Adaptation du Protocole RSVP sur les Réseaux ATM

Olivier Fourmaux [1]

22 juillet 1996

[1] Olivier.Fourmaux@masi.ibp.fr

Résumé

Dans ce document, nous étudions des problèmes liés à l'adaptation de RSVP sur les réseaux ATM. En particulier, dans un schéma classique d'utilisation de RSVP sur le modèle "Classical IP over ATM", nous nous intéressons à la correspondance entre les styles de réservations RSVP et les VC ATM ainsi qu'à la mise en œuvre du modèle multicast dynamique à récepteurs hétérogènes de RSVP avec les VC multicast à QoS unique et statique d'ATM.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction

Nous débutons ce document par une présentation des services offerts pour les des deux architectures que nous utilisons, IP et ATM, puis nous étudions leurs interactions lors de l'utilisation de IP sur ATM.

1.1 Services IP

La gestion de la qualité de service dans l'Internet utilise le protocole de réservation de ressources RSVP. Ce protocole est adapté au modèle de réseaux basé sur IP avec fonctionnalités multicast.

1.1.1 Le protocole de l'Internet

Le principe de base du réseau des réseaux, l'Internet, est de fonctionner avec un protocole de niveau 3 unique : le protocole IP. L'Internet utilise actuellement IP version 4, et nous utiliserons bientôt son évolution, IP version 6. Nous allons présenter ces deux versions ainsi que la gestion du multicast avec IP.

IPv4

IPv4 est un protocole qui fonctionne en mode non fiable non connecté. Le protocole ne gère donc pas d'état du réseau dans les nœuds intermédiaires. Tout paquet est acheminé vers son destinataire de manière indépendante. Il est impossible de garantir à un flot de données particulier un délai d'acheminement borné. Il est donc nécessaire d'ajouter à IP un protocole de signalisation qui pourra gérer un minimum d'états tel que RSVP.

La résolution des adresses du niveau inférieur se fait avec un protocole dépendant de cette couche : ARP[1]. Cette résolution est dépendante du médium utilisé.

le multicast avec IPv4

Dans sa forme de base, IPv4 ne gère que le mode point-à-point. La définition des adresses de groupes et de la signalisation associée IGMP[2] a permis l'utilisation du mode multipoint-à-multipoint ou multicast (voir plus bas).

[1]Address Resolution Protocol
[2]Internet Group Management Protocol

IPv6

IPv6 [15] est une évolution de IPv4 dont il reprend les principaux mécanismes. Les changements de taille de certains champs, notamment celui des adresses, nécessitent la modification des API[3] et des protocoles développés pour IPv4. La gestion du multicast a été intégrée au protocole de contrôle (IGMP fait partie de ICMP[4]).

La résolution d'adresses fait maintenant partie de la base du protocole IPv6 et est donc située dans le niveau réseau plutôt que dans les niveaux inférieurs. Les mécanismes d'adaptation d'IPv4 sur ATM ne pourront être traduits directement pour IPv6 à cause de ce transfert de fonctionnalité. Le protocole ND[5], qui fait partie intégrante de IPv6, remplit par exemple les fonctionnalités de résolution d'adresses assurées sous IPv4 par ARP.

Un champ pour un identificateur de flot est intégré dans l'entête pour pouvoir gérer les trafics qui nécessitent un traitement particulier (association avec RSVP par exemple). Ce champ permet juste une meilleure gestion du trafic dans les routeurs et n'introduit pas la notion d'état dans le réseau IPv6.

1.1.2 Le multicast Internet

Sous IP, une partie de l'espace d'adressage est dévolu aux adresses de groupes multicast[6]. Cette abstraction entre le contenu et le contenant permet d'adresser virtuellement toutes les interfaces Internet qui auront décidées de se joindre au groupe.

Le multicast IP [4], est orienté récepteur par le découplage entre l'émission et la réception que génèrent les adresses de groupe. Un émetteur émet vers un groupe multicast sans avoir d'information sur ses participants et un récepteur appartient à un groupe sans explicitement connaitre les émetteurs du groupe.

L'attachement d'un membre, ou son départ, sont des primitives de IGMP[7] appelées par le membre lui-même. Le processus est local, c'est le protocole de routage multicast qui gère la distribution des paquets selon les groupes. Le système est basé sur des envois périodiques de messages afin d'assurer la dynamique du système.

Le service fourni initialement était symétrique sur toutes les branches étant donné qu'il n'y en avait qu'un : le service "au mieux". Le but avec la spécification de service est de fournir un multicast avec une QoS[8] adaptée sur chaque branche de l'arbre de distribution.

1.1.3 Les services Internet

Le groupe de travail IETF[9] IntServ[10] a défini quatre classes de service [2, 9] basées principalement sur le délai d'acheminement. Les pertes ne sont pas prises en compte, elles sont supposées faibles. Le tableau 1.1 décrit ces classes.

[3] Application Programming Interface
[4] Internet Control Message Protocol
[5] Neighbor Discovery
[6] Adresses de classe D avec ipv4
[7] Ou ICMP dans IPv6
[8] Quality of Service
[9] Internet Engineering Task Force
[10] Internet Services

Classe	*Description*
“Guaranteed Delay”	La classe à *délai garanti* fournit des bornes strictes sur le délai maximum d'acheminement des paquets de bout en bout.
“Controlled Delay”	La classe à *délai contrôlé* fournit différents niveaux de service hiérarchisés (le plus petit étant le meilleur), tous meilleurs que le service “au mieux” pour les délais. Pour chacun de ces niveaux, le réseau fournit une estimation du délai maximum.
“Predictive Service”	La classe à *service prévisible* est identique à la classe à *délai garanti* excepté que la perte ou le retard occasionnel de paquets est toléré. Comme pour la classe à *délai contrôlé*, des niveaux de service sont définis.
“Controlled Load”	La classe à *charge contrôlée* procure le service minimal : identique à celui qui serait procuré en utilisant le service “au mieux” habituel sur le réseau sans charge.

Tableau 1.1: Classes de service Internet

1.1.4 La réservation de ressources dans l'Internet

RSVP [1, 2, 3] est un protocole de signalisation de réservation de ressources sur IP. Il est utilisé pour indiquer la qualité de service à fournir pour un flot de données.

Fonctionnalités de RSVP

Les caractéristiques du protocole RSVP sont réunis dans le tableau 1.2.

Mécanismes de RSVP

Les réservations sont basées sur des *sessions* identifiant un flot de données pour une destination et un protocole de transport. RSVP utilise deux messages principalement : *RESV* et *PATH*.

- Les messages PATH sont émis des émetteurs vers les récepteurs, établissent un état dans les nœuds traversés et transmettent la description des flots des émetteurs aux récepteurs. Les messages PATH suivent les mêmes chemins que les données et construisent un arbre de parcours inverse que suivront les messages RESV.

- Les messages RESV sont émis des récepteurs vers les émetteurs par les chemins définis avec les messages PATH. La réservation est réalisée à l'aide de ces messages RESV en une passe ou mieux, une passe avec indication en retour (*OPWA*[11]) par le champ *Adspec* des messages PATH.

Les messages sont émis périodiquement afin de maintenir dynamiquement les chemins et réservations utilisés. Deux messages, PTEAR et RTEAR, permettent de libérer les réservations explicitement.

[11] One Path With Advertising

Réservation pour un flot de données **simplex** (requêtes de réservation dans un sens).
Protocole de **signalisation IP** (fonctionne sur IPv4 ou IPv6 au niveau du protocole de transport sans transmission de données).
Orientation **récepteur** (les récepteurs d'un flot initient et entretiennent les réservations de celui-ci).
Gestion des réservations du modèle point-à-point au modèle **multipoint-à-multipoint**.
Adaptation **dynamique** aux changements de routages et de groupes.
RSVP n'est pas un protocole de routage (il doit être intégré avec les fonctionnalités de routage, de contrôle d'admission, de filtrage et d'ordonnancement dans les routeurs).
Gestion d'un **état du réseau** (*Soft State*) périodiquement mis à jour.
Gestion de différents **styles de réservation** adaptés aux applications permettant la fusion des flots.
Prise en compte transparente des routeurs non RSVP.

Tableau 1.2: Fonctionnalités RSVP

Une réservation RSVP est associée à un descripteur de flot, composé d'un *flowspec* et d'un *filter spec*.

- Un *flowspec* contient une description de la classe de service et deux valeurs numériques : *Rspec* et *Tspec* représentant respectivement la qualité de service et le trafic.
- Un *filter spec* est utilisé pour choisir un sous-ensemble de paquets du flot (la sélection est effectuée par rapport à l'émetteur ou par un protocole de niveau supérieur).

Une réservation est associée à un style qui permet d'effectuer des agrégations entre les différents flots d'une session. Les paramètres sont fonction de la capacité des flots à partager une même ressource par émission alternative ou non (*shared* ou *distinct*) et de l'expression explicite des émetteurs ou non (*explicit* ou *wildcard*). Trois styles sont définis, nous les décrivons dans le tableau 1.3 [12].

Architecture RSVP

Une architecture basée sur RSVP pour fournir des services intégrés repose sur différents éléments représentés sur la figure 1.1 : module de routage, module RSVP, module de filtrage, ordonnanceur de paquets et contrôleur d'admission.

Lors de l'arrivée d'un message RESV, le module de contrôle d'admission accepte ou non la requête. En cas de succès le *filter spec* est confié au module de filtrage des paquets et le *flowspec* est donné à l'ordonnanceur de paquets pour assurer la bonne qualité de service aux bons paquets.

[12] Astérisque pour l'émetteur quelconque, $\mathcal{Q}$ pour le flowspec demandé, S_n les émetteurs nommés et $\mathcal{Q}_n$ pour le flowspec correspondant.

Style	*Représentation*	
“Wildcard-Filter”	**WF(**$*Q$**)**	Une réservation est partagée par un groupe d’émetteurs (la réservation correspond à la demande la plus importante).
“Shared-Explicit”	**SE(**$(S_1, S_2, ...)Q$**)**	Idem que précédemment mais les émetteurs sont déclarés explicitement.
“Fixed-Filter”	**FF(**S_1Q_1**), FF(**S_2Q_2**), ...**	Une réservation distincte par émetteur (pas de partage, la réservation correspond à la somme des demandes).

Tableau 1.3: Styles RSVP

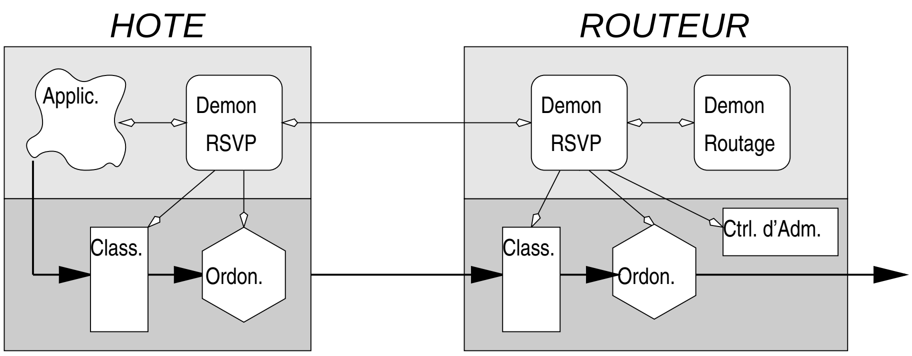

Figure 1.1: Architecture des nœuds du réseau avec RSVP.

1.1.5 Le modèle IP

Les caractéristiques du modèle IP que nous utilisons sont les suivantes :

- initiées par les récepteurs,
- de styles différents[13],
- gérées dynamiquement,
- hétérogènes.

Nous étudions l’adaptation de chacune sur les réseaux ATM[14].

[13] pour l’agrégation des flots
[14] Asynchronous Transfert Mode

1.2 Services ATM

Les réseaux ATM sont de type maillé (liaison point-à-point possible entre tous les hôtes). Les communications s'effectuent en mode connecté. Les connections appelées VC[15] sont donc initialisées avant d'être utilisées. Les caractéristiques d'un VC étant statiques pendant sa durée de vie, la dynamicité du modèle ATM dépend du temps de création d'un VC.

L'accès aux services ATM se fait à travers l'interface UNI[16]. La norme actuelle est UNI 3.1 et la version 4.0 est en cours de développement. Nous étudions ces interfaces et plus particulièrement le service multicast qui, de part sa modélisation différente de celle d'Internet, va poser des problèmes d'adaptation.

1.2.1 UNI 3.0 et 3.1

UNI 3.x fournit les PVC[17] et les SVC[18] avec trois types services (CBR[19], VBR[20] et UBR[21]) et deux types de liaison (point-à-point bidirectionnel ou point-à-multipoint unidirectionnel).

Pour le multicast, ATM n'offre ni la diffusion, ni l'abstraction des adresses de groupes, ni le routage dynamique du monde IP. Le service multicast se limite à des **VC point-à-multipoint unidirectionnels**. De plus, seule la racine (émetteur) de l'arbre multicast peut gérer l'ajout ou le retrait de feuilles (récepteurs). Un ensemble d'hôtes desservis par le service multicast est appelé un *cluster*. Deux modèles ont été définis pour gérer le *multicast* au niveau ATM :

Maillage multicast

Le maillage multicast crée des VC multicast de chaque émetteur vers tous les autres membres du groupe. Les problèmes proviennent de la consommation de ressources et du coût de l'ajout d'un membre (en pointillé sur la figure 1.2 à gauche).

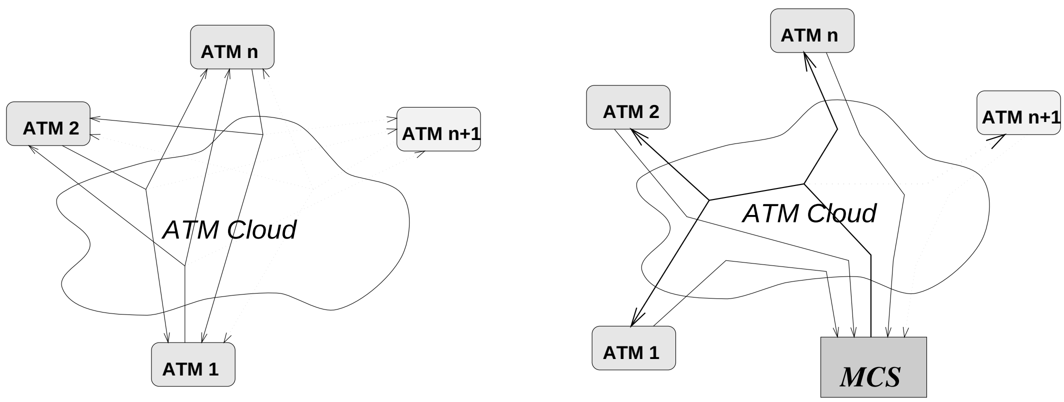

Figure 1.2: Maillage multicast et serveur multicast

[15]Virtual Circuit
[16]User-Network Interface
[17]Permanent Virtual Circuit
[18]Switched Virtual Circuit
[19]Constant Bit Rate
[20]Variable Bit Rate
[21]Unspecified Bit Rate

Serveur multicast

Le serveur multicast (MCS[22]) nécessite la création d'un VC point-à-point de chaque source vers un MCS. Le MCS établit un VC point-à-multipoint vers les destinations. Le rôle du MCS est de réassembler les paquets arrivants sur les VC entrants et de les mettre dans une file d'attente avant de les émettre sur le VC point-à-multipoint. La gestion de groupes est facilitée par la structure centralisée, par contre cela produit des risques de congestion à l'entrée du serveur. Ce modèle est mieux adapté aux groupes dynamiques de récepteurs (ajout économique d'un membre : en pointillé sur la figure 1.2 à droite) mais le temps de réassemblage des paquets et le double transit va augmenter le temps de latence.

1.2.2 UNI 4.0

UNI 4.0 est une évolution de l'interface précédente avec des apports permettant de mieux s'adapter aux demandes de l'informatique. De nouvelles catégories de services ont été définies : ABR[23], rt-VBR[24]et nrt-VBR[25]. Parmi les nouvelles fonctionnalités, se trouve le LIJ[26], qui va permettre d'orienter le multicast coté récepteur. Le LIJ permet à un récepteur de se joindre à un arbre multicast (VC point-à-multipoint) sans en avertir la racine (l'émetteur). Cela permet de se rapprocher du modèle multicast IP, mais il n'y a toujours pas d'abstraction de groupe (il faut connaitre l'émetteur pour rejoindre son arbre multicast), de QoS hétérogène sur les VC point-a-multipoint, ni de renégociation de la QoS des VC existants possible.

1.2.3 Le modèle ATM

Nous retiendrons que les VC ATM sont :

- initiés au niveau de l'émetteur,
- avec spécification explicite des récepteurs multicast,
- avec une QoS non renégociable,
- avec la même QoS sur chaque branche de l'arbre multicast.

Nous allons donc faire correspondre les services IP avec la création de VC en spécifiant une QoS, un type de services et un mode (point-à-point ou point-à-multipoint).

1.3 IP sur ATM

Différents modèles ont été proposés[10] pour utiliser IP sur ATM. Le seul actuellement disponible est le modèle "classical IP over ATM" [5, 11] que nous décrivons dans la suite.

[22]MultiCast Server
[23]Available Bit Rate
[24]Real Time Variable Bit Rate
[25]Non Real Time Variable Bit Rate
[26]Leaf Initiated Join

1.3.1 Modèle "Classical IP"

Ce modèle utilise IP et ARP[27] de manière classique sur un réseau ATM configuré en LIS [28] (les LIS sont des entitées administratives disjointes et correspondent à un lien dans un réseau à diffusion). La correspondance adresses-IP/adresses-ATM dans un LIS est fournie par le service ATMARP (ARP adapté à un niveau 2 de type ATM). Pour les communications vers des hôtes externes au LIS, un routeur IP est utilisé. Si le réseau ATM peut être divisé en plusieurs LIS, la communication inter LIS devra passer par des routeurs IP, même si un VC entre deux hôtes de deux LIS différents peut être établi. La figure 1.3 représente le chemin entre deux hôtes à travers plusieurs LIS.

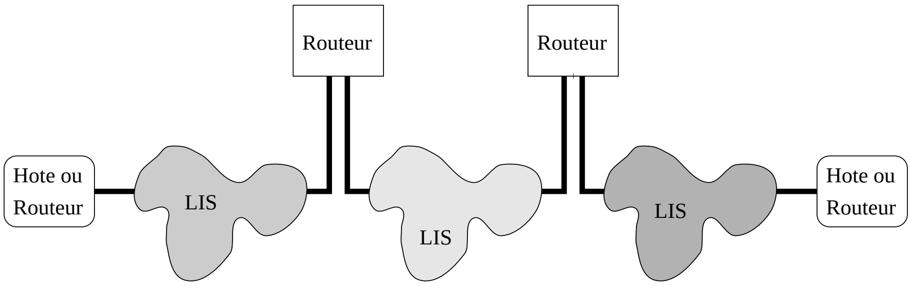

Figure 1.3: Le Modèle "Classical IP" avec trois LIS.

1.3.2 Multicast dans "Classical IP"

Pour le multicast, les hôtes et les routeurs IP connectés à ATM utilisent MARS[29] qui leur fournit le service de niveau 2 du multicast IP [4] pour la résolution des adresses. A l'intérieur d'un *cluster*, MARS s'occupe de la gestion des groupes multicast et permet au niveau 3 du multicast IP d'utiliser le modèle du *maillage multicast* ou le modèle du *serveur multicast* au niveau ATM.

MARS est une évolution du serveur ATMARP. celui-ci utilisait une table de conversion d'adresses IP pour tous les membres du LIS :

$$\boxed{\{\ \text{Adresse-IP} \Longleftrightarrow \text{Adresse-ATM}\ \}}$$

MARS étend la table et généralise l'adresse d'entrée :

$$\boxed{\{\ \text{Adresse-de-niveau-3} \Longleftrightarrow \text{Adr-ATM.1, Adr-ATM.2, ... Adr-ATM.n}\ \}}$$

MARS peut résider dans tous nœuds directement accessibles par les utilisateurs de ce serveur. Les interfaces qui veulent se joindre à un groupe doivent connaitre l'adresse du nœud où leur MARS réside. MARS n'est qu'un système de résolution d'adresses, il reste a créer les VC ATM point-à-multipoint pour obtenir un service multicast. Les limitations d'ATM apparaissent ici : création de VC à QoS identiques à partir de l'émetteur sous UNI 3.x ou diffusion de l'adresse de l'émetteur et de la référence de l'arbre multicast aux récepteurs pour qu'ils créent les VC sous UNI 4.0 avec la même QoS.

[27] Address Resolution Protocol
[28] Logical IP Subnetwork
[29] Multicast Address Resolution Server

1.3.3 Intégration de IPv6 sur ATM

Les modèles précédents ont été développés pour IPv4. Des travaux d'adaptation de ces modèles ou de nouveaux pour IPv6 sont en cours. Le transfert au niveau IP de la fonctionnalité de résolution d'adresse ARP ne permet plus d'utiliser ATMARP. La résolution d'adresse se trouve dans ND pour IPv6, il faut donc étudier l'adaptation de ND sur ATM. On peut citer une proposition [12] pour gérer de manière hiérarchique ND (adaptation à un voisinage de taille plus grande).

La limitation principale de IPv4 sur ATM était l'impossibilité de faire des VC directs à l'extérieur du LIJ (appelé "shortcuts"). ND gère ces nouveaux hôtes en introduisant la notion de lien logique. La résolution d'adresse peut-être effectuée par le protocole NHRP [14]. De nombreuses discutions ont lieu car l'utilisation des shortcuts pose des problèmes au niveau de l'adaptabilité aux grands groupes de récepteurs.

Chapitre 2

RSVP sur ATM

Plusieurs travaux ont déjà abordé l'adaptation des services IP sur ATM à travers RSVP. Nous les présentons en insistant sur leur manière de traiter le multicast. Nous étudions aussi les différentes problèmes posés par l'utilisation de RSVP sur ATM et l'environnement sur lequel nous nous basons.

2.1 Travaux apparentés

Plusieurs propositions d'adaptation de RSVP sur ATM ont déjà été présentées sous forme d'Internet Draft :

- présentation de la problématique dans le draft "Issues for RSVP and Integrated Services over ATM" [8],
- proposition utilisant des shortcuts dans le draft "Support for RSVP-based Service over an ATM Network" [7],
- proposition utilisant des éléments intermédiaires gérant les fonctionnalités RSVP au niveau d'un cluster MARS dans le draft "A Framework for Supporting RSVP Flows Over ATM Networks" [9].

2.1.1 Intégration RSVP et ATM

Nous allons considérer dans un premier temps les liaison point-à-point. Les informations pour gérer les VC sont fournies par les messages RSVP.

Modèle classique RSVP sur ATM

Ce modèle "classical RSVP over ATM" [7] se positionne sur le modèle "classical IP over ATM" vu précédemment. Les routeurs dans le LIS intègrent les fonctionnalités RSVP. Les messages PATH vont de l'émetteur (A) vers le récepteur (B) en construisant le *soft state* dans les nœuds intermédiaires R1 et R2. Les messages RESV parcourent le chemin inverse de B vers A. Lorsque R2 reçoit le message, il a suffisamment d'information pour faire sa réservation entre R2 et B, c'est à dire, initialiser un VC entre R2 et B avec la QoS correspondante. Même chose pour R1 vers R2 et A vers R1 comme le représente la figure 2.1.

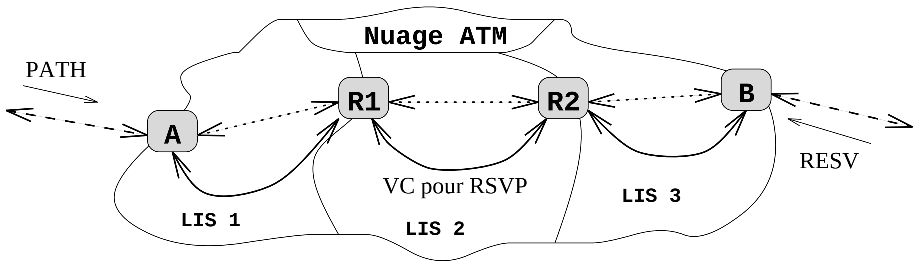

Figure 2.1: Réservation classique à travers 3 LIS.

Modèle avec liaisons ATM directes

Pour pouvoir utiliser la QoS de bout en bout, il est intéressant d'éviter la remonté au niveau IP dans les routeurs intermédiaire. Pour cela il faut que les routeurs connaissent les adresses ATM des routeurs d'entrée et de sortie du nuage pour pouvoir créer un ATM *shortcut* [7]. Lorsqu'un message PATH va arriver dans un routeur entrant (A), il va transmettre l'adresse ATM de ce routeur en plus. Les routeurs intermédiaires dans le nuage ATM ne vont ni créer d'état ni modifier le champ nœud précédent. Lorsque le routeur de sortie reçoit le PATH il créé un état et y ajoute l'adresse ATM contenue dans le PATH. Il est alors possible de créer un *shortcut*, soit en renvoyant un message RESV au routeur entrant (A) avec l'adresse ATM du routeur sortant (B) par le chemin classique pour lui indiquer de créer un VC entre A et B (figure 2.2) avec la QoS demandée, soit en créant le VC entre B et A directement par le routeur sortant (B). Les VC étant bidirectionnels, la réservation de ressources se fait dans les deux sens. La réservation sera spécifiée seulement pour A vers B.

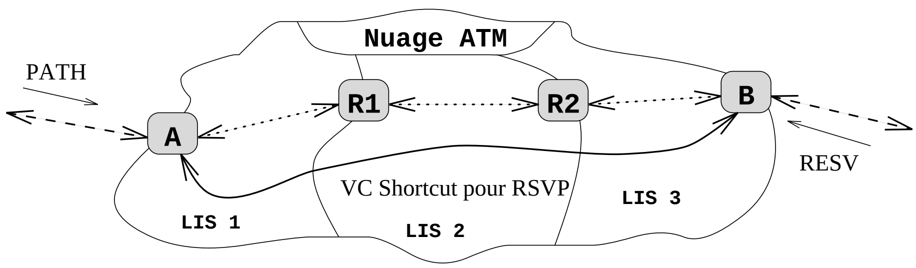

Figure 2.2: Réservation d'un “shortcut” à travers 3 LIS.

Finalement, un *shortcut* a été créé entre A et B avec la QoS désirée, utilisant ainsi au maximum les capacités du réseaux ATM. Les nœuds R1 et R2 sont évités sauf pour la signalisation. Deux problèmes apparaissent ici :

- La gestion des shortcuts est un problème de routage, or les spécifications de RSVP insistent sur le fait que RSVP ne s'occupe pas du routage. C'est au niveau IP de gérer les shortcuts.
- Ce que l'on gagne au niveau ATM avec les shortcuts est perdu au niveau optimisation de l'utilisation des ressources : moins il a d'élément RSVP intermédiaire, moins l'agrégation de flots est possible.

2.1.2 Intégration de RSVP sur ATM pour le Multicast

Nous étudions maintenant comment est traité le multicast avec les shortcuts. La restitution du service multipoint-à-multipoint de RSVP dans ATM passe par l'utilsation des fonctionnalités multicast de celui-ci. Deux types de solutions sont proposées : basées sur des *shortcuts* ATM ou basées sur des éléments RSVP intermédiaires.

Voyons dans un premier temps deux méthodes [7] utilisant des *shortcuts* multicast. Chacune des méthodes est adaptée à un environnement : les *shortcuts* créés par l'émetteur correspondent à UNI 3.1 et les *shortcuts* créés par les récepteurs correspondent à UNI 4.0.

ATM shortcuts créés par l'émetteur

Les messages PATH utilisent le parcours des paquets IP à travers les LIS et un procédé local est utilisé pour résoudre la diffusion des paquets *multicast*. Les PATH arrivent dans le nuage ATM par le routeur entrant A. A insère son adresse dans les PATH et fait suivre les paquets. Les routeurs intermédiaires dans ATM ne créent pas d'état et ne modifient pas le champ nœud précédent du PATH. Le Message PATH va arriver aux routeurs sortants B et C. Supposons que B reçoive le message en premier. L'adresse de B se trouvera dans les messages RESV à destination de A. Les routeurs intermédiaires R1 et R2 ne modifient pas le message. L'arrivée de ce message au routeur entrant A permet l'établissement d'un VC multicast ayant pour racine A et pour feuille B. Le second message RESV provenant de C arrive en A. A va ajouter au VC multicast existant une nouvelle feuille, C, grâce au message de signalisation ATM “ ADD PARTY”. La figure 2.3 nous indique l'état final de la réservation.

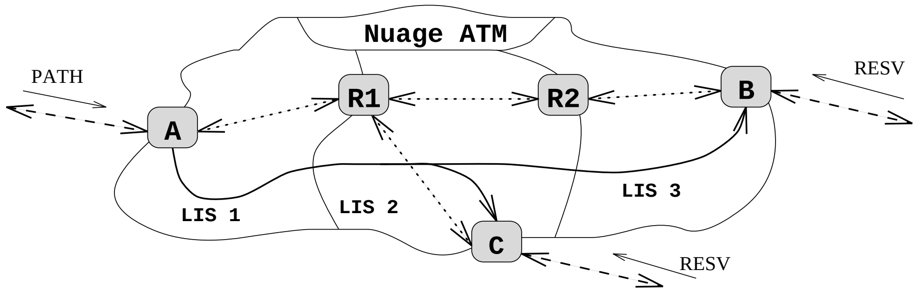

Figure 2.3: Réservation d'un “shortcut” Multicast à travers 3 LIS.

Cette solution permet d'utiliser du multicast avec RSVP sur ATM mais est limité au modèle multicast d'ATM, c'est à dire arbre multicast avec des branches aux QoS homogènes et pas de dynamicité (si un récepteur demande une QoS plus importante, il faut renégocier tout l'arbre).

ATM shortcuts créés par les récepteurs

Les messages PATH sont échangés comme précédemment et continuent d'utiliser les mécanismes IP multicast sur ATM. Ces messages vont transporter en plus de l'adresse ATM de A, un identificateur de l'arbre multicast auquel les récepteurs devront se joindre avec la requête LIJ. Lorsqu'un routeur sortant doit générer un message RESV (après réception d'un message RESV en aval), il peut être le premier à le faire. B, par exemple, envoie une requête LIJ. Le VC multicast est créé à cet instant avec A comme racine et B comme première feuille. Les routeurs sortants suivants, tel que C, vont rejoindre l'arbre en rajoutant des branches à l'arbre multicast avec des requêtes LIJ.

La racine A n'en sera pas avertie évitant ainsi la congestion à la racine, mais les modèles que nous avons décrits ne permettent pas, à l'intérieur du nuage ATM, l'agrégation de flots a cause de l'utilisation de *shortcuts* entre routeurs entrants et routeurs sortants. Cela posera des problèmes lors de l'utilisation avec des groupes multicast de grande taille. Les mêmes limitations que précédemment se retrouvent au niveau du multicast, cette solution n'apporte que l'utilisation des VC multicast initiés par les feuilles.

Entités de coordination RSVP

Une autre approche a été développée pour pouvoir mieux utiliser le multicast avec RSVP : un modèle avec des entités intermédiaires [9]. Le principe de base de ce modèle est d'intégrer des réseaux de routeurs et d'hôtes RSVP dans le nuage ATM. MARS gère la résolution des adresses IP de groupe. Nous supposerons que l'étendue du nuage ATM est la même que celle du cluster MARS.

Les fonctions réseau sont supportées par une RCE[1], Entité de coordination RSVP qui s'interface avec ATM via UNI et gère les fonctionnalités RSVP tel que :

- Agrégation de flots RSVP (un ou plusieurs RCE sur un chemin)
- Établissement et gestion d'arbre *multicast* au niveau ATM
- Support de l'hétérogénéité des récepteurs
- Conversion des *Tspec* en paramètres de trafic ATM
- Conversion des *Rspec* en qualité de service ATM

Il peut y avoir un ou plusieurs RCE dans un cluster MARS. La figure 2.4 représente une session RSVP avec deux émetteurs (S1 et S2), un RCE (RCE1), trois récepteurs (R1, R2, R3). Ce modèle est facilement extensible à plusieurs RCE. Les émetteurs et les récepteurs peuvent ne pas être des éléments terminaux et dans ce cas font suivre les messages PATH et RESV.

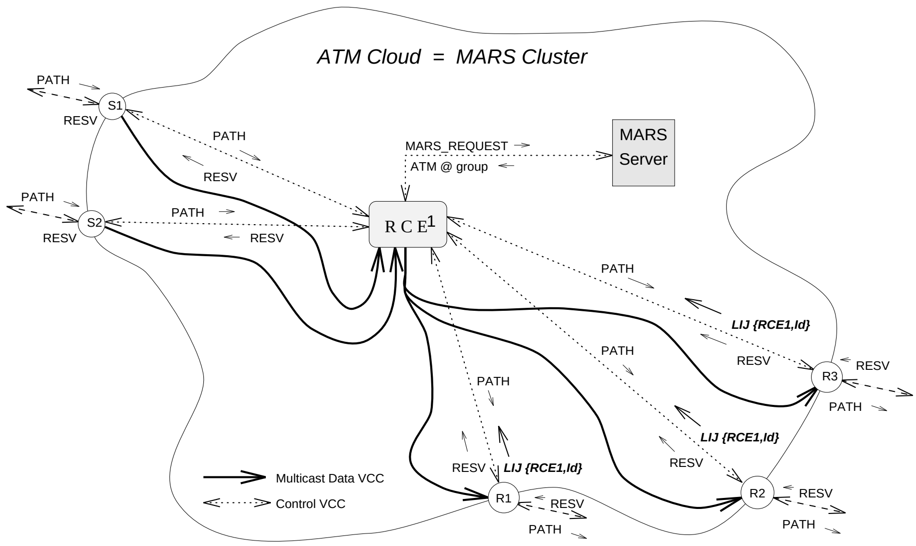

Figure 2.4: Gestion d'un flot par les RCE.

Les fonctionnalités de filtrage et d'agrégation de flots sont réalisées au niveau du RCE. La gestion de plusieurs RCE sur un même chemin est en cours de développement.

[1]RSVP Coordination Entity

2.2 Les problèmes de RSVP sur ATM

2.2.1 Shortcuts ATM

La question de savoir quand réaliser un *shortcut* reste ouverte. Faut-il favoriser l'agrégation de flot ou la création de shortcuts de taille maximum pour avoir les meilleures performances. Plus on utilise de shortcuts moins on remonte au niveau trois ; mais on ne peut plus gérer les agrégations de flot à cause du découplage chemin de donnée et signalisation.

L'utilisation de *shortcuts* impose la modification du protocole RSVP : Les messages PATH et RESV doivent transporter des informations pour la création de ces VC dont ils ne pourront peut être pas résoudre l'adresse à l'autre extrémité. Une gestion adaptée dans les nœuds intermédiaires est aussi nécessaire pour que les nœuds évités par le shortcut ne modifient pas les messages qui continuent à transiter par le chemin IP. La prise en compte des shortcuts au niveau RSVP n'est donc pas transparente.

La gestion des *shortcuts* est ici associée à RSVP, or c'est un problème de routage. On peut donc s'interroger sur l'orthogonalité du protocole de réservation et du protocole de routage et se demander si l'on traite bien les shortcuts au bon niveau sachant que dans le modèle initial de RSVP, la gestion des flots est en fonction du routage et non l'inverse. L'utilisation des shortcuts dans RSVP à été mise en place a ce niveau pour pallier un manque des protocoles de routages. Les nouveaux protocoles tel que la version 6 d'IP intègrent les shortcuts.

2.2.2 Elément intermédiaire

Plus une solution a d'éléments RSVP intermédiaires, plus son potentiel d'agréger des flots augmente. La solution avec intégration d'éléments intermédiaires (RCE) permet d'agréger au mieux les flots RSVP. Cette fonction est réalisée au niveau 3 à cause du filtrage, pour identifier les flots. Mais pour pouvoir utiliser pleinement l'agrégation, il faut étendre l'interface ATM pour pouvoir inclure deux nouvelles fonctionnalités :

- Avec des réservations partagées, il faut prendre la réservation maximum. Un nouveau récepteur qui rejoint le groupe avec une demande supérieure nécessite la possibilité de changement dynamique de QoS du VC utilisé.
- La gestion de récepteurs hétérogènes implique l'utilisation d'une QoS adaptée par branche. Il faut donc un nouveau modèle de VC multicast ou l'on pourra spécifier la QoS de chaque branche que l'on crée.

Avec l'agrégation des flots, la gestion du modèle multipoint-à-multipoint devient possible, mais la gestion n'est pas transparente pour l'ATM et la mise en œuvre est lourde.

2.2.3 Adaptation du Trafic IntServ à la QoS ATM

Les spécifications du trafic ou de la QoS ne sont pas intégrées à RSVP qui ne s'occupe pas du contenant des réservations qu'il opère. Un service du type à délai garanti est réalisé selon le principe du seau à jeton (*token bucket*) avec une profondeur et un débit moyen. Des travaux sur la conversion de ces valeurs en QoS ATM ont été réalisés.

2.2.4 Adaptation du Soft State RSVP au Hard State ATM

Les réseaux ATM gèrent un état global du système ("Hard State"). RSVP, par contre, met en œuvre un état global relaché ("Soft State") pour pallier le manque d'état global dans les réseaux

IP. L'adaptation de RSVP devra en tenir compte ; par exemple,la gestion des envois périodiques de messages pour le maintient du “Soft State” n'est plus utile dans le nuage ATM.

2.2.5 Limitation du modèle multicast ATM

Les solutions proposées dans les modèles précédents ne traitent que de l'architecture de RSVP sur ATM sans discuter des problèmes liés au multicast. Trois problèmes majeurs subsistent :

modèle multicast ATM

Pour identifier une session multicast avec ATM, on utilise la racine de distribution, car le modèle est orienté émetteur. Les sessions multicast ATM sont donc 1 vers N (point-à-multipoint) . RSVP suppose un modèle sous-jacent orienté récepteur qui va lui permettre de gérer des sessions N vers M (multipoint-à-multipoint). Le modèle utilisé par RSVP est celui d'IP avec abstraction des groupe : on découple les adresses des groupes des récepteurs eux-même pour en faire une entité virtuelle dynamique sur laquelle n'importe quel émetteur peut envoyer des informations et à laquelle les récepteurs peuvent appartenir à n'importe quel moment. Les sessions multicast RSVP sont identifiées par les groupes de récepteurs.

Dynamicité

Le modèle de réservation de RSVP est dynamique : il permet la renégociation à tout moment des réservations associées aux flots d'une session. Avec ATM, les flots correspondent à des VC et les réservations aux QoS des VC. La dynamicité du modèle est basée sur la dynamicité de la QoS des VC or la QoS d'un VC ATM est statique. La gestion de la renégociation d'un VC est donc basée sur le remplacement du VC avec la QoS obsolète par un autre VC avec la QoS demandée. Deux solutions pour cela :

- ouverture d'un nouveau VC et fermeture de l'ancien (coût de la renégociation basé sur la création d'un VC);
- changement de VC dans le cas ou différents VC avec différentes QoS sont initialement créés (coût de la renégociation basé sur le changement de VC).

QoS homogène

Le modèle RSVP est orienté récepteurs et permet a chacun de ceux-ci de demander une réservation adaptée a leur besoins, de manière indépendante les uns des autres. La construction d'arbre de distribution avec des branches aux QoS différentes est donc implicite. Or, au niveau ATM, les interfaces UNI 3.x et 4.0 spécifient uniquement la création de VC multicast avec des QoS Homogènes. La création d'un VC multicast est séquentielle : création d'une branche initiale (VC point-à-multipoint) vers un premier récepteur avec une QoS donnée, puis ajout des autres branches vers les autres récepteurs avec la même QoS.

2.3 Dans quel environnement se placer ?

Nous présentons les bases sur les quelles nous développons des solutions pour résoudre les problèmes de multicast de RSVP sur ATM.

2.3.1 Architecture RSVP sur ATM

Le modèle RSVP utilisé est semblable à celui sur IP, à ceci près que la gestion des réservations est faite au niveau d'ATM. Le classificateur de paquets fait correspondre les paquets entrants à leurs VC de sortie aux lieux de les affecter à des files d'attentes. L'ordonnanceur de paquets s'occupe du partage des VC selon le style de réservation choisi. La structure d'un élément du réseau avec RSVP est représentée dans la figure 2.5 (Le trafic IP normal et donc la signalisation IP ne sont pas représentés dans la figure 2.5).

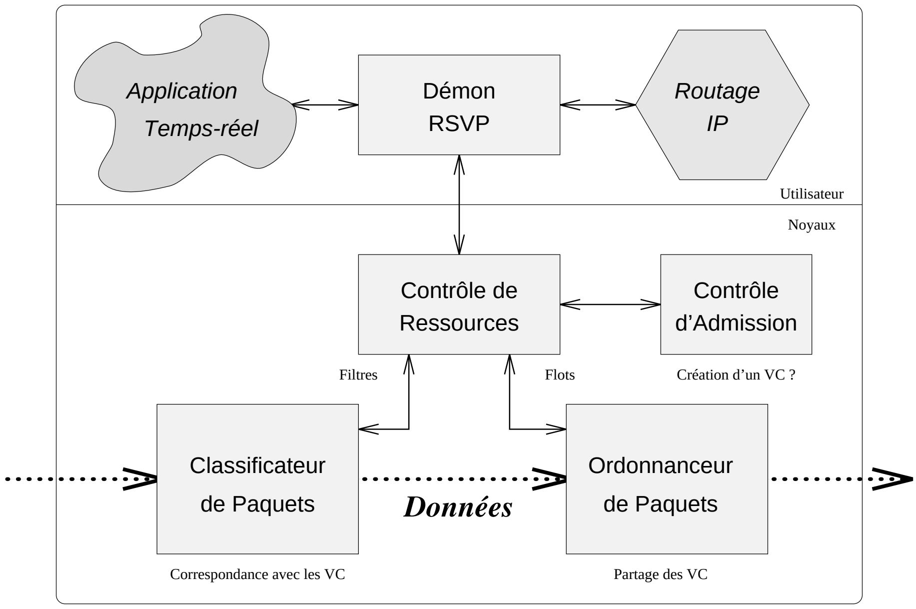

Figure 2.5: RSVP dans un hôte ou un routeur

2.3.2 classical IP

Pour commencer, nous nous plaçons dans un modèle simple : le modèle "classical IP over ATM" avec un réseau ATM segmenté en plusieurs LIS. Dans chaque LIS, les services de résolution d'adresses sont gérés par ATMARP et MARS (pour les adresses multicast, on peut considérer qu'un cluster correspond à un LIS). Le réseau ATM que nous utilisons peut n'être qu'un élément intermédiaire dans une topologie IP plus vaste : les éléments en bordure du réseau ATM peuvent être terminaux (hôtes directement rattachés à ATM) ou intermédiaires (éléments d'accès au réseau ATM : routeurs entrants ou sortants). Tous les éléments de la partie ATM (en bordure et internes) traitent RSVP.

La configuration de la figure 2.6 nous sert d'exemple. Elle est composée de deux éléments entrants (A et B) et de deux éléments sortants (C et D). Ces éléments peuvent être terminateurs ou non, nous les appelons respectivement émetteurs et récepteurs. La partie ATM représentée est composée de trois LIS reliés entre eux par des routeurs (R1 et R2) gérant RSVP.

L'intérêt de ce modèle est de représenter simplement une topologie multipoint-à-multipoint et d'étudier quelles réservations faire selon leur style et la demande des récepteurs. C'est ce modèle qui à incité l'utilisation des shortcuts.

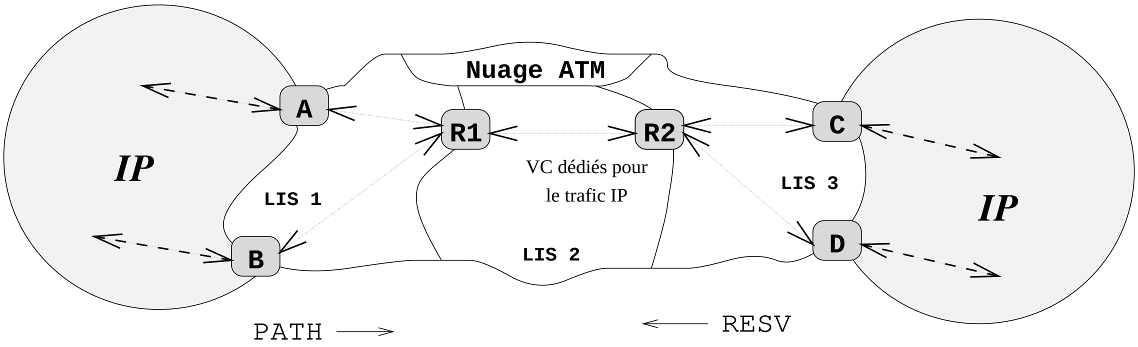

Figure 2.6: Modèle de base

2.3.3 Connectivité ATM globale

Il faut aussi envisager l'utilisation optimum de la connectivité du réseau ATM et la topologie qui en découle (absence de division en LIS). Les seuls éléments à gérer sont des routeurs, hôtes entrants ou sortants sur le schéma 1 vers N. Les agrégations ont lieu au niveau des flots arrivant dans le routeur entrant et sont transmis aux N routeurs sortants en un saut au niveau IP à travers ATM.

La figure 2.7 nous montre deux éléments entrants (A et B) et deux éléments sortants (C et D). Ces éléments peuvent être terminateurs ou non, nous les appelons respectivement émetteurs et récepteurs. Tous les membres attachés au réseau ATM global sont directement accessibles. Les shortcuts ne sont plus nécessaires, ils sont intégrés au niveau routage. Le protocole ND d'IPv6 peut, par exemple, gérer ces hôtes.

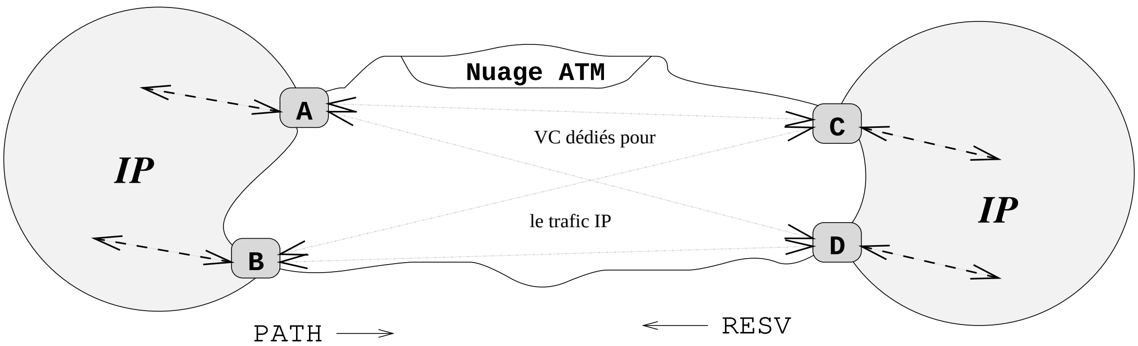

Figure 2.7: Modèle à connectivité globale

C'est vers ce genre de solution que s'oriente la gestion de la connectivité ATM à travers IP, rendant par la même la gestion des shortcuts au niveau RSVP obsolète. Nous nous plaçons dans la suite du document sur un modèle avec LIS, ce modèle étant facilement simplifiable à un modèle à connectivité directe.

2.3.4 Modèle serveur sous classical IP

Les fonctionnalités de RSVP sont de niveau 3. Pour pouvoir les intégrer dans un réseau de niveau 2 comme ATM, il faut créer des entités qui assurent cette remontée protocolaire. Dans le modèle RSVP sur ATM avec des LIS, les fonctionnalités RSVP sont assurées par routeurs inter-LIS qui ont assuré la remontée des paquets au niveau IP pour pouvoir traiter RSVP. Dans le modèle serveur, ce sont des entitées indépendantes qui gèrent les fonctions RSVP.

Nous nous intéressons à un modèle du même type que celui proposé dans [9]. Le serveur est jumelé avec un MARS pour résoudre le multicast. Il peut y avoir plusieurs serveur dans un LIS ou dans un cluster selon le niveau sur lequel on se place. Les hôtes en bordure du nuage ATM sont configurés pour accéder à un serveur proche. Ils envoient les requêtes RSVP vers le serveur qui va gérer la création des VC correspondant aux réservations

La figure 2.8 nous montre deux éléments entrants (A et B) et de deux éléments sortants (C et D). Ces éléments peuvent être terminateurs ou non, nous les appelons respectivement émetteurs et récepteurs. Nous avons intégré un serveur RSVP (S.R.) par LIS. Les connections avec les R.S. représentent les liens pour la signalisation. Les messages PATH et RESV vont passer sur le réseau des serveurs RSVP.

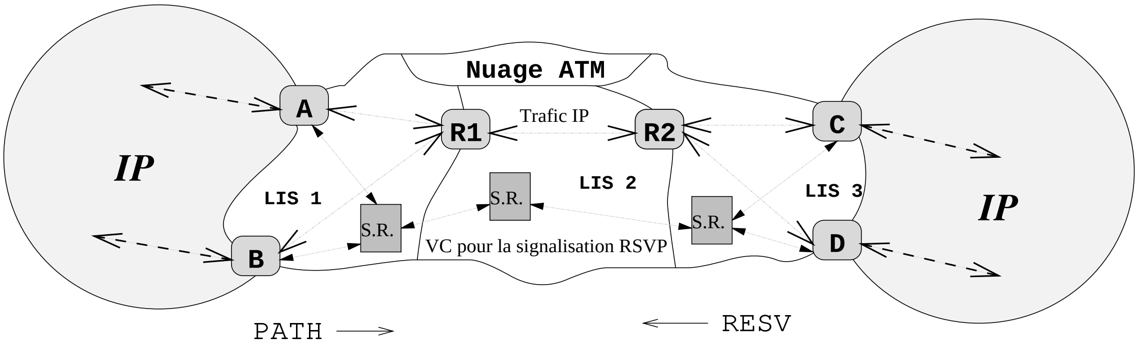

Figure 2.8: Modèle serveur avec LIS

Différents mécanismes sont à mettre en œuvre pour gérer le serveur et ses communications avec les autres serveurs tel que : la réplication (sécurité), détection, reconfiguration, élection et diffusion de l'adresse du serveur ; le partage de charge ; l'accès direct aux serveurs des autres LIS sans passer par un routeur intermédiaire (dans ce modèle on dissocie complètement le flux IP de celui des serveurs RSVP).

Ce modèle nécessite la mise en place d'un protocole assez lourd qui n'est peut être pas justifié dans un modèle avec LIS ou les fonctionnalités RSVP peuvent être facilement intégrées dans les routeurs.

2.3.5 Modèle serveur en connectivité globale

Le modèle serveur en cas de connectivité globale permet d'intégrer des éléments intermédiaires pour obtenir les fonctionnalités RSVP dans le nuage ATM.

La figure 2.9 nous montre deux éléments entrants (A et B) et deux éléments sortants (C et D). Ces éléments peuvent être terminateurs ou non, nous les appelons respectivement émetteurs et récepteurs. Tous les membres attachés au réseau ATM global sont directement accessibles. Nous introduisons une structure de serveurs RSVP (S.R.) composée ici de deux serveurs. Les liens sur les S.R. sont utilisés pour la signalisation RSVP séparée du flot IP normal.

Nous pouvons faire les mêmes remarques sur la mise en œuvre que précédemment. Par contre, la nécessité d'utiliser des serveurs en connectivité globale est justifiée car ici on n'a plus d'élément intermédiaire comme dans le modèle précédent. Une structure hiérarchisée avec plusieurs serveurs pourra en plus d'intégrer les fonctionnalités RSVP éviter des problèmes de congestion.

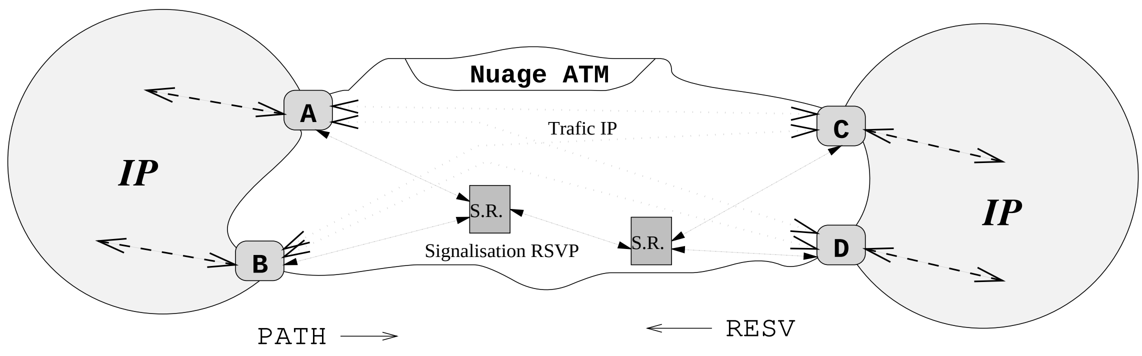

Figure 2.9: Modèle serveur en connectivité globale

2.3.6 UNI 3.x ou 4.0

L'interface UNI 3.x est actuellement disponible sur le matériel ATM. L'interface 4.0 n'est qu'une évolution dont l'avancée nous concernant le plus est le LIJ. Cette fonctionnalité ne remet pas en cause les mécanismes que nous mettons en œuvre sous UNI 3.x. En effet, le LIJ évite de faire remonter une demande de création de VC jusqu'à la racine, mais à notre niveau, pour la signalisation, les demandes de signalisation remontent l'arbre multicast pour pouvoir diffuser les messages de réservations en amont.

2.4 Réservations de VC ATM selon les Styles RSVP

Dans notre modèle RSVP sur ATM, les réservations correspondent à des VC. Nous étudions comment les agrégations se partagent les VC.

Les styles indiquent le type d'agrégation possible entre les flots d'une session[2]. Les filtres[3] sont utilisés pour attribuer les paquets des trafics émetteurs à leurs flots RSVP respectifs dans une session. Dans le modèle RSVP sur ATM, on fait correspondre un VC pour un flot.

2.4.1 Style "Wildcard-Filter"

Une réservation est partagée par un groupe d'émetteurs (la réservation correspond à la demande la plus importante).

Représentation : **WF**$(*\mathcal{Q})$

Astérisque pour l'émetteur quelconque et $\mathcal{Q}$ pour le flowspec demandé.

Dans la figure 2.10, nous représentons une session avec M émetteurs. Le style de la réservation est WF, l'agrégation des flots produit un flot avec pour réservation le "maximum" des réservations demandé par le récepteur. Au niveau ATM cela se traduit par un VC avec la QoS "maximum" demandée par le récepteur. Le changement du maximum (nouvelles réservations, arrivées ou départs de récepteur ou d'émetteur) peut amener à renégocier ce VC.

2.4.2 Style "Shared-Explicit"

Une réservation est partagée par un groupe d'émetteurs mais les émetteurs sont déclarés explicitement (la réservation correspond à la demande la plus importante).

[2]Session = (Adresse destination IP, Protocole, Port destination)

[3]Filtre = (Adresse source IP, Port source)

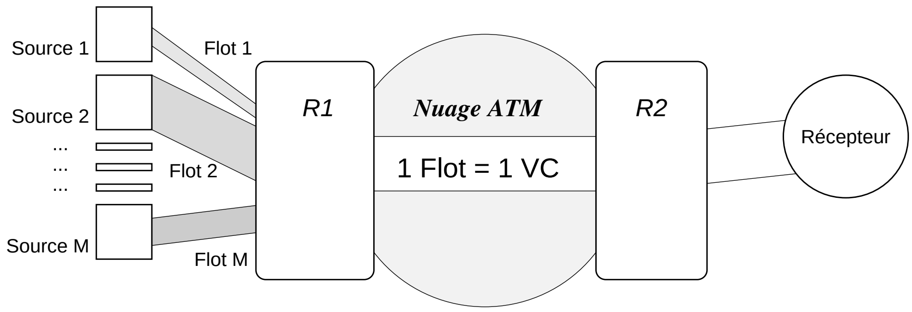

Figure 2.10: VC pour le style Wildcard-Filter

Représentation : **SE**$((S_1, S_2, ...)\mathcal{Q})$
S_n les émetteurs nommés et $\mathcal{Q}$ pour le flowspec demandé.

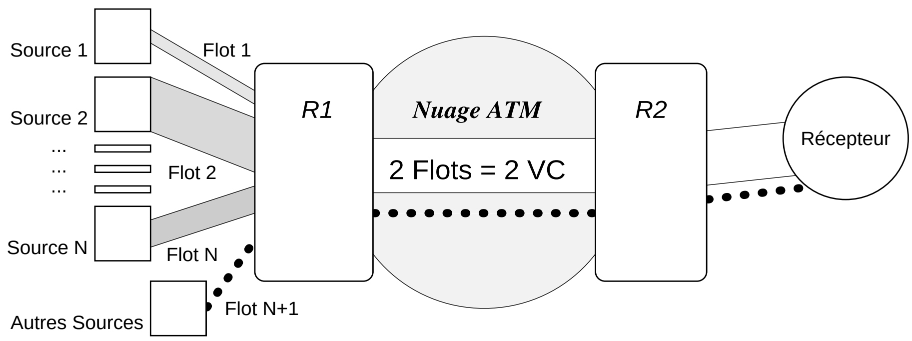

Figure 2.11: VC pour le style Shared-Explicit

Dans la figure 2.11, nous représentons une session avec M émetteurs. Le style de la réservation est SE, l'agrégation des flots se produit entre les N flots (N $\leq$ M) explicitement sélectionnés dans le filtre, donnant un flot avec pour réservation le "maximum" des réservations demandées par le récepteur par rapport à ces N émetteurs. Les autres flots sont agrégés dans un flot à la QoS non spécifiée. Au niveau ATM cela se traduit par deux VC : un VC avec la QoS "maximum" demandée par le récepteur par rapport aux N émetteurs choisis et un flot de type UBR sans QoS pour les paquets des autres émetteurs (en pointillé sur la figure 2.11. Le changement du maximum des N émetteurs (nouvelles réservations, arrivées ou départs de récepteur ou d'émetteur) peut amener à renégocier le VC avec QoS.

2.4.3 Style "Fixed-Filter"

Une réservation distincte par émetteur (pas de partage, la réservation correspond à la somme des demandes).
Représentation : **FF**$(S_1\mathcal{Q}_1)$, **FF**$(S_2\mathcal{Q}_2)$, ...
S_n les émetteurs nommés et $\mathcal{Q}_n$ pour le flowspec correspondant.

Dans la figure 2.12, nous représentons une session avec M émetteurs. Le style de la réservation est FF, il n'y a pas d'agrégation entre les N flots (N $\leq$ M) explicitement sélectionnés dans le filtre, ce qui produit N réservations distinctes. Les autres flots sont agrégés dans un flot à la QoS non spécifiée. Au niveau ATM cela ce traduit par N+1 VC : N VC avec les QoS demandées par le récepteur respectivement pour N émetteurs choisis et un flot de type UBR sans QoS pour les paquets des autres émetteurs (en pointillé sur la figure 2.12). Les changements parmi les N

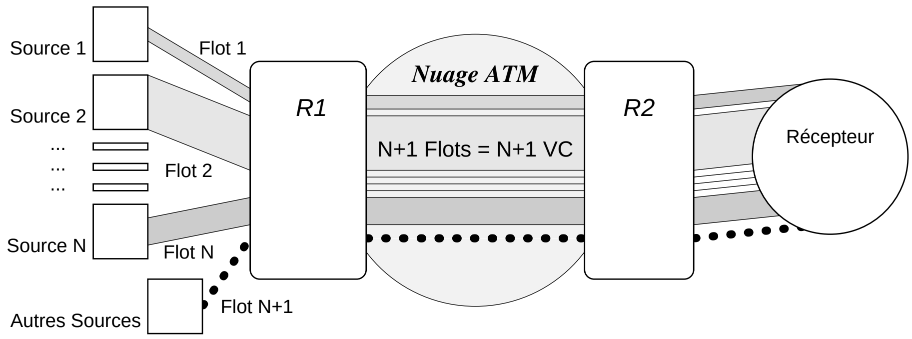

Figure 2.12: VC pour le style Fixed-Filter

émetteurs (nouvelles réservations, arrivées ou départs de récepteur ou d'émetteur) peuvent amener à ajouter, détruire ou renégocier des VC avec QoS.

2.4.4 Sans Style

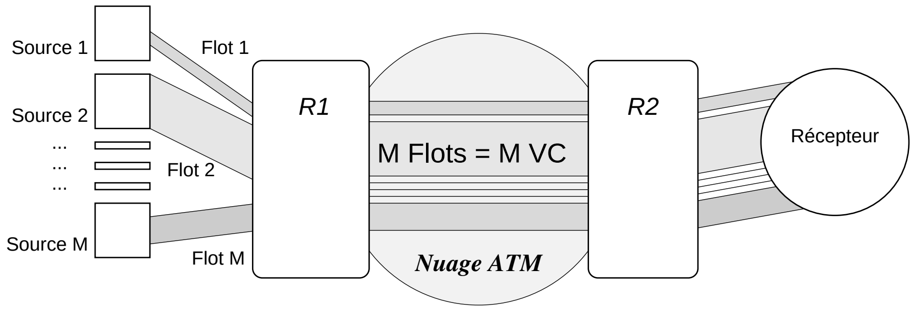

Figure 2.13: VC sans style

Dans la figure 2.13, nous représentons une session avec M émetteurs. Il n'y a pas de style particulier pour la réservation. Il n'y a pas d'agrégation entre les M flots, ce qui produit M réservations distinctes. Au niveau ATM cela se traduit par M VC avec les QoS demandées par le récepteur pour les M émetteurs respectivement. Les changements parmi les M émetteurs (nouvelles réservations, arrivées ou départs de récepteur ou d'émetteur) peut amener à ajouter, détruire ou renégocier des VC.

Chapitre 3

Multicast asymétrique et dynamique

3.1 Granularité de service multicast

Dans le modèle RSVP, les réservations doivent permettre l'utilisation de connections multicast avec des branches aux réservations variées. L'adaptation de ce modèle à ATM pose le problème des VC multicast à QoS variées, sachant que les seuls VC multicast disponibles actuellement sont à QoS homogène (identiques) sur toutes les branches. Nous définissons la granularité de service multicast comme la capacité à proposer des QoS proches de celle demandée sur chaque branche. Nous étudions ici trois types de granularités dans un environnement utilisant UNI 3.x dans un nuage ATM segmenté en plusieurs LIS.

3.1.1 Granularité faible – Réservations homogènes

Le cas le plus simple préconise l'utilisation directe du multicast ATM. L'utilisation de réservations homogènes implique que nous ayons des récepteurs qui demandent les mêmes réservations ou que nous nous basions sur la QoS "maximum" de manière à satisfaire tous les récepteurs.

Asymétrie Nous utilisons le modèle multicast ATM, il n'y a donc pas d'asymétrie. Il faut mettre en place une politique pour évaluer la QoS globale allouée à toutes les branches.

Dynamicité Nous utilisons un VC multicast ATM, nous avons donc la dynamicité d'un VC multicast ATM. En cas de changement de QoS, il faut fermer et recréer tous les VC multicast après avoir calculé une nouvelle QoS. Suivant la politique mise en œuvre on peut laisser le VC multicast inchangé, si la branche concernée est toujours sur-dimensionnée. L'ajout et la suppression d'un récepteur acceptant la QoS choisie pour le premier VC multicast créé, est par contre dynamique.

Exemple Nous avons représenté les VC ainsi créés sur la figure 3.1 : il n'y a que des VC multicast *simplex* (même s'il n'y a qu'un nœud suivant, en prévision de l'ajout d'un nouveau successeur). Nous supposons qu'il y a une agrégation de type WF entre R1 et R2.

Avantages Une granularité de service multicast faible correspond à une grande simplicité et a une signalisation légère grâce à l'utilisation du multicast fourni par l'interface actuelle d'ATM.

Défauts La mise en place d'une politique pour trouver la QoS globale est difficile. Le modèle RSVP étant par nature dynamique, un élément intermédiaire reçoit des messages RESV à n'importe quel moment. Doit-il attendre un certain nombre de messages avant de réserver,

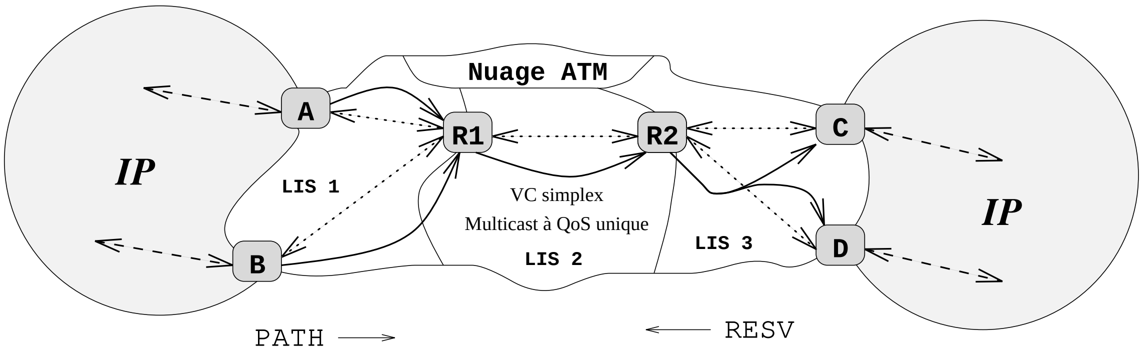

Figure 3.1: Réservation avec une QoS unique

doit-il se baser sur le premier message RESV, avec le risque d'initialiser un VC sous dimensionné ?

Toutes les branches de l'arbre multicast ne vont peut être pas accepter des VC avec cette QoS "maximum". Le résultat de la négociation à l'aide des messages PATH et RESV nous fournit la QoS à choisir : avec les messages PATH, chaque récepteur obtient la QoS demandable aux émetteurs majorée par celle que le chemin peut lui fournir. En retour le nœud concerné devrait allouer la QoS "maximum" demandée dans toutes les branches du VC multicast majorée par la plus petite QoS disponible sur ces branches,*or c'est une information dont il ne dispose pas.* Il est donc nécessaire d'allouer un VC multicast avec la QoS "minimum" pour s'assurer de la création complète de celui-ci.

Ce modèle de réservation par maximum a les mêmes défauts que l'agrégation WF : il faut renégocier tout le VC multicast si le maximum de QoS change. Le choix de cette solution dépend du nombre de récepteurs, de leur variété et de leur dynamicité. Nous pouvons aussi prévoir de sur-dimensionner les VC en prévision mais nous nous retrouvons dans un modèle à allocation statique avec une sur-allocation qui risque d'être très importante avec beaucoup de récepteurs.

Ce modèle de réservation, envisagé pour l'utilisation sous le multicast IP, ne satisfait ni la dynamicité ni l'hétérogénéité nécessaire au modèle RSVP.

3.1.2 Granularité fine – Réservations par VC unicast

Le modèle de réservation opposé au précédent est celui des réservations totalement hétérogènes : pour pouvoir s'adapter au plus près à chaque réservation, on réalise au niveau ATM un VC unicast avec la QoS correspondante pour chaque successeur du nœud concerné. Pour ce modèle, le passage des messages PATH et RESV nous donne toutes les informations pour allouer les VC.

Asymétrie Nous n'utilisons plus les VC multicast ATM, mais des VC point-à-point pour obtenir la meilleure granularité : la QoS est négociée sur la base un récepteur, une QoS. Chaque branche ayant la QoS désirée si elle est disponible.

Dynamicité Nous utilisons des VC point-à-point ATM, nous avons donc la dynamicité de ces VC ATM. Or les VC ATM sont avec une QoS fixée. Pour conserver notre QoS ajustée, il nous faut fermer l'ancien VC et en ouvrir un autre avec la QoS demandée. La changement est plus léger que précédemment, mais on est encore loin de la dynamicité du modèle RSVP qui est basé sur des changement de réservations internes au routeur et non sur des réservations de circuits comme sur ATM.

Exemple Dans la figure 3.2, nous avons représenté les VC unicast *duplex* avec QoS désirée, entre les nœuds successifs de l'exemple. Nous supposons qu'il y a une agrégation de type WF entre R1 et R2.

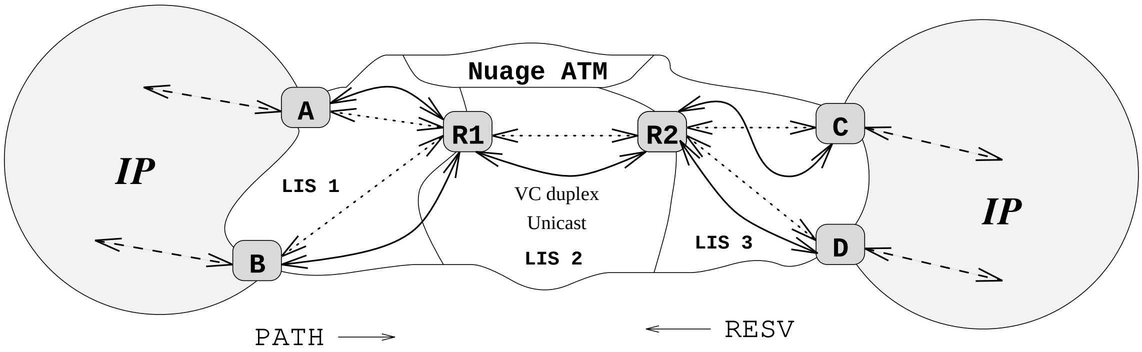

Figure 3.2: Réservation avec QoS multiples

Avantages L'intérêt de ce modèle, en plus d'être complètement asymétrique, est qu'en cas de renégociation, nous ne recréons que le VC unicast de la réservation qui a évolué.

Défauts Une granularité de service multicast élevée correspond à une signalisation élevée et une inadaptabilité au facteur de taille des groupes multicast. Un autre défaut de ce modèle est la sur-utilisation de ressources qu'entraine la création de N connections unicast par rapport à une connection *multicast* (mais cette ressource sera adaptée au mieux aux besoins de la session). De plus, nous définissons ici des VC bidirectionnels alors que les réservations RSVP sont unidirectionnelles (basées sur des flots simplex).

Ce modèle de réservation, même s'il est hétérogène, ne répond pas a la dynamicité nécessaire au modèle RSVP. De plus la perte des capacités multicast au niveau ATM empêche l'adaptation à des groupes multicast avec un nombre de récepteurs élevé.

3.1.3 Granularité intermédiaire – Réservations par multiple VC multicast

Les deux modèles de réservations précédents représentent les deux manières extrêmes d'adapter des réservations hétérogènes sur ATM. En observant qu'en général, les quantités de réservations demandées ont un nombre de niveaux discret ou que nous pouvons les classer en paliers, nous définissons un modèle mixte. Celui-ci combine plusieurs arbres multicast ATM homogènes avec des QoS correspondant à ces paliers.Une plus grande finesse d'adaptation à la réservation demandée est obtenue par rapport au modèle s'appuyant sur un arbre multicast.

Pour chaque niveau de QoS choisi, nous allons donc définir un arbre multicast, matérialisé par un VC multicast avec cette QoS. Un récepteur change dynamiquement d'arbre multicast selon la QoS dont il a besoin. Nous envisageons deux modèles :

modèle à flots séparés le récepteur utilise une branche sur l'arbre dans lequel se trouve la QoS qu'il demande (les récepteurs rejoignent l' arbre multicast avec la QoS recherchée);

modèle à codage hiérarchique le récepteur utilise une branche de **tous** les arbres dont il a besoin (les récepteurs rejoignent les N premiers arbres multicast pour réaliser le niveau de QoS voulu).

La gestion de ces arbres multicast doit être la plus transparente possible au niveau RSVP. Le protocole de routage multicast ainsi que le rattachement des hôtes à ces arbres garantissent leur construction dynamique.

Asymétrie La possibilité d'avoir plusieurs QoS est assurée par les différents niveaux de QoS des arbres multicast (et complémentaires si codage hiérarchique).

Dynamicité La dynamicité est assurée par la facilité de changement d'arbre multicast. Les arbres étant créés initialement, un récepteur peut demander son rattachement aux arbres qui correspondent à la QoS qu'il demande.

Exemple Dans la figure 3.3, nous avons représenté deux arbres multicast avec deux QoS différentes initiées par A et B vers les récepteurs. Les deux flots arrivent, au niveaux de C et D, à partir desquels l'on peut sélectionner la QoS demandée par les éléments en aval. Nous supposons qu'il y a une agrégation de type WF entre R1 et R2.

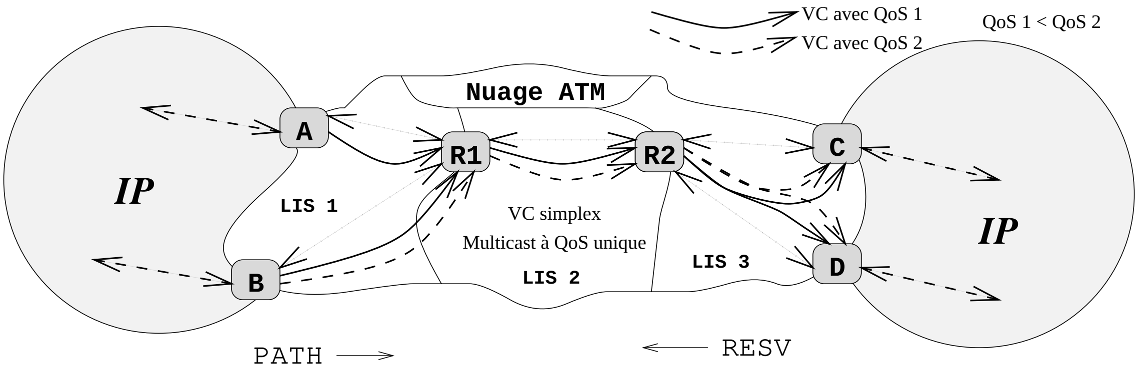

Figure 3.3: Réservation avec plusieurs VC multicast

Avantages La dynamicité et l'hétérogénéité se rapprochent de celles recherchées pour RSVP autant que pour les deux modèles précédents. Les mécanismes multicast du niveau ATM sont conservés. En gérant les rattachements aux arbres multicast à l'aide de gestion de groupe comme nous l'expliquons plus loin, nous restons transparents par rapport à RSVP, de manière à diminuer les modifications nécessaires.

Défauts Tous les niveaux de QoS sont disponibles, il faut faire des choix par rapport au nombre d'arbre et aux paramètres de QoS à leur attribuer. Pour les agrégations, il faut que les niveaux de QoS choisis correspondent entre les différents émetteurs. La gestion de plusieurs arbres multicast dans le nuage ATM entraine une surconsommation des ressources.

Ce modèle pallie les deux problèmes du multicast ATM . Le problème de la QoS homogène des branches des arbres multicast ATM en créant plusieurs arbres avec des QoS homogènes mais de différents niveaux entre les arbres. La dynamicité est alors gérée par changement d'arbre et correspond à celle attendue par RSVP. Nous allons étudier ces deux modèles en détail dans la suite.

3.2 Réservations hétérogènes par multiples VC multicast

Nous développons les solutions à granularité de multicast intermédiaire.

3.2.1 Mécanismes préalables

Nous définissons des mécanismes de base commun aux deux modèles.

Rôle de l'application

Pour garantir l'indépendance du modèle avec multiples arbres multicast par rapport aux applications, nous générons la séparation des flots au niveau de l'application elle-même. L'application doit pouvoir fournir en sortie le même flot avec des QoS différentes. Nous ne nous préoccupons pas de la manière de générer ces flots pour rester le plus générique possible. L'intelligence reste à l'extérieur du réseau pour être le plus générique possible et pouvoir s'adapter à tout type de trafic. Dans l'optique RSVP, l'application ne génère qu'un flot que l'on peut diviser par filtrage, mais cela implique de pouvoir filtrer au bon moment, sinon le flot unique risque d'être rejeté par un contrôle d'admission si on ne l'a pas divisé avant. Avec des flots divisés au niveau des applications, on peut envoyer progressivement les flots en commençant par les plus importants (plus petits en terme de ressources) pour leur assurer plus de chance d'avoir leurs réservations. De plus avec des flots séparés, le changement d'arbre pour les récepteur va pouvoir être gérer à l'aide d'adhésion à groupe.

Gestion de groupe

Pour avoir une solution simple à mettre en œuvre nous essayons d'utiliser au maximum les mécanismes RSVP tel qu'ils sont spécifiés actuellement. Or la gestion de plusieurs arbres multicast ne se fait pas dans RSVP. Nous pouvons la gérer indépendamment en associant chaque arbre à une adresse de groupe. il faut ensuite associer les QoS aux groupes, l'application n'ayant plus qu'à s'abonner aux groupes qui correspondent à la QoS demandée et gérer la synchronisation des différents flux reçus.

Création des arbres multicast

Si l'on peut conserver la transparence vis a vis de RSVP pour les arbres multicast à branches hétérogènes, il n'en va pas de même pour la dynamicité. Le modèle conventionnel RSVP sur ATM associe les VC au réservation et donc crée les VC au moment de la réservation. Si nous supposons que les VC multicast sont crées avant la demande de réservation, cela permet les changements de VC sans attente de création de circuit. Cette modification par rapport à la gestion habituelle RSVP sur ATM est étudiée dans la partie suivante.

3.2.2 Multiples VC multicast à flots séparés

L'application génère autant de flots que de niveaux de QoS désirés. Chaque flot a une QoS différente (par exemple du son, avec un de flot qualité téléphonique et un autre de qualité Hi-Fi) et est indépendant. Les différents flots sont routés à travers le réseau vers les destinataires qui demandent la QoS du flot. Pour atteindre les récepteurs, nous assignons une adresse de groupe multicast par flot. Cela permet au protocole de routage multicast d'ajuster les arbres multicast au mieux selon la répartition des récepteurs.

Sur la figure 3.4, l'émetteur A envoie un flot FA1 avec la QoS 1 et l'émetteur B, qui génère un flot avec une meilleure QoS, envoie deux flots FB1 et FB2, avec respectivement la QoS 1 et la QoS 2 (Le flot FB1 transporte les mêmes données que FB2 mais avec la QoS 1, plus faible que la QoS 2). Ces flots sont agrégés dans le routeur R1 et sont transmis dans deux VC multicast assurant les QoS 1 et 2 vers R2, Les récepteurs C et D reçoivent les deux flots : deux VC multicast

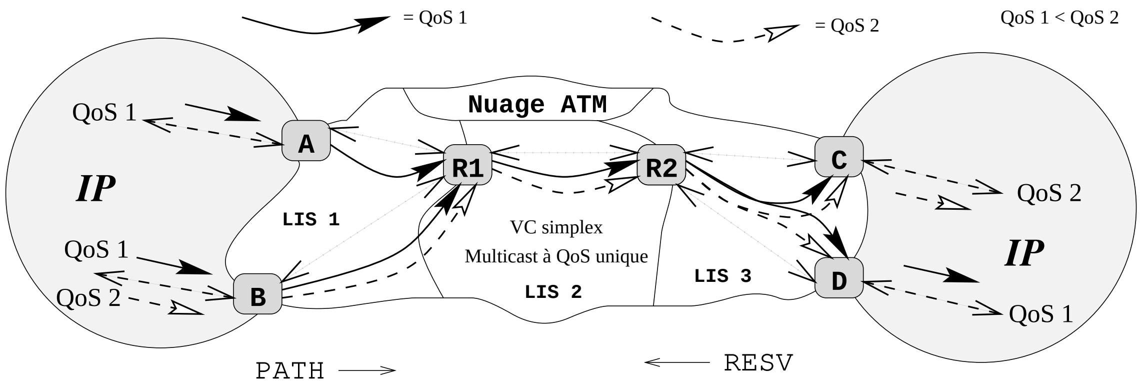

Figure 3.4: Réservation avec des VC multicast à flots séparés

avec les deux QoS vont vers C et D. Les applications peuvent récupérer au niveau de C et D l'une ou l'autre des QoS en adhérant au groupe de FA1 et FB1 pour la QoS 1 ou au groupe de FB2 pour la QoS 2 Nous supposons une agrégation de type WF dans les routeurs.

Un problème apparait au niveau de l'agrégation : si des applications gérant des flots de données de différents niveaux de QoS veulent utiliser un style partage de réservations, comment peut-on traiter l'agrégation sachant que pour gérer les différents niveaux de QoS, les niveaux sont associés à des adresses destinations différentes (adresses de groupe). Or l'agrégation ne peut avoir lieu que dans une même session dans RSVP. Par exemple, l'élément intermédiaire R1 reçoit deux flots de données FB1 et FB2 (issu du même flot généré par une application, mais avec les QoS respective QoS 1 et QoS 2, et la QoS 1 inférieure à la QoS 2), puis un émetteur lui envoie un autre flot FA1 de QoS 1 uniquement. Tous ces flots sont sensés appartenir à la même session RSVP, mais pour gérer nos arbres multiples nous dissocions leurs adresses destinations en leur assignant différentes adresses de groupes. Nous pouvons agréger FA1 avec FB1 qui possèdent la même QoS, mais pas FA1 avec FB2 pour les récepteurs qui réclament la QoS 2 et implicitement les QoS inférieures. Il faut donc être capable de gérer l'agrégation de flots **inter-session**.Il faut modifier le modèle de RSVP ou propager les flots de QoS inférieure avec le flot de QoS demandée. donc l'applications réceptrice doit s'abonner à tous les groupe de QoS inférieure ou égale à celle qu'elle demande, à la manière de l'abonnement dans le codage hiérarchique que nous allons voir maintenant, excepté que l'on évite la sur allocation de réservation.

3.2.3 Multiples VC multicast à codage hiérarchique

L'application doit pouvoir fournir en sortie un flot découpé en plusieurs autres avec des QoS moindres selon le principe du codage hiérarchique. Nous ne nous préoccupons pas de l'obtention de ces flots pour rester le plus générique possible.

Les différents flots sont routés à travers le réseau vers les destinataires qui en ont besoin. Pour atteindre les récepteurs, nous assignons une adresse de groupe multicast par flots. Cela permet au protocole de routage multicast d'ajuster les arbres multicast au mieux selon la répartition des récepteurs à la sortie du nuage ATM. Pour une QoS donnée, un récepteur peut avoir besoin de N flots. A la réception, l'application doit reconstruire le flot avec la QoS demandée selon le principe du codage hiérarchique utilisé, avec les N flots obtenus en s'abonnant aux N groupes correspondants.

Au niveau de l'agrégation, le procédé est transparent pour RSVP étant donné qu'il a lieu sur chaque sous-flot de manière indépendante.

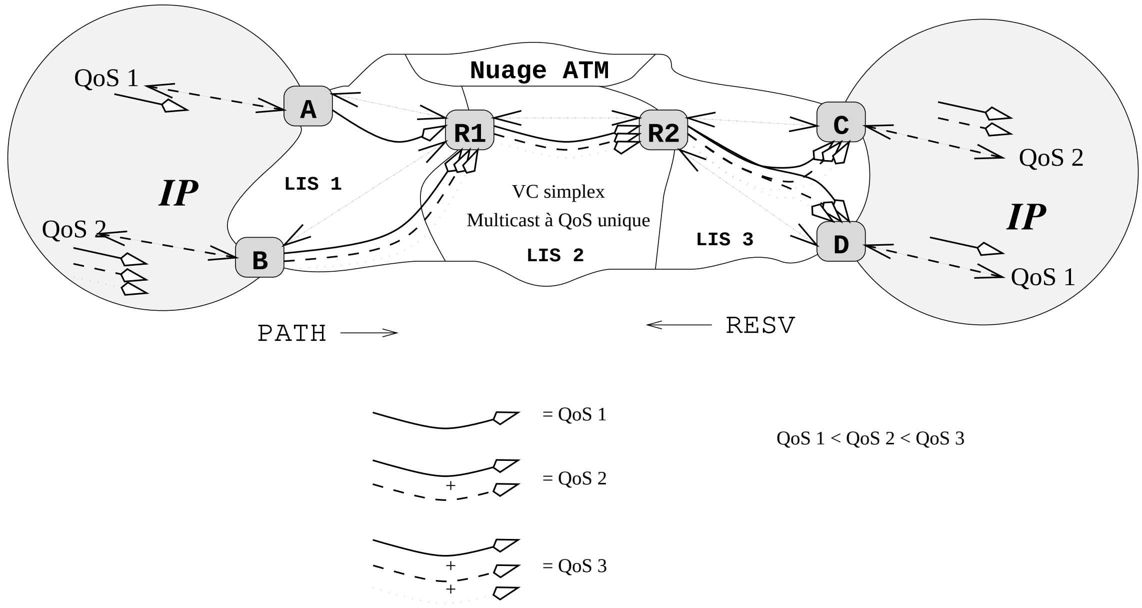

Figure 3.5: Réservation avec des VC multicast à codage hiérarchique

Sur la figure 3.5, l'émetteur B génère un flot avec une QoS 3 max qu'il hiérarchise en 3 flots de QoS moindre : FB1, FB2 et FB3 (en utilisant les trois flots, la QoS 3 est obtenue ; avec FB1 et FB2, la QoS 2 ; et avec FA1 seulement, la QoS 1). B envoie ces trois flots vers R1. L'émetteur A envoie un flot FA1 avec la QoS 1. Les deux flots avec la même QoS, FA1 et FB1, appartenant à la même session, sont agrégés dans le routeur R1 dans un VC de QoS 1 et sont transmis vers R2 (nous supposons ici une agrégation de type WF). Ce flot ainsi que FB2 et FB3 sont transmis vers R2 sur trois VC multicast. Les récepteurs C et D peuvent fournir la QoS 3 grâce aux VC multicast originaires de R2. Sur la figure 3.5, C transmet vers l'aval le flot constitué de l'agrégation de FA1 et FB1 pour générer un flot de QoS 1 ; D transmet les flots pour former des flots de QoS 2 dans les applications. Il y a une sur-allocation au niveau du nuage ATM, mais elle est nécessaire pour permettre à D de fournir un les flots pour la QoS 3 dynamiquement.

3.3 Mise en œuvre de multiple VC multicast

La mise en œuvre de multiple VC multicast doit avoir un minimum d'impact sur l'implantation de RSVP sur ATM. Les modifications minima que nous avons relevées dans ce qui précède concernent la création des VC sur les message PATH pour les créer avant la demande de réservation. ATM étant basé sur un état global statique (Hard State) alors que RSVP met en œuvre un état global dynamique ou “mou” (Soft State) pour ces réservations. Notre politique est de conserver la dynamicité du modèle RSVP au niveau des réservations tout en gérant le niveau ATM de manière adaptée, en prévoyant le comportement des réservations.

3.3.1 Initialisation des multiples arbres

Les arbres multicast doivent être crées le plus vite possible, dès l'arrivée des messages PATH dans un routeur frontière, de manière à mettre la ressource disponible pour les retours de messages RESV.

Pour que les arbres se construisent au maximum dans le nuage ATM, il faut que les applications adhérent à toutes les adresses de groupe correspondant aux QoS dont elle a potentiellement besoin. De cette manière, les messages PATH vont créer les arbres que pourra utiliser cette application dans le nuage ATM. Cela n'implique pas une utilisation de ces ressources mais permet de les utiliser rapidement si besoin est.

Un autre problème à régler est le passage en trafic IP normal des données ne passant pas les filtres des réservations. Or dans notre cas nous nous abonnons pour créer les arbres dans le nuage ATM, pas pour recevoir systématiquement toutes les données par le canal IP. Donc au niveau ATM, nous ne transmettons plus les données ne correspondant pas à une réservation explicite.

Au niveau de l'application, une politique d'envoi des messages PATH correspondant au partie du flot la plus petite en premier est à mettre en œuvre pour assurer le passage d'un minimum d'abord (un VC de taille supérieur peut être rejeté), le reste du flots ensuite si l'on peut créer les autres VC.

Les niveaux de QoS générés par les applications doivent être identiques entre toutes les applications qui ont l'intention de faire des agrégations ensuite.

3.3.2 Utilisation des multiples arbres

Les récepteurs sélectionnent le niveau de QoS désiré en adhérant ou en quittant les groupes multicast correspondants. Si de nouveaux récepteurs se joignent ou quittent la session, cela modifie le routage dans le nuage ATM, les tables de MARS changent et les messages PATH sont diffusés vers le nouveau groupe de récepteurs en même temps que l'on créer ou détruit des VC multicast.

Les messages PATH et RESV sont gérés de manière normale à l'extérieur du nuage ATM comme l'intérieur, les messages utilisant les circuits réservés à IP (non fiable).

3.3.3 Fermeture des multiples arbres

Ne pas fermer les VC à cause d'un arrêt de réservation, la ressource doit rester disponible pour une autre réservation. On ne peut non plus fermer les VC sur demande explicite des récepteurs pour pouvoir s'adapter aux groupes multicast. Il faut mettre en œuvre une gestion basée sur les messages PATH. Les arbres sont créés par rapport aux groupes auxquels ont adhéré les récepteurs et ceux-ci sont répercutés dans la base de MARS. C'est par rapport à cette table que l'on sait s'il y a encore des membres concernés par un arbre multicast.

3.4 Validation des modèles

3.4.1 Implémentation

Type de modèle à valider

Nous validons le modèle à partir des plate-formes ATM existantes. Notre modèle fonctionne sur les normes existantes :

- Interface ATM UNI 3.1
- Utilisation de "Classical IP over ATM"
- Utilisation de MARS pour la résolution du multicast
- Structuration du nuage ATM en LIS
- IPv4 avec multicast en dehors du nuage ATM

Ce dont on a besoin

Pour expérimenter RSVP sur ATM sur la plate-forme précédente, il nous faut :

- RSVP pour IP.
- Classificateur de paquets.
- Ordonnanceur de paquet pour IP sur réseau à diffusion (l'ordonnanceur pour un lien Ethernet doit s'occuper de la gestion de la QoS pour laquelle les niveaux inférieurs ne garantissent rien).
- Ordonnanceur de paquet pour ATM (l'ordonnanceur pour ATM s'appuie sur une gestion de la QoS au niveau inférieur et a juste à répartir les paquets entre les différents VC).
- Contrôleur d'admission.

Pour les modèle avec des arbres multiples, nous avons besoin :

- Applications produisant des données sous formes de plusieurs flots indépendants appartenant à des sessions différentes avec des QoS variées.
- Applications produisant des données sous formes de plusieurs sous-flots appartenant à des sessions différentes et capable de reconstruire le flot initial pour le codage hiérarchique.
- RSVP pour IP modifié pour ATM.

3.4.2 Moyens disponibles

Nous présentons ici les moyens matériels et logiciels disponible pour réaliser des expérimentations.

Niveau laboratoire (Matériel)

Structure de test envisageable au sein du MASI. Deux types de réseaux sont accessible localement : Ethernet et ATM. Trois types de machines sont utilisés : Station Sun sous SunOS ou Solaris et PC sous Linux. Exemple avec la figure 3.6.

Problèmes à résoudre avant d'installer des modules pour une architecture RSVP :

- Configuration du réseau IP en multicast à compléter.
- Configuration du réseau ATM en deux LIS
- Plusieurs architectures matérielles : les programmes doivent fonctionner sur SunOS, Solaris et Linux.

Niveau communauté Internet (Logiciel)

Impléméntation de la Distribution ISI de RSVP selon les spécification du draft 12. Seulement disponible pour Solaris et seulement la partie RSVP (signalisation) sont disponibles. Des architectures complètes RSVP ont été testées au niveau universitaire, des classificateurs et des ordonnanceurs de paquets sont certainement disponibles en domaine public. Vérifier l'existence d'expérimentation de RSVP sur ATM et voir quels produits de mappage de réservation sur des VC existent.

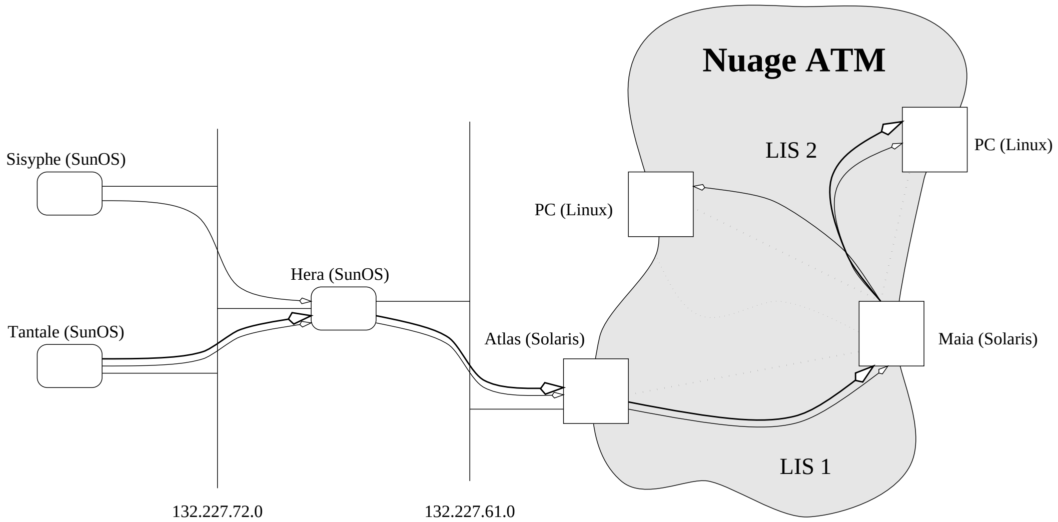

Figure 3.6: Expérimentation au MASI

3.4.3 Réalisations futures

Niveau IP pur

- Installations d'une plate-forme RSVP sur IP complète.
- Test des modèles avec plusieurs arbres sur ce modèle (gestion de la QoS avec l'appartenance à un groupe multicast).
- Créations de flots et réassemblage au niveau des applications.

Niveau IP sur ATM

- Test des fonctionnalités IP sur ATM et de MARS.
- Extension de la plate-forme RSVP à ATM.
- Modification de RSVP sur ATM
- Validation des modèles avec plusieurs arbres multicast.

3.5 Evolutions des modèles à arbres multiples

Les modèles définis précédemment par rapport aux moyens disponibles actuellement sont-ils évolutifs, s'adaptent-ils aux évolutions des protocoles et interfaces en cours de développement ?

3.5.1 Arbres multiples et IPv6

Adaptation transparente, on n'utilise que des mécanismes standard identiques sur les deux versions. On n'utilise pas les mécanismes de résolution d'adresse et on ne manipule pas directement les nouvelles adresses.

3.5.2 Arbres multiples et UNI 4.0

Nous construisons les arbres multicast à partir des émetteurs, au passage des messages PATH, les nouvelles fonctionnalités tel que le LIJ ne nous intéressent pas.

Chapitre 4

conclusion

Les modèles basés sur plusieurs arbres *multicast* pour gérer les flots 1 vers N dynamiquement avec des QoS différentes pour les N récepteurs, permettent d'éviter les granularités de contrôle extrême.

Des voies à explorer :

- Interaction des modèles avec la topologie (répartition dense ou dispersée), des protocoles de routages tels que PIM peuvent prendre en compte une partie de ces problèmes.
- Rapport de coût de la renégociation VC *multicast* par rapport au VC unicast, rapport de coût des réservations *multicast* par rapport aux multi-réservations unicast, sont à connaitre pour évaluer les différentes solutions.
- Les mécanismes que nous développons ne sont-t-ils pas du ressort du protocole de routage avec QoS, car nous pallions des manques du niveau deux à l'aide de la signalisation de niveau RSVP et des mécanismes de gestion de groupe de niveau IP.

Bibliographie

[1] L. Zhang, S. Deering, D. Estrin, S. Shenker & D. Zappala. *RSVP : A New Resource ReSerVation Protocol*. IEEE Networks, Septembre 1993.

[2] R. Braden, D. Clark & S. Shenker. *Integrated Services in the Internet Architecture : an Overview*. RFC 1633, juillet 1994.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog & S. Jasmin. *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*. Internet Draft, draft-ietf-rsvp-spec-12, Mai 1996.

[4] S. Deering, *Host Extensions for IP Multicasting.* RFC 1112, Stanford University, Août 1989.

[5] M. Laubach. *Classical IP and ARP over ATM*. RFC 1577, Hewlett-Packard Laboratories, Décembre 1993.

[6] G. Armitage. *Support for Mulicast over UNI 3.0/3.1 based ATM Network*. Internet Draft, draft-ietf-ipatm-ipmc-12, 6 mai 1996.

[7] A. Birman, R. Guerin & D. Kandlur . *Support for RSVP-based Service over an ATM Network*. Internet Draft, draft-birman-ipatm-rsvpatm-00, 22 fevrier 1996.

[8] F. Baker, S. Berson, M. Borden, E. Crawley & J. Krawczyk. *Issues for RSVP and Integrated Services over ATM*. Internet Draft, draft-crawley-rsvp-over-atm, 22 fevrier 1996.

[9] R. O. Onvural & V. Srinivasan. *A Framework for Supporting RSVP Flows Over ATM Networks*. Internet Draft, draft-onvural-rsvp-atm, 26 fevrier 1996.

[10] R. Cole, D. Shur & C. Villamizar. *IP over ATM : A Framework Document*. RFC 1932, Avril 1996.

[11] M. Laubach & J. Halpern. *Classical IP and ARP over ATM*. Internet Draft, draft-ietf-ion-classic2-00, 7 juin 1996.

[12] P. Schulter. *A Framework for IPv6 over ATM*. Internet Draft, draft-schulter-ipv6atm-framework-01, 12 février 1996.

[13] G. Armitage. *IPv6 and Neighbour Discovery over ATM*. Internet Draft, draft-ietf-ion-ipv6nd-00, 13 juin 1996.

[14] D. Katz, D. Piscitello, B. Cole & J. Luciani. *NBMA Next Hop Resolution Protocol (NHRP)*. Internet Draft, draft-ietf-rolc-nhrp-09, 18 juillet 1996.

[15] C. Huitema. *IPv6 : The New Internet Protocol*. Prentice Hall, 1996.

[16] Leslie Lamport. *LaTeX. A Document Preparation System*. Addison-Wesley, September 1986.