Multicast for RSVP Switching

An Extended Multicast Model with QoS for Label Swapping in an IP over ATM Environment

Olivier Fourmaux [a] Serge Fdida [a]

[a] *Laboratoire d'Informatique de Paris 6 (LIP6)*
Université Pierre et Marie Curie (Paris 6) / CNRS
4, place Jussieu 75252 Paris Cedex 05 FRANCE
Ph./Fax: +33 1 4427 7126/6286
E-mail: Olivier.Fourmaux@lip6.fr

The ubiquity of IP associated with the acknowledgment of ATM as a key switching technology has motivated an increasing interest towards the design of a more efficient way of operating IP over ATM networks. This approach is known under the name Label Swapping. A few studies have addressed the primary issue of providing simultaneously quality of service and multicast. We propose a solution where we mix an RSVP architecture with one Label Swapping technique called IP Switching. We discuss problems that arise when using cut-through associated with an RSVP multicast model and propose an application for an IPv6 environment over an ATM switching hardware.

Keywords: Multicast, QoS, IP Switching, Label Swapping, ATM

1. Introduction

The original design of the Internet was thought for the support of a single service, called Best Effort unicast. It achieves point-to-point data transfer using first-in first-out scheduling at each hop of the network. This solution was sufficient for basic data transfer but is limited to enable emerging real-time or group communication applications. Therefore, new services are required from the network, namely, efficient multipoint-to-multipoint communication and Quality-of-Service (QoS) support.

Two major extensions were added to IP. The former is the multicast extension [9] that introduces a multipoint-to-multipoint communication model with group abstraction. The latter is the introduction of new integrated services to support more than Best-Effort flows. The QoS signaling protocol is RSVP (Resource ReSerVation Protocol) [33,6]. Although the utilization of RSVP is controversial, we recognize that some end-to-end signaling is mandatory to provide per-flow QoS. We also understand that scalability issues appears when using RSVP but this problem is limited in local environments or virtual networks.

The IntServ (Integrated Services) [5,32] workgroup has proposed two classes of service : the Controlled Load [31] and the Guaranteed [29] services. However, many applications as well as the existing ones will be satisfied with the ability of using a better service than Best Effort. Therefore, solutions based on Differentiated Services are under discussion but are not considered in this paper.

The "RSVP Multicast" model provides a multipoint-to-multipoint communication model with QoS. It enables flows to request several types of reservations with aggregation of resources and filtering. Receivers can dynamically access an RSVP session by joining the corresponding multicast groups. They can choose which packet and QoS they want by doing filtering and specifying reservations. So far, providing efficient multicast communication with QoS support is not achieved today in the IP world.

On the other hand, the ATM technology was sold as a solution for providing a powerful set of QoS virtual circuits (VC) to satisfy all needs. Nevertheless, the ATM multicast scheme is quite simple: it only allows point-to-multipoint connections with explicit receivers when initiating the multicast tree. Point-to-multipoint VC QoS is identical for all receivers and defined only once. Again, multicast support with QoS is far to be supported in the ATM environment.

The ubiquity of IP associated with the acknowledgment of ATM as a key switching technology has motivated an increasing interest towards the design of a more efficient way of operating IP over ATM networks. Different solutions have been proposed by the ATM Forum and the IETF (Internet Engineering Task Force):

- ATM Forum has standardized LAN Emulation (Local Area Network Emulation) [17,18] and MPOA (MultiProtocol Over ATM) [22].
- IETF has proposed the "Classical IP" (CLIP) standard [14,19,20], using Logical IP subnets connected by routers and based on ATM technology that provides direct connectivity. Additional protocols provide multicast functionalities over ATM, like MARS [3] or EARTH [30], and shortcuts capabilities, like NHRP [21].

Unfortunately, the above mentioned solutions are poor in providing both QoS and multicast. Some works related to perform Integrated Services with multicast have been done, particularly for Integrated Services over CLIP with MARS [7,13] or Multicast Integrated Services with EARTH [28]. The major disadvantage of these approaches is that they all keep the intricate and inefficient ATM virtual circuit mode under IP datagrams.

Solutions were designed in order to simplify the integration of ATM and IP, speed-up forwarding while using ATM switching fabrics and keep the IP signaling to permit a graceful evolution of routing. They are presented under the generic name "Label Swapping". They all use a fast label matching principle to provide cut-through at the link layer. It enables to increase considerably the forwarding capability of intermediate network elements. We can classify Label Swapping solutions in two main categories. The first one is based on local data flows to decide when to cut-through and is called "flow driven" [25,10,2]. The second one is based on label distribution among the network associated to routing signaling and is called "topology driven" [26]. These two techniques manage the multicast differently as described in the sequel.

To provide real-time guaranty, global state must be introduced in the network. RSVP do it fine in a connectionless environment. It was proposed to use this protocol in the Label Swapping model. This third approach based on the association of RSVP sessions with labels is called "session driven" and provide QoS. There exist a few draft taking natively this approach [4].

RSVP can be used in any previous solution categories, *i.e.* topology or flow driven. It then produces two RSVP based solutions:

- The first solution comes with MPLS (MultiProtocol Label Switching) that was originally called "topology driven". Since this model was extended to all control protocols, we name it "control driven". In the MPLS architecture, RSVP is seen as one of the supported control protocol for carrying label information. Therefore MPLS proposes a way of using RSVP in its framework [8].
- The second solution comes from the "flow driven" approach. With this approach we produce a "session driven" solution, like MPLS does, but by following a different method. They are important differences with the previous solution, particularly in the multicast and associated protocols, as explained in the sequel.

The "flow driven" approach to RSVP is the one we choose to provide QoS with a full RSVP multicast model. Other solutions with a similar approach, like IS-IPSO[1] [1], only partially address the multipoint-to-multipoint communication model of RSVP as we propose (with aggregation and filtering).

The organization of this paper is as follows. We discuss and compare multicast models in Label Swapping environments. Then, we introduce QoS and present a global view of "RSVP Switching". We develop a generic solution for dealing with routing layer functionality in a switching environment, and we particularly emphasize on RSVP aggregation and filtering. Finally we present how to use our solution in an IPv6 environment over an ATM switching hardware and introduce our test-bed.

2. Label Swapping Multicast

The key issue is to provide a multicast solution also able to support QoS. We first discuss the way multicast is provided both in "flow driven" and "control

[1] IS-IPSO is a proposition for using IPSOFACTO with Integrated Services and RSVP

driven" approaches.

IPsilon has proposed a "flow driven" approach where multicast is naturally integrated through native IP multicast. Every flow starts by being routed, so it evolves in a robust and well understood environment. Thereafter, the labeling and cut-through are taking place smoothly. Unicast and multicast are treated in the same way and share the same label space.

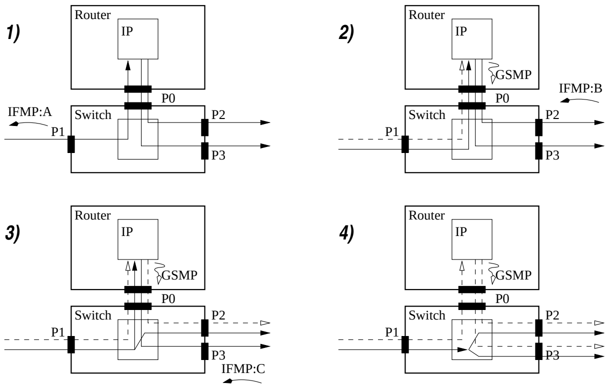


Figure 1. IPswitching multicast cut-through

Figure 1 shows the creation of a multicast cut-through with the IPsilon approach. All shortcuts are considered as being potentially multicast and can be seen as branches of the corresponding potential multicast tree. The cut-throughs are set by adding branches, equivalent to connections in the switching matrix, from the labeled input ports to the labeled output ports. If no connections are already set at an input port, a first multicast branch is set and is considered as a unicast connection, otherwise a multicast tree already exist from this input port, and it results in adding a branch to it. This example involves an IP over ATM hardware environment, as assumed with the IPsilon approach, because using the "Add Branch" request at the switching level is typically performed in ATM hardware. In the figure, the flows are initially routed, and then progressively switched, after being labeled. The signaling stay at the routing level and is described by doted lines. Label informations are exchanged with IFMP (IPsilon Flow Management Protocol) [24]. The communication between the switch and the router is managed by GSMP (General Switch Management Protocol) [23]. The "flow driven" approach handles partially routed trees but can not deal with aggregation.

The MPLS working group proposes, in its "control driven" approach, to handle the multicast separately from the unicast traffic. According to MPLS, the label table is filled by a specific label protocol or by usual protocols themselves using piggybacking. Label distribution is done with LDP (Label Distribution Protocol) [11] for unicast. The multicast is based on spanning trees with specific labels bound to each tree. The label distribution is done with the multicast routing protocol PIM (Protocol Independent Multicast) [15].

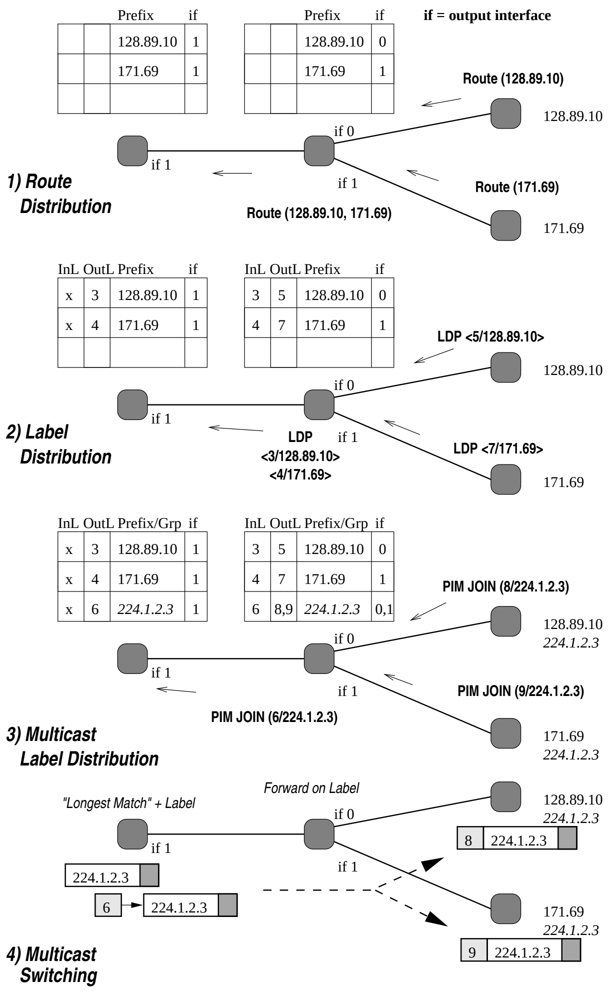


Figure 2. Creating a MPLS multicast

Multicast and unicast cut-through setups are presented in figure 2. The routing table augmented with label informations is depicted for the two left nodes. In part *1)*, the prefix information is distributed by the routing protocol, so we can see in routing tables prefixes and corresponding output interfaces. In part *2)* LDP adds label informations (Input Label and Output Label) for unicast traffic. In part *3)* PIM is used to create a multicast line in the table and add piggybacked label informations. The last part shows a multicast switching example, where a multicast packet is labeled with label “6” in the first node after a longest prefix match in the routing/label table. In the next node, this packet is switched according to the local label table: it is multicasted to the output interfaces “if 0” and “if 1” with the respective labels “8” and “9”.

With MPLS, the problems related to aggregation, for multipoint-to-multipoint model are not resolved.

MPLS is initially based on a “topology driven” approach that makes it adapted for backbone environments. Extended use of MPLS including local environments is less interesting because it will generate large label tables. The separation of control protocols implies, in order to run RSVP, the adaptation of all the related protocols: LDP for unicast, PIM for multicast and RSVP itself.

The use of the “flow driven” approach is basically dynamic and is well adapted to protocols related to group or session management. RSVP needs a solution that handles dynamic flows efficiently. The solution proposed by IPsilon is well suited but is restricted to local areas, like campus networks or VPN (Virtual/Private Network). No modifications to existing protocols are needed for unicast and multicast data traffic. RSVP will run directly with unmodified protocols, as described in the next section.

We think this approach is more efficient by using the same scheme for both unicast and multicast traffics. Keeping together data and control before doing cut-through will result in a more transparent use of RSVP. This utilization of multicast is also well suited to our approach where multicast cut-through are only used with RSVP session cut-through while other data are kept at the routing layer.

3. RSVP Switching

Our objective is to provide a RSVP Multicast model with QoS in a Label Swapping architecture. The model we propose is akin to the IPsilon one. And is called “RSVP Switching”. Local decisions are used but they are driven by RSVP signaling and not by local data traffic. This model is called “session driven” because RSVP builds sessions on which we will base switching decisions. We have decided to always switch guaranteed flows. Best Effort flows could be switched with classical IPsilon techniques or simply routed.

Our solution is different from the other solutions aiming at using RSVP with Label Swapping because it comes from the “flow driven” approach and takes advantage of its efficient multicast. We also resolve different problems due to the cut-through management issue as discussed below.

At the switching level, ATM provides point-to-multipoint connections with homogeneous QoS while at the routing layer we have the RSVP Multicast model that provides a dynamic, heterogenous multipoint-to-multipoint solution.

Table 1 provides a comparison of both models. We assume that the switching level is of NBMA (Non Broadcast Multiple Access) type, as an ATM switch.

The “State” line of table 1 describes the way the information about resource

	Routing layer	Switching layer
State	Dynamic (soft state)	Dynamic (soft state)
Multicast QoS	Heterogenous (filtering)	Homogeneous (switching hardware)
Multicast model	N to M (aggregation)	1 to M (switching hardware)

Table 1
Comparison of routing and switched layer

reservation is kept in a network element. We can see that both models rely on a dynamic state. At the routing layer, RSVP soft state is used to maintain reservations in the network. At the switching layer, cut-throughs are managed by an equivalent exchange message protocol with timers. In the IPsilon solution, this protocol is called IFMP [24]. At this level, there are no mapping problems and we can merge functionalities by holding dynamic states for both reservations and cut-throughs. We use RSVP for that double usage, with a new object to manage labeling.

The "Multicast QoS" line of table 1 indicates whether all leaves can choose independently their QoS. At the routing layer, with RSVP, each receiver can choose a different QoS. This proceeds from the receiver oriented model where each receiver specifies the desired QoS. At the switching level, however, there is only a basic duplication permitting the same QoS on each branch of the multicast tree. A first problem due to filtering is that if we operate at the switching layer, we can not provide the same functionality as found at the routing layer. For instance, the ability to provide a different QoS to different receivers. Therefore, we propose to do filtering before the switching area.

The "Multicast Model" line of table 1 indicates if a "N to M" (multipoint-to-multipoint) multicast model is possible. Usually, it is provided by the routing layer in the IP conventional model and RSVP manages the aggregation to deal with the QoS. Like with the filtering, the aggregation is a routing layer feature we want to use at the switching layer. Unfortunately, sending different flows to the same port at the switching layer could result in an unmanageable multiplexing if we use a cell technology, and the loss of all data. In order to avoid this problem, we propose to move the aggregation after the switching area.

RSVP switching is presented in the sequel. It is developed to support the RSVP functionalities in a switching environment as well as to minimize existing protocol modifications.

4. Cut-through

In this section, we describe the basic operation of the RSVP Switch in order to handle cut-throughs. At the initial state, data and signaling are routed. In figure 3, we present a routed path through three RSVP Switches for the RSVP session **S**. Before and after this area, we find IP networks with possibly RSVP, other RSVP Switches or directly attached hosts. We present three RSVP Switches composed of IP and RSVP elements at the routing layer, and a switching hardware at the link layer.

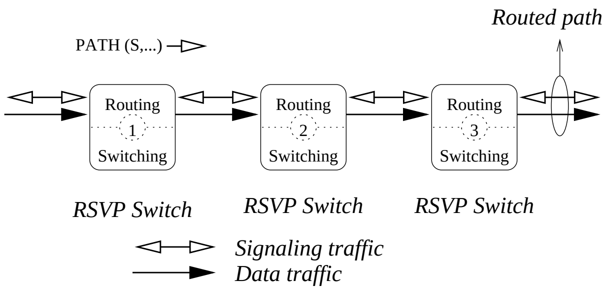
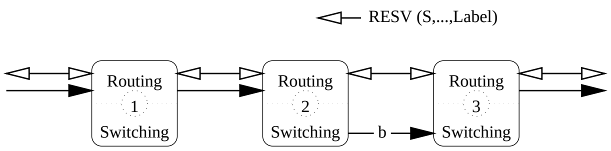


Figure 3. Initial routing path

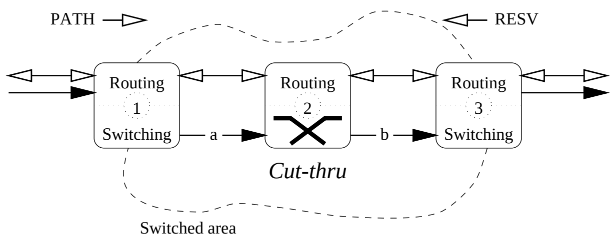
After the PATH message exchange phase, reservations are taking place. Data are separated from signaling and are directed to a dedicated cut-through at the switching level. Informations related to the cut-through are transmitted in RESV messages with reservation requests. A new specific object is added to the RSVP specification for this purpose. We call this object "Label" for label swapping information. After exchanging messages with label reservation we can enable a cut-through at the switching level. Different reservations use different labels corresponding to different cut-through.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S	"b"	(3)out			

Figure 4. Starting of labeling

In figure 4, a RESV message is propagating from the receiver back to the sender. The RESV messages hold the Label object to built the cut-through. In the first part of the picture, between RSVP Switch 1 and 2, data are still routed. In the second part, a reservation is done. A corresponding labeled flow is established and data, even if they are still routed, belong to a different virtual circuit from the routed traffic. Labeled flows are waiting for a cut-through. It is represented by an arrow at the switching level with a corresponding label "b". Labels are local to a link and may differ from hop to hop.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S	"b"	(3)out	**S**	"a" (1)	"b" (3)
S	"a"	(1)in			

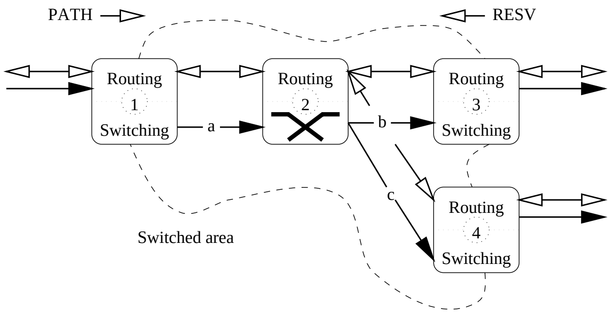
Figure 5. Enabling unicast cut-through

In figure 5, there exist a label "a" for session **S** between RSVP Switches 1 and 2. A cut-through can be initiated. In all figures, we assume that a link or port from node N is referred by the notation (N). In the switching table of the RSVP Switch 2, incoming data arrives from the input port (1) with label "a" and are switched to the corresponding output port (3) with label "b". At this point, we do not consider filtering or aggregating issues.

In figure 6, a simple multicast flow is established by adding a new entry in the switching table. This is achieved using RESV messages and results in the label "c" being allocated to this flow directed to (4).

The switching boundaries are defined by ingress and egress nodes. An "ingress" (respectively "egress") node is a RSVP Switch from which dataflows are entering the switching area (respectively leaving). In figure 6, node 1 is an ingress node, nodes 3 and 4 are egress nodes.

The use of cut-through has an impact on the data header. For instance, packet TTL are not modified while the packet is crossing the switch, therefore we need to compute it from signaling packets that are always routed. By keeping conventional header calculation in the routed path, we avoid network loops in a usual way.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S	"b"	(3)out	**S**	"a" (1)	"b" (3)
S	"c"	(4)out		"a" (1)	"c" (4)
S	"a"	(1)in			

Figure 6. Enabling multicast cut-through

Problems such as enabling cut-through are alike those that we face with all others IP over ATM solutions. They are related to the restricted model of the link layer that can not be extended without going to an upper layer. To provide RSVP switching, we have to solve the routing outside the switching area.

Similarly, filtering and aggregation can not be done at the link layer in the switching area. Therefore, they will also be provided outside the switching area. We present our solution in the next sections, addressing filtering and aggregation successively.

5. Filtering

In RSVP, filtering permits to select subsets of packets in a session. Subsets are defined by any field in any protocol headers. For example, filtering can be done by senders or by different sub-flows for a hierarchically-encoded flow. In the first RSVP specification, only sender IP addresses are checked for simplicity, but it can be very simply extended to manage sub-flows.

We present in the following the way we handle filtering for an RSVP sessions with sub-flows. We propose to use a pre-filtering before the cut-through area, achieved by reservation signaling. The signaling follows the routing path and PATH messages are not altered. The multicast tree is built by the PATH message as usually done. RESV messages are extended to carry label information on data traffic. In order to be consistent with the ability to provide receiver-oriented QoS, we set-up multiple multicast trees at the link level. These trees are set during the reservation phase and each of them will be used by a sub-flow. Every sub-flow exhibits an homogeneous QoS as required by the ATM link layer.

In order to avoid the problem of internal filtering, when the cut-through is initiated, the filtering takes place at the beginning of the switching area. The ingress node provides this functionality by selecting and forwarding on the corresponding multicast trees. If an egress node needs several multicast trees to serve multiple flows of a session, they will be bundled together in this node to create the output session flow.

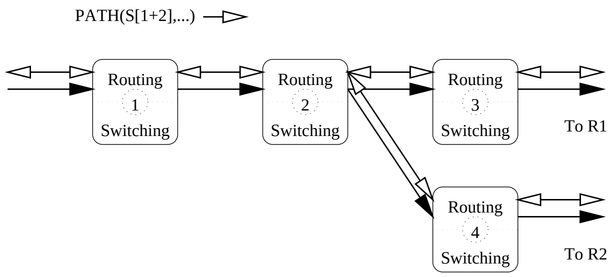
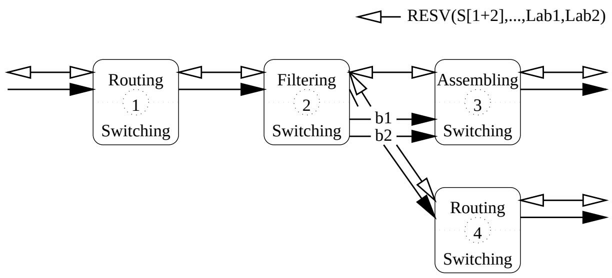


Figure 7. Initial routing path for filtering

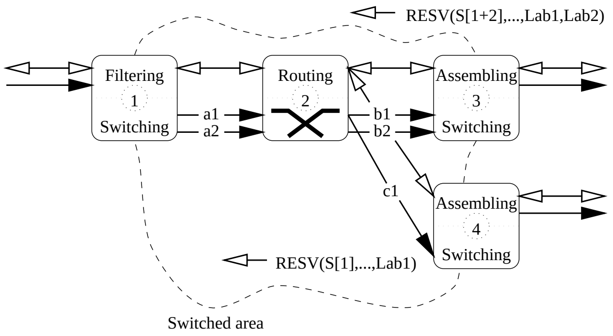
We present in figure 7, the scenario where a sender issue a data-flow in an RSVP session (S). This dataflow is composed of two sub-flows 1 and 2. Receivers can ask either for a complete flow by receiving both sub-flows or a subset by receiving only a single sub-flow. PATH messages carry the informations used by receivers to decide which reservations , *i.e.* which sub-flows they need to satisfy their QoS requirements.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S sub 1	“b1”	(3)out			
S sub 2	“c2”	(3)out			

Figure 8. Starting labeling for filtering

In figure 8, receiver R1 asks for the complete flow and reserves resources for sub-flow 1 and 2. RSVP Switches manage the cut-through by labeling data of each sub-flow with label “b1” and “b2”.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S sub 1	“b1”	(3)out	**S sub 1**	“a1” (1)	“b1” (3)
S sub 2	“b2”	(3)out		“a1” (1)	“c1” (4)
S sub 1	“c1”	(4)out	**S sub 2**	“a2” (1)	“b2” (3)
S sub 1	“a1”	(1)in			
S sub 2	“a2”	(1)in			

Figure 9. Enabling cut-through with filtering

In figure 9, receiver R2 asks for a lower quality by reserving resources for sub-flow 1 only. This sub-flow receives the label “c1”. The reservation done between nodes 1 and 2 is the “maximum” of the two previous reservations *i.e.* reservation for sub-flows 1 and 2. We assign label “a1” and “b1” to the sub-flows.

The flows of each sides of the RSVP Switch 2 being labeled, cut-throughs can be initialized. For the sub-flow 1 of session **S**, a multicast cut-through from (1) with label “a1” to (3) and (4) is built with respective labels “b1” and “c1”. For the sub-flow 2 of session **S**, a unicast cut-through is built from (1) with label “a2” to (3) with label “b2”.

The filtering functionality is performed in node 1, the ingress node of the switching area, instead of node 2 as it should be if done at the routing layer. This translation of functionality allows to use a simple multicast in the switching area while keeping the RSVP filtering capability.

In the egress node 3, both sub-flows are needed to construct the complete flow. In the egress node 4, only sub-flow 1 is needed. The exit of the switching area is responsible for computing the new IP header after it was frozen in the short-cut.

6. Aggregation

Aggregation is needed in the RSVP architecture. This feature allows to use the multipoint-to-multipoint model with QoS efficiently. It also reduces resource consumption for RSVP multicast alike in point-to-multipoint Best Effort IP multicast. Associated with the filtering functionality, aggregation allows to merge flows from several senders and to separate them later.

The aggregation issue is complex because of the lack of multipoint-to-point functionality at the link layer. The issue to address is how to do labeling and switching without interleaving at the link layer because we use the same label for two entering flows. To identify them, we must keep them separated in the switching area. We use the same principle as for filtering, and move the functionality out of the switching area. It is done after the cut-through to allow receivers to access any node of the short-cut. When getting off the switching area, the different flows are moved to the routing layer to be aggregated.

Separating flows does not mean that we have to make over reservations in the switching area by avoiding the data multiplexing. We only use some more signaling in the switching area in order to perform the short-cut. At the network resource level we associate a different signaling *i.e.* labels to only one reservation. For example, with the two main filtering styles of RSVP:

- Shared reservations like WF (Wildcard Filter) or SE (Shared Explicit) style reservations: we assign a label to each sender but only one reservation corresponding to the "maximum" resources that are needed. The number of senders with the Wildcard Filter style is limited by the RSVP SCOPE object.
- Distinct reservations like FF (Fixed Filter): we assign a label to each sender and a reservation corresponding to the "sum" of all resources needed.

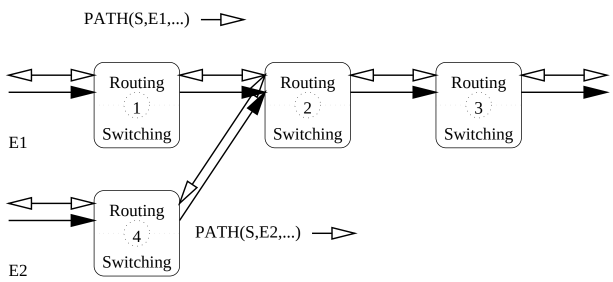
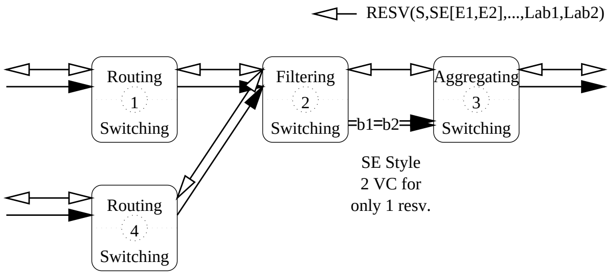


Figure 10. Initial routing path with aggregation

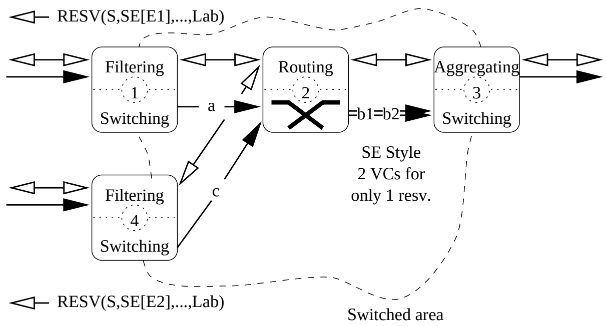
In figure 10, two senders, E1 and E2 are sending PATH messages in the RSVP session **S**. Their data traffics sent to receiver R are merged. PATH messages carry identification of their sources.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S(E1)	"b1"	(3)out			
S(E2)	"b2"	(3)out			

Figure 11. Starting labeling with aggregation

The reply to the PATH messages is a reservation request with two labels, "b1" and "b2", one for each flow of the two senders involved in session **S** (figure 11). Only one global reservation is done for both flows. At the link level, two labeled short-cuts are managed.



Information in node 2					
Label table			*Cut-through table*		
Session	Label	Port	Session	From	To
S(E)1	"b1"	(3)out	**S (E1)**	"a" (1)	"b1" (3)
S(E)2	"b2"	(3)out			
S(E1)	"a"	(1)in	**S (E2)**	"c" (4)	"b2" (3)
S(E2)	"c"	(4)in			

Figure 12. Enabling cut-through with aggregation

When the labeling of both sides of node 2 is completed (figure 12), the cut-through is established. Data traffic from each sender has a different label and go in a separate short-cut to avoid link level frame interleaving. Each flow has its own cut-through in node 2. Flows are aggregated at the exit of the switching area.

The aggregation functionality is translated from node 2 to the egress nodes 3 and 4. This mechanism allows the use of RSVP aggregation in a switched environment. Alike with filtering, the exit of the switching area is responsible for computing the new IP header.

Aggregation and filtering are done in each intermediate node in the switching area, in order to provide the dynamic service required by IP and RSVP. The switching area may be variable, so each node must be able to perform ingress and egress functionalities. An ingress node at the entry of the switching area manages sub-flows separation as well as an egress node deals with sub-flows reassembly and flow aggregation at the exit. The nodes also have to provide all the basic functionalities associated with Switching cut-through, IP and RSVP architectures.

The management of sub-flows is identical to managing SE style reservations in the RSVP Switching model.

7. IPv6 and RSVP Switching

We propose to implement RSVP switching in an IPv6 environment. IPv6 [16] is the new version of the Internet Protocol. It uses a fixed length header with daisy-chained extension headers. It makes more efficient use of header fields but It is also harder to use upper layer fields like with RSVP on IPv4 because of chained header and encryption possibilities. We give a short description to explain how IPv6 fields are used to identify RSVP sessions.

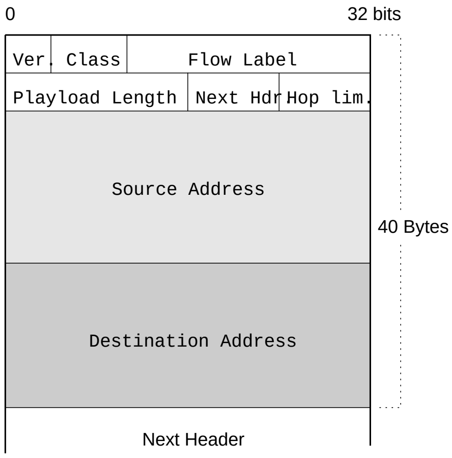


Figure 13. Ipv6 Header

With IPv6 we can use header fields without layer violation like with IPv4. Figure 13 shows IPv6 header fields that are used as follows :

- Identifying a RSVP session is done with `Destination address` and `Flow label`.
- Filtering identification is done with `Source address`.
- Sub-flow identification[2] could be done with a part of the `Flow label` field, that we call prio (reducing the size of the previous `flow label` field identifying the session).

Following is an example of the utilization of IPv6 fields for RSVP with *Filtering* (Label Table in node 2):

- "b1" port(3)out $\mapsto$ @dest(S), Flow(S), sub1
- "b2" port(3)out $\mapsto$ @dest(S), Flow(S), sub2

and *Aggregation* (Label Table in node 2):

- "b1" port(3)out $\mapsto$ @dest(S), Flow(S), @src(E1)
- "b2" port(3)out $\mapsto$ @dest(S), Flow(S), @src(E2)

This scheme shows which fields are associated to which label (per port). "@dest" is destination address, "Flow" is the flow label associated to the RSVP session, "sub1" and "sub2" are fields used to distinguish sub-flows and "@src" is the source address.

Label tables should contain the four following fields: destination address, flow label, source address and sub-flow label. With these fields, the classification of packets is fully achieved with the IPv6 header.

8. ATM and RSVP Switching

We need to select a switching hardware at the link layer to provide label swapping. We choose ATM as the leading switching technology. With ATM, the labels take place in the VPI/VCI fields of the ATM header cell. The label swapping table at the link layer is the switching table of the ATM hardware and the management of the shortcuts is similar to the one of ATM Virtual Circuits.

Alike IPsilon, we use directly the ATM switching hardware, but there is no need to use a specific protocol like IFMP to manage VPI/VCI exchange. The management of VCI/VPI is associated to the new "Label" object of RESV messages. We use RSVP as the unique state management protocol, because it

[2] Previously, this identification could be done by the source relative `Priority` field. This field of the IPv6 header is now replaced by the `Class` field, designed for network relative priorities use and cannot be used for sub-flows.

maintains a local network state and provides receiver oriented mechanisms like IFMP. Introducing RSVP close to the ATM hardware provides the RSVP and IP multicast management capacities to the ATM link layer multicast.

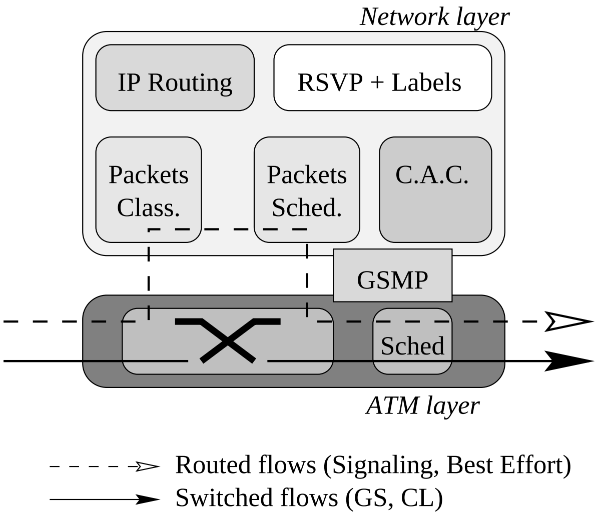


Figure 14. RSVP Switch architecture

Figure 14 presents the RSVP Switch architecture. It is the integration of both RSVP and IPsilon architectures. At the network layer, we find the classical RSVP elements with an extended RSVP daemon dealing with label exchange. It uses the new object "Label" and manage the ATM switch through IPsilon GSMP. At the link layer, we find a general ATM hardware, where the routed flows use dedicated VCs to the routing part of the RSVP Switch. The switched flows go only through the ATM matrix as ATM VCs do.

The ATM hardware requires scheduling capabilities, namely, per flow queuing and per class queuing. The former is dedicated to manage per flow QoS while the latter is designed to carry aggregated flows. This solution enables to keep signaling separated while using merged resources. In this case we associate multiple labels to aggregated flows *i.e.* to a class. Discussion of class based queueing may be found in [12]. This allows us to work without VC merge capable switch and to keep low latency. Moreover, using VC merge suppose to be able to identify sender flows after merging which is rather complex.

RSVP switching is not designed to support all type of applications but only those that request hard guarantees. This requires to maintain a per flow state in the network. A major problem appears due to the limited VPI/VCI space and the fact that we need a VCI/VPI mapping per flow or sub-flow. Other scalable solutions like DiffServ (Differentiated Services) are far simpler but can only provide classes of service. The RSVP solution is known to be not scalable and its utilization will be limited to LAN or Virtual network environments. The mapping of local RSVP networks with wide-area networks is therefore becoming a major issue.

9. RSVP Switching Testbed

The RSVP Switch architecture will be demonstrated on top of MIRIHADE, a French nationwide IP over ATM multi-megabit testbed. Figure 15 shows where RSVP Switching will be introduced in the testbed.

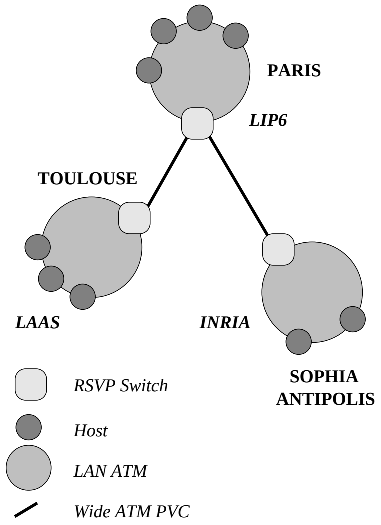


Figure 15. MIRIHADE Testbed

QoS mapping between reservations and ATM parameters is achieved with GSMP[3]. The way to support Guaranteed Service (GS) and Controlled Load (CL) services on a RSVP Switch architecture is presented in [27]. We introduce a new service called Bandwidth Recovery (BR) to implement CL as a way to access the bandwidth left unused by the GS flows. We implement a WFQ (Weighted Fair Queuing) scheduler, a queue per GS or BR flows and an admission control mechanism based on:

- $\sum R_{GS} \leq R_{LinkBandwidth} - \epsilon_{BR}$
- $\sum R_{\overline{GS}} + \sum R_{\overline{BR}} \leq R_{LinkBandwidth}$

[3] GSMPv2 manage various QoS parameters associated with the internal switch QoS components.

where R_{GS} is the peak rate of GS flow reservations, $R_{LinkBandwidth}$ the link rate capacity, ϵ_{BR} a small amount of bandwidth to avoid BR flows starvation, $R_{\overline{GS}}$ and $R_{\overline{BR}}$ are the average rate of respectively GS and BR flow reservations.

The required reservation is allocated to each GS flow and the remainder bandwidth is fairly shared between all the BR flows.

10. Conclusion

The interest of using ATM with RSVP is to avoid the connection-oriented mode of ATM to exploit the high forwarding power of the ATM switching hardware, and to provide a good latency trough the cell based ATM layer.

The utilization of the ATM hardware at the link layer associated with IPv6 at the routing layer enables the “RSVP multicast” model and add new possibilities for the label swapping approach. The advantages are: the use of IP with high forwarding rate and low delay; the dynamic use of ATM and the availability of the complete extended RSVP multicast model using a “flow driven” approach.

This solution suits well local environments like VPN or campus networks, because it is based on RSVP limited in terms of scalability. This approach should scale in environments where guarantees are more important than the number of flows. The per VC queuing should not be a drawback since new ATM hardware are now presented with more than hundred thousand policed queues.

We are now in the process of implementing a prototype over our ATM network testbed. The objective is to validate our “RSVP Switching” approach and demonstrate the ability of this model to satisfy various application requirements. A DIS (Distributed Interactive Simulation) application with multicast and QoS requirements aim at being demonstrated before the end of December.

References

[1] Arup Acharya, Furquan Ansari, Max Ott, and Henning Sanneck. Dynamic QoS for IP switching using RSVP over IPSOFACTO. In *Proceedings of SYBEN '98*, Zurich, May 1998.

[2] Arup Acharya, Rajiv Dighe, and Furquan Ansari. IP Switching Over Fast ATM Cell Transport (IPSOFACTO): Switching multicast flows. In *Proceedings of IEEE GLOBECOM '93*, volume 3, pages 1850–1854, Austin, Texas, November 1997.

[3] Grenville Armitage. IP Multicasting over ATM Networks. *IEEE Journal on Selected Areas in Communications*, 15(3):445–457, April 1997.

[4] E. Basturk, A. Birman, G. Delp, R. Guérin, et al. Design and Implementation of QoS Capable Switch-Router. Technical Report RC 20848 (01/31/97) Computer Science, IBM Research Division, January 1997.

[5] R. Braden, D. Clark, and D. Shenker. Integrated Services in the Internet Architecture: an Overview. IETF Informational Request For Comment 1633, July 1994.

[6] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jasmin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. IETF Request For Comment 2205, September 1997.

[7] E. Crawley, L. Berger, S. Berson, F. Baker, M. Borden, and J. Krawczyk. A Framework for Integrated Services and RSVP over ATM. Internet Draft `<draft-ietf-issll-atm-framework-02.txt>`, work in progress, February 1998.

[8] B. Davie, Y. Rekhter, A. Viswanathan, S. Blake., V. Srinivasan, and E. Rosen. Use of Label Switching With RSVP. Internet Draft `<draft-ietf-mpls-rsvp-00.txt>`, work in progress, March 1998.

[9] S. Deering. Host Extensions for IP Multicasting. IETF Request For Comment 1112, Stanford University, August 1989.

[10] H. Esaki, Y. Katsube, S. Matsuzawa, A. Mogi, K. Nagami, and T. Jinmei. Cell switch router: High performance packet forwarding architecture over internet and intranet using atm technology. White paper, Toshiba Corporation, Japan, November 1996.

[11] N. Feldman, P. Doolan, A. Fredette, and L. Andersson. Label Distribution Protocol (LDP) Specification. Internet Draft `<draft-feldman-ldp-spec-00.txt>`, work in progress, November 1997.

[12] Sally Floyd and Van Jacobson. Link-Sharing and Ressource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386, August 1995.

[13] Olivier Fourmaux and Christophe Deleuze. Qualité de service hétérogène pour liaisons ATM point-à-multipoint. In *Actes de CFIP'97*, pages 383–398, Liège, Belgium, September 1997. Université de Liège, Hermes.

[14] J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5. IETF Request For Comment 1483, Telecom Finland, July 1993.

[15] Christian Huitéma. *Le Routage dans l'Internet.* Eyrolles, April 1995.

[16] Christian Huitéma. *IPv6, The New Internet Protocol.* Prentice Hall, second edition, 1998.

[17] Lan emulation over atm – version 1.0 specification. Technical Report af-lane-00021.000, ATM Forum, January 1995.

[18] Lan emulation over atm – version 2.0, luni specification. Technical Report af-lane-00084.000, ATM Forum, July 1997.

[19] M. Laubach and J. Halpern. Classical IP and ARP over ATM. IETF Request For Comment 2225, Com21, Inc. and Newbridge Networks, Inc., April 1998. *Remplace les RFC 1577 et 1626.*

[20] F. Liaw, A. Mankin, E. Hoffman, D. Grossman, and A. Malis. ATM Signaling Support for IP over ATM. IETF Request For Comment 1755, February 1997.

[21] J. Luciani, D. Katz, D. Piscitello, B. Cole, and N. Doraswamy. NBMA Next Hop Resolution Protocol (NHRP). IETF Request For Comment 2332, April 1998.

[22] Multiprotocol over atm – version 1.0. Technical Report af-mpoa-00087.000, ATM Forum, July 1997.

[23] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. Ipsilon's General Switch Management Protocol Specification Version 2.0. IETF Informational Request For Comment 2297, Nokia, Sprint, Fiberlane, March 1998.

[24] Peter Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. Ipsilon Flow Management Protocol Specification for IPv4 – Version 1.0. IETF Informational Request For Comment 1953, IPsilon, May 1996.

[25] Peter Newman, Greg Minshall, and Tom Lyon. IP Switching: ATM Under IP. *IEEE/ACM Transactions on Networking*, November 1997.

[26] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow. Cisco Systems' Tag Switching Architecture Overview. IETF Informational Request For Comment 2105, Cisco Systems Inc., February 1997.

[27] Simon-Pietro Romano, Christophe Deleuze, José Rezende, and Serge Fdida. Integrated qos architecture for ip switching. In *6th Open Workshop on High Speed Networks Proceedings*, Stuttgart, October 1997.

[28] L. Salgarelli, A. Corghi, M. Smirnow, H. Sanneck, and D. Witaszek. Supporting IP Multicast Integrated Services in ATM Networks. Internet Draft `<draft-salgarelli-issll-mis-00.ps>`, work in progress, CEFRIEL Politecnico di Milano and GMD-FOKUS, November 1997.

[29] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. IETF Request For Comment 2212, September 1997.

[30] M. Smirnow. Scalable and efficient multiprotocol ip multicast over atm. In *Proceedings of SPIE VV'97*.

[31] J. Wroclawski. Specification of the Controlled-Load Network Element Service. IETF Request For Comment 2211, MIT LCS, September 1997.

[32] J. Wroclawski. The Use of RSVP with IETF Integrated Services. IETF Request For Comment 2210, MIT LCS, September 1997.

[33] Lixia Zhang, Stephen Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network Magazine*, pages 8–18, September 1993.