Network Working Group
Internet Draft
Expiration Date: October 1999

Bruce Davie
Jeremy Lawrence
Keith McCloghrie
Yakov Rekhter
Eric Rosen
George Swallow
Cisco Systems, Inc.

Paul Doolan
Ennovate Networks, Inc.

April 1999

MPLS using LDP and ATM VC Switching

draft-ietf-mpls-atm-02.txt

Status of this Memo

Abstract

The MPLS Architecture [1] discusses a way in which ATM switches may be used as Label Switching Routers. The ATM switches run network layer routing algorithms (such as OSPF, IS-IS, etc.), and their data forwarding is based on the results of these routing algorithms. No

ATM-specific routing or addressing is needed. ATM switches used in this way are known as ATM-LSRs.

This document extends and clarifies the relevant portions of [1] and [2] by specifying in more detail the procedures which to be used when distributing labels to or from ATM-LSRs, when those labels represent Forwarding Equivalence Classes (FECs, see [1]) for which the routes are determined on a hop-by-hop basis by network layer routing algorithms.

This document also specifies the MPLS encapsulation to be used when sending labeled packets to or from ATM-LSRs, and in that respect is a companion document to [3].

Contents

1. Introduction

The MPLS Architecture [1] discusses the way in which ATM switches may be used as Label Switching Routers. The ATM switches run network layer routing algorithms (such as OSPF, IS-IS, etc.), and their data forwarding is based on the results of these routing algorithms. No ATM-specific routing or addressing is needed. ATM switches used in this way are known as ATM-LSRs.

This document extends and clarifies the relevant portions of [1] and [2] by specifying in more detail the procedures which are to be used for distributing labels to or from ATM-LSRs, when those labels represent Forwarding Equivalence Classes (FECs, see [1]) for which the routes are determined on a hop-by-hop basis by network layer routing algorithms. The label distribution technique described here is referred to in [1] as "downstream-on-demand". This label distribution technique MUST be used by ATM-LSRs that are not capable of "VC merge" (defined in section 3), and is OPTIONAL for ATM-LSRs that are capable of VC merge.

This document does NOT specify the label distribution techniques to be used in the following cases:

- the routes are explicitly chosen before label distribution begins, instead of being chosen on a hop-by-hop basis as label distribution proceeds,

- the routes are intended to diverge in any way from the routes chosen by the conventional hop-by-hop routing at any time,

- the labels represent FECs that consist of multicast packets,

- the LSRs use "VP merge".

Further statements made in this document about ATM-LSR label distribution do not necessarily apply in these cases.

This document also specifies the MPLS encapsulation to be used when sending labeled packets to or from ATM-LSRs, and in that respect is a companion document to [3]. The specified encapsulation is to be used for multicast or explicitly routed labeled packets as well.

This document uses terminology from [1].

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

3. Definitions

A Label Switching Router (LSR) is a device which implements the label switching control and forwarding components described in [1].

A label switching controlled ATM (LC-ATM) interface is an ATM interface controlled by the label switching control component. When a packet traversing such an interface is received, it is treated as a labeled packet. The packet's top label is inferred either from the contents of the VCI field or the combined contents of the VPI and VCI fields. Any two LDP peers which are connected via an LC-ATM interface will use LDP negotiations to determine which of these cases is applicable to that interface.

An ATM-LSR is a LSR with a number of LC-ATM interfaces which forwards cells between these interfaces using labels carried in the VCI or VPI/VCI field.

A frame-based LSR is a LSR which forwards complete frames between its interfaces. Note that such a LSR may have zero, one or more LC-ATM interfaces.

Sometimes a single box may behave as an ATM-LSR with respect to certain pairs of interfaces, but may behave as a frame-based LSR with respect to other pairs. For example, an ATM switch with an ethernet interface may function as an ATM-LSR when forwarding cells between its LC-ATM interfaces, but may function as a frame-based LSR when forwarding frames from its ethernet to one of its LC-ATM interfaces. In such cases, one can consider the two functions (ATM-LSR and frame-based LSR) as being coresident in a single box.

It is intended that an LC-ATM interface be used to connect two ATM-LSRs, or to connect an ATM-LSR to a frame-based LSR. The use of an LC-ATM interface to connect two frame-based LSRs is not considered in this document.

An ATM-LSR domain is a set of ATM-LSRs which are mutually interconnected by LC-ATM interfaces.

The Edge Set of an ATM-LSR domain is the set of frame-based LSRs which are connected to members of the domain by LC-ATM interfaces. A frame-based LSR which is a member of an Edge Set of an ATM-LSR domain may be called an Edge LSR.

VC-merge is the process by which a switch receives cells on several incoming VCIs and transmits them on a single outgoing VCI without causing the cells of different AAL5 PDUs to become interleaved.

4. Special Characteristics of ATM Switches

While the MPLS architecture permits considerable flexibility in LSR implementation, an ATM-LSR is constrained by the capabilities of the (possibly pre-existing) hardware and the restrictions on such matters as cell format imposed by ATM standards. Because of these constraints, some special procedures are required for ATM-LSRs.

Some of the key features of ATM switches that affect their behavior as LSRs are:

- the label swapping function is performed on fields (the VCI and/or VPI) in the cell header; this dictates the size and placement of the label(s) in a packet.

- multipoint-to-point and multipoint-to-multipoint VCs are generally not supported. This means that most switches cannot support 'VC-merge' as defined above.

- there is generally no capability to perform a 'TTL-decrement' function as is performed on IP headers in routers.

This document describes ways of applying label switching to ATM switches which work within these constraints.

5. Label Switching Control Component for ATM

To support label switching an ATM switch MUST implement the control component of label switching. This consists primarily of label allocation, distribution, and maintenance procedures. Label binding information is communicated by several mechanisms, notably the Label Distribution Protocol (LDP) [2]. This document imposes certain requirements on the LDP.

This document considers only the case where the label switching control component uses information learned directly from network layer routing protocols. It is presupposed that the switch participates as a peer in these protocols (e.g., OSPF, IS-IS).

In some cases, LSRs make use of other protocols (e.g. RSVP, PIM, BGP) to distribute label bindings. In these cases, an ATM-LSR would need to participate in these protocols. However, these are not explicitly considered in this document.

Support of label switching on an ATM switch does NOT require the switch to support the ATM control component defined by the ITU and ATM Forum (e.g., UNI, PNNI). An ATM-LSR may OPTIONALLY respond to OAM cells.

6. Hybrid Switches (Ships in the Night)

The existence of the label switching control component on an ATM switch does not preclude the ability to support the ATM control component defined by the ITU and ATM Forum on the same switch and the same interfaces. The two control components, label switching and the ITU/ATM Forum defined, would operate independently.

Definition of how such a device operates is beyond the scope of this document. However, only a small amount of information needs to be consistent between the two control components, such as the portions of the VPI/VCI space which are available to each component.

7. Use of VPI/VCIs

Label switching is accomplished by associating labels with Forwarding Equivalence Classes, and using the label value to forward packets, including determining the value of any replacement label. See [1] for further details. In an ATM-LSR, the label is carried in the VPI/VCI field, or, when two ATM-LSRs are connected via an ATM "Virtual Path", in the VCI field.

Labeled packets MUST be transmitted using the null encapsulation, as defined in Section 5.1 of RFC 1483 [5].

In addition, if two LDP peers are connected via an LC-ATM interface, a non-MPLS connection, capable of carrying unlabelled IP packets, MUST be available. This non-MPLS connection is used to carry LDP packets between the two peers, and MAY also be used (but is not required to be used) for other unlabeled packets (such as OSPF packets, etc.). The LLC/SNAP encapsulation of RFC 1483 [5] MUST be used on the non-MPLS connection.

It SHOULD be possible to configure an LC-ATM interface with additional VPI/VCIs that are used to carry control information or non-labelled packets. In that case, the VCI values MUST be in the

0-32 range. These may use either the null encapsulation, as defined in Section 5.1 of RFC 1483 [5], or the LLC/SNAP encapsulation, as defined in Section 4.1 of RFC 1483 [5].

7.1. Direct Connections

We say that two LSRs are "directly connected" over an LC-ATM interface if all cells transmitted out that interface by one LSR will reach the other, and there are no ATM switches between the two LSRs.

When two LSRs are directly connected via an LC-ATM interface, they jointly control the allocation of VPIs/VCIs on the interface connecting them. They may agree to use the VPI/VCI field to encode a single label.

The default VPI/VCI value for the non-MPLS connection is VPI 0, VCI 32. Other values can be configured, as long as both parties are aware of the configured value.

A VPI/VCI value whose VCI part is in the range 0-32 inclusive MUST NOT be used as the encoding of a label.

With the exception of these reserved values, the VPI/VCI values used in the two directions of the link MAY be treated as independent spaces.

The allowable ranges of VCIs are communicated through LDP.

7.2. Connections via an ATM VP

Sometimes it can be useful to treat two LSRs as adjacent (in some LSP) across an LC-ATM interface, even though the connection between them is made through an ATM "cloud" via an ATM Virtual Path. In this case, the VPI field is not available to MPLS, and the label MUST be encoded entirely within the VCI field.

In this case, the default VCI value of the non-MPLS connection between the LSRs is 32. The VPI is set to whatever is required to make use of the Virtual Path.

A VPI/VCI value whose VCI part is in the range 0-32 inclusive MUST NOT be used as the encoding of a label.

With the exception of these reserved values, the VPI/VCI values used in the two directions of the link MAY be treated as independent spaces.

The allowable ranges of VPI/VCIs are communicated through LDP. If more than one VPI is used for label switching, the allowable range of VCIs may be different for each VPI, and each range is communicated through LDP.

7.3. Connections via an ATM SVC

Sometimes it may be useful to treat two LSRs as adjacent (in some LSP) across an LC-ATM interface, even though the connection between them is made through an ATM "cloud" via a set of ATM Switched Virtual Circuits.

The current document does not specify the procedure for handling this case. Such procedures can be found in [4]. The procedures described in [4] allow a VCID to be assigned to each such VC, and specify how LDP can be used used to bind a VCID to a FEC. The top label of a received packet would then be inferred (via a one-to-one mapping) from the virtual circuit on which the packet arrived. There would not be a default VPI or VCI value for the non-MPLS connection.

8. Label Distribution and Maintenance Procedures

This document discusses the use of "downstream-on-demand" label distribution (see [1]) by ATM-LSRs. These label distribution procedures MUST be used by ATM-LSRs that do not support VC-merge, and MAY also be used by ATM-LSRs that do support VC-merge. The procedures differ somewhat in the two cases, however. We therefore describe the two scenarios in turn. We begin by describing the behavior of members of the Edge Set of an ATM-LSR domain; these "Edge LSRs" are not themselves ATM-LSRs, and their behavior is the same whether the domain contains VC-merge capable LSRs or not.

8.1. Edge LSR Behavior

Consider a member of the Edge Set of an ATM-LSR domain. Assume that, as a result of its routing calculations, it selects an ATM-LSR as the next hop of a certain FEC, and that the next hop is reachable via a LC-ATM interface. The Edge LSR uses LDP to request a label binding for that FEC from the next hop. The hop count field in the request is set to 1 (but see the next paragraph). Once the Edge LSR receives the label binding information, it may use MPLS forwarding procedures to transmit packets in the specified FEC, using the specified label as an outgoing label. (Or using the VPI/VCI that corresponds to the specified VCID as the outgoing label, if the VCID technique of [4] is being used.)

Note: if the Edge LSR's previous hop is using downstream-on-demand label distribution to request a label from the Edge LSR for a particular FEC, and if the Edge LSR is not merging the LSP from that previous hop with any other LSP, and if the request from the previous hop has a hop count of h, then the hop count in the request issued by the Edge LSR should not be set to 1, but rather to h+1.

The binding received by the edge LSR may contain a hop count, which represents the number of hops a packet will take to cross the ATM-LSR domain when using this label. If there is a hop count associated with the binding, the ATM-LSR SHOULD adjust a data packet's TTL by this amount before transmitting the packet. In any event, it MUST adjust a data packet's TTL by at least one before transmitting it. The procedures for doing so (in the case of IP packets) are specified in section 10. The procedures for encapsulating the packets are specified in section 9.

When a member of the Edge Set of the ATM-LSR domain receives a label binding request from an ATM-LSR, it allocates a label, and returns (via LDP) a binding containing the allocated label back to the peer that originated the request. It sets the hop count in the binding to 1.

When a routing calculation causes an Edge LSR to change the next hop for a particular FEC, and the former next hop was in the ATM-LSR domain, the Edge LSR SHOULD notify the former next hop (via LDP) that the label binding associated with the FEC is no longer needed.

8.2. Conventional ATM Switches (non-VC-merge)

When an ATM-LSR receives (via LDP) a label binding request for a certain FEC from a peer connected to the ATM-LSR over a LC-ATM interface, the ATM-LSR takes the following actions:

- it allocates a label,

- it requests (via LDP) a label binding from the next hop for that FEC;

- it returns (via LDP) a binding containing the allocated incoming label back to the peer that originated the request.

For purposes of this procedure, we define a maximum hop count value MAXHOP. MAXHOP has a default value of 255, but may be configured to a different value.

The hop count field in the request that the ATM-LSR sends (to the next hop LSR) MUST be set to one more than the hop count field in the request that it received from the upstream LSR. If the resulting hop count exceeds MAXHOP, the request MUST NOT be sent to the next hop, and the ATM-LSR MUST notify the upstream neighbor that its binding request cannot be satisfied.

Otherwise, once the ATM-LSR receives the binding from the next hop, it begins using that label.

The ATM-LSR MAY choose to wait for the request to be satisfied from downstream before returning the binding upstream. This is a form of "ordered control" (as defined in [1] and [2]), in particular "ingress-initiated ordered control". In this case, as long as the ATM-LSR receives from downstream a hop count which is greater than 0 and less than MAXHOP, it MUST increment the hop count it receives from downstream and MUST include the result in the binding it returns upstream. However, if the hop count exceeds MAXHOP, a label binding MUST NOT be passed upstream. Rather, the upstream LDP peer MUST be informed that the requested label binding cannot be satisfied. If the hop count received from downstream is 0, the hop count passed upstream should also be 0; this indicates that the actual hop count is unknown.

Alternatively, the ATM-LSR MAY return the binding upstream without waiting for a binding from downstream ("independent" control, as defined in [1] and [2]). In this case, it specifies a hop count of 0 in the binding, indicating that the true hop count is unknown. The correct value for hop count will be returned later, as described below.

Note that an ATM-LSR, or a member of the edge set of an ATM-LSR domain, may receive multiple binding requests for the same FEC from the same ATM-LSR. It MUST generate a new binding for each request (assuming adequate resources to do so), and retain any existing binding(s). For each request received, an ATM-LSR MUST also generate a new binding request toward the next hop for the FEC.

When a routing calculation causes an ATM-LSR to change the next hop for a FEC, the ATM-LSR MUST notify the former next hop (via LDP) that the label binding associated with the FEC is no longer needed.

When a LSR receives a notification that a particular label binding is no longer needed, the LSR MAY deallocate the label associated with the binding, and destroy the binding. In the case where an ATM-LSR receives such notification and destroys the binding, it MUST notify the next hop for the FEC that the label binding is no longer needed. If a LSR does not destroy the binding, it may re-use the binding only if it receives a request for the same FEC with the same hop count as

the request that originally caused the binding to be created.

When a route changes, the label bindings are re-established from the point where the route diverges from the previous route. LSRs upstream of that point are (with one exception, noted below) oblivious to the change.

Whenever a LSR changes its next hop for a particular FEC, if the new next hop is reachable via an LC-ATM interface, then for each label that it has bound to that FEC, and distributed upstream, it MUST request a new label binding from the new next hop.

When an ATM-LSR receives a label binding for a particular FEC from a downstream neighbor, it may already have provided a corresponding label binding for this FEC to an upstream neighbor, either because it is using independent control, or because the new binding from downstream is the result of a routing change. In this case, unless the hop count is 0, it MUST extract the hop count from the new binding and increment it by one. If the new hop count is different from that which was previously conveyed to the upstream neighbor (including the case where the upstream neighbor was given the value 'unknown') the ATM-LSR MUST notify the upstream neighbor of the change. Each ATM-LSR in turn MUST increment the hop count and pass it upstream until it reaches the ingress Edge LSR. If at any point the value of the hop count equals MAXHOP, the ATM-LSR SHOULD withdraw the binding from the upstream neighbor. A hop count of 0 MUST be passed upstream unchanged.

Whenever an ATM-LSR originates a label binding request to its next hop LSR as a result of receiving a label binding request from another (upstream) LSR, and the request to the next hop LSR is not satisfied, the ATM-LSR SHOULD destroy the binding created in response to the received request, and notify the requester (via LDP).

If an ATM-LSR receives a binding request containing a hop count that exceeds MAXHOP, it MUST not establish a binding, and it MUST return an error to the requester.

When a LSR determines that it has lost its LDP session with another LSR, the following actions are taken. Any binding information learned via this connection MUST be discarded. For any label bindings that were created as a result of receiving label binding requests from the peer, the LSR MAY destroy these bindings (and deallocate labels associated with these binding).

An ATM-LSR SHOULD use 'split-horizon' when it satisfies binding requests from its neighbors. That is, if it receives a request for a binding to a particular FEC and the LSR making that request is,

according to this ATM-LSR, the next hop for that FEC, it should not return a binding for that route.

It is expected that non-merging ATM-LSRs would generally use "conservative label retention mode" [1].

8.3. VC-merge-capable ATM Switches

Relatively minor changes are needed to accommodate ATM-LSRs which support VC-merge. The primary difference is that a VC-merge-capable ATM-LSR needs only one outgoing label per FEC, even if multiple requests for label bindings to that FEC are received from upstream neighbors.

When a VC-merge-capable ATM-LSR receives a binding request from an upstream LSR for a certain FEC, and it does not already have an outgoing label binding for that FEC (or an outstanding request for such a label binding), it MUST issue a bind request to its next hop just as it would do if it were not merge-capable. If, however, it already has an outgoing label binding for that FEC, it does not need to issue a downstream binding request. Instead, it may simply allocate an incoming label, and return that label in a binding to the upstream requester. When packets with that label as top label are received from the requester, the top label value will be replaced with the existing outgoing label value that corresponds to the same FEC.

If the ATM-LSR does not have an outgoing label binding for the FEC, but does have an outstanding request for one, it need not issue another request.

When sending a label binding upstream, the hop count associated with the corresponding binding from downstream MUST be incremented by 1, and the result transmitted upstream as the hop count associated with the new binding. However, there are two exceptions: a hop count of 0 MUST be passed upstream unchanged, and if the hop count is already at MAXHOP, the ATM-LSR MUST NOT pass a binding upstream, but instead MUST send an error upstream.

Note that, just like conventional ATM-LSRs and members of the edge set of the ATM-LSR domain, a VC-merge-capable ATM-LSR MUST issue a new binding every time it receives a request from upstream, since there may be switches upstream which do not support VC-merge. However, it only needs to issue a corresponding binding request downstream if it does not already have a label binding for the appropriate route.

When a change in the routing table of a VC-merge-capable ATM-LSR causes it to select a new next hop for one of its FECs, it MAY optionally release the binding for that route from the former next hop. If it doesn't already have a corresponding binding for the new next hop, it must request one. (The choice between conservative and liberal label retention mode [1] is an implementation option.)

If a new binding is obtained, which contains a hop count that differs from that which was received in the old binding, then the ATM-LSR must take the new hop count, increment it by one, and notify any upstream neighbors who have label bindings for this FEC of the new value. Just as with conventional ATM-LSRs, this enables the new hop count to propagate back towards the ingress of the ATM-LSR domain. If at any point the hop count exceeds MAXHOP, then the label bindings for this route must be withdrawn from all upstream neighbors to whom a binding was previously provided. This ensures that any loops caused by routing transients will be detected and broken.

9. Encapsulation

The procedures described in this section affect only the Edge LSRs of the ATM-LSR domain. The ATM-LSRs themselves do not modify the encapsulation in any way.

Labeled packets MUST be transmitted using the null encapsulation of Section 5.1 of RFC 1483 [5].

Except in certain circumstances specified below, when a labeled packet is transmitted on an LC-ATM interface, where the VPI/VCI (or VCID) is interpreted as the top label in the label stack, the packet MUST also contain a "shim header" [3].

If the packet has a label stack with n entries, it MUST carry a shim with n entries. The actual value of the top label is encoded in the VPI/VCI field. The label value of the top entry in the shim (which is just a "placeholder" entry) MUST be set to 0 upon transmission, and MUST be ignored upon reception. The packet's outgoing TTL, and its CoS, are carried in the TTL and CoS fields respectively of the top stack entry in the shim.

Note that if a packet has a label stack with only one entry, this requires it to have a single-entry shim (4 bytes), even though the actual label value is encoded into the VPI/VCI field. This is done to ensure that the packet always has a shim. Otherwise, there would be no way to determine whether it had one or not, i.e., no way to determine whether there are additional label stack entries.

The only ways to eliminate this extra overhead are:

- through apriori knowledge that packets have only a single label (e.g., perhaps the network only supports one level of label)

- by using two VCs per FEC, one for those packets which have only a single label, and one for those packets which have more than one label

The second technique would require that there be some way of signalling via LDP that the VC is carrying only packets with a single label, and is not carrying a shim. When supporting VC merge, one would also have to take care not to merge a VC on which the shim is not used into a VC on which it is used, or vice versa.

While either of these techniques is permitted, it is doubtful that they have any practical utility. Note that if the shim header is not present, the outgoing TTL is carried in the TTL field of the network layer header.

10. TTL Manipulation

The procedures described in this section affect only the Edge LSRs of the ATM-LSR domain. The ATM-LSRs themselves do not modify the TTL in any way.

The details of the TTL adjustment procedure are as follows. If a packet was received by the Edge LSR as an unlabeled packet, the "incoming TTL" comes from the IP header. (Procedures for other network layer protocols are for further study.) If a packet was received by the Edge LSR as a labeled packet, using the encapsulation specified in [3], the "incoming TTL" comes from the entry at the top of the label stack.

If a hop count has been associated with the label binding that is used when the packet is forwarded, the "outgoing TTL" is set to the larger of (a) 0 or (b) the difference between the incoming TTL and the hop count. If a hop count has not been associated with the label binding that is used when the packet is forwarded, the "outgoing TTL" is set to the larger of (a) 0 or (b) one less than the incoming TTL.

If this causes the outgoing TTL to become zero, the packet MUST NOT be transmitted as a labeled packet using the specified label. The packet can be treated in one of two ways:

- it may be treated as having expired; this may cause an ICMP message to be transmitted;

- the packet may be forwarded, as an unlabeled packet, with a TTL that is 1 less than the incoming TTL; such forwarding would need to be done over a non-MPLS connection.

Of course, if the incoming TTL is 1, only the first of these two options is applicable.

If the packet is forwarded as a labeled packet, the outgoing TTL is carried as specified in section 9.

When an Edge LSR receives a labeled packet over an LC-ATM interface, it obtains the incoming TTL from the top label stack entry of the generic encapsulation, or, if that encapsulation is not present, from the IP header.

If the packet's next hop is an ATM-LSR, the outgoing TTL is formed using the procedures described in this section. Otherwise the outgoing TTL is formed using the procedures described in [3].

The procedures in this section are intended to apply only to unicast packets.

11. Optional Loop Detection: Distributing Path Vectors

Every ATM-LSR MUST implement, as a configurable option, the following procedure for detecting forwarding loops. We refer to this as the LDPV (Loop Detection via Path Vectors) procedure. This procedure does not prevent the formation of forwarding loops, but does ensure that any such loops are detected. If this option is not enabled, loops are detected by the hop count mechanism previously described. If this option is enabled, loops will be detected more quickly, but at a higher cost in overhead.

11.1. When to Send Path Vectors Downstream

Suppose an LSR R sends a request for a label binding, for a particular LSP, to its next hop. Then if R does not support VC-merging, and R is configured to use the LDPV procedure:

- If R is sending the request because it is an ingress node for that LSP, or because it has acquired a new next hop, then R MUST include a path vector object with the request, and the path vector object MUST contain only R's own address.

- If R is sending the request as a result of having received a request from an upstream LSR, then:

  * if the received request has a path vector object, R MUST add its own address to the received path vector object, and MUST pass the resulting path vector object to its next hop along with the label binding request;

  * if the received request does not have a path vector object, R MUST include a path vector object with the request it sends, and the path vector object MUST contain only R's own address.

An LSR which supports VC-merge SHOULD NOT include a path vector object in the requests that it sends to its next hop.

If an LSR receives a label binding request whose path vector object contains the address of the node itself, the LSR concludes that the label binding requests have traveled in a loop. The LSR MUST act as it would in the case where the hop count exceeds MAXHOP (see section 8.2).

This procedure detects the case where the request messages loop though a sequence of non-merging ATM-LSRs.

11.2. When to Send Path Vectors Upstream

As specified in section 8, there are circumstances in which an LSR R must inform its upstream neighbors, via a label binding response message, of a change in hop count for a particular LSP. If the following conditions all hold:

- R is configured for the LDPV procedure,

- R supports VC-merge,

- R is not the egress for that LSP, and

- R is not informing its neighbors of a decrease in the hop count,

then R MUST include a path vector object in the response message.

If the change in hop count is a result of R's having been informed by its next hop, S, of a change in hop count, and the message from S to R included a path vector object, then if the above conditions hold, R MUST add itself to this object and pass the result upstream. Otherwise, if the above conditions hold, R MUST create a new object with only its own address.

If R is configured for the LDPV procedure, and R supports VC merge, then it MAY include a path vector object in any label binding response message that it sends upstream. In particular, at any time that R receives a label binding response from its next hop, if that response contains a path vector, R MAY (if configured for the LDPV procedure) send a response to its upstream neighbors, containing the path vector object formed by adding its own address to the received path vector.

If R does not support VC merge, it SHOULD NOT send a path vector object upstream.

If an LSR receives a message from its next hop, with a path vector object containing its own address, then LSR MUST act as it would if it received a message with a hop count equal to MAXHOP.

LSRs which are configured for the LDPV procedure SHOULD NOT store a path vector once the corresponding path vector object has been transmitted.

Note that if the ATM-LSR domain consists entirely of non-merging ATM-LSRs, path vectors need not ever be sent upstream, since any loops will be detected by means of the path vectors traveling downstream.

By not sending path vectors unless the hop count increases, one avoids sending them in many situations when there is no loop. The cost is that in some situations in which there is a loop, the time to detect the loop may be lengthened.

12. Security Considerations

The use of the procedures and encapsulations specified in this document does not have any security impact other than that which may generally be present in the use of any MPLS procedures or encapsulations.

13. Intellectual Property Considerations

Cisco Systems may seek patent or other intellectual property protection for some or all of the technologies disclosed in this document. If any standards arising from this document are or become protected by one or more patents assigned to Cisco Systems, Cisco intends to disclose those patents and license them under openly specified and non-discriminatory terms, for no fee.

14. References

[1] Rosen E., Viswanathan, A., Callon R., "Multi-Protocol Label Switching Architecture", Work in Progress, April 1999.

[2] Andersson L., Doolan P., Feldman N., Fredette A., Thomas R., "Label Distribution Protocol", Work in Progress, April 1999.

[3] Rosen, E., Rekhter, Y., Tappan, D., Farinacci, D., Fedorkow, G., Li, T., Conta, A., "MPLS Label Stack Encoding", Work in Progress, April 1999.

[4] Nagami, K., Demizu N., Esaki H., Doolan P., "VCID Notification over ATM link", Work in Progress, April 1999.

[5] Heinanen, J., "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 1483, July 1993

15. Acknowledgments

Significant contributions to this work have been made by Anthony Alles, Fred Baker, Dino Farinacci, Guy Fedorkow, Arthur Lin, Morgan Littlewood and Dan Tappan. We thank Alex Conta for his comments.

16. Authors' Addresses

Bruce Davie
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: bsd@cisco.com

Paul Doolan
Ennovate Networks Inc.
330 Codman Hill Rd
Boxborough, MA 01719

E-mail: pdoolan@ennovatenetworks.com

Jeremy Lawrence
Cisco Systems, Inc.
99 Walker St.
North Sydney, NSW, Australia

E-mail: jlawrenc@cisco.com

Keith McCloghrie
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134

E-mail: kzm@cisco.com

Yakov Rekhter
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134

E-mail: yakov@cisco.com

Eric Rosen
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: erosen@cisco.com

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: swallow@cisco.com