MPLS Working Group
Internet Draft
Expiration Date: August 1999

Liwen Wu
Pierrick Cheval
Christophe Boscher
Alcatel USA

Eric Gray
Lucent

Februray 1999

LDP State Machine
draft-ietf-mpls-ldp-state-00.txt

0. Status of this Memo

1. Abstract

In the current LDP draft [Ref5], there is no state machine specified for processing the LDP messages. We think that defining a common state machine is very important for interoperability between different ldp implementations.

This draft provides state machine tables for ATM switch LSRs. We begin in section 2 by defining a list of terminologies. Then in section 3, we propose two sets of state machine tables for ATM switch LSRs, one for non-vc merge ATM LSRs and one for the vc merge ATM LSRs. Finally, in section 4, we outline the possible future working items.

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

2. Terminologies

- 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-NAK: It is a LDP notifications which is used to abort a LSP setup or reject a LSP Label Request message.

3. State Machine

This draft focuses on the LDP state machines and the associated control blocks. It does not describe the "LDP controller" which is in charge of LDP session initialization, address mapping messages management, routing interface, etc.

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

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

The loop detection and loop prevention message will be processed in the way as specified in [Ref5]. The impact of loop detection and loop prevention messages on a state transition is left for further study.

3.0 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, the LSR ignores the new next hop change, and the LSP stays unchanged. The LSR may decide to send an LDP-MAPPING which contains the attribute about this New Next Hop (NH) change.

3.1. ATM Switch LSR with NO VC-merge

In an MPLS domain where some ATM LSRs do not have VC-merge capability, downstream-on-demand mode is required for these ATM LSRs[Ref1]. Also, "conservative label retention mode" is required in this case[Ref1].

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

1) the LSP Control Block and its state machine which 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 which 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.

3.1.1 LSP Control Block

For each LSP, an LSP_Control_Block is defined which may contain the following information:

- Upstream Label Request ID(assigned by the upstream LSR), which is the 'Return Message Id' in the Label Request Message received from the upstream LSR.

- Downstream Label Request ID (assigned by this LSR itself), which is 'Return Message Id' in the Label Request Message sent to the downstrearm LSR.

- Upstream LDP Session Identifier

- Downstream LDP Session 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)which points to the control block that triggers setting up this LSP or tearing down this LSP.

- Next Hop Control Block Pointer, which points to the control block which 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 Session Identifier, or

- Upstream Label and Upstream LDP Session Identifier, or

- Upstream Label Request ID and Upstream LDP Session Identifier

Here is the relationship between different control blocks, the detail definition of Next Hop Trigger Block is described in section 3.1.6.

For example, an LSP which transit through (LSR-A, LSR-B, LSR-C,LSR-D):

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

The control blocks in LSR-A are:

```
+-----------------------+
| Trigger Control Block |
|  (e.g, by config)     |
+-----------------------+
           ^
           |(Trigger Control block pointer)
           |
           |
+-----------------------+
| LSP Control Block     |
+-----------------------+
```

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:

```
   +------------------------+
   | LSP Control Block      |
   | (original LSP)         |
   +------------------------+
(LSP      ^  |
 Control  |  | (Next Hop Control Block Pointer)
 Block    |  |
 Pointer  |  v
   +---------------------------------+
   | Next Hop Trigger Control Block |
   +---------------------------------+
          ^  |
(Trigger  |  | (New Next Hop LSP
  Control |  |  Control Block Pointer)
  Block   |  |
  Pointer)|  |
          |  v
   +-------------------------+
   | LSP Control Block       |
   | (for LSP: LSR-B,LSR-E,  |
   |   LSR-D)                |
   +-------------------------+
```

3.1.2   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 message from the downstream LSR.

-- ESTABLISHED

This state means that the LSR has received the LDP-MAPPING message

from the downstream LSR and the LSP is up and operational.

-- RELEASE_AWAITED

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

3.1.3 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 Nak

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

-- LDP Downstream Nak

The LSR receives an LDP-NAK from a 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.

-- Internal New NH

The LSR decides to switch over the better next hop.

3.1.4 State Transitions

The following diagram describes briefly the state transitions.

```
               +-------------------+
               |                   |
     +-------->|  IDLE             |<--------------+
     |         |                   |               |
     |         +-------------------+               |(LDP Release)
     |(LDP Upstream Nak) |                         |
     |(LDP DownstreamNak)|(LDP Request)            |(LDP Upstream Nak)
     |(Downstream Lost)  |(Internal SetUp)         |
     |                   |                         |
     |(Upstream Lost)    v                         |(Upstream Lost)
     |         +-------------------+               |
     |         |                   |               |
     +---------|  RESPONSE_AWAITED |               |
               |                   |               |
               +-------------------+               |
                        |                          |
                        |(LDP Mapping)             |
                        |                          |
                        v                          |
               +-------------------+               |
               |                   |               |
               |  ESTABLISHED      |------------>--+
               |                   |               |
               +-------------------+               |
                        |                          |
                        |                          |
                        |(LDP Withdraw)            |
                        |(Downstream Lost)         |
                        |                          |
                        v                          |
               +-------------------+               |
               |                   |               |
               | RELEASE_AWAITED   |------------>--+
               |                   |
               +-------------------+
```

3.1.5 State Machine

3.1.5.1 State -- "IDLE"

```
State:          IDLE

Event:          LDP Request

New States:     Depends upon the action routine

Actions:

  If this LSR is the LSP Egress or Proxy Egress [Ref2]

  Then:
    Choose an upstream label, connect this upstream label to the local
    IP forwarding module, allocate the resources, send the LDP-MAPPING
    message 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 message, propagate the LDP-REQUEST message towards the
    obtained next hop, and go to the new state `RESPONSE_AWAITED'.

    If the LSR uses the independent control mode[Ref2], choose an
    upstream label, connect this upstream label to the local IP
    forwarding module, and send an LDP-MAPPING message upstream with
    the upstream label.

  If unable to process the request for any reason, issue an LDP-NAK
  message to the sender with the appropriate error code and delete the
  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 Nak

 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 implementaion error.

3.1.5.2 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 message 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 message downstream and an LDP-Nak upstream, 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-NAK message downstream and delete the LSP_Control_Block.

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 Nak

New State: IDLE

Actions:

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

Send an LDP-NAK message 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.

2. Else If the LSR uses the independent control mode, send an LDP-WITHDRAW message upstream and go to state 'RELEASE_AWAITED'.

If the LSR uses the ordered control mode, send an LDP-NAK message upstream. Then 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.

Send an LDP-NAK downstream 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.

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 and then delete the 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-NAK downstream 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-NAK to the old downstream, and send LDP-REQUEST to the new next hop.

3.1.5.3 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 message, which may contain the new attributes

of the label mapping and then propagate the LDP-MAPPING message 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 Block if it was in the middle of switching over to the better next hop.

Propagate the LDP-RELEASE message downstream if the LSR is not the end of LSP 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 triggerd to be setup by itself(e.g it is the ingress LSR of this LSP), send event "Internal LSP NAK" to the trigger control block and delete the control block

3.2) else, if it is triggered by the incoming LDP Request, Disconnect the upstream label from the downstream label. And Propagate the LDP-WITHDRAW message 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.

State: ESTABLISHED

Event: LDP Upstream Nak

New State: IDLE

Actions:

Disconnect the upstream label from the downstream label.

Free the upstream label.

Send event "Internal Destroy" if the LSR was in the middle of switching over to the better next hop.

Propagate an LDP-RELEASE message downstream and delete the LSP_Control_Block.

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 msg 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 message downstream and delete the LSP_Control_Block.

State: ESTABLISHED

Event: Downstream Lost

New State: Depends on the actions

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 and delete the control block.

2) Else, disconnect the upstream label from the downstream label. Propagate an LDP-WITHDRAW message upstream and go to state '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 message downstream 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 which points this control block. And then pass "Internal New NH" to this control block.

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:

Ignore the event. It is an protocol error from the downstream LDP peer.

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: IDLE

Actions:

Ignore the event. It is an protocol error from the downstream LDP peer.

State   RELEASE_AWAITED

Event   LDP Upstream Nak

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.

3.1.6 Handling the Next Hop Change

When an LSR detects a better next hop, it may decides 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.

3.1.6.1 Next Hop Trigger Control Block

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

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

3.1.6.2 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 message.

3.1.6.3 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.

3.1.6.4 State Transition for next hop change

```
                   |
                   |
                   |
                   | Internal New NH
                   |
                   v
          +---------------------+
          |                     | Internal destroy
          | NEW_NH_RETRY        |--------------
          |                     |             |
          +---------------------+             |
                   |                          |
                   |                          |
                   | (Internal retry timeout) |
                   |                          |
                   v                          |
          +---------------------+             |
          |                     | Internal    |
          | NEW_NH_RESPONSE     | Destroy     |
          | _AWAITED            |-------------|
          |                     |             |
          +---------------------+             |
                   |                          |
                   | (Internal LSP UP)        |
                   | (Internal LSP NAK)       |
                   v                          v
                    the control block is deleted
```

3.01.3 State Machine

3.01.3.1 State -- "IDLE"

State: IDLE

Event: Internal New NH

New States: NEW_NH_RETRY

Actions:

Start the LSP retry timer and go to the 'NEW_NH_RETRY' state.

State: IDLE

Event: Internal retry timeout

New States: IDLE

Actions:

Ignore. It is an internal implementation error.

State: IDLE

Event: Internal LSP UP

New States: IDLE

Actions:

Ignore. It is an internal implementation error.

State: IDLE

Event: Internal LSP NAK

New States: IDLE

Actions:

Ignore. It is an internal implementation error.

State: IDLE

Event: Internal destroy

New States: IDLE

Actions:

Ignore. It is an internal implementation error.

3.01.3.2 State -- "NEW_NH_RETRY"

State: NEW_NH_RETRY

Event: Internal New NH

New States: NEW_NH_RETRY

Actions:

Restart the LSP retry timer.

State: NEW_NH_RETRY

Event: Internal retry timeout

New States: NEW_NH_RESPONSE_AWAITED

Actions:

If the new next hop is the same one as the old next hop, delete the control block.

Otherwise, create an LSP control block which will try to establish a new LSP through the new next hop. And send event "Internal Setup" to its state machine.

State: NEW_NH_RETRY

Event: Internal LSP UP

New States: NEW_NH_RETRY

Actions:

Ignore. It is an internal implementation error.

State: NEW_NH_RETRY

Event: Internal LSP NAK

New States: NEW_NH_RETRY

Actions:

Ignore. It is an internal implementation error.

State: NEW_NH_RETRY

Event: Internal destroy

New States: IDLE

Actions:

Stop the timer and delete the control block.

3.01.3.3 State -- "NEW_NH_RESPONSE_AWAITED"

State: NEW_NH_RESPONSE_AWAITED

Event: Internal New NH

New States: 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 States: NEW_NH_RESPONSE_AWAITED

Actions:

Ignore. It is an internal implementation error.

State: NEW_NH_RESPONSE_AWAITED

Event: Internal LSP UP

New States: 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 delete the control block.

State: NEW_NH_RESPONSE_AWAITED

Event: Internal LSP NAK

New States: IDLE

Actions:

Delete the control block.

State: NEW_NH_RESPONSE_AWAITED

Event: Internal destroy

New States: IDLE

Actions:

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

3.1.7 LDP Related Message Handling

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

a) If this is a duplicate request, discard the msg. A duplicate request means that there is a LSP_Control_Block which has same FEC, same 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 message.

b) Find an LSP_Control_Block which 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 which has the same Downstream Lable 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 message.

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

a) Find an LSP_Control_Block which 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 which 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 message.

- 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 which 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-NAK from an upstream LSR:

  a) Extract the LDP Request ID value from the LDP-NAK message.

  b) Find an LSP_Control_Block which 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 Nak' 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 which 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

  d) Otherwise, ignore the message.

3.2. ATM Switch LSR with VC-merge

3.2.1 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 [Ref2], 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.

3.2.2 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 of an FEC. This LSR must assign a unique upstream label and it can not merge this upstream label into an existing downstream label of this FEC.

There is one Next Hop Trigger Control Block for each FEC which was detected with a better next hop and the LSR decides to switch over 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, which points to the one which 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 which 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.

```
    +---------------------+
    |                     |
    | Upstream_LSP_Control|
    | _Block  for Upstream|-------+
    | LSR1                |       |
    +---------------------+       |
                                  |
                                  |
                                  |
                                  |
                                  |
    +---------------------+       |
    |                     |       |
    | Upstream_LSP_Control|       | (merged into)
    | _Block for Upstream |-------+
    |  LSR2               |       |
    +---------------------+       |
                                  |
                                  |
                                  |
                                  |   +------------------------------+
                                  |   |                              |
                                  +-->| Downstream_LSP_Control_Block |
                                  |   |   for Downstream LSR4        |
                                  |   |                              |
                                  |   +------------------------------+
                                  |
    +---------------------+       |
    | Next_Hop_Trigger_   |       |
    | LSP_Control_Block   |       |
    |                     |-------+
    |                     |
    +---------------------+
```

3.2.3   State Machines for Downstream-on-demand Mode

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

3.2.3.1   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 message from the downstream LSR.

-- ESTABLISHED

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

-- RELEASE_AWAITED

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

3.2.3.2 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 message 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 message by the Downstream_LSP_Control_Block's state machine.

-- LDP Upstream Nak

The LSR receives an LDP-NAK 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 message 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 happens when it is switched over to a better LSP through the new next hop.

3.2.3.3 State Transitions of the Upstream_LSP_Control_Block's State Machine for Downstream-on-demand Mode

```
               +-------------------+
               |                   |
     +-------->|  IDLE             |<------------+
     |         |                   |             |
     |         +-------------------+             |
     |(LDP Upstream Nak)|                        |
     |(Internal         |(LDP Request)           |
     | Downstream Nak)  |                        |
     |(Upstream Lost)   |                        |
     |                  v                        |
     |         +-------------------+             |
     |         |                   |             |
     +---------|  RESPONSE_AWAITED |             |
               |                   |             |(Upstream Lost)
               +-------------------+             |
                        |                        |
                        |(Internal Downstream    |
                        |  mapping)              |(LDP Release)
                        |                        |
                        v                        |
               +-------------------+             |
               |                   |             |(LDP Upstream Nak)
               |  ESTABLISHED      |------->-----+
               |                   |             |
               +-------------------+             |
                        |                        |
                        |                        |
                        |(Internal Downstream Withdraw)
                        |(Internal Downstream Nak)
                        v
               +-------------------+             |
               |                   |             |
               |RELEASE_AWAITED    |------->-----+
               |                   |
               +-------------------+
```

3.2.3.4 Upstream_LSP_Control_Block's State Machine for Downstream-on-demand Mode

3.2.3.4.1 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 [Ref2],

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 which 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 message 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[Ref2], choose an upstream label, and send an LDP-MAPPING message 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 message to the sender with the appropriate error code 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 Nak

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.
```

3.2.3.4.2 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 message upstream with the upstream label.

Then go to the state 'ESTABLISHED'.

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

State: RESPONSE_AWAITED

Event: LDP Release

New State: RESPONSE_AWAITED

Actions

Ignore the event. It is an 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 Nak

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 message upstream.

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

3.2.3.4.3 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 message 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 Nak

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: ESTABLISHED

Event: Internal Downstream Nak

New State: Depends on the actions

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 lable to the new downstream lable.

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

3.2.3.4.4 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 implementatin error.

State: RELEASE_AWAITED

Event: LDP Release

New State: IDLE

Actions:

Free the upstream lable 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 Nak

New State: IDLE

Actions:

Free the upstream lable 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 msg.

State: RELEASE_AWAITED

Event: Upstream Lost

New State: IDLE

Actions:

Free the upstream lable 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 msg.

State: RELEASE_AWAITED

Event: Internal Re-Cross-Connect

New State: RELEASE_AWAITED

Actions:

Ignore the event. It is an internal implementation error.

3.2.3.5 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 message from the downstream LSR.

-- ESTABLISHED

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

3.2.3.6 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 message from a downstream LSR.

-- LDP Withdraw

The LSR receives an LDP-WITHDRAW message 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.

3.2.3.7 State Transitions of the Downstream_LSP_Control_Block'S State Machine for Downstream-on-demand mode

```
               +-------------------+
               |                   |
               |  IDLE             |<--------------+
               |                   |               |(last Internal
               +-------------------+               | DeleteUpstream)
                        |                          |(LDP Withdraw)
                        |(1st Internal AddUpstream)|
                        |                          |(LDP Downstream
                        v                          | Nak)
               +-------------------+               |(Downstream
               |                   |               |   Lost)
               |  RESPONSE_AWAITED |---------->----^
               |                   |               |
               +-------------------+               |
                        |                          |
                        |(LDP Mapping)             |
                        |                          |
                        v                          |
               +-------------------+               |
               |                   |               |
               |  ESTABLISHED      |-------->------^
               |                   |
               +-------------------+
```

3.2.3.8 Downstream_LSP_Control_Block'S State Machine for Downstream-on-demand Mode.

3.2.3.8.1 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.

3.2.3.8.2 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-NAK downstream, 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

Ignore the event. It is a protocol error from the downstream LDP peer.

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.

3.2.3.8.3 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, and then delete the control block.

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.

State: ESTABLISHED

Event: LDP Downstream Nak

New State: ESTABLISHED

Actions

It is a protocol error from the downstream LDP peer.

3.2.3.9 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 message from the downstream LSR.

3.2.3.10 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 message 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 message 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.

3.2.3.11 State Transitions of the Next_Hop_Trigger_Control_Block's State Machine for Downstream-on-demand Mode

```
                   |
                   |
                   |
                   | Internal New NH
                   |
                   v
          +---------------------+
          |                     | Internal destroy
          | NEW_NH_RETRY        |--------------
          |                     |             |
          +---------------------+             |
                   |                          |
                   |                          |
                   | (Internal retry timeout) |
                   |                          |
                   v                          |
          +---------------------+             |
          |                     | Internal    |
          | NEW_NH_RESPONSE     | Destroy     |
          | _AWAITED            |-------------|
          |                     |             |
          +---------------------+             |
                   |                          |
                   | (Internal Downstream     |
                   |   Mapping                |
                   | (Internal Downstream     |
                   |    Nak)                  |
                   v                          v
                    the control block is deleted
```

3.2.3.12 State Machine

3.2.3.12.1 State -- "IDLE"

State: IDLE

Event: Internal New NH

New States: NEW_NH_RETRY

Actions:

Start the LSP retry timer and go to the 'NEW_NH_RETRY' state.

```
 State:          IDLE

 Event:          Internal retry timeout

 New States:     IDLE

 Actions:

   Ignore. It is an internal implementation error.

 State:          IDLE

 Event:          Internal Downstream Mapping

 New States:     IDLE

 Actions:

   Ignore. It is an internal implementation error.

 State:          IDLE

 Event:          Internal Downstream Nak

 New States:     IDLE

 Actions:

   Ignore. It is an internal implementation error.


 State:          IDLE

 Event:          Internal destroy

 New States:     IDLE

 Actions:

   Ignore. It is an internal implementation error.
```

3.2.3.12.2 State -- "NEW_NH_RETRY"

```
 State:          NEW_NH_RETRY

 Event:          Internal New NH

 New States:     NEW_NH_RETRY
```

Actions:

Restart the LSP retry timer.

State: NEW_NH_RETRY

Event: Internal retry timeout

New States: NEW_NH_RESPONSE_AWAITED

Actions:

If the new next hop is the same one as the old next hop, delete the control block.

Otherwise, find an downstream LSP control block which go through the same next hop for the same FEC, if there is no one, create one, and pass "Internal AddUpstream'event to its state machine.

State: NEW_NH_RETRY

Event: Internal Downstream Mapping

New States: NEW_NH_RETRY

Actions:

Ignore. It is an internal implementation error.

State: NEW_NH_RETRY

Event: Internal Downstream NAK

New States: NEW_NH_RETRY

Actions:

Ignore. It is an internal implementation error.

State: NEW_NH_RETRY

Event: Internal destroy

New States: IDLE

Actions:

Stop the timer and delete the control block.

3.2.3.12.3 State -- "NEW_NH_RESPONSE_AWAITED"

State: NEW_NH_RESPONSE_AWAITED

Event: Internal New NH

New States: 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 States: NEW_NH_RESPONSE_AWAITED

Actions:

Ignore. It is an internal implementation error.

State: NEW_NH_RESPONSE_AWAITED

Event: Internal Downstream Mapping

New States: 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.

Then delete the control block.

State: NEW_NH_RESPONSE_AWAITED

Event: Internal Downstream NAK

New States: IDLE

Actions:

Delete the control block.

State: NEW_NH_RESPONSE_AWAITED

Event: Internal destroy

New States: IDLE

Actions:

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

3.2.4 LDP Related Message Processing

- If an LSR receives an LDP-REQUEST message:

a) If this is a duplicate request, discard the msg. A duplicate request means that there is a LSP_Control_Block which has same FEC, same Upstream Label Request ID and same 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 message:

Locate a Downstream_LSP_Control_Block which 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 'Lable Request Message ID' field in the LDP-MAPPING MSG) 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 message:

Locate an Upstream_LSP_Control_Block which 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 message:

  Find a Downstream_LSP_Control_Block which 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 the downstream LDP peer is that LSR.

- If a next hop of an FEC is changed:

  For all the Upstream_LSP_Control_Blocks which are infected by this change, pass the event 'Internal New NH' to their state machines.

- If an LSR receives an LDP-NAK 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 Upstream Nak' 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.

3.2.4 Downstream Unsolicited

The following sections describe the state machines for the ATM-LSR which 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 independant 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 allowed 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 loop condition.

3.2.4.0 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 correspondng 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 withdrawed.

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

Downsteam LDP Session Identifier can be used to locate a unique Downstream_LSP_Control_Block.

3.2.4.1 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 message 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 msg in respond to the LDP-WITHDRAW sent by this LSR.

-- RESOUCES_AWAITED

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

3.2.4.2 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 message 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 message 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 either the FEC is removed from the forwarding table.

-- Upstream Lost

This event means that the upstream LDP peer is lost.

3.2.4.3 State Transitions of Upstream_LSP_Control_Block's State Machine for Downstream Mode

```
                            |
                            |(created when
                            |a label is to be distributed
                            | to the LDP peer)
                            v
                  +-------------------+
                  |                   |
                  |  IDLE             |<--------------+
                  |                   |               |
                  +-------------------+               |
                           |                          |(LDP Release)
                           |                          |
                           |                          |
                           |                          |
                           |(Internal Downstream      |
       +-------------------|  Mapping)                |
       |                   |                          |
       |(no label resource)v                          |
       |          +-------------------+               |
       |          |                   |               |
       |    +-----|  ESTABLISHED      |---------------+
       |    |     |                   |               ^
       |    |     +-------------------+               |
       |    |(delete FEC)   ^                         |
       |    |(Internal      |(Resource Available)     | (LDP Release)
       |    |  Withdraw)    |                         | (Internal
       |    |               |                         |  Downstraem
       |    |               |                         |    Withdraw)
       |    |     +-------------------+               |
       +--------->|                   |               |
            |     |RESOURCES_AWAITED  |---------------+
            |     |                   |               |
            |     +-------------------+               |
            |                                         |
            | (Internal Downstream Withdraw)          |(LDP Release)
            |     +-------------------+               |
            |     |                   |               |
            +---->|  RELEASE_AWAITED  |---------------+
                  |                   |
                  +-------------------+
```

3.2.4.4 Upstream_LSP_Control_Block's State Machine for Downstream Mode

3.2.4.4.1 : State -- "IDLE"

State: IDLE

Event: Internal Downstream mapping

New State: Depends on the action.

Actions

Choose an upstream label, connect the upstream label with the downstream label. And go to state 'ESTABLISHED'

Propagate the LDP-MAPPING upstream.

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.

3.2.4.4.2 : State -- "ESTABLISHED"

State: ESTABLISHED

Event: Internal Downstream Mapping

New State: ESTABLISHED

Actions

Process the new attributes of the new mapping msg.

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.

3.2.4.4.2 : State -- "RELEASE_AWAITED"

State: RELEASE_AWAITED

Event: Internal Downstream Mapping

New State: RELEASE_AWAITED

Actions

Ignore the msg.

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.

3.2.4.4.2 : State -- "RESOURCE_AWAITED"

State: RESOURCE_AWAITED

Event: Internal Downstream Mapping

New State: RESOURCE_AWAITED

Actions

Ignore the msg.

State: RESOURCE_AWAITED

Event: LDP Release

New State: RESOURCE_AWAITED

Actions

Ignore the msg. 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.

3.2.4.5 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 msg from the downstream LSR.

3.2.4.6 Events of the Downstream_LSP_Control_Block's State Machine for Downstream Mode

-- LDP Mapping

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

-- LDP Withdraw

The LSR receives an LDP-WITHDRAW msg 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.

3.2.4.7 State Transitions of Downstream_LSP_Control_Block's State Machine for Downstream Mode

```
                           |
                           |(FEC is being added into the forwarding
   table)
                           v
                  +-------------------+
                  |                   |
                  |  IDLE             |<--------------+
                  |                   |               |
                  +-------------------+               |
                           |                          |
                           |                          |(LDP Withdraw)
                           |                          |(Internal New NH)
                           |                          |(Downstream Lost)
                           |  (LDP Mapping)           |
                           |                          |
                           v                          |
                  +-------------------+               |
                  |                   |               |
                  |  ESTABLISHED      |---------------+
                  |                   |
                  +-------------------+
                           |
                           |(FEC is deleted from the forwarding table)
                           v
```

3.2.4.8 Downstream_LSP_Control_Block's State Machine for Downstream Mode

3.2.4.8.1 : State -- "IDLE"

State: IDLE

Event: LDP mapping

New State: ESTABLISHED

Actions

For all the LDP peers except the downstream LSR which 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.

3.2.4.8.1 : 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.

3.2.4.5 LDP Related Message Processing for downstream mode.

- If an LSR receives an LDP-REQUEST message:

  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 message:

  Locate a Downstream_LSP_Control_Block which 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 message:

  Locate an Upstream_LSP_Control_Block which 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 message:

  Find a Downstream_LSP_Control_Block which 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 which 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 msg. An LDP-NAK should never appear in the downstream-mode LSR

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

  Ignore the msg. It is an protocol error from the downstream LSR.

4. The Possible Future Works

The following is a list of items that might need to be worked on in the future:

1) Support senario such that: in a downstream-on-demand mode, a Label request contains multiple FEC items, whereas the corresponding Label Withdraw only withdraw one FEC item.

2) The impact of loop detection, loop recovery and loop prevention

3)Once the LSP is operational, whether the LSR can receive an LDP-NAK message is also debatable.

4)In this draft, for the VC-merge case, the LDP-REQUEST message only goes up to the merge point. Whether the LDP-REQUEST need to go all the way to the egress and back in order for the path to be setup (brute force way) is debatable.

5) Define all the necessary information for the control block. For example, the ER-HOP list information used to re-build a Label Request in case of "local repair".

5. Security Considerations

Security considerations will be addressed in a future revision of this document.

6. Acknowledgements

The authors would like to acknowledge the helpful comments and suggestions of the following people:

7. Authors's Address

Liwen Wu

Alcatel
44983 Knoll Square
Ashburn, VA. 20147
U.S.A
Phone: 703-724-2619
Email:liwen.wu@adn.alcatel.com

Pierrick Cheval

Alcatel
44983 Knoll Square
Ashburn, VA. 20147
U.S.A
Phone: 703-724-2080
Email: pierrick.cheval@adn.alcatel.com

Christophe Boscher

Alcatel CIT
Le Mail
44700 Nantes-Orvault
France
Phone: (33) 251781828
Email: christophe.boscher@or.cit.alcatel.fr

Eric Gray

Lucent Technologies, Inc.
1600 Osgood Street
North Andover, MA 01845
Email: ewgray@lucent.com

8. References

1."Use of Label Switching With ATM", Bruce Davie, Jeremy Lawrence,

Keith McCloghrie, Yakov Rekhter, Eric Rosen, George Swallow, Paul Doolan, work in progress, Internet Draft, <draft-ietf-mpls-atm-01.txt>

2."Multiprotocol Label Switching Architecture", Eric C Rosen, Arun Viswanathan, Ross Callon, work in progress, Internet Draft, <draft-ietf-mpls-arch-02.txt>

3."Definition of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)", Joan Cucchiara, Hans Sjostrand, James V. Luciani, work in progress, Internet Draft, <draft-ietf-mpls-ldp-mib-00.txt>

4."Requirements for Traffic Engineering Over MPLS", Daniel O.Awduche, Joe Malcolm, Johnson Agogbua, Mike O'Dell, Jim McMaus, work in progress,Internet Draft, <draft-ietf-mpls-traffic-eng-00.txt>

5. "LDP Specification", Loa Andersson, Paul Doolan,Nancy Feldman, Andre Fredette, Bob Thomas, work in progress, Internet Draft, <draft-ietf-mpls-ldp-03.txt>

6. "MPLS Loop Prevention Mechanism", Yoshihiro Ohba, Yasuhiro Katsube, Eric Rosen, Paul Doolan, work in progress, Internet Draft, <draft-ohba-mpls-loop-prevention-02.txt>