Network Working Group
Internet Draft
Expiration Date: September 1998

Bruce Davie
Cisco Systems, Inc.

Yakov Rekhter
Cisco Systems, Inc.

Eric Rosen
Cisco Systems, Inc.

Arun Viswanathan
Lucent Technologies

Vijay Srinivasan
IBM Corp.

Steven Blake
IBM Corp.

March 1998

Use of Label Switching With RSVP

draft-ietf-mpls-rsvp-00.txt

Status of this Memo

Abstract

Multiprotocol Label Switching (MPLS) allows labels to be bound to various granularities of forwarding information, including application flows. In this document we present a specification for allocating and binding labels to RSVP flows, and to distributing the appropriate binding information using RSVP messages.

Contents

1. Introduction

The purpose of this document is to propose a standard method for hosts and routers that support both label switching [1] and RSVP [4] to associate labels with RSVP flows. The goal is to enable label switching routers (LSRs) to be able to identify the appropriate reservation state for a packet based on its label value. To this end, the document describes a set of procedures for allocating and binding labels, and a way to distribute the bindings using RSVP messages. It also defines two new RSVP Objects: RSVP_LABEL, to carry a label in an RSVP message, and HOP_COUNT, to enable TTL processing for RSVP flows which pass through ATM-LSRs.

While there are several alternatives to mapping RSVP flows to labels, this document specifies a model in which, on a given link, each

sender to a single RSVP session is associated with one label. (There is one exception, described below.) The rationale for this choice is discussed below.

2. Specification

As mentioned above, in a label switching environment it is desirable to associate each RSVP flow with a label. An RSVP flow [4] is a simplex flow from a sending application to a set of receiving applications identified by an IP address (and perhaps a transport protocol port), and a session may contain several flows. An RSVP reservation may be flow specific (fixed filter) or shared across flows (shared explicit and wildcard).

For the purposes of this specification, we assume that all routers on a given link are capable of sending and receiving labeled packets over that link. Thus, for example, one would not enable labeling on some routers on a LAN but not on others.

2.1. Label Allocation

The association between RSVP flows and labels involves the allocation of a label to a flow, which could in principle be initiated by either the upstream or downstream node. However, there are some strong arguments in favor of downstream allocation that arise from the need to coordinate label allocation for RSVP with other label allocation schemes, e.g., the allocation of labels for best effort traffic. If some labels were allocated by the receiver of data and some by the sender, race conditions could arise in which both sender and receiver allocated the same label for different purposes. This can most easily be avoided by leaving allocation to one party. Since the receiver of data allocates labels for best effort traffic, we believe this is also the best choice for traffic with resource reservations.

Even when label allocation is performed by the downstream nodes, it may be necessary to communicate label bindings from upstream to downstream. For example, if two routers on a shared media LAN are receiving data for the same session, that data should be sent with the same label to both receivers. The best way to accomplish this while retaining downstream allocation is for one of the receivers to allocate a label, communicate the label-to-session binding to the sender(s) on the LAN using RESV messages, and then have the sender(s) communicate the binding to the receivers in PATH messages. The PATH message should be sent as soon as the reservation with the label binding is installed (rather than waiting for the normal PATH refresh

interval), so that receivers of labeled data will be notified at once of the fact that labeled data is about to arrive.

It is possible in this case that two or more receivers might try to allocate the label for a single session. In this case, the upstream node(s) will receive RESV messages for the session advertising different labels. Any node receiving conflicting labels in this way must break the tie in some way. The only requirement on the tie-breaking is that it be consistent (i.e., once a choice has been made, it should not be reversed at some later time). Simply choosing the label in the first RESV message received is an adequate approach. Having broken the tie, the selected label will appear in subsequent PATH messages, and the recipients of these PATH messages must accept the result. If for some reason (e.g. hardware limitation) the assigned LABEL value was not acceptable to a recipient, it would need to generate a PATH error message. Methods outside the scope of this document (e.g. LDP) may be used to determine acceptable label ranges.

One unfortunate consequence of this method of label distribution on shared media is that even nodes which do not wish to make reservations for some session may receive PATH messages corresponding to that session indicating that some other node has made a reservation and that data for that session will now arrive with a new (non-best effort) label. It seems necessary for the node that made no reservation to accept the new label. At present, it is not clear how to make the sender of data aware that all nodes are ready to receive data with the new label. This is an area for further study.

2.2. Choice of label for data forwarding

As soon as a LSR has installed a reservation on one of its interfaces and has received a label binding for that reservation (either for the whole session or for some flows in the session) it should use the chosen label for all appropriate flows. Any flows or sessions for which label bindings have not been received must be sent using the appropriate best effort label. This best effort label will have been advertised by some other mechanism, such as PIM (for multicast) or LDP (for unicast). When a router is forwarding packets out multiple interfaces, it may be the case that reservations are installed on some interfaces and not others. In this case, the best effort allocated label should be used on those interfaces for which no reservation is present.

2.3. Label space partitioning on shared media

To ensure that a single label is not allocated twice for different purposes by different routers, it is necessary to partition the label space among routers on a shared media, just as described in [2]. In fact, such partitioning is only needed for multicast sessions, and thus the exact mechanism described in [2] can be used. A router which has thus obtained a portion of the label space can decide unilaterally which labels from this space to use for multicast of best effort traffic and which to use for RSVP sessions. Similarly, in the unicast case, a router decides locally which labels it will allocate for best effort traffic and which for RSVP sessions.

2.4. Label withdrawal

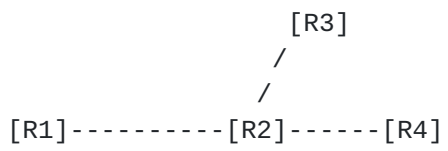
When the original allocator of a label no longer wishes to have a reservation for the corresponding flow or session, or if the allocator crashes, it will stop refreshing the reservation with RESV messages. It may also issue a ResvTear message. Upstream nodes which had been redistributing that label using PATH messages must stop doing so when the reservation times out or is torn down. They will thus resume sending PATH messages with no labels, and any recipient of those PATH messages will be at liberty to allocate a new label and place it in a RESV message. However, it may be that the nodes that did not crash will keep refreshing the reservation using the old label. It is important that a router that is newly rebooted does not try to assign that label; this should be possible, since it will receive the PATH messages once it reboots.

A label may be withdrawn without removing the reservation by sending a RESV message which contains no label. This would similarly be propagated via PATH messages to other receivers, who would have the option of allocating a new label.

2.5. Reservation Styles

So far we have glossed over the exact mapping between labels and sessions or labels and flows. It seems clear that for fixed filter (FF) style reservations, a label per sender is needed, since each sender has its own allocated resources. Because of the merging rules for SE reservations, we believe a label per sender is needed in this case also. The following example illustrates the point.

Consider the following arrangement of LSRs:



where data is flowing from left to right and there are at least 2 senders to the session, S1 and S2. Suppose one of the receivers downstream of R3 makes a shared explicit (SE) reservation for data coming from two senders S1 and S2, while a receiver downstream of R4 makes a reservation for data coming from one sender S1. These would be merged at R2 as a single SE reservation before forwarding to R1. So, if we used a single label per session on the link from R1 to R2, there would be no way for R2 to distinguish packets from sender S1 (which are covered by a reservation on both outgoing links) from those from S2 (which are covered by a reservation only on the link to R3. Thus, we need a label per sender for SE reservations.

Finally, for the WF case, we might imagine that a single label could be used for the session, since all senders to the session are covered by the reservation. For a shared tree, this is true, but for source specific trees we need different labels for different senders since the fact that two trees share a link at some point does not mean they will not diverge at some later point on the way to a receiver. If we were to use a single label for all senders to a WF session on source-specific trees, it would be impossible to determine the appropriate forwarding action at a point where the trees diverge.

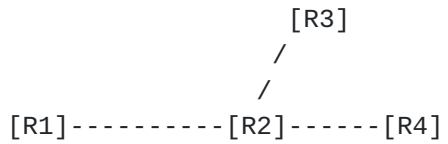
Thus, the general rule is one label per sender to a session, with the exception being that one label can be used for all senders to a session with a WF reservation who are using a shared tree. Note that some senders to a session may use a shared tree while others may be on the source specific tree. The router allocating labels and sending them in RESV messages needs to know which senders are on which type of tree; it can find this out using the interface to routing described in [5].

2.6. ATM-LSR considerations

In most respects, an ATM-LSR behaves like any other LSR that is connected to its neighbors with point to point links. One minor difference is that that, on ATM-LSRs which do not support VC-merge, a label per sender is needed for all reservation styles. In theory, this could be reduced to a label per ingress router per session for WF reservations on a shared tree, but procedures to allocate labels appropriately have not yet been defined.

Note that, in WF and SE styles, resources are allocated to reservations, not to specific senders. An ATM-LSR therefore needs to be able to allocate resources to a collection of labels to support these filter styles correctly.

More significantly, ATM-LSRs which cannot perform VC merge create a problem when some but not all of their downstream neighors make reservations. For example, in the following arrangement of four LSRs:



Assume R2 receives a reservation from R3 but not from R4. R2 will bind a label to the reservation and advertise it to R1. Packets from R1 which match that reservation will arrive at R2 carrying the label R2 assigned. Best effort packets from R1 will arrive at R2 carrying the best effort label. Both sets of packets should be sent to R4 with the best effort label. However, if R2 is not capable of VC merge, best effort packets and reserved packets will become interleaved on the way to R4.

The problem could be averted by assigning an extra label for use on the link between R2 and R4 for each label that R2 creates on the link to R1. Since this label is for best effort traffic, it could be allocated using the Label Distribution Protocol in the downstream on demand mode. This enables R2 to force the label allocation without introducing the complexity of mixing upstream and downstream allocation. Note that this may cause allocation of numerous labels for best effort traffic on the R2-R4 link as a label per sender per session will be allocated on the R1-R2 link.

When IP packets are label switched by ATM-LSRs, the TTL value in the IP header cannot be decremented, and no TTL is available in the ATM header. To enable TTL to be decremented by the number of ATM-LSR hops, the proposed HOP_COUNT Object is used to count the number of consecutive LSR hops. The object is inserted into the Path message by a non-ATM LSR whose next hop for the session is an ATM-LSR, and initialized with a hop count of 1. Subsequent ATM-LSRs increment the hop-count only if there is a label-switched path for that sender flow through that LSR. All LSRs maintain the hop count in the Path State. The `egress' LSR, i.e., the first frame-based LSR to receive the HOP_COUNT object, uses the count to decrement the TTL on packets for that sender flow, and removes the HOP_COUNT object from the PATH message.

2.7. RSVP Object Definitions

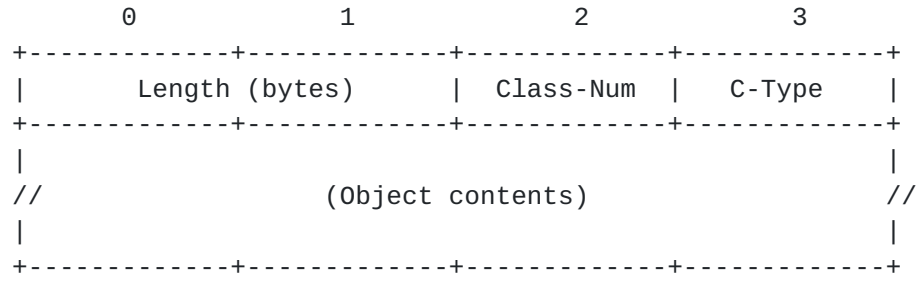
As discussed above, labels may be carried in both PATH and RESV messages. When a label is to be associated with a single sender, it must immediately follow the FILTER_SPEC for that sender in the RESV or the SENDER_TEMPLATE in the PATH message.

The wildcard filter case is the most complicated. If all senders are using the shared tree, then only one label is needed, and can be placed immediately following the FLOW_SPEC in the RESV. In this case, all PATH messages must contain the same label, again following the SENDER_TEMPLATE.

If some senders to a WF session are not using the shared tree, then seperate labels need to be allocated for those senders and the bindings distributed. It is necessary to enumerate the senders who are using source specific trees and associate a label with each one; this can be done by including a FILTER_SPEC object followed by an RSVP_LABEL object for each such sender. All senders using the shared tree will use the label that follows the single FLOW_SPEC in the message.

The RSVP_LABEL object class conforms to the standard RSVP object format:

RSVP_LABEL class = 16, C_Type = 1



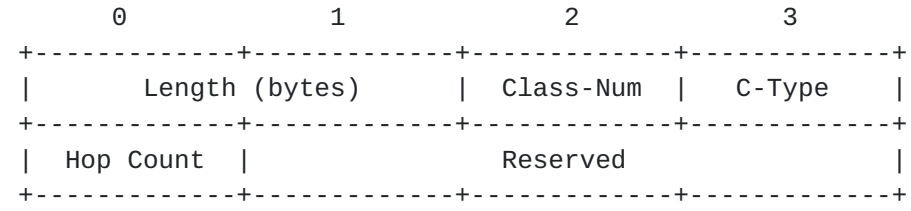
The contents of a Label object is a stack of labels, where each label is encoded right aligned in 4 octets. The top of the stack is in the rightmost 4 octets of the object contents. The label stack can be carried in packets using an encoding such as decribed in [7]. When an ATM link is used, the low order 28 bits of the top label in the stack are carried in the VPI/VCI field of the ATM cells.

When no labels have been allocated to a session, the PATH messages for that session must contain no RSVP_LABEL object. If labels have been allocated for some senders but not others in a session, then RSVP_LABEL objects should be included only after the SENDER_TEMPLATEs of those senders for who labels are assigned. This enables receivers of PATH messages to determine if a label has been assigned or if a label assignment is required.

A node receiving a PATH message containing a label must use that label in subsequent RESV messages for the same sender or session. If for some reason it is unable to do this, it must generate a PATH error message.

The HOP_COUNT object class conforms to the standard RSVP object format:

HOP_COUNT object: Class = 17, C-Type = 1



Hop Count
Counts the length (in ISR hops) of the switched path.

2.8. Non RSVP routers

RSVP is designed to cope gracefully with non-RSVP routers anywhere in the path between senders and receivers. However, non-RSVP routers will not be able to receive label bindings conveyed in PATH or RESV messages. This means that if a LSR has a downstream neighbor who is not RSVP capable, it must not use labels advertised by RSVP messages when forwarding data to that neighbor. This includes the case where some routers on a LAN are RSVP capable and some are not; if an RSVP capable router on the LAN advertises a label binding in a RESV message, the recipient of that message cannot send labeled data using that label if there are any non-RSVP routers on the LAN that have joined the multicast group for that session.

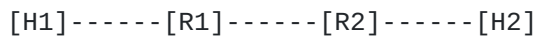
Also, when RESV messages are received by a non-RSVP router, it unwittingly passes them on towards the previous hop RSVP router. This

could result in a label being advertised to a router which was not directly connected to the advertiser of the label. Such a label would be useless for data forwarding. Thus, RESV messages containing label binding information must not be sent toward a previous hop when it would pass through non-RSVP routers on the way. [4] describes how routers may determine the presence of non-RSVP routers in a path.

3. Examples

3.1. Unicast

The figure below shows a simple example network in which two hosts H1 and H2 communicate through a sequence of label switched routers (R1, R2).

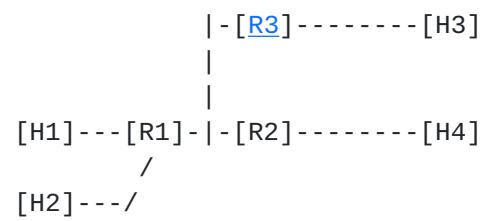


Following RSVP procedures, H1 sends a RSVP PATH messages to H2. The PATH messages traverse through R1 and R2.

When H2 determines that it would like to setup a reservation for this particular session, it allocates a label, and sends a RESV message containing this label to R2. H2 stores this label as an identifier for the session, and can use it to demultiplex arriving data packets to the appropriate application or device. When R2 receives the RESV message, along with normal RSVP processing, it stores the value of the label as part of the reservation state for this session and interface. This will be the outgoing label for data packets sent by R2 to H2. R2 then allocates a label, and sends a RESV message containing this label to R1. When R1 receives this message, it behaves similarly to R2, storing the label received from R2 and allocating a new label which it sends in a RESV message to H1. Upon receiving the message, H1 proceeds to start sending the session's data with the label received from R1. R1 forwards the data to R2 using the label it received from R2, and R2 sends data to H2 using the label received from H2.

3.2. Multicast

The figure below shows a network in which two senders H1 and H2 send traffic to two receivers H3 and H4. Routers R1, R2 and R3 are on a shared media LAN.



Assume that H1 and H2 are using the shared multicast tree to send data. H1 and H2 send PATH messages toward H3 and H4. Assume H3 makes a WF reservation by sending a RESV to R3. H3 allocates a label and includes it in the RESV message. R3 stores this label as the label to use for data traffic to H3 for this session. R3 allocates a label and includes it in the RESV that it sends to R1. R1 stores this label and includes it in subsequent PATH messages to R2 and R3. R1 allocates a label for the session and sends it in RESV messages to H1 and H2 (different labels may be used on different interfaces, as a matter of implementation choice).

Assume H4 then makes a WF reservation. H4 allocates a label and sends it in a RESV message to R2. R2 stores this label and will use it for data packets to H4. R2 now sends a RESV to R1 using the label contained in the PATH message from R1.

4. Security Considerations

Security considerations are not addressed in this version of the document. We presume that the security procedures defined for RSVP will handle any security issues that arise with coupling label switching with RSVP.

5. References

[1] Rosen, E. et al. A Proposed Architecture for MPLS, Internet Draft, draft-ietf-mpls-arch-00.txt, Aug. 1997.

[2] Farinacci, D. Partitioning Tag Space among Multicast Routers on a Common Subnet, Internet Draft, draft-farinacci-multicast-tag-part-00.txt, Dec. 1996.

[3] Farinacci, D. Multicast Tag Binding and Distribution using PIM, Internet Draft, draft-farinacci-multicast-tagsw-00.txt, Dec. 1996.

[4] Braden, R. et al. Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification, RFC 2205, Sep. 1997.

[5] Zappala, D. RSRR: A Routing Interface For RSVP, Internet Draft, draft-ietf-rsvp-routing-01.txt, Nov. 1996.

[6] Davie, B. et al. Use of Label Switching With ATM, Internet Draft, draft-davie-mpls-atm-00.txt, Nov. 1997.

[7] Rosen, E. et al. Label Switching: Label Stack Encodings, Internet Draft, draft-rosen-tag-stack-03.txt, July, 1997.

6. Author's Addresses

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

E-mail: bsd@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

Vijay Srinivasan
IBM Corporation
P. O. Box 12195
Research Triangle Park, NC 27709

E-mail: vijay@raleigh.ibm.com

Arun Viswanathan
Lucent Technologies
101 Crawford Corner Rd., #4D-537
Holmdel, NJ 07733

E-mail: arunv@dnrc.bell-labs.com

Steven Blake
IBM Corporation
PO Box 12195
Research Triangle Park, NC 27709

Email: slblake@raleigh.ibm.com