Downstream NAK' to its state machine.

Release the resources, and delete the control block.

**2.3.3.8.3 State -- "ESTABLISHED"**

**State:** ESTABLISHED

**Event:** Internal AddUpstream

**New State:** ESTABLISHED

**Actions**

Add the pointer to new Upstream LSP Control Block to the Upstream LSP Control Block pointers list.

**State:** ESTABLISHED

**Event:** Internal DeleteUpstream

**New State:** Depends on the action routine.

**Actions**

Delete the pointer of Upstream LSP Control Block from its Upstream LSP Control Block pointers list.

If the list becomes empty, release the resources, send an LDP-RELEASE downstream, go to IDLE and then delete the control block.

Otherwise, remain in the ESTABLISHED state.

**State:** ESTABLISHED

**Event:** LDP Mapping

**New State:** ESTABLISHED

**Actions**

For each Upstream LSP Control Block in the Upstream LSP Control Block pointers list, pass the event 'Internal Downstream mapping' to its state machine.

**State:** ESTABLISHED

**Event:** LDP Withdraw

**New State:** IDLE

### Actions

For each Upstream LSP Control Block in the Upstream LSP Control Block pointers list, pass the event 'Internal Downstream withdraw' to its state machine.

Release the resources, and delete the control block and send LDP-RELEASE downstream.

State: ESTABLISHED

Event: LDP Downstream NAK

New State: ESTABLISHED

### Actions

It is a protocol error from the downstream LDP peer.

#### 2.3.3.9 State of the Next\_Hop\_Trigger\_Control\_Block's State Machine for Downstream-on-demand Mode

##### -- IDLE

This is the initial LSP state.

##### -- NEW\_NH\_RETRY

This is the state where an LSR waits for a retry timer to expire and then tries to establish an LSP through a new next hop.

##### -- NEW\_NH\_RESPONSE\_AWAITED

This state means that the LSR has sent a new LDP-REQUEST towards a downstream LSR. The LSR is waiting for the LDP-MAPPING from the downstream LSR.

#### 2.3.3.10 Events of the Next\_Hop\_Trigger\_Control\_Block's State Machine for Downstream-on-demand Mode

##### -- Internal New NH

Trigger to setup an LSP through a better next hop.

**-- Internal Downstream Mapping**

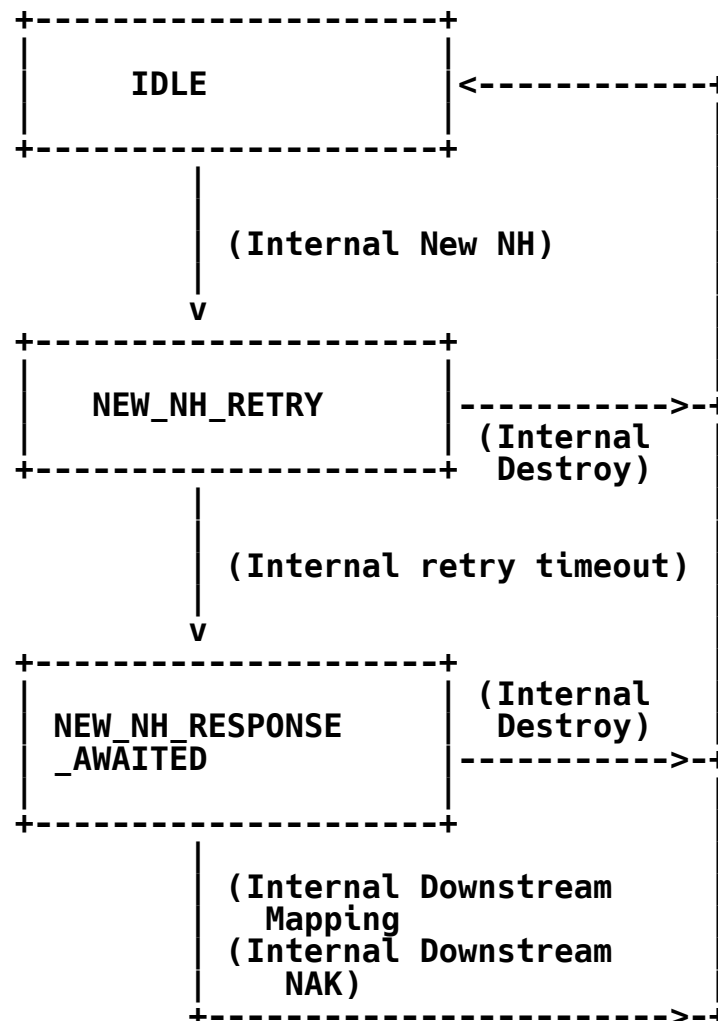
This event is sent by one Downstream LSP Control Block's state machine. This Downstream LSP Control Block is the merged Downstream LSP Control Block of this Upstream LSP Control Block. The event is the result of receiving an LDP-MAPPING by the Downstream LSP Control Block's state machine.

**-- Internal Downstream NAK**

This event is sent by one Downstream LSP Control Block's state machine. This Downstream LSP Control Block is the merged Downstream LSP Control Block of this Upstream LSP Control Block. The event is the result of receiving an LDP-NAK by the Downstream LSP Control Block's state machine, or it detects an error.

**-- Internal Destroy** This event is used to stop the procedure of switching over to the better next hop.

### 2.3.3.11 State Transitions of the Next\_Hop\_Trigger\_Control\_Block's State Machine for Downstream-on-demand Mode



### 2.3.3.12 State Machine

#### 2.3.3.12.1 State -- "IDLE"

State: IDLE  
 Event: Internal New NH  
 New State: NEW\_NH\_RETRY

**Actions:**

Start the LSP retry timer and go to the `NEW\_NH\_RETRY' state.

**State:** IDLE

**Event:** Internal retry timeout

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

**State:** IDLE

**Event:** Internal Downstream Mapping

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

**State:** IDLE

**Event:** Internal Downstream NAK

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.

**State:** IDLE

**Event:** Internal destroy

**New State:** IDLE

**Actions:**

Ignore. It is an internal implementation error.



### 2.3.3.12.2 State -- "NEW\_NH\_RETRY"

State: NEW\_NH\_RETRY

Event: Internal New NH

New State: NEW\_NH\_RETRY

Actions:

Restart the LSP retry timer.

State: NEW\_NH\_RETRY

Event: Internal retry timeout

New State: Depends on the action routine.

Actions:

If the new next hop is the same one as the old next hop, go to IDLE and delete the control block.

Otherwise, go to NEW\_NH\_RESPONSE\_AWAITED, find a downstream LSP control block that goes through the same next hop for the same FEC, if there are none, create one, and pass 'Internal AddUpstream' event to its state machine.

State: NEW\_NH\_RETRY

Event: Internal Downstream Mapping

New State: NEW\_NH\_RETRY

Actions:

Ignore. It is an internal implementation error.

State: NEW\_NH\_RETRY

Event: Internal Downstream NAK

New State: NEW\_NH\_RETRY

Actions:

Ignore. It is an internal implementation error.

State: NEW\_NH\_RETRY  
Event: Internal destroy  
New State: IDLE  
Actions:

Stop the timer and delete the control block.

#### 2.3.3.12.3 State -- "NEW\_NH\_RESPONSE\_AWAITED"

State: NEW\_NH\_RESPONSE\_AWAITED  
Event: Internal New NH  
New State: NEW\_NH\_RETRY  
Actions:

Restart the LSP retry timer and send event 'Internal destroy' to the control block of the LSP for the new next hop.

State: NEW\_NH\_RESPONSE\_AWAITED  
Event: Internal retry timeout  
New State: NEW\_NH\_RESPONSE\_AWAITED  
Actions:

Ignore. It is an internal implementation error.

State: NEW\_NH\_RESPONSE\_AWAITED  
Event: Internal Downstream Mapping  
New State: IDLE  
Actions:

Send event 'Internal Re-cross-connect' event to the upstream LSP control block of the new next hop.

Send event 'DeleteUpstream' event to the downstream LSP control block of the the new next hop, since the upstream has spliced into the new next hop.

Delete the control block.

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal Downstream NAK

New State: IDLE

Actions:

Delete the control block.

State: NEW\_NH\_RESPONSE\_AWAITED

Event: Internal destroy

New State: IDLE

Actions:

Send event 'Internal DeleteUpstream' the control block for the new LSP through the new next hop.

#### 2.3.4 LDP Related Message Processing

- If an LSR receives an LDP-REQUEST:

a) If this is a duplicate request, discard the message. A duplicate request means that there is a LSP Control Block that has the same FEC, Upstream Label Request ID and Upstream LDP Session Identifier.

b) Otherwise, create a new Upstream LSP Control Block. Then pass the event 'LDP Request' to this Upstream LSP Control Block's state machine.

- If an LSR receives an LDP-MAPPING:

Locate a Downstream LSP Control Block that has the same FEC, the same Downstream LDP Session Identifier and the same Downstream Label. If a Downstream LSP Control Block is found, pass the event 'LDP Mapping' to its state table. This could mean that the attributes of label binding have changed.

Otherwise, use the Downstream LDP request ID (the 'Label Request Message ID' field in the LDP-MAPPING) and Downstream LDP Session Identifier to locate the Downstream LSP Control Block and pass the

event 'LDP Mapping' to its state machine. If no Downstream LSP Control Block is found, ignore the message.

- If an LSR receives an LDP-RELEASE:

Locate an Upstream LSP Control Block that has the same FEC, the same Upstream Label, the same Upstream LDP Session Identifier. If no Upstream LSP Control Block is found, ignore the message. If an Upstream LSP Control Block is found, send the event 'LDP Release' to its state machine.

- If an LSR receives an LDP-WITHDRAW:

Find a Downstream LSP Control Block that has the same FEC, the same Downstream LDP Session Identifier and the same Downstream Label. Pass the event 'LDP Withdraw' to its state machines.

- If an Upstream LDP peer is lost:

Pass the event 'Upstream Lost' to the state machines of all the Upstream LSP Control Blocks whose upstream LDP peer is that LSR.

- If a Downstream LDP peer is lost:

Pass the event 'Downstream Lost' to the state machines of all the Downstream LSP Control Blocks whose downstream LDP peer is that LSR.

- If a next hop of an FEC is changed:

For all the Upstream LSP Control Blocks that are affected by this change, pass the event 'Internal New NH' to their state machines.

- If an LSR receives an LDP-ABORT from an upstream LSR:

Use the Upstream LDP Request ID and Upstream LDP Session Identifier to locate the Upstream LSP Control Block and pass the event 'LDP Abort' to its state machine.

- If an LSR receives an LDP-NAK from a downstream LSR:

Use the Downstream LDP Request ID and Downstream Session Identifier to locate a Downstream\_LSP\_control\_block and pass the event 'LDP Downstream NAK' to its state machine.

### 3. State Machine for Downstream Unsolicited

The following sections describe the state machines for the ATM-LSR that uses downstream unsolicited mode.

While both independent LSP control and ordered LSP control modes are possible, only the ordered mode is taken into account, because the independent LSP control mode uses the liberal label retention mode and so is considered burning too many ATM resources.

In downstream unsolicited mode, multiple path is not supported in this version and will be For Further Study (FFS). We suspect with multiple next hops and Downstream mode, it is easy to get into a loop condition.

#### 3.1 Control Block

There are 2 types of control blocks involved: Upstream LSP Control Block, Downstream LSP Control Block.

There is a list of Upstream LSP Control Blocks for each FEC in the routing table, with each one corresponding to a LDP peer. A Upstream LSP Control Block is created for each FEC when there is a label ready to be distributed to that upstream. It is deleted when the FEC is deleted from the FEC table, or the LDP peer disappears, or the downstream label is withdrawn.

There is one Downstream LSP Control Blocks for each FEC in the routing table. It is created when the FEC is inserted into the forwarding table and deleted when the FEC is removed from the forwarding table.

An Upstream LSP Control Block may contain the following information:

- Upstream LDP Session Identifier
- State
- Upstream Label (assigned by this LSR)
- FEC

Upstream Label and Upstream LDP Session Identifier, or FEC and Upstream LDP Session Identifier can be used to locate a unique Upstream LSP Control Block.

A Downstream LSP Control Block may contain the following information:

- FEC
- State
- Downstream LDP Session Identifier
- Downstream Label (assigned by the downstream LSR)
- Downstream Label Request ID (assigned by the LSR itself)

Downstream Label and Downstream LDP Session Identifier, or FEC and Downstream LDP Session Identifier can be used to locate a unique Downstream LSP Control Block.

### 3.2 States of the Upstream LSP Control Block's State Machine for Downstream Mode

#### -- IDLE

This is the initial LSP state.

#### -- ESTABLISHED

This state means that the LSR has received the LDP-MAPPING from the downstream LSR and the LSP is up and operational.

#### -- RELEASE\_AWAITED

This state means that the LSR is waiting for the LDP-RELEASE in respond to the LDP-WITHDRAW sent by this LSR.

#### -- RESOURCES\_AWAITED

This state means that the LSR is waiting for the label resources.

### 3.3 Events of the Upstream LSP Control Block's State Machine for Downstream Mode

#### -- Internal Downstream Mapping

This event is sent by one Downstream LSP Control Block's state machine. The event is the result of receiving an LDP-MAPPING by the Downstream LSP Control Block's state machine. Or when the LDP peer is discovered and there is a downstream Label available for this FEC.

**-- LDP Release**

The LSR receives an LDP-RELEASE from an upstream LSR.

**-- Internal Withdraw**

This event is sent by Downstream LSP Control Block's state machine. The event is the result of receiving an LDP-WITHDRAW by the Downstream LSP Control Block's state machine.

**-- Resource Available**

This event means the local resource (such as label) becomes available.

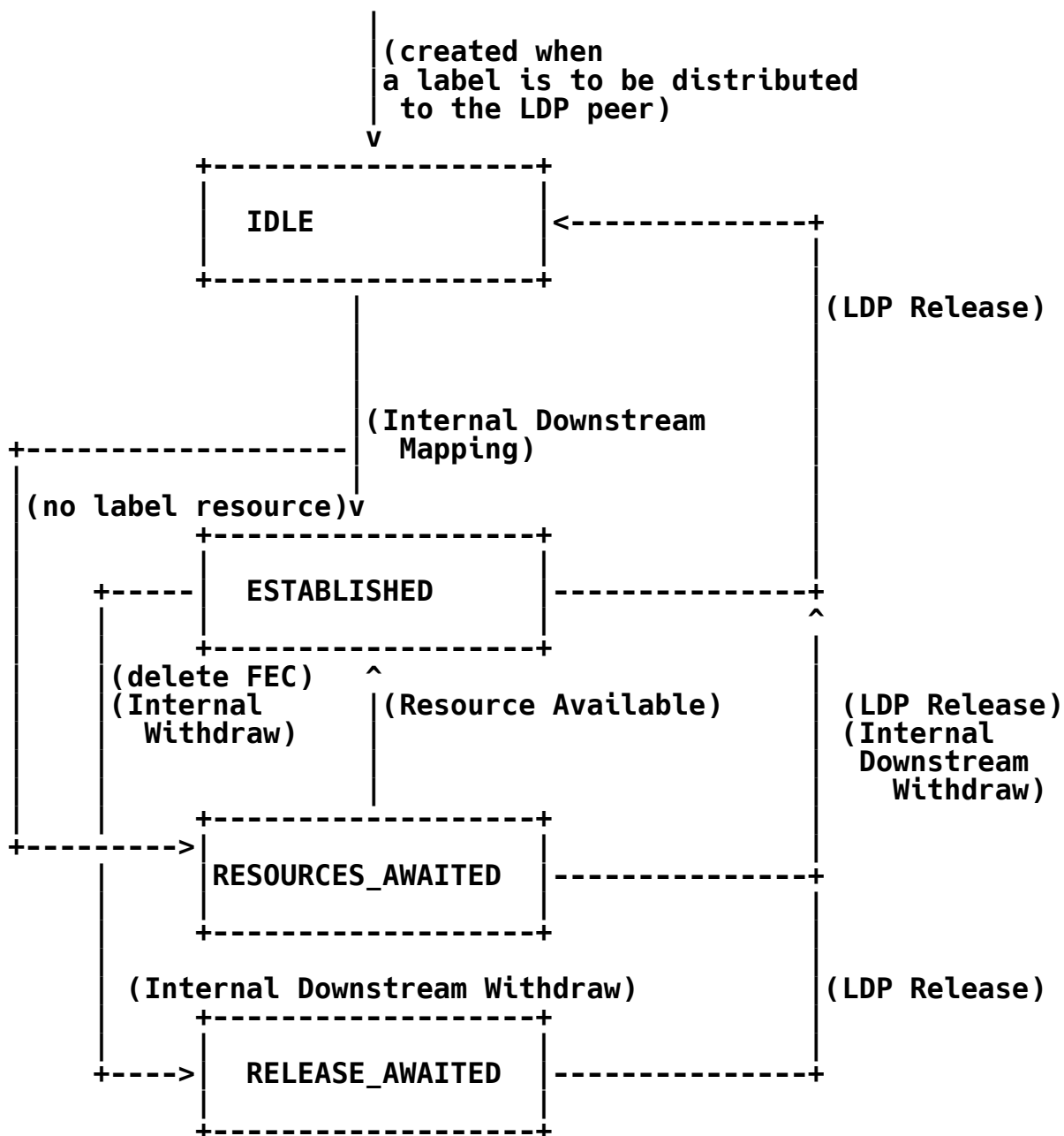
**-- Delete FEC**

This event means that the FEC is removed from the forwarding table.

**-- Upstream Lost**

This event means that the upstream LDP peer is lost.

### 3.4 State Transitions of Upstream LSP Control Block's State Machine for Downstream Mode





### 3.5 Upstream LSP Control Block's State Machine for Downstream Mode

#### 3.5.1 : State -- "IDLE"

State: IDLE

Event: Internal Downstream mapping

New State: Depends on the action routine.

##### Actions

Choose an upstream label, connect the upstream label with the downstream label, propagate the LDP-MAPPING upstream and go to state 'ESTABLISHED'.

If there is no resource for the upstream label, go to state 'RESOURCE\_AWAITED'.

State: IDLE

Event: LDP Release

New State: IDLE

##### Actions

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Internal Downstream Withdraw

New State: IDLE

##### Actions

Ignore the event. It is an internal implementation error.

State: IDLE

Event: Resource Available

New State: IDLE

##### Actions

Ignore the event. It is an internal implementation error.

**State: IDLE**

**Event: Delete FEC**

**New State: IDLE**

**Actions**

**Delete the control block.**

**State: IDLE**

**Event: Upstream Lost**

**New State: IDLE**

**Actions**

**Delete the control block.**

**3.5.2 : State -- "ESTABLISHED"**

**State: ESTABLISHED**

**Event: Internal Downstream Mapping**

**New State: ESTABLISHED**

**Actions**

**Process the new attributes of the new mapping message.**

**Propagate the LDP-MAPPING upstream.**

**State: ESTABLISHED**

**Event: LDP Release**

**New State: IDLE**

**Actions**

**Disconnect upstream label from downstream label.**

**Release the upstream label resource**

**Delete the control block.**

State: ESTABLISHED  
Event: Internal Downstream Withdraw  
New State: RELEASE\_AWAITED

**Actions**

Disconnect upstream label from downstream label.

Propagate the LDP-WITHDRAW upstream.

State: ESTABLISHED  
Event: Resource Available  
New State: ESTABLISHED

**Actions**

Ignore the event. It is an internal implementation error.

State: ESTABLISHED  
Event: Delete FEC  
New State: RELEASE\_AWAITED

**Actions**

Send a LDP-WITHDRAW upstream.

State: ESTABLISHED  
Event: Upstream Lost  
New State: IDLE

**Actions**

Release the upstream label and delete the control block.

**3.5.3 : State -- "RELEASE\_AWAITED"**

State: RELEASE\_AWAITED  
Event: Internal Downstream Mapping

**New State:** RELEASE\_AWAITED

**Actions**

Ignore the message.

**State:** RELEASE\_AWAITED

**Event:** LDP Release

**New State:** IDLE

**Actions**

Release the upstream label and delete the control block.

**State:** RELEASE\_AWAITED

**Event:** Internal Downstream Withdraw

**New State:** RELEASE\_AWAITED

**Actions**

Ignore the event.

**State:** RELEASE\_AWAITED

**Event:** Resource Available

**New State:** RELEASE\_AWAITED

**Actions**

Ignore the event. It is an internal implementation error.

**State:** RELEASE\_AWAITED

**Event:** Delete FEC

**New State:** RELEASE\_AWAITED

**Actions**

Do nothing.

**State:** RELEASE\_AWAITED

Event: Upstream Lost

New State: IDLE

Actions

Release the upstream label and delete the control block.

3.5.4 : State -- "RESOURCE\_AWAITED"

State: RESOURCE\_AWAITED

Event: Internal Downstream Mapping

New State: RESOURCE\_AWAITED

Actions

Ignore the message.

State: RESOURCE\_AWAITED

Event: LDP Release

New State: RESOURCE\_AWAITED

Actions

Ignore the message. It is an internal implementation error.

State: RESOURCE\_AWAITED

Event: Internal Downstream Withdraw

New State: IDLE

Actions

Delete the control block.

State: RESOURCE\_AWAITED

Event: Resource Available

New State: ESTABLISHED

**Actions**

Allocate an upstream label, connect the upstream label with the downstream label, and send LDP-MAPPING upstream.

State: RESOURCE\_AWAITED

Event: Delete FEC

New State: IDLE

**Actions**

Delete the control block.

State: RESOURCE\_AWAITED

Event: Upstream Lost

New State: IDLE

**Actions**

Delete the control block.

### 3.6 State of the Downstream LSP Control Block's State Machine for Downstream Mode

**-- IDLE**

This is the initial LSP state.

**-- ESTABLISHED**

This state means that the LSR has received the LDP-MAPPING from the downstream LSR.

### 3.7 Events of the Downstream LSP Control Block's State Machine for Downstream Mode

**-- LDP Mapping**

The LSR receives an LDP-MAPPING from a downstream LSR.

**-- LDP Withdraw**

The LSR receives an LDP-WITHDRAW from a downstream LSR.

**-- Delete FEC**

The FEC is deleted from the forwarding table.

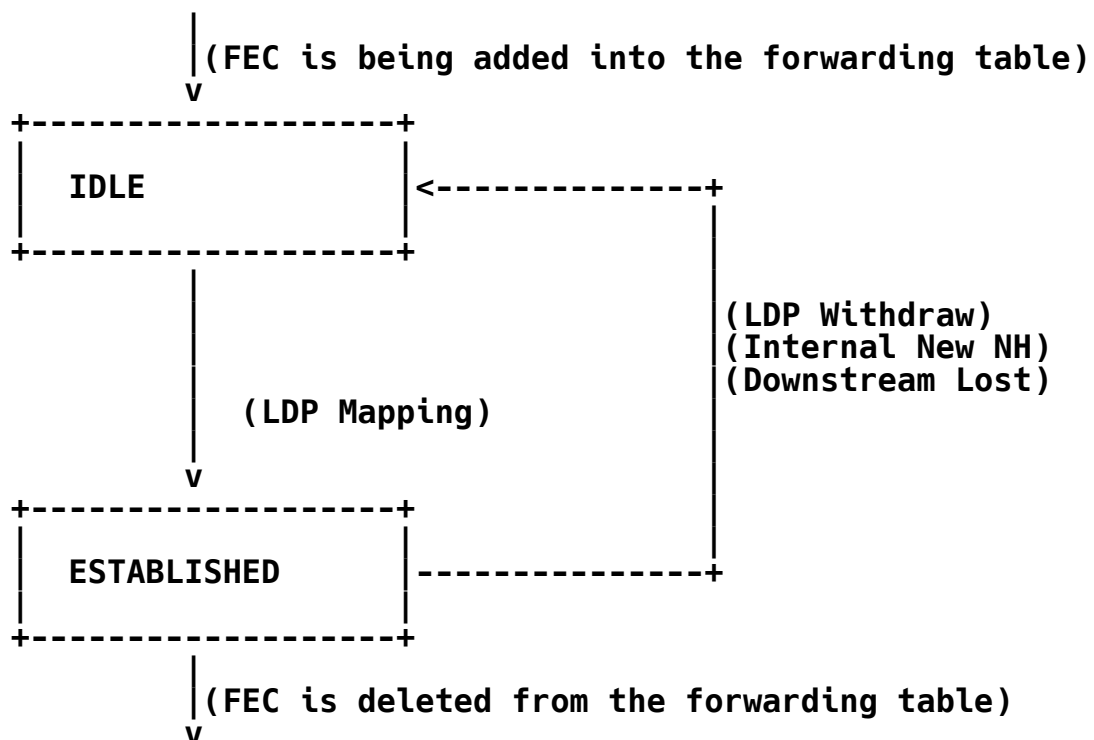
**-- Next Hop Change**

The next hop for this FEC is change to different LSR.

**-- Downstream Lost**

The downstream peer is gone.

### 3.8 State Transitions of Downstream LSP Control Block's State Machine for Downstream Mode



### 3.9 Downstream LSP Control Block's State Machine for Downstream Mode

#### 3.9.1 : State -- "IDLE"

State: IDLE  
 Event: LDP mapping  
 New State: ESTABLISHED

**Actions**

For all the LDP peers except the downstream LSR that assigned the label, create an Upstream LSP Control Block, and pass the event 'Internal Downstream Mapping' to each of the Upstream LSP Control Block's state machines.

**State:** IDLE

**Event:** LDP withdraw

**New State:** IDLE

**Actions**

Ignore the event. It is an internal implementation error.

**State:** IDLE

**Event:** Delete FEC

**New State:** IDLE

**Actions**

Delete the control block.

**State:** IDLE

**Event:** Next Hop Change

**New State:** IDLE

**Actions**

Ignore the event.

**State:** IDLE

**Event:** Downstream Lost

**New State:** IDLE

**Actions**

Ignore the event.



**3.9.2 : State -- "ESTABLISHED"**

**State:** ESTABLISHED

**Event:** LDP mapping

**New State:** ESTABLISHED

**Actions**

For each Upstream\_LSP\_control\_block of this FEC, pass event  
'Internal downstream mapping' to its state machine.

**State:** ESTABLISHED

**Event:** LDP withdraw

**New State:** IDLE

**Actions**

For each Upstream\_LSP\_control\_block for this FEC, pass event  
'Internal downstream Withdraw' to its state machine.

Send a LDP Withdraw downstream.

**State:** ESTABLISHED

**Event:** Delete FEC

**New State:** IDLE

**Actions**

Send LDP-RELEASE downstream and delete the control block.

**State:** ESTABLISHED

**Event:** Next Hop Change

**New State:** IDLE

**Actions**

For each Upstream\_LSP\_control\_block for this FEC, pass event  
'Internal downstream Withdraw' to its state machine.

Send LDP-REQUEST to the new next hop.

State: ESTABLISHED  
Event: Downstream Lost  
New State: IDLE

#### Actions

Send LDP-WITHDRAW to all Upstream\_Control\_Block's state machine of this FEC.

### 3.10 LDP Related Message Processing for downstream mode.

- If an LSR receives an LDP-REQUEST:

If there is a next hop for this FEC and there is a Downstream\_Control\_Block for this FEC whose state is 'ESTABLISHED', create a new Upstream\_Control\_Block and pass 'internal Mapping' event to its state machine.

- If an LSR receives an LDP-MAPPING:

Locate a Downstream LSP Control Block that has the same FEC, the same Downstream LDP Session Identifier and the same Downstream Label. If a Downstream LSP Control Block is found, pass the event 'LDP Mapping' to its state table. This could mean that the attributes of label binding have changed.

Otherwise, if there is no matching Downstream LSP Control Block found, find a Downstream LSP Control Block of this FEC and its next hop is the this downstream peer, pass the event 'LDP Mapping' to its state machine.

- If an LSR receives an LDP-RELEASE:

Locate an Upstream LSP Control Block that has the same FEC, the same Upstream Label, the same Upstream LDP Session Identifier. If no Upstream LSP Control Block is found, ignore the message. If an Upstream LSP Control Block is found, send the event 'LDP Release' to its state machine.

- If an LSR receives an LDP-WITHDRAW:

Find a Downstream LSP Control Block that has the same FEC, the same Downstream LDP Session Identifier and the same Downstream Label. Pass the event 'LDP Withdraw' to its state machines.

- If an Upstream LDP peer is lost:

Pass the event 'Upstream Lost' to the state machines of all the Upstream LSP Control Blocks whose upstream LDP peer is that LSR.

- If a Downstream LDP peer is lost:

Pass the event 'Label Withdraw' to the state machines of all the Downstream LSP Control Blocks whose the downstream LDP peer is that LSR.

- If a next hop of an FEC is changed:

Find all the Downstream LSP Control Blocks that has the same FEC and the same next hop and pass the event 'Next Hop Change' to their state machine

- If there is a FEC being added to the forwarding table

Create a new Downstream LSP Control Block with state 'IDLE'

- If the FEC is deleted from the forwarding table

Send the 'Delete FEC' event to the its control block.

- If an LSR receives an LDP-NAK from an upstream LSR:

Ignore the message. An LDP-NAK should never appear in the downstream-mode LSR

- If an LSR receives an LDP-NAK from a downstream LSR:

Ignore the message. It is a protocol error from the downstream LSR.

#### 4. Security Considerations

This document is provided as an informational extension of the LDP specification [4]. State machines presented here are intended to clarify procedures defined in the LDP specification, but do not supplant or override definitions and procedures provided there.

Implementations of a state machine may be vulnerable to spurious events generated by an external source. In this document, events fall in two categories: internal events and external events caused by receipt of an LDP message.

LDP messages may be protected using mechanisms described in the LDP specification. See "Security Considerations" in the LDP specification [4].

Security considerations relating to generation of spurious internal events are not addressed in this document.

## 5. Acknowledgements

The authors would like to acknowledge the helpful comments and suggestions of the following people: Bob Thomas, Myunghee Son and Adrian Farrel.

## 6. References

- [1] Davie, B., Lawrence, J., McCloghrie, K., Rosen, E., Swallow, G., Rekhter, Y. and P. Doolan, "MPLS Using LDP and ATM Switching", RFC 3035, January 2001.
- [2] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [3] Cucchiara, J., Sjostrand, H. and J. Luciani, "Definition of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)", Work in Progress.
- [4] Andersson, L., Doolan, P., Feldman, F., Fredette, A. and B. Thomas, "LDP Specification", RFC 3036, January 2001.
- [5] Jamoussi, B., Ed., O., Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Heinanen, J., Kilty, T. and A. Malis, "Constraint-Based LSP Set up Using LDP", RFC 3212, January 2002.

## 7. Authors' Address

Christophe Boscher  
Alcatel  
Le Mail  
44700 Orvault  
France

Phone: (33) 251781828  
EMail: christophe.boscher@alcatel.fr

Pierrick Cheval  
Alcatel  
5 rue Noel-Pons  
92734 Nanterre Cedex  
France

Phone: (33) 146524027  
EMail: pierrick.cheval@space.alcatel.fr

Liwen Wu  
Cisco Systems, Inc.  
3550 Cisco Way  
San Jose, CA 95134  
U.S.A

Phone: 408-853-4065  
EMail: liwwu@cisco.com

Eric Gray  
Sandburst Corporation  
600 Federal Drive  
Andover, MA 01810

Phone: (978) 689-1610  
EMail: eric.gray@sandburst.com

## 8. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

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

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

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

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.