Déploiement de service multicast dans des environnements hétérogènes

Fethi Filali

UNIVERSITÉ DE NICE SOPHIA-ANTIPOLIS - UFR SCIENCES

École Doctorale STIC

THÈSE

Présentée pour obtenir le titre de :
Docteur en SCIENCES
de l'Université de Nice Sophia-Antipolis

Spécialité : Informatique

par

Fethi FILALI

Déploiement de service multicast dans des environnements hétérogènes

Soutenue publiquement le 27 Novembre 2002 devant le jury composé de :

Président :	Jean-Paul	RIGAULT	UNSA
Rapporteurs :	Horst	CLAUSEN	Université de Salzburg
	Jean-Jacques	PANSIOT	ULP Strasbourg
Examinateurs :	Ernst	BIERSACK	Eurécom
	Benoît	GARNIER	Alcatel Space Industries
	Farouk	KAMOUN	ENSI
Directeur de thèse :	Walid	DABBOUS	INRIA

(15:00 - INRIA Sophia-Antipolis)

Déploiement de Service Multicast dans des Environnements Hétérogènes

Fethi FILALI

Multicast Service Deployment in Heterogeneous Environments

Dédicaces

À ma famille ...

Remerciements

Je voudrais tout d'abord remercier M. Jean-Jacques Pansiot, Professeur à l'Université Louis Pasteur de Strasbourg, et M. Horst Clausen, Professeur à l'Université de Salzburg, qui m'ont fait l'honneur d'avoir accepté d'être les rapporteurs de ma thèse.

J'exprime ma profonde gratitude à M. Walid Dabbous, Directeur de Recherche et chef du projet Planète à l'INRIA Sophia-Antipolis, pour m'avoir proposé ce sujet, pour son assistance lors de la réalisation de ce travail, pour sa disponibilité qui m'a permis de mener à terme cette thèse. La liberté qu'il m'a accordée et les responsabilités qu'il m'a confiées ont beaucoup contribué à la formation de ma personnalité et à mon autonomie dans le travail.

Je tiens à remercier M. Jean-Paul Rigault, Professeur à l'ESSI, pour avoir accepté de présider mon jury.

Mes remerciements les plus vifs vont aussi à M. Ernst Biersack, Professeur à l'Eurécom, et M. Benoît Garnier, Ingénieur de recherche à Alcatel Space Industries, qui se sont intéressés à mes travaux et ont bien voulu participer à mon jury de thèse.

Un grand merci à M. Farouk Kamoun, Professeur à l'ENSI, qui n'a pas cessé de m'encourager et de me supporter tout le long de mes études. Ses conseils et suggestions m'ont été très bénéfiques. Je le remercie d'avoir accepté l'invitation pour assister à ma soutenance de thèse et participer au jury.

Mes remerciements vont à tous les membres du projet Planète pour leur soutien permanent et leur ambiance de travail.

J'adresse également mes chaleureux remerciements à ma famille pour la patience dont ils ont fait preuve à mon égard et pour leur aide et leurs encouragements.

Un très grand merci à ma femme Lamia qui m'a soutenu tout le long de ma thèse surtout dans les moments délicats. Son amour et sa patience m'ont motivés à mener à terme ce travail.

Enfin, j'exprime ma grande reconnaissance à tous mes camarades, et tout particulièrement Imed, Saber, Mohamed, Miled, Naceur, Mahfoudh, Ali, et Ahmed pour leur suivi de près de l'avancement de ma thèse pour les conseils judicieux qu'il n'ont cessé de me prodiguer.

Fethi Filali
Sophia-Antipolis
27 Novembre 2002

Présentation des Travaux

Il est certain que l'Internet est et elle restera la technologie la plus utilisée dans l'infrastructure mondiale de communication. Durant les deux dernières décennies, l'Internet a évolué d'un simple projet académique de recherche à un réseau de milliers des réseaux indépendants, publiques, et privés interconnectés via le protocole IP (Internet Protocol) [97]. La charge de trafic de données échangées dans l'Internet a augmenté exponentiellement durant cette période, et tous les centres de recherche s'accordent sur le fait que cette évolution dramatique de l'Internet continuera dans les prochaines décennies. Cette croissance peut être expliquée par le grand potentiel de l'Internet à remplacer les réseaux traditionnels téléphoniques en offrant les mêmes services avec une qualité comparable.

Au cours de ses premières années d'apparition, l'Internet a servi un groupe des utilisateurs dans les universités, laboratoires, et agences gouvernementales via des applications unicast de transfert des données comme la messagerie électronique, les news-groups, FTP, et telnet. Cependant, avec l'apparition de World Wide Web (WWW) vers les années 1990, l'Internet commençait à être populaire à l'extérieur de la communauté de recherche scientifique d'une manière exponentielle, et son application est devenue extrêmement diversifiée. Actuellement, l'Internet se transforme à un répertoire d'information (e.g. pages web industriels et personnels, news, etc.) et services à valeurs ajoutées (e.g. dotcoms), un support des nouveaux médias (e.g. téléphonie IP, live audio/video streaming, conférences multimédias), et une technologie pour des nouveaux types de communication (e.g. calcul/interaction/collaboration distribuée basée sur la réalité virtuelle, accès/découverte des services à distance, calcul mobile).

Ces tendances nécessitent une conception et des améliorations massives de l'architecture Internet existante afin que les utilisateurs puissent accéder à des services qui passent à l'échelle et qui sont en plus flexibles et plus sophistiqués. Dans ce contexte, nous considérons que le service "multicast" ou "le transfert point à multipoint" est l'un des blocs clefs qui doit être révisé afin d'aboutir aux objectifs désirés.

Le multicast permet de délivrer des datagrammes à des groupes d'hôtes Internet, éventuellement dispersé sur le réseau qui se sont déclarés membres du même groupe.

L'IP multicast est basé sur le protocole IGMP (Internet Group Management Protocol) décrit dans le RFC 1112 [34]. Ce protocole permet à une station de s'enregistrer dynamiquement pour recevoir différents types de trafic multicast IP. Toute station voulant accéder à un groupe prend l'adresse multicast IP de ce groupe et s'abonne à ce "pseudo réseau" et elle est considérée membre du groupe, tant qu'elle ne le quitte pas. L'adresse de groupe multicast IP est une adresse IP de classe D comprise dans le rang d'adresses de 224.0.0.0 à 239.255.255.255. Dans cette adresse, les 28 bits de poids faible identifient le groupe lui-même. Une partie (23 bits de droite) de ces 28 bits seront utilisés pour former l'adresse multicast Ethernet. Ainsi, une adresse multicast Ethernet permet de couvrir un ensemble de groupes multicast IP.

Tout hôte est libre de rejoindre ou de quitter un groupe multicast à tout moment. Il n'y

a pas de restriction sur le nombre d'hôtes ni sur la localisation de ces derniers. Un hôte peut être membre de plusieurs groupes multicast simultanément.

À la différence des applications unicast utilisant le protocole de transport TCP (Transmission Control Protocol) [98] par exemple, le multicast n'est pas connecté. Un datagramme de trafic multicast est délivré aux membres du groupe avec les mêmes garanties qu'un paquet IP en unicast, i.e.: une trame n'est pas certaine d'atteindre sa destination et l'ordre d'envoi des paquets n'est pas non plus forcément respecté.

Le multicast a l'avantage de préserver la bande passante et de simplifier la conception des applications réparties. Supposons qu'une application ait besoin d'envoyer la même information à un groupe de N utilisateurs. La solution unicast consiste à envoyer sur le réseau local de transmission (au moins) N fois le même paquet d'information, ce qui constitue un gaspillage évident de bande passante et peut engorger le réseau au niveau de l'hôte émetteur.

La solution broadcast nécessite un seul envoi de données. Mais les paquets sont alors traités par toutes les machines ce qui occasionne un gaspillage de CPU. Par ailleurs, le broadcast n'est pas très raisonnable en dehors des réseaux locaux. La solution multicast consiste à envoyer un seul paquet au groupe multicast que les routeurs se chargent de distribuer sur le réseau en multipliant les paquets seulement quand c'est nécessaire.

L'avantage du multicast est donc d'économiser la bande passante en essayant de réduire la multiplication des paquets introduits sur le réseau. Il en découle un second intérêt, celui de simplifier l'envoi de données à un groupe d'utilisateurs : on a pas besoin d'envoyer explicitement la même information autant de fois qu'il y a d'utilisateurs ni de savoir exactement quels sont les utilisateurs. La visioconférence et le tableau blanc sont des applications typiques où apparaît clairement l'intérêt du multicast.

Dès son apparition à la fin des années 1980, le multicast a été l'un des composants clefs de l'Internet Protocol, et il y a eu des énormes activités de recherche au tour des mécanismes de routage multicast et de gestion et construction des arbres de diffusion [10, 22, 33, 1, 40, 55, 68, 86], ainsi qu'autour de transfert multicast fiable [20, 79, 102, 108, 116]. Cependant, la technologie de multicast dans l'Internet n'a pas été encore développée jusqu'à son plein degré. En effet, malgré les grands intérêts d'utilisation de multicast dans l'Internet, il ne permet en aucun cas de résoudre le problème majeur de l'Internet à savoir la congestion du réseau. Ceci du au fait que le développement de l'Internet dans le monde connaît une croissance véritablement exponentielle. La conséquence de cette croissance phénoménale est que le réseau Internet souffre actuellement de nombreux goulots d'étranglement, et cette situation risque encore de s'aggraver.

Le service multicast est loin d'être déployé partout dans l'Internet à cause des problèmes qui restent encore sans des solutions qui soient à la fois efficaces et cohérentes avec les protocoles existants et notamment le protocole TCP qui est le protocole de transport le plus utilisé dans l'Internet.

L'un des principaux problèmes qui limitent le déploiement universel du service multicast dans l'Internet terrestre est le partage des ressources réseaux, et principalement la bande passante, entre les flux unicast et les flux multicast. La question que se pose est faut-il imposer un partage équitable de la bande passante entre une session multicast et une connexion TCP utilisant le même lien de communications ? Ou faut-il donner "un peu plus" pour les sessions multicast car elle sont destinées à servir plusieurs membres. Une autre question qu'en découle est comment garantir ce partage: faut-il se baser sur une solution de bout-en-bout ou avoir un support des routeurs dans le réseau? Faut-il ajouter un service spécifique au service multicast existant afin de pouvoir déployer cette solution? Si oui, comment faut-il procéder?

En fait, il y a actuellement un grand débat au sein de la communauté Internet sur le fait que l'argument "bout-en-bout" est toujours la règle d'or qu'il faut toujours respecter, ou elle est juste morte [26]. En effet, l'Internet actuel est construit sur la philosophie "de bout en bout". Les systèmes à l'extrémité du réseau sont responsables de la construction d'un service mieux que le service best-effort (service au-mieux), donc le réseau est souvent considéré comme un grand nuage avec des systèmes à ses extrémités (PCs, etc.) qui lui sont attachés et auxquels son fonctionnement interne est totalement caché et imprédictible. Cependant, ce n'est pas toujours vrai dans l'Internet actuel, où les serveurs et les proxies pour des applications spécifiques (spécialement le web) sont installés à l'intérieur du réseau dans le but d'améliorer ses performances. Par conséquent, plusieurs personnes croient que le modèle de service "de bout en bout" est en voie de disparition ou il a même déjà disparu, vu que plusieurs services peuvent être implémentés plus efficacement à l'intérieur du réseau qu'à ses bordures.

En plus des difficultés de partage de ressources entre les flux multicast et unicast, un autre problème de déploiement de service multicast multimédias se pose. En effet, la plupart des technologies de communication traditionnelles ne supportent pas parfaitement les nouvelles applications multicast multimédia vu qu'elles ne répondent pas aux contraintes et aux caractéristiques inhérentes de ces dernières telles que le délai de transfert, le débit et la perte.

De plus, l'apparition et la disponibilité de nouvelles applications multimédia telles que la vidéo à la demande, la téléconférence, la télé-médecine, la téléphonie IP, etc. nécessitent la mise en place, de façon économique et à grande échelle, de nouveaux supports de communication. En effet, les contraintes imposées par ce genre d'applications diffèrent de celles des applications traditionnelles (ftp, telnet, web, etc.).

Afin de supporter ce type d'applications, on pense de plus en plus à utiliser d'autres technologies de communication. Outre les technologies terrestres (DSL-*Digital Subscriber Line* et Câble Modems), l'utilisation de satellites apparaît de plus en plus comme une solution convaincante pour les fournisseurs de services multimédia à large bande. En effet, en plus du fait qu'ils offrent une bande passante importante, ils ont un coût de mise en place très modéré et une souplesse d'utilisation très avantageuse [5, 7, 120].

Etant intrinsèquement des liens de diffusion, les liaisons satellites nous permettent de transmettre en multipoint à moindre coût. Contrairement aux connexions multipoints routées sur des réseaux terrestres, les ressources réseaux consommées ne dépendent pas du nombre de récepteurs.

Le succès des standards DVB-S/MPEG-2 (Digital Video Brodcasting) [42, 43, 44] est tel qu'ils sont presque universellement adoptés dans les réseaux satellitaires, et en conséquence, les prix des équipements des consommateurs et de distribution en sont sérieusement affectés. En tant que plate-forme mondialement répandue pour les services de télévision numérique, de radio et données par satellite, l'approche DVB-S/MPEG2 apporte nécessairement de grandes économies d'échelle. Les réseaux basés sur les standards DVB-S/MPEG-2 ont de plus un très grand potentiel pour accepter les nouveaux types d'applications et de services nécessitant de hauts débits pour délivrer des contenus multimédias directement à l'utilisateur final.

Grâce à leur capacité inhérente à couvrir de vaste zone sans contrainte de distances, les systèmes basés sur des satellites ont été traditionnellement orientés pour répondre aux applications de diffusion, par exemple DVB, ou à la fourniture d'accès à des places isolées ou des utilisateurs disséminés dans de larges zones.

Pour transporter des applications interactives, le système satellite doit fournir des réseaux maillés de grande capacité. Les principales difficultés rencontrées sont alors :

- Le délai de propagation,
- Le coût du terminal,
- La largeur de bande limitée (en bande C et Ku).

Le développement de nouvelles technologies de transmission satellite, utilisant la bande Ka et une charge utile régénératrice, permet d'éliminer la plupart de ces difficultés :

- L'utilisation de la bande Ka offre une largeur bande additionnelle au satellite, et permet l'utilisation de plus petites antennes avec un meilleur gain;
- La régénération à bord du signal de communication afin d'améliorer la qualité du signal sur le lien descendant, d'offrir la possibilité de commuter les signaux à bord, et d'établir à bord des liens directs entre utilisateurs;
- La couverture multi-faisceaux est très large et, grâce à la capacité de commuter à bord, ne présente aucune restriction de connectivité. En plus, celle-ci permet une utilisation plus efficace de la fréquence (réutilisation de fréquence), ce qui accroît considérablement la capacité du système. Par conséquent, les satellites avec un processeur bord, conforme à la norme DVB-S sur le lien descendant, sont particulièrement bien adaptés aux applications à large bande DTH (Direct To Home, à destination des utilisateurs finaux) et aux applications de diffusion de données interactives.

Bénéficiant d'une capacité unique de diffusion, les réseaux par satellites peuvent supporter les communications multicast de façon très efficace et peuvent ainsi prétendre jouer un rôle important dans le développement rapide de l'IP Multicast dans le réseau Internet.

Une couverture multi faisceaux est particulièrement adaptée au scénario où les serveurs d'applications multimédias sont répartis géographiquement et où les stations sols éloignées demandent des services à ces serveurs via le satellite. De plus, pour assurer une connectivité sans restriction entre des terminaux utilisateurs petits et abordables avec de faibles délais de transmission, il sera préférable d'avoir la possibilité de commuter à bord les flux de données individuels.

Pour répondre à tous ces besoins, la nouvelle génération des satellites devra donc utiliser des faisceaux multiples, tout en gardant la connectivité entre les utilisateurs et ceci en implémentant un processeur à bord du satellite ayant la capacité de régénération et commutation du signal. Le satellite doit aussi opérer en bande Ka et sa définition sera optimisée afin de répondre aux services identifiés. La tendance qui consiste à apporter des services multimédias interactifs dans les foyers, doit conduire à construire un système satellite Géostationnaire qui supporte un scénario bidirectionnel de haute capacité, flexible et efficace, tout en gardant à l'esprit la contrainte la plus importante qui est le coût du terminal utilisateur.

La définition du système reposera fondamentalement sur un standard DVB/MPEG-2 (au niveau transmission et transport) fournissant la réception TV et l'accès Internet. Etant donné que pratiquement toutes les applications des utilisateurs finaux reposent sur IP, le système devra supporter des services IP multicast sur DVB.

L'intégration des satellites dans l'Internet a fait l'objet de plusieurs travaux de recherche dans plusieurs centres et laboratoires. Ces travaux visent à adapter les protocoles du standard TCP/IP aux caractéristiques inhérentes des satellites. Ils concernent dans une grande partie l'adaptation des protocoles de routage (au sein du groupe de travail UDLR-*UniDirectionel*

Link Routing de l'IETF-*Internet Engineering Task Force* [39]) pour supporter les liens unidirectionnels et le développement de nouveaux mécanismes et techniques de contrôle de congestion et de correction d'erreurs (au sein du groupe de travail *tcpsat* de l'IETF[1]) améliorant les performances de TCP dans les liens satellitaires.

Cette thèse a pour objectif d'étudier le déploiement de service multicast dans des réseaux hétérogènes. En premier lieu, nous nous intéressons au problème de partage de ressources entre les flux unicast et multicast aussi bien dans des réseaux best-effort que dans des réseaux supportant une architecture de différentiation de service telle que DiffServ [15]. Nous développons des mécanismes qui permettent de garantir au mieux ce partage, et restent simples et efficaces, et de plus passent à l'échelle. En effet, le passage à l'échelle est un facteur très important auquel nous faisons plus particulièrement attention dans nos choix de conception des différents mécanismes.

Dans un deuxième lieu, et tout en restant dans le contexte de support des applications multicast dans l'Internet, nous nous focalisons sur le routage multicast dans des réseaux IP intégrant des liens satellitaires aussi bien unidirectionnels que bidirectionnels. Nous nous intéressons à la fois aux satellites géostationnaires (GEO) transparents et aux satellites GEO de nouvelle génération supportant une commutation à bord du satellite et servant des zones multiples et auxquelles nous proposons des mécanismes de commutation multicast à bord de satellite.

Ce rapport de thèse commence par une introduction générale au sujet de transfert multicast dans l'Internet. Cette introduction a pour objectif de présenter les différents mécanismes spécifiques au multicast dans la pile protocolaire et de définir la notion de l'allocation des ressources et le support du multicast sur satellite qui sont les principaux thèmes de ce travail. Dans le chapitre 2, nous faisons un survol des travaux en rapport avec les notre et nous donnerons les briques de base de nos propositions en termes d'hypothèses prises, d'objectifs fixés, et des moyens dont on dispose.

Notre première étude a pour vocation de proposer un nouveau mécanisme de gestion active de files d'attentes qui permet de garantir une équité dans le partage de ressources entre les flux multicast. L'idée principale derrière l'approche adoptée est l'utilisation d'une extension de table de routage multicast pour qu'elle intègre des informations utiles pour effectuer ce partage d'une manière efficace. L'ISP aura la possibilité de choisir la fonction d'équité appropriée. Ainsi, dans le chapitre 3 on s'intéresse à l'allocation de ressources réseaux pour les flux multicast. Notre objectif est de proposer des mécanismes simples qui permettent de fournir une qualité de servie au trafic multicast et ceci en prenant en compte ses propres caractéristiques ainsi que ses différences par rapport au trafic unicast. Dans le chapitre 4, on étudie l'intégration de trafic multicast avec le trafic unicast dans les réseaux supportant la qualité de service et plus particulièrement dans les réseaux utilisant l'architecture DiffServ [15] comme architecture de différentiation. Suite à nos résultats de la première partie, nous proposons des méthodes et des différentes alternatives pour coupler les mécanismes proposés avec l'architecture DiffServ [15]. Nous proposons dans le chapitre 5 une extension au service multicast qui permet de compter d'une manière efficace le nombre de membres dans les groupes multicast aussi bien au niveau des sources multicast que dans les routeurs intermédiaires de l'arbre de diffusion multicast.

Dans notre deuxième étude, nous intéressons au problème de support de multicast dans les réseaux hétérogènes incluant aussi bien des liens terrestres que des liens satellites de nou-

[1] `http://tcpsat.grc.nasa.gov/tcpsat/`

velle génération. Dans le chapitre 6, on se focalise sur le problème de routage multicast sur ce type de réseaux et on détermine les comportements que pourront avoir les protocoles de routage multicast dans un tel environnement de transmission. Dans le chapitre 7, des solutions concernant le support de multicast sur un satellite GEO de nouvelle génération avec un processeur à bord et supportant des spots multiples seront présentées. Ces solutions concernent la commutation des paquets multicast à bord et la gestion de messages de routage multicast par les différents composants du système. Le chapitre 8 présente un nouveau mécanisme de commutation entre les deux modes de protocole de routage multicast PIM-SM [55], à savoir le mode basé sur la source et le mode basé sur l'arbre partagé. Notre mécanisme est applicable aussi bien aux réseaux complètement terrestres qu'aux réseaux utilisant des liens satellitaires.

Dans ce qui suit, on présente les travaux achevés pour chaque partie ainsi que des résumés des résultats obtenus. On termine la présentation par quelques perspectives sur notre futur travail autour de la qualité de service pour les flux multicast et le support de multicast par la nouvelle génération des satellites géostationnaires.

1 Allocation des ressources entre les flux multicast et les flux unicast dans des environnements terrestres hétérogènes

À la différence des applications multicast au niveau applicatif, le multicast au niveau des routeurs permet de résoudre un ensemble des problèmes liés entre autres à la facteur d'échelle. Ce thème de recherche est assez riche et promoteur car il invalide un certain nombre d'hypothèses des protocoles existants et pose par la même occasion des problèmes qui sont toujours ouverts aujourd'hui.

Dans notre étude, on s'intéresse au problème d'allocation dynamique des ressources réseaux pour les flux multicast. Dans ce thèse, nous détaillerons trois contributions majeures:

- un mécanisme de gestion active de files d'attentes pour les flux multicast appelé MFQ (Multicast Fair Queuing) qui permet de partager équitablement la bande passante entre les flux multicast concurrents et ayant des caractéristiques hétérogènes;
- un mécanisme de partage de bande passante entre les flux multicast et unicast dans les réseaux best-effort (service au mieux) ainsi que son intégration dans l'architecture DiffServ [15];
- une extension au service multicast qui permet de compter le nombre des membres dans les groupes multicast aussi bien au niveau de la source multicast qu'au niveau des routeurs intermédiaires dans l'arbre de diffusion.

On détaille dans ce qui suit ces différentes contributions.

1.1 Partage de la bande passante entre les flux multicast

L'IETF a défini un guideline très stricte pour développer des mécanismes de contrôle de congestion multicast [82] sans tenir compte de nombre de membres dans la session multicast. Cependant, d'un coté, plusieurs travaux de recherche ont montré que les protocoles de contrôle de congestion utilisant un principe multiplicative decrease/linear increase, et en particulier TCP, conduisent à une équité proportionnelle [74] qui consiste à donner plus de bande passante

aux flux ayant le plus faible RTT (Round Trip Time). D'un autre coté, Chiu [25] a prouvé que si on considère un flux multicast comme étant un flux unicast, alors l'application de modèle de Kelley et Tan [74] montre que les groupes multicast ayant un grand nombre de membres obtiendront une plus faible bande passante que celles qui ont moins de membres.

Nous partageons l'idée des auteurs de [108] dans la mesure où la définition de l'équité inter-muticast (équité entre les flux multicast) doit tenir compte de nombre de groupes multicast, de nombre de flux par groupe, et de nombre de membres par flux. Les auteurs de [75] ont défini trois différentes stratégies d'allocation de bande passante pour les flux multicast ainsi que des critères de comparaison de leur performance. Ils ont montré que la politique LogRD[2] conduit toujours à un meilleur compromis entre la satisfaction des récepteurs et l'équité inter-multicast. Afin d'implémenter leur proposition dans des réseaux réels, ils ont proposé d'ajouter leurs stratégies d'allocation de la bande passante dans le scheduler général en configurant les poids d'un scheduler GPS (Generalized Processor Sharing) [96]. L'objectif peut être aussi atteint en réservant d'une manière explicite la bande passante dans le réseau soit pour des connections individuelles ou pour un groupe de connexions (une classe de service). Cependant, les deux méthodes sont complexes et nécessitent soit l'utilisation des mécanismes de Fair Queuing (FQ) [36, 96] dans chaque routeur du réseau soit l'utilisation des protocoles de réservation de ressources comme RSVP (ReserVartion Protocol) [18] qui nécessitent une coordination et une intégration parfaite entre les routeurs adjacents tout au long du chemin entre la source et la destination.

Afin de garantir l'équité entre les flux multicast, nous proposons une nouvelle approche basée sur un mécanisme de gestion active de files d'attente (AQM-Active Queue Management) au lieu de l'utilisation de mécanisme FQ ou la réservation explicite de la bande passante. Nous développons un nouveau mécanisme de gestion active de files d'attente dans les routeurs qui :

- garantit le partage efficace de la bande passante entre les flux multicast de la manière souhaitée,
- s'adapte au changement dans les tailles de groupes multicast, le nombre de flux actifs, et la stratégie d'allocation de la bande passante utilisée,
- améliore le taux d'utilisation du lien de communication.

Deux motivations principales sont derrières notre proposition. Premièrement, le transfert multicast d'une source à plusieurs récepteurs demandent moins de bande passante que l'unicast, essentiellement pour les applications avec un groupe large et dense. Deuxièmement, l'utilisation d'un mécanisme simple de gestion active de files d'attentes pour distribuer la bande passante d'une manière équitable entre les sessions multicast, encouragera les ISPs (Internet Service Providers) à déployer le multicast dans leurs réseaux vu que leur nombre de clients augmentera tout en maintenant une utilisation faible de ressources réseaux. Quant aux clients, ils seront plus motivés à choisir un ISP qui supporte ce type de service qu'un autre car ils pourront obtenir un service d'une bonne qualité et probablement avec le même ou un plus faible coût. Par conséquent, les clients seront de plus en plus intéressés à former des groupes larges ce qui est une grande motivation pour les fournisseurs des applications multicast afin de développer et d'encourager le déploiement du service multicast.

[2]La politique LogRD consiste à donner au flux multicast numéro i une fraction de la bande passante égale à $\frac{1+\log n_i}{\sum_j (1+\log n_j)}$, où n_j est le nombre de récepteurs du flux j.

Dans notre travail nous proposons un nouveau AQM pour le trafic multicast. Nous ne cherchons pas la fonction optimale d'équité inter-multicast car il est certain qu'en existe pas une seule vu les facteurs économiques et sociaux qui influent d'une façon directe ou indirecte sur le choix de cette fonction. Ces facteurs poussent généralement les ISPs à implémenter des différentes politiques d'équité en tenant compte aussi bien de leur stratégie commerciale qu'aux attentes de leurs clients. Pour être déployer sur les réseaux des différents ISPs, notre mécanisme est totalement indépendant de la fonction d'équité à adopter. Cependant, nous pensons que le nombre de membres de sessions multicast dans chaque routeur intermédiaire doit être considéré comme un paramètre essentiel dans la définition de la fonction d'équité inter-multicast.

À notre connaissance, il n'y a pas des travaux similaires dans le domaine de partage de la bande passante en utilisant un mécanisme de gestion active de files d'attentes dans la littérature. De plus, nous considérons notre approche un axe promoteur pour le développement des protocoles de contrôle de congestion pour le multicast utilisant une aide du réseau.

Nous appelons notre mécanisme MFQ (Multicast Fair Queuing), un mécanisme de rejet par-flux (per-flow dropping mechanism) et qui interagit avec un module d'allocation de la bande passante calculant pour chaque flux multicast actif sa fraction de bande passante à ne pas dépasser. MFQ appartient à la classe de mécanismes faisant un traitement par flux, comme le cas de FRED (Flow Random Early) [78]. Les opérations faites par MFQ ne sont pas complexes comme la manipulation de priorités dans le mécanisme FQ (Fair Queing) [36], vu qu'il consiste tout simplement de décider d'accepter ou de rejeter le paquet arrivé en appliquant un algorithme simple.

Il est très important à noter à ce stade de rapport que le fait de considérer que les mécanismes basés sur le flux sont complexes et ne passent pas à l'échelle n'est valide que pour ceux utilisés pour les flux unicast pour lesquels les routeurs ne gardent pas un état dans leur table de routage. En multicast, les tables de routage maintiennent une entrée par session active, essentiellement l'adresse de la source, l'adresse de groupe, la liste des interfaces d'entrées, la liste des interfaces de sorties. Par conséquent, l'ajout d'une information supplémentaire par session active n'augmentera la table de routage que d'une petite fraction. Nous montrerons qu'un état de taille très modeste et un petit calcul dans les routeurs permettent de fournir des gains en performances très importants pour les applications multicast.

MFQ permet d'obtenir le partage souhaité de la bande passante de lien entre tous les flux en compétition en utilisant une seule file d'attente. Il a été conçu pour être indépendant de la politique de partage de la bande passante. En effet, le module d'allocation de la bande passante peut implémenter soit une fonction d'équité multicast comme celles décrites dans [75] soit un modèle de tarification multicast comme ceux proposés dans plusieurs travaux de recherche [24, 32, 65, 66]. Notre mécanisme nécessite que les routeurs maintiennent un état et exécute des opérations par flux. Il utilise le concept Multicast Allocation Layer (MAL), une nouvelle notion d'allocation de la bande passante qui nous utilisons pour obtenir l'allocation attendue via un service paquet-par-paquet et pour garantir un partage très fin de la bande passante entre les flux concurrents.

Nous utilisons des simulations avec le simulateur des réseaux NS-2 [84] pour évaluer les performances de MFQ pour des différents scénarios de configuration. Dans l'absence des mécanismes similaires au notre, les performances de MFQ seront comparées aux résultats obtenus analytiquement.

Nous avons tout d'abord évalué les performances de notre mécanisme pour des sources multicast non-adaptatives; c'est à dire pour des sources qui ne modifient pas leur comportement

quand un paquet est perdu. Ensuite, nous avons considéré le cas quand ces sources implémentent le protocole Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL) [20] comme mécanisme de contrôle de congestion, et donc des sources adaptatives. En particulier, nous montrons que MFQ permet d'éviter le rejet de paquet des couches prioritaires par rapport aux couches à faible priorité et il permet d'obtenir un partage équitable de la bande passante entre les différentes sources FLID-DL. Enfin, nous expérimentons MFQ pour des sources hétérogènes où il y a des sources CBR et FLID-DL. Pour les trois cas, nous montrons que le partage de la bande passante obtenu en utilisant MFQ est très proche de celui attendu. Sans perdre en généralité, nous validons MFQ pour des différents schémas d'allocation de la bande passante pour les flux multicast utilisant des fonctions linéaire, indépendant et logarithmique qui dépendent de nombre de membres dans les groupes multicast. De plus, nous montrons que notre mécanisme s'adapte au changement dynamique dans le nombre de membres en raison d'arrivé ou de départ des abonnés au groupe multicast. Nous validons MFQ pour des sources très hétérogènes ayant différents débits, nombre de membres, temps de début de la transmission, et temps de fin de la transmission.

1.2 Partage de la bande passante entre flux unicast et multicast

Parmi les facteurs qui bloquent le déploiement de l'IP multicast est l'absence des mécanismes de contrôle de congestion qui soient bien évalués et validés via des expérimentations réelles et à grande échelle.

Les conditions requises pour le contrôle de congestion multicast sont encore ouvertes à la discussion, mais il est très évident qu'un flux multicast n'est acceptable que si la bande passante consommée sur le chemin menant à un membre n'excède pas celle qui aurait été consommée par un flux TCP entre l'émetteur et le membre en question. Cette condition pourrait être atteinte soit en utilisant un seul groupe si l'émetteur envoie à un débit unique qui correspond à celui du membre le plus lent, soit via un schéma de transmission multicast en couches qui permet aux différents récepteurs de recevoir les données avec des différents débits qui correspondant à leur capacité de réception.

Dans ce travail, nous développons une nouvelle approche qui apporte de l'aide aux mécanismes de contrôle de congestion dans la mesure où ils pourront coexister équitablement avec TCP sans pour autant implémenter des techniques complexes. Notre approche est basée sur une nouvelle notion d'équité appelée "l'équité inter-service" (inter-service fairness), qui est utilisée pour partager les ressources réseaux entre les services unicast et multicast. Dans cette définition, le trafic multicast agrégé doit rester globalement TCP-freindly dans chaque lien de communication. En d'autres termes, le débit de service multicast ne doit pas dépasser en aucun cas la somme des débits TCP-friendly de tous les flux multicast. Cette définition permet aux ISPs d'utiliser leur propre stratégie de partage de la bande passante qui, comme nous avons déjà signalé, pourra implémenter soit une politique de facturation [65] soit une stratégie d'allocation de la bande passante comme celles décrites dans [75].

Afin d'implémenter notre définition d'équité, nous proposons d'utiliser un scheduler qui ressemble à CBQ/WRR [59] et qui utilise que deux files d'attentes: une pour la classe unicast et l'autre pour la classe multicast. Nous appelons notre mécanisme: Service-Based Queuing (SBQ) puisqu'il distingue entre deux différents services de transfert: unicast et multicast. SBQ intègre une méthode pour varier dynamiquement les poids deux files d'attente afin d'atteindre le partage de la bande passante souhaité entre les flux unicast et multicast. Pour chaque paquet arrivé, le routeur identifie la nature du paquet (unicast ou multicast) et l'envoi vers la

file d'attente adéquate pour être ensuite servi par le scheduler.

A noter que l'identifiant de la nature du paquet nécessite de voir si le poids fort de l'adresse destination. Il n'est pas alors nécessaire d'examiner tous les champs de l'entête IP.

Il est important d'avoir à l'esprit que SBQ n'est qu'un élément parmi les composants de l'architecture globale de différentiation pour le multicast. En fait, le but de ce scheduler est de partager la bande passante entre les flux unicast et multicast. Afin de pouvoir partager la bande passante réservée pour le service multicast entre les flux multicast, nous utilisons notre mécanisme de gestion des files d'attentes MFQ présenté dans la section précédente et qui représente un autre composant clef.

Là aussi, nous n'avons pas trouvé des travaux similaires au notre et qui permettent de partager explicitement les ressources réseaux entre les services unicast et multicast.

Nous utilisons des simulations pour évaluer les performances de notre scheduler pour plusieurs types des sources incluant non seulement des sources unicast TCP, UDP, et des sources Multicast CBR, mais aussi des sources multicast implémentant le protocole de contrôle congestion en couches FLID-DL |80|. Les simulations ont été effectuées pour des caractéristiques des réseaux et des liens très hétérogènes avec des différents temps de début et terminaison des sessions multicast, taille de paquets, débits, et nombre de membres dans les différentes sessions.

Dans la deuxième partie de notre étude sur le partage de ressources entre les flux unicast et multicast on s'est intéresser au problème d'intégration de SBQ dans l'architecture DiffServ. Nous proposons trois différents méthodes de re-marquage de paquets multicast dans les nœuds dans le réseau DiffServ. Le scheduler SBQ permet de partager la bande passante entre les flux unicast et multicast dans les classes Best-effort et les classes AF (Assurated Forwarding).

Vu que l'état par flux dans la table de routage multicast est nécessaire pour la réplication des paquets multicast à la liste des interfaces de sortie, nous proposons de profiter des avantages de l'existence de cet état pour atteindre deux objectifs:

- améliore le partage de la bande passante entre les flux unicast et multicast, et
- améliore le partage de ressources réseaux entre les flux multicast dans les réseaux DiffServ.

Le premier objectif est obtenu via l'intégration de SBQ dans l'architecture DiffServ, alors que le deuxième est garantit avec le développement des méthodes de re-marquage des flux multicast qui prend en compte le nombre des membres dans chaque groupe multicast. A noter que re-marquer un paquet multicast signifie de changer son DSCP (DiffServ Code Point) sans pour autant changer la classe DiffServ mais que le "drop precedence".

Notre proposition est basée sur le remplacement de la seule file d'attente de chaque classe DiffServ (BE ou AF) avec notre scheduler SBQ. De plus, nous proposons et évaluons trois méthodes (le mapping LIN, le mapping LOG top, et le mapping LOG bottom) basé sur le nombre de membres dans chaque interface de sortie pour re-marquer les paquets multicast dans le but de partage de ressources réseaux entre les flux multicast. Chaque méthode utilise une façon spécifique pour mapper le nombre de récepteurs d'un groupe multicast à un nouveau drop-precedence, et donc un nouveau valeur de DSCP.

Nous pensons que nos propositions peuvent aider au déploiement des services multicast dans les réseaux best-effort ainsi que dans les réseaux DiffServ. En effet, notre contributions encourageront les ISPs à déployer le multicast dans leurs réseaux vu qu'ils pourront déployer

leur modèle de tarification multicast qui sont basés sur le nombre de membres. En effet, le concept de re-marquage que nous avons proposés peut être couplée avec une stratégie de tarification pour sélectionner la méthode appropriée.

1.3 Calcul de nombre de membres dans les communications multicast

Malgré une décennie de recherche et de développement, le multicast n'a jamais été déployé à grande échelle. Parmi les difficultés que rencontre le déploiement de multicast dans l'infrastructure actuelle de l'Internet est l'allocation des adresses, la gestion, et le support de la facturation des services multicast.

Nous pensons que la connaissance de nombre de membres dans les routeurs pour un groupe spécifique peut aider à résoudre plusieurs problèmes de recherche liés au déploiement du multicast dans l'Internet. En effet, il sera possible de résoudre par exemple le problème d'implosion de feedbacks qui apparaît dans le cas d'utilisation des messages NACKs ou IGMP dans certains types de support de transmission comme les réseaux satellites. De plus, les ISPs seront capables d'établir un modèle économique pour le multicast qui prend en considération le nombre de membres dans chaque session multicast active.

Dans le modèle standard de service multicast [34], ni la source ni les routeurs intermédiaires ne peuvent connaître le nombre de récepteurs descendant. Plusieurs travaux de recherche concernant l'estimation de taille des groupes multicast au niveau de la source multicast et qui utilisent des modèles analytiques et des techniques de probing ont été présentés dans la littérature [6, 17, 61, 77, 92]. Ces propositions ont trois limitations majeures. Premièrement, ils nécessitent de bien configurer un certain nombre de paramètres qui dépendent des conditions de réseau et des membres. Deuxièmement, ils permettent qu'à la source d'estimer la taille des groupes multicast et non aux routeurs intermédiaires. En effet, ces routeurs pourront utiliser cette information pour certaines applications comme les techniques de partage de la bande passante ou la facturation de service multicast. Finalement, ces méthodes ne passent pas à l'échelle car elles utilisent des messages périodiques envoyés par les membres vers l'émetteur même dans le cas où il n'y aurait aucun changement dans la taille de groupe.

Nous proposons une autre approche basée sur des messages explicites des mises à jour de nombre de membres et non sur techniques de probing. Le but de notre protocole est de déterminer le nombre de récepteurs pour chaque session multicast et pour chaque interface de sortie de chaque router appartenant à l'arbre de diffusion multicast. Notre protocole aidera le routeurs désigné (DR- Designated Router) de la source, et par la suite la source elle même, à connaître les nombres d'hôtes dans leurs réseaux locaux qui sont abonnés à chaque session multicast active. De plus, il permet aux routeurs intermédiaires de déterminer le nombre de membres dans chaque interface de sortie. Pour ce faire, nous n'avons pas besoin un nouveau message entre le routeur et les hôtes car dans IGMPv3 [22] et MLDv2 [117] chaque hôte doit s'abonner et désabonner explicitement des sessions multicast.

2 Déploiement de l'IP multicast dans les réseaux hybrides (satellites-terrestres)

La diffusion de flux dans Internet requiert, en général, des débits importants. Cela concerne la diffusion de vidéos, mais aussi le téléchargement massif de logiciels et la création de sites Web miroirs. Pour cela, nous tirons parti de la large bande passante dont disposent les satellites

de télédiffusion. Ces satellites géostationnaires, conçus exclusivement pour la diffusion de programmes télévisuels, ont vu leur champ d'action élargi à la diffusion de tous les types de supports multimédia. L'émission nécessite un investissement important. Mais, pour la réception des données, il suffit d'une antenne parabolique de 60 centimètres couplée à un ordinateur personnel ou à un réseau local. Pour parvenir à ce résultat, il a fallu résoudre un problème majeur. En effet, le fonctionnement de certains protocoles Internet repose sur une communication à double sens entre deux stations, alors que la liaison par antenne de réception est unidirectionnelle. Il faut donc assurer une voie de retour par liaison terrestre. Ce problème a été résolu dans le groupe de travail UDLR (UniDirectional Link Routing)[3] de l'IETF en proposant un mécanisme appelé LLTM (Link Layer Tuneling Mechanism) [39] qui permet d'utiliser les liens terrestres de retour d'une manière transparent en émulant ainsi des liens satellites bidirectionnels. Néanmoins, des problèmes liés au fonctionnement des protocoles de routage dynamiques multicast restent encore ouverts.

Dans une première étape dans notre étude sur le support de l'IP multicast, nous avons fixé comme objectifs:

- d'étudier le comportement des protocoles de routage multicast dans un réseau hybride satellite-terrestre utilisant des satellites transparents, et
- de proposer de mécanismes et des techniques afin d'optimiser le comportement de ces protocoles dans un tel système.

Dans une deuxième étape, nous nous focalisons sur les satellites géostationnaires de nouvelle génération. Ces satellites se différencient des satellites transparents par le fait qu'ils supportent plusieurs zones de diffusion et qu'ils ont des processeurs intelligents à bord qui sont capables de commuter les données reçus vers un ou plusieurs zones. Supporter le multicast d'une manière efficace dans des réseaux hétérogènes incluant des liens satellitaires de nouvelle génération est le problème auquel nous avons apporté des solutions qui sont à la fois déployable et efficace.

Nous résumons dans ce qui suit les différents travaux achevés au cours de ces deux étapes.

2.1 Le routage multicast sur les satellites géostationnaires transparents

Le multicast permet d'envoyer des paquets des données à plusieurs sites en même temps. L'idée derrière ça est la capacité d'envoyer un seul message à un ou plusieurs nœuds dans une seule opération. Ceci permet de fournir un gain énorme en bande passante en le comparant à la transmission unicast traditionnelle qui consiste à envoyer plusieurs copies de même message à chaque nœud du réseau. En plus de l'amélioration des performances par rapport à la transmission unicast, le multicast permet la construction réelle des applications distribuées. En effet, il permet aux développeurs des applications d'ajouter plus des fonctionnalités sans pour autant influencer énormément sur le réseau. Il ressort alors que les applications et les services multicast joueront un rôle important dans la future de l'Internet vu que le déploiement de multicast encourage leur développement et utilisation.

Parallèlement à l'évolution des services Internet, l'infrastructure Internet intègre de plus en plus plusieurs types des liens de communications filières et sans fils. Un des composants majeurs de cette infrastructure est les liens satellitaires [101]. En fait, durant les trois dernières décennies, les satellites ont joué un rôle très important dans le déploiement des services multimédias interactifs. Les applications potentielles sont la téléconférence, l'enseignement à

[3] `http://www.udcast.com/udlr`

distance, le transfert des images à haute résolution, la vidéo à la demande, la diffusion de TV, radio, des journaux et des données. La nature des ces services nécessite l'adoption d'une transmission avec un débit T1 et même plus. Afin de supporter ces applications demandant une grande bande passante, il est évidant que la nouvelle génération des systèmes de communication satellites sera différente des systèmes traditionnels en utilisant un processeur intélligent à bord du satellite et en utilisant la bande ka ou la bande V, ainsi que la technologie de diffusion séparée sur des multiples spots. Tous ces aspects peuvent poser plusieurs questions et éventuellement des problèmes de déploiement de l'Internet par satellite.

Vu que les protocoles Internet ont été conçus sans prendre en compte les caractéristiques de support physiques, l'intégration efficace des systèmes satellite de nouvelle génération dans l'Internet nécessite l'étude et l'adaptation de ces protocoles. Le problème de routage dynamique sur les liens unidirectionnels, et en particulier les liens satellitaires, a été résolu dans le groupe du travail UDLR de l'IETF. En fait, il a été proposé un mécanisme appelé LLTM (Link Layer Tunneling Mechanism) [39] qui permet d'émuler le lien unidirectionnel. Ce mécanisme utilise le protocole Dynamic Tunnel Configuration Protocol (DTCP) qui fournit un moyen aux récepteurs satellite pour découvrire dynamiquement la présence de feeds (les stations satellitaire émettrices) et de maintenir la liste de tunnels terrestres opérationnels. Les feeds annoncent sur le lien unidirectionnel périodiquement les adresses de bout de leur tunnel. Les récepteurs écoutent ces annonces et maintiennent une liste des end-points de tunnels.

Alors que le routage dynamique unicast sur les liens unidirectionnels a été résolu, il reste plusieurs autres problèmes concernant l'intégration des liens satellite dans l'Internet qui sont encore sans solutions ou même qu'ils n'ont été jamais posés comme le passage à l'échelle du mécanisme LLTM, le routage multicast sur liens satellites, et le transfert multicast fiable sur satellites. Dans notre travail, nous étudierons le comportement de l'IP multicast dans la nouvelle génération des réseaux hybrides satellite-terrestre. Nous discutons les problèmes de déploiement des liens satellitaires dans l'Internet pour la diffusion multicast et nous présentons des solutions à court et à long terme pour remédier à ces problèmes.

Nous examinons les protocoles de routage basés sur la source comme DVMRP [118] et PIM-DM [1] et pour ceux utilisant un arbre partagé, nous nous focalisons sur PIM-SM [40]. Pour chaque protocole, nous déterminons un ou plusieurs comportements nous souhaitables dans le segment satellitaire et nous proposons des solutions pour résoudre ces problèmes. Pour DVMRP et PIM-DM, nous identifions quelques scénarios de configuration où les récepteurs satellite peuvent recevoir des paquets dupliqués et nous proposons une méthode pour remédier à ce problème. Pour PIM-SM, nous détaillons une politique de configuration qui a deux avantages principaux. Premièrement, elle construit un arbre de diffusion où les membres reçoivent les données envoyées par les sources multicast via le satellite en utilisant un arbre partagé basé sur le point de redez-vous (RP - Rendez-vous Point) et non un arbre basé sur la source. Deuxièmement, elle minimise la charge du trafic multicast sur le lien terrestre ce qui donne un gain important dans la bande passante terrestre qui pourra être utilisée par les applications terrestres, ou plus généralement par les applications ayant des besoins et des exigences qui ne peuvent pas être garantis par les liens satellites.

Nous étudions un système satellitaire concret développé dans le cadre du projet RNRT DIPCAST[4]. Ce système utilise un satellite GEO transparent pour fournir le service multicast pour les utilisateurs finaux. Nous avons proposé et décrit un ensemble des configurations et des adaptations pour le protocole PIM-SM afin d'optimiser le transfert multicast dans le

[4] `http://www.dipcast-satellite.com`

système DIPCAST en réduisant les messages de signalisation et des données sur le segment satellite.

2.2 Le support efficace de l'IP multicast dans les satellites géostationnaires de nouvelle génération

L'intégration des satellites dans l'Internet a fait l'objet de plusieurs travaux de recherche dans plusieurs centres et laboratoires. Ces travaux visent à adapter les protocoles du standard TCP/IP aux caractéristiques inhérentes des satellites. Jusque là, la totalité des travaux, supposent l'utilisation d'un satellite transparent, un satellite qui ne fait que l'amplification du signal reçu et sa diffusion sur le lien descendant. Ceci présentait différents problèmes, et essentiellement, lorsque la diffusion se fait sur des zones non concernées par ces données. Dans ce cas, on se trouvait avec deux constations:

- le gaspillage de l'énergie interne du satellite,
- le manque de sécurité du réseau satellitaire,
- et enfin, la nécessité d'avoir un récepteur assez sensible pour réceptionner les données utiles.

Pour tous ces problèmes, le fait d'utiliser un nouveau type de satellite avec processeur à bord est intéressant. D'une part, ce type de satellite va nous permettre de profiter de l'information contenue dans le signal transmis, afin d'exécuter une tâche précise comme la commutation, d'autre part, il permet de concentrer la puissance transmise dans des spots limités. D'un autre côté, et vu l'aspect de diffusion caractérisant les réseaux satellitaires, le satellite se voit le moyen le plus adapté pour les applications multicast. En effet, les applications multicast, sont des sessions point-à-multipoints, définies par une adresse source, et une adresse de groupe identifiant l'ensemble des membres. Ce genre d'applications exige une large bande passante, une contrainte que le satellite peut satisfaire facilement. Le satellite diffuse les flux reçus sur ces spots de sorties, et ainsi, tous les terminaux au sol, peuvent accéder à ce flux, ce qui permet de consommer la même bande passante indépendamment du nombre de terminaux.

Afin de profiter au mieux du satellite, et le rendre un composant essentiel dans la transmission du flux multicast dans l'Internet, il faut résoudre deux problèmes:

- identifier les sessions sur le lien satellitaire, afin de permettre d'une part, de procéder à la commutation à bord du satellite, en se basant sur sa destination, et d'autre part, permettre aux terminaux au sol de faire le filtrage des données utiles parmi les données reçus.
- minimiser l'utilisation de la bande passante sur le lien montant, et cela en mettant en place une nouvelle manière de signalisation entre les différents composants du réseau satellitaire, réduisant le trafic de contrôle et de signalisation.

Ces deux dernières années, les chercheurs se sont intéressés à l'étude des méthodes de transmission du flux IP multicast sur des systèmes DVB (Digital Video Broadcasting) [42, 44, 43, 46, 45]. Parmi eux, il y a des groupes qui se sont intéressés au transport sur le réseau terrestre câblé, comme ATSC (Advanced Television Systems Committee) [2] et le SCTE (Society of

Cable Telecommunications Engineers) [112], et d'autres à l'étude du même sujet, mais spécifiquement pour le satellite regénératif [3]. Cependant, jusqu'à ce jour, il n'existe aucune étude, qui utilise la possibilité de commutation à bord d'un satellite multi-spots permettant de couvrir plusieurs zones, chacune desservie par un spot donné. Cela n'empêche que les études déjà faites vont profiter énormément pour les nouvelles études.

Dans ce travail, réparti en deux grandes parties, nous avons essayé de résoudre les différents problèmes soulevés précédemment. La première tâche consistait en la mise en place d'une nouvelle couche liaison, appelée *IP-Optimized Adaptaion Layer,* qui fournie une nouvelle manière d'encapsulation des paquets IP et leur segmentation en des segments de données MPEG-2 sur les liens satellitaires, et présentant trois intérêts:

- un nouveau mode d'adressage remplaçant l'adressage IP sur le lien satellitaire,
- la commutation, qui devient possible grâce aux différents champs précisant les ports de sorties à bord du satellite.
- et enfin, le filtrage de paquets au niveau des terminaux satellitaires, non seulement selon l'adresse destination, mais aussi selon l'adresse source du paquet reçu.

Notre méthode d'encapsulation optimisée pour l'IP multicast a deux modes permettant d'utiliser deux différent approches de commutation des paquets à bord du satellite: l'approche self-routing et l'approche label-switching. La première approche utilise une table de commutation maintenue dans le processeur satellite et mise à jour par le NCC alors que la deuxième approche utilise une information de commutation qui est incluse au préalable par le RCST émetteur dans chaque segment de données MPEG-2.

La deuxième tâche était de proposer un nouveau protocole de routage, que nous avons nommé SMRP (Satellite Multicast Routing Protocol), qui permet d'étendre l'aspect et les caractéristiques du protocole de routage PIM-SM (Protocol Independent Multicast - Sparse Mode) [40, 55], sur les liens satellitaires. Le protocole PIM-SM, comme protocole de routage multicast, gère l'abonnement et la transmission des flux entre les différents routeurs composant le réseau multicast. Le protocole SMRP exécute les mêmes tâches tout en veillant à minimiser les flux de signalisation, et surtout sur le lien montant du réseau satellitaire.

Le protocole SMRP, a été conçu pour la transmission du flux multicast, mais il pourrait être utilisé aussi pour les flux unicast. En effet, le satellite est jusqu'à ce jour très utilisé pour la télédiffusion, et pour cela, le protocole SMRP, que nous proposons tient compte de divers applications et protocoles multicast et unicast existants.

2.3 Un mécanisme de commutation entre les deux modes de PIM-SM

PIM-SM est probablement le protocole de routage multicast le plus déployé [35, 40, 55] actuellement dans l'Internet. PIM-SM crée un arbre de distribution multicast partagé basé sur un point de Rendez-vous permettant d'atteindre tous les membres du groupe multicast en question. De plus, il donne la possibilité aux membres de passer vers l'arbre basé sur la source (SPT - Shortest Path Tree), mais il ne spécifie pas d'une manière définitive l'algorithme qui doit être utilisé obligatoirement par les membres. En effet, la politique recommandée par le standard de PIM-SM et de migrer vers le SPT dès que la taille des données reçues via l'arbre partagé dépasse un certain seuil. Cette heuristique de la décision de la migration entre le RPT

et le SPT est loin d'être efficace et suffisante. En effet, les routeurs décident de migrer ou non indépendamment des autres routeurs appartenant à l'arbre de distribution.

A notre connaissance, il y a un seul travail de recherche décrivant un mécanisme amélioré pour la commutation entre l'arbre partagé et l'arbre basé sur la source décrit dans [68]. En effet, les auteurs ont proposé une extension de PIM-SM nommée PIM-Switch et qui est basée sur l'estimation de la densité de groupe multicast dans le réseau. Cependant, PIM-Switch a trois inconvénients majeurs. Premièrement, il propose d'utiliser d'une manière exclusive soit le RPT soit le SPT et ceci pour tous les membres dans le groupe. Deuxièmement, il ne tient pas en compte ni les besoins en qualité de service des récepteurs, ni les exigences du réseau. Finalement, il n'utilise aucune coordination entre les récepteurs pour décider comment et quand commuter entre les deux modes de PIM-SM.

Nous pensons que l'utilisation d'un mécanisme basé sur une coordination entre les membres concernés pour commuter entre le RPT et le SPT est plus efficace et permet d'accomplir les intentions originales de PIM-SM et le rendre plus efficace en le comparent avec d'autres protocoles de routage et principalement le protocole CBT [10].

La conception d'un nouveau mécanisme de commutation pour PIM-SM renferme trois parties majeures. La première partie est le développement d'un algorithme de décision qui permet aux récepteurs de décider quand ils envoient des requêtes de commutation vers le RPT ou vers le SPT. Dès que cette décision est prise, la deuxième partie décrira la procédure d'acceptation de cette requête par les routeurs intermédiaires appartenant au chemin entre le membre et la source. Cette acceptation doit tenir compte non seulement de besoins des membres en QoS mais aussi d'autres exigences liées au réseau comme la concentration de trafic dans les liens de communication et l'utilisation de ressources réseaux par les protocoles de signalisation. La troisième partie comprend la syntaxe et la spécification du mécanisme à mettre en place afin de prendre efficacement la décision de commutation en intégrant le nouveau mécanisme dans le protocole PIM-SM.

Dans notre mécanisme, nous explorons les trois parties. Il est claire que nous aimerions améliorer les paramètres de performance qui sont utiles à la fois pour le RP/le réseau et les membres. Pour ce faire, nous étudions et nous comparons le compromis entre la complexité et l'efficacité de notre mécanisme de commutation.

3 Perspectives

Il y a plusieurs extensions importantes et prometteuses des problèmes examinés dans cette thèse. En effet, un nombre des problèmes restent encore ouverts pour la conception des mécanismes efficaces afin de distribuer efficacement les ressources réseaux entre les flux unicast et multicast. Nous continuerons notre approche de conception tenant en compte à la fois les exigences des ISPs et les contraintes imposées par les applications et les réseaux.

Premièrement, dans l'architecture de MFQ, nous avons supposé que la fonction d'équité dépende de nombre des membres dans chaque groupe multicast. Nous planifions de développer une définition générique de la fonction d'équité multicast dont nous avons donné des premiers éléments au cours de la description de MFQ et ceci en incluant d'autres paramètres tels que le débit du trafic de chaque flux ou les modèles de tarification multicast. D'autres travaux futurs dans ce domaine pourront inclure aussi le développement d'un mécanisme de contrôle de congestion qui se basera sur MFQ pour garantir l'équité entre les flux multicast à l'intérieur du réseau et utilise une approche en couches pour la transmission des flux multicast.

Deuxièmement, il existe plusieurs domaines possibles pour des travaux futurs dans la thèse de partage de ressources réseaux entres les flux unicast et multicast et quelques aspects de performance demeurent encore à évaluer. Une extension possible serait de développer un mécanisme de contrôle de congestion de bout-en-bout qu'utilisera notre scheduler SBQ pour partager efficacement la bande passante entre les flux unicast et multicast.

Troisièmement, le service de comptage peut être intégré et évalué dans l'implémentation SSM décrite et disponible dans [8]. Il est très important de mentionner que notre proposition est très utile pour d'autres applications telles que les services de tarification et de statistiques, le contrôle de congestion pour le multicast, la suppression de feed-back, etc.

Supportant l'IP multicast sur les satellites GEO est un axe de recherche excitant intégrant plusieurs problèmes ouverts. Notre travail dans ce domaine a été focalisé principalement sur le problème de routage multicast. Cependant, d'autres problèmes demandent davantage des investigations. Par exemple, l'intégration des satellites GEO avec des satellites LEO ou des réseaux sans fils est un problème intéressant d'examiner. Etant donné un réseau GEO avec plusieurs zones des couvertures (spot beams), comment réseau peut-il décider sur quelle passerelle à utiliser pour sortir du réseau ? Quel type d'équilibrage de charge, à intégrer autour des terminaux satellitaires, est-il nécessaire ? Quelles sont les tailles optimales des files d'attente des commutateurs à bord du satellite?

Le mécanisme de commutation entre l'arbre basé sur le RP et l'arbre basé sur la source que nous avons proposé pourrait être étendu en évaluant d'autres métriques de performance telles que le coût en terme de bande passante et de délai ainsi que la satisfaction des récepteurs. Une autre possibilité d'extension de ce mécanisme serait de voir comment il pourrait être étendu afin d'offrir la commutation inverse : de l'arbre basé sur la source vers l'arbre partagé.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet is the underpinning communication infrastructure of the future. During the last couple of decades, the Internet has evolved from an academic research project into a network of thousands of independently administered, public and private networks interconnected via the Internet Protocol (IP) |97|. The traffic on the Internet has been growing exponentially during this period, and industry research firms forecast that the dramatic growth of the Internet will continue in the next decades. This growth alludes to the potential of the Internet to replace the traditional telephone and the pervasive public network.

At its early stage, the Internet has served a confined group of users at universities, laboratories, and government agencies via unicast, bulk-data-based applications such as E-mail, News group, FTP, and Telnet. However, with the advent of the World Wide Web (WWW) |12| in the early 1990's, the Internet has become incredibly popular outside research community at an exponential rate, and its application has become extremely diversified. Today the Internet is evolving into a repository of information (e.g. personal/industrial webpages, news, stock quotes) and value-added services (e.g. dotcoms), a vehicle for new media (e.g. IP-telephony, live audio/video streaming, multimedia conference), and an enabling technology for new types of communications (e.g. virtual-reality based distributed computing/interaction/collaboration, remote service discovery/invocation, mobile computing, home-area wireless network).

Internet applications profit from three main transfer modes: unicast, multicast, and broadcast. The multicast transfer mode fits between the widely-used unicast point-to-point and the broadcast communication techniques. In unicast, data is sent from one source to one destination. In broadcast, data is sent from the source to all other hosts in the network simultaneously (or, more typically, in the same subnetwork; an Ethernet subnet is a physical broadcast medium as a result of its shared bus). For group applications, where more than two users are exchanging messages and routers maintain shared multicast routing state, the number of unicast connections required increases rapidly as the number of users increases. To prevent applications from needing to know about all users in the group, or needing to be responsible for maintaining all these connections, and to decrease network load, we require multicast. Multicast is therefore the efficient emulation of a "broadcast" service to interested users, within the constraints of a network environment.

Multicast allows a source to send data simultaneously to all hosts on the internetwork where users are interested in receiving the data, but in a more efficient manner than simply flooding the entire internetwork with redundant broadcast packets.

The set of all hosts with interested users or participants forms a multicast group. Uninterested hosts that are not in the group do not see the data, perhaps because there is no need for the data to be sent across their subnetwork, or, if they do see it, discard it. An example of this is the logical level of Ethernet, where packets not addressed to the cards network interface or associated groups are ignored.

To communicate the data efficiently to all hosts in the group, each network or internetwork must set up a spanning tree connecting the subnetworks of all interested users, along which the multicast messages can be sent and replicated at tree branches. Construction of spanning trees for multicast protocols has been considered in depth previously for networks, notably in [33] and [34]. Group management is generally separated from tree construction, and becomes an "internetwork" function in Deering's IP multicast group model [33].

Since its introduction in the late 80's [34], multicast has been one of the key features of the Internet Protocol, and there has been a flurry of research activities in such areas as multicast routing and tree construction/management [10, 22, 33, 1, 40, 55, 68, 86], and reliable multicast which adds reliable delivery to a multicast tree [20, 79, 102, 108, 116]. These multicasting protocols are many and varied in their approaches to handling multicast. [19] summarizes early implementations, while [38], [4] and [37] overview the development of multicast and provide a taxonomy of protocols. Reliable multicast transport protocols are categorized in [94]. However, the multicast technology in the Internet has not been developed to its full extent yet; in fact, the concept of multicast is still evolving as we gain more understanding on the nature of multicast communications and their applications.

The new trends on the Internet applications mandate massive redesign and enhancements to the existing Internet architecture so that the users can enjoy scalable, flexible, and more sophisticated services. Moreover, next generation networks are expected to provide service guarantees to those customers willing to pay for the service, but there are complexities for Internet Service Providers (ISPs) wishing to deploy and operate wide scale multicast networks.

In this context, we identify that "multicast" or "one-to-many data delivery" service is one of the key building blocks that needs to be critically reexamined and extended in order to achieve the desired goals. Specifically, multicast technology is one of the key service components for enabling scalable and large deployable services for conventional data-oriented applications (e.g. content distribution [30, 52]), adaptive web caching [121], replicated database update, software upgrade), as well as emerging multimedia applications (e.g. Internet TV, networked classroom, multimedia conference, virtual-reality-based remote interaction, distributed interactive simulation). A number of issues have hindered multicast deployment in terrestrial networks, and are likely to continue to do so, at least for the foreseeable future. These include:

- Difficulties in providing the multicast service across a wide geographical area without affecting the unicast applications;
- Difficulties to invent a scalable multicast routing protocol for use in large multicast networks;
- Difficulties to incrementally integrating the multicast protocols in network routers;
- Difficulties to develop end-to-end multicast protocols for heterogeneous traffic and networks.

In addition to these difficulties, traditional transmission channels have demonstrated their physical, and above all, economic limits for the provision of multimedia services. Compared

with the availability of cable systems and ISDN lines, other technologies (e.g., satellite, xDSL, radio, etc.) can provide these new applications to businesses and households quickly. However despite significant local and backbone improvements using these technologies, there is no evidence that problems will disappear.

The new technologies deployed recently in the Internet such as satellite, radio, HFC (Hybrid Fiber/Coaxial), and WDM (Wavelength Division Multiplexing) have many characteristics that differ from those that the Internet was initially designed for. In particular: loss rate, error rate, asymmetry and unidirectionality affect the performance of Internet protocols and special improvement should be done to overcome problems caused by these inherent characteristics.

In this thesis, we study the deployment of IP multicast in **heterogeneous environments**. We define a heterogeneous environment as a data communication network having distinct characteristics that could be related either to the communication media (generally a static heterogeneity) or to the traffic transmitted in this network (a dynamic heterogeneity).

We first consider the problem of the multicast support in terrestrial networks and mainly the coexistence challenges between unicast and multicast traffic in the Internet. In this large topic, we examine the problem of network resource sharing between both types of services where unicast and multicast competing flows have **heterogeneous traffic characteristics** in terms of applications requirements, multicast group size, receiving capacity, etc. Our target is to develop and to evaluate a set of **complementary mechanisms** that enable the dynamic bandwidth sharing between unicast and multicast flows not only in best-effort networks but also in networks that provide service differentiation.

Then, we focus on the problem of enabling a large deployment of IP multicast over **heterogeneous communication mediums**. Given that the satellites may have an important role in providing the multicast service to complementing next generation terrestrial networks [5, 7, 120], we consider the problem of the multicast support over GEO (Geosynchronous Earth Orbit) satellites. This work is performed in the context of the RNRT French government funded DIPCAST Project[1]. In a first step, we study the behavior of IP multicast routing protocols in transparent GEO satellite systems and we propose a set of techniques to enhance the construction of multicast delivery trees in such architecture. In a second step, we emphasize in the next-generation of satellite-terrestrial hybrid networks that provide On Board Switching (OBS) (where the satellite digitally regenerates the signal and performs data segments switching) and multiple spot beams (where the satellite may send data to one or more antennas directed at different parts of the satellite coverage area). We propose and evaluate a set of **mechanisms and protocols** for ground stations as well as for the on-board processor to allow an efficient multicast forwarding in such type of environment by reducing the load of control and data messages in the satellite segment, while building efficient multicast delivery trees reaching only the spot beams containing the members.

This dissertation presents the motivation of the high level goals, the detailed description of the design of each mechanism and protocol, and the performance evaluation from a set of simulations and measurements in a test-bed. The following section summarizes the key contributions of this thesis.

[1] `http://www.dipcast-satellite.com`

1.1 Thesis Contributions

Regarding the several issues addressed in this dissertation, we were able to make a number of contributions, which we summarize here and discuss in more detail in the following chapters:

- **A Simple and Scalable Bandwidth Sharing Mechanism for Multicast flows:** We have designed and extensively evaluated a simple and scalable single FIFO queue-based active queue management mechanism called MFQ (Multicast Fair Queuing) to achieve the requested inter-multicast bandwidth sharing. For each flow, we assign a dynamic weight which is computed using an inter-multicast fairness function that may implement either a bandwidth allocation strategy or a multicast pricing policy. In our case, MFQ interacts with an external multicast bandwidth allocation module which implements a pre-defined inter- multicast fairness function. To guarantee a fine-grained packet queuing/dropping, MFQ uses a novel bandwidth sharing notion, called "Multicast Allocation Layer" (MAL). Based on this notion, MFQ classifies multicast packets into layers and adjusts their weights in order to provide a bandwidth sharing being as close as possible to that given by the fluid model algorithm.
  Simulation results demonstrate that MFQ achieves the expected allocation for both responsive and non-responsive multicast flows. Without loss of generality, MFQ has been evaluated for both linear and logarithmic bandwidth allocation functions. We have validated our findings by analyzing the impact of network and groups dynamics on the expected bandwidth allocation and comparing it to that obtained by MFQ. Furthermore, we have shown that MFQ converges very fast to a stable state and easily adapts itself to the dynamic change of the flows weights.
  We believe that MFQ combined with a well-accepted definition of the inter-multicast fairness provides a significant step towards a complete and scalable congestion control algorithm for multicast applications.

- **Efficient Network Resource Sharing between Unicast and Multicast Flows:** We have designed, developed, and evaluated a simple scheduler called SBQ (Service-Based Queuing) to share the bandwidth fairly between unicast and multicast flows according to a new definition of fairness between unicast and multicast flows referred as the *inter-service fairness*. We utilized our proposed active queue management mechanism MFQ (Multicast Fair Queuing) to "fairly" share the bandwidth among all competing flows in the multicast queue.
  We have also designed a framework, based on the SBQ scheduler, for enhancing the support of multicast in DiffServ-enabled networks. Our proposal allows the differentiation between multicast flows belonging to the same DiffServ Class while keeping the class identifier unchangeable. This discrimination could be done according to the number of downstream receivers of each active multicast flow.
  The simulation results that we have obtained for heterogeneous sources and links characteristics show that, on the one hand, our scheduler achieves the expected aggregated bandwidth sharing among unicast and multicast service, and on the other hand it allows the multicast flows to be globally TCP-friendly which is a hard criterion for multicast congestion control schemes.

- **Counting the Number of Group Members:** We have proposed and validated a simple extension to the multicast service to explicitly and efficiently count the number of

group members. Each intermediate router consolidates the number that it receives from its children and passes it up the multicast tree until the root receives the total number in the corresponding multicast group. Our proposal, namely, counting the number of members for a multicast group is very useful for service providers in order to implement network mechanisms such as multicast pricing policies and bandwidth sharing strategies like our proposed schemes : MFQ and SBQ. We have extended the host and the router part of the multicast service model to include our proposed extension.
We have validated our proposal through simulation and we have shown that the routers as well as the senders are able to compute the number of downstream members effectively with a reasonable cost.

- **Enhancing the Multicast Routing Protocols over Transparent GEO Satellites:** Concerning the satellite multicast support, we have studied the IP multicast protocols behavior over transparent satellite-terrestrial hybrid networks.
We first presented some undesirable behavior of multicast routing protocols such as DVMRP, PIM-DM, and PM-SM in these networks. For DVMRP and PIM-DM, we identified some configurations where the satellite receivers may receive duplicated packets and we proposed a method to enhance their efficiency. We then developed a configuration policy of PIM-SM in hybrid networks concerning the choice of the list of Rendezvous Points (RPs) and the switching from the RP-rooted tree to the shortest path tree.
We studied the multicast delivery in a concrete transparent GEO satellite-based network (DIPCAST transparent system). We developed a set of adaptation schemes for PIM-SM to enhance its deployment in this network .

- **Supporting IP Multicast in the Next Generation of GEO Satellite Systems:** The new generation of GEO satellites are characterized by the support of on-board switching and multiple spot beams. We proposed a new encapsulation scheme that provides an efficient segmentation of IP packets into MPEG2-TS segments and allows the on-board satellite processor to switch all receiving segments to the appropriate spot beams. Two approaches have been proposed and compared: the self routing approach which consists in switching the incoming data segments based on a switching table maintained by the satellite and the label switching approach which uses a label already included in each data segment by a terrestrial-satellite router to enable the on-board switching. We also designed a new protocol called SMRP (Satellite Multicast Routing Protocol), which is implemented in routers connected to the satellite links and it interoperates with terrestrial PIM-SM routers. Thanks to SMRP, the system entities can make possible the management of multicast sessions, and the switching of multicast IP packets on board a multi-spot GEO satellite having OBS (On-Board Switching) capability. This protocol allows an efficient and transparent integration of satellite links in the Internet.

- **An Enhanced PIM-SM Switching Mechanism:** PIM-SM is the only deployed intra-domain multicast routing protocol that builds both shared and source- based trees. However, it does not provide an efficient mechanism to switch between the two modes. The PIM-SM switching mechanism that we have proposed aims to fulfill both network and receiver requirements. It is a coordination-based mechanism in the sense that all concerned receivers contribute to the switching decision and not only the receiver requesting the switching. Furthermore, the mechanism may use information about the temporary

available network resource provided by an underlying QoS-based unicast routing protocol (if available) or using periodic switching experiments to decide when and how to switch between the two modes based on the receivers QoS requirements. Simulation results have shown that our mechanism achieves its original intentions and provides the inter-receiver fairness.

During this Ph.D., we focused on the deployment of IP multicast in heterogeneous environments in both terrestrial and GEO satellite networks given that the involvement of satellite in IP networks is a direct result of new trends in global telecommunications where Internet traffic will hold a dominant share in the total network traffic.

We believe that the set of protocols and mechanisms that we have designed and extensively evaluated will encourage ISPs to efficiently support the multicast technology in their networks.

1.2 Dissertation Outline

The remainder of this dissertation is organized as follows. Chapter 2 reviews the work related to this dissertation. We first discuss the latest development of the multicast architecture, recent research in network resource allocation for multicast flows, multicast over DiffServ-enabled networks, and IP multicast over both transparent and next-generation GEO satellites. Then, we derive the remaining challenges for the coexistence of unicast and multicast flows in heterogeneous terrestrial networks and the IP multicast support in GEO-satellite based networks which are the two main research problems addressed in this thesis.

From Chapter 3 to Chapter 5, we focus on the first research problem namely: the network resource sharing between unicast and multicast flows in heterogeneous environments.

In Chapter 3, we consider a network environment where there are only multicast competing flows. We propose and evaluate a new active queue management mechanism called MFQ (Multicast Fair Queuing) which shares the available bandwidth fairly between the flows using on a pre-defined inter-multicast fairness function based on the number of members in each multicast group. The queuing and the dropping decision are designed in a manner to provide a bandwidth sharing which is as close as possible to that achieved by the fluid model algorithm.

In Chapter 4, we address the problem of network resource sharing among unicast and multicast flows. We present and evaluate a new simple scheduler called SBQ (Service Based Queuing) that uses two queues for both type of services and through a dynamic configuration of queue weights it is able to share fairly the bandwidth. We use a new notion of unicast and multicast fairness called inter-service fairness that aims to take into account both max-min fairness and TCP-fairness criterion. The bandwidth sharing between multicast flows is guaranteed by adopting MFQ, that we detail in Chapter 3, in the multicast queue. We also propose a technique to efficiently use SBQ in DiffServ-enabled network and to re-mark IP Multicast packets in DiffServ core routers according to the number of downstream members.

In Chapter 5, we investigate the challenge of counting the number of members in multicast groups. While this information is useful for many network services, in our case we consider it as a complementary building block of MFQ and SBQ. Indeed, in MFQ, this information is needed to implement the inter-multicast fairness function and in SBQ is useful for re-marking multicast packets in DiffServ Networks. We extended the host and the router parts of the multicast service model to allow both senders and intermediate routers to collect the exact number of downstream members.

From Chapter 6 to Chapter 8, we focus on the impact of the heterogeneity of communication mediums on the large multicast deployment.

In Chapter 6, we consider the use of transparent GEO satellites as a transmission medium in order to provide multicast delivery to users connected through terrestrial networks. We mainly focus on the problem of multicast routing in such type of environment. Basically, we examine the behavior of PIM-SM, PIM-DM, and DVMRP protocols and we identify some of their undesirable behaviors. We propose a set of adaptation techniques to avoid these behaviors so as to build efficient multicast delivery trees.

In Chapter 7, we turn our attention to the multicast support in next-generation of GEO satellites. We develop a framework that enables the support of multicast in GEO satellite-based hybrid networks. Our proposed framework is composed of two complementary blocks. The first one deals with the encapsulation scheme used by satellite uplink stations to segment IP multicast packets into MPEG data segments, while the second one implements an adaptation of PIM-SM multicast routing protocol in the satellite segment.

In Chapter 8, we emphasize on the switching between the two modes of PIM-SM namely; the source-based tree mode and the shared tree mode. We propose a new switching mechanism which can be applied in both terrestrial and satellite networks. Our mechanism coordinates between all members concerned by the switching in order to make the switching decision. Moreover, it may take into account the available information about the links characteristics and the members QoS requirements in the switching procedure. We develop the specifications of all new PIM-SM messages needed to take advantages from our proposal and we explore the issue of its incremental deployment in the current networks where routers implement the standard PIM-SM protocol.

In Chapter 9, we present a general summary of the work achieved and the conclusions concerning the results obtained during this thesis. Some perspectives and open questions are given for the continuation of this work in the area of multicast deployment in both terrestrial and satellite networks.

Chapter 2

Trends and Challenges in IP Multicast Deployment

There has been a wide spectrum of work during the last decade on multicast in packet switched networks and a special interest in using satellite links for data transmission. This chapter only presents a survey of related trends and issues that serves as background to our research work.

We provide in Section 2.1 a brief overview of IP multicast and we review the recent development in IP multicast routing and the challenges of the coexistence of unicast and multicast in terrestrial networks. After giving some historic about satellite communications, we describe in Section 2.2, the architecture of both transparent and next-generation GEO satellites. In Section 2.3, we address the current status of multicast development in terrestrial networks and we report the main problems that we examine in the first part of this thesis. The advances and the remaining challenges of the IP multicast deployment over GEO satellites will be discussed in Section 2.4. We also identify in this section the issues related to this area on which we focus in the second part of this thesis. Section 2.5 summarizes this chapter.

2.1 IP Multicast - An overview

2.1.1 A Brief Description

The transmission service with which most network users are familiar is point-to-point, or unicast service. This is the standard form of service provided by networking protocols such as HDLC and TCP [98]. Somewhat less commonly used is broadcast service. Over a large network, broadcasts are unacceptable (because they use network bandwidth everywhere, regardless of whether individual subnets are interested in them or not), and so they are usually restricted to LAN-wide use (broadcast services are provided by low-level network protocols such as IP). Even on LANs, broadcasts are often undesirable because they require all machines to perform some processing in order to determine whether or not they are interested in the broadcast data.

Group communication applications involve multiple participants and typically requires delivering the same data to all or some subset of the participants. Examples of such applications include interactive games (e.g. Quake), audio/video conferencing, TV/video broadcast (e.g. netcast of Rolling Stone concert), collaborative work environments (e.g. Whiteboard), file distributions (e.g. Gnutella), and content distributions (e.g. stock updates). As more and

more users come on-line, there has been a significant increase in the number of group applications. Unfortunately, network services to support group applications are severely lacking in today's Internet. Today's network architecture supports best-effort unicast delivery, which means the packets are addressed to a single machine, and the job of network routers is to route and forward packets towards their destinations. Unicast cannot provide efficient data delivery for group communication applications because the same data are sent multiple times so the application's bandwidth consumption increases linearly with group size. Large bandwidth consumption creates congestion in the network and hampers the performance of the application. In addition, the sender application must create and maintain connection channels to all receivers, which can consume a prohibitive amount of host resources for large sized groups. Today's unicast architecture is inadequate to support large scale group communication applications.

IP multicast was introduced more than 12 years ago by Steve Deering [33, 34]. IP multicast delivers a single packet, originated from a sender, to multiple receivers in a group. Multicast packets are duplicated in the network only when necessary, therefore no bandwidth is used to transmit redundant data. IP multicast addresses occupies a special part of IP address space (i.e. Class D IP address), where each multicast address represents a single multicast group. The multicast address scheme shields the multicast sender from group membership information and group management tasks. The multicast backbone (Mbone), a virtual multicast-capable inter-network situated on top of the conventional network, debuted in 1992. The Mbone provides a test-bed for researchers to test and develop multicast protocols and applications.

Individual hosts can join or leave a multicast group at any time. There are no restrictions on the physical location or the number of members in a multicast group. A host can be a member of more than one multicast group at any given time and does not have to belong to a group to send packets to members of a group.

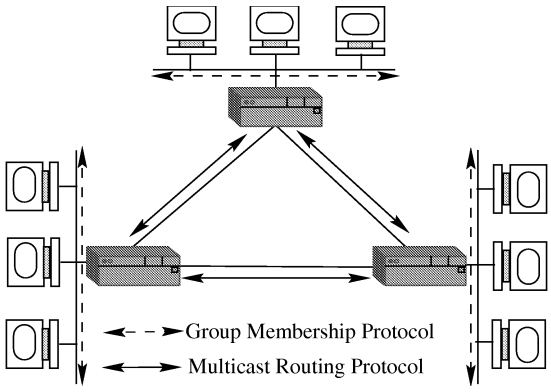


Figure 2.1: IP multicast standard model: the Internet Group Management Protocol (IGMP) protocol is used between routers and directly connected hosts and the multicast routing protocols run between connected routers.

As shown in Figure 2.1, IP multicasting protocols rely on the Internet Group Management Protocol (IGMP) [22, 34, 53] to manage multicast groups. When a host joins a multicast group, it transmits a group membership protocol message for the group or groups that it wants to receive and sets its IP process and network interface card to receive frames addressed to the multicast group. On the other hand, the multicast routing protocols are responsible

for building the multicast delivery tree that reach all members in the multicast session.

Within the vast domain of IP multicast technology, the issues that we address in this thesis require the review of two main multicast-related areas:

- the coexistence of unicast and multicast flows in the Internet from the point of view of network resource sharing, and
- the widespread deployment of existing multicast routing protocols.

These two key elements will be detailed in the next two sub-sections.

2.1.2 Coexistence of Unicast and Multicast Flows

A multicast datagram is delivered to destination group members with the same best-effort reliability as a standard unicast IP datagram. This means that multicast datagrams are not guaranteed to reach all members of a group or to arrive in the same order in which they were transmitted. However, having both unicast and multicast applications sharing the network resource raises many open issues.

First, giving that the network resource are in general very limited in the Internet backbone, it is not trivial that they will be shared "fairly" among competing flows. When the input traffic of a communication link exceeds its capacity a well know-phenomena appears which called: *congestion*. This problem can be well solved if each application uses a congestion control mechanism that detects and reacts to the congestion events. The typical flow/congestion control mechanism in the Internet is TCP (Transmission Control Protocol) [98]. Recent statistics predict that more than 80% of the Internet traffic is generated by applications that use the TCP protocol. In fact, the unicast applications which are the most popular in the daily uses are the Web, e-mail, etc. which require the TCP protocol for reliability and rate and congestion control. A natural criterion when developing a new congestion control mechanism is to respect the behavior of TCP under the same conditions. This can be done if this mechanism always guarantee that the achieved throughput does not exceed that of a TCP connection. Research work in this area include for example TFRC (TCP-Friendly Rate Control) protocol [57] for unicast applications and TFMCC (TCP-Friendly Multicast Congestion Control) protocol [119] for multicast applications which is an equation-based mechanism that extends the TCP-friendly TFRC protocol and it improves upon the well-known approach of using exponentially weighted random timers by biasing feedback in favor of low-rate receivers while still preventing a response implosion.

Second, there is a strong application demand for reliable multicast. Widespread use of the Internet makes the economy of multicast transport attractive. The current Internet multicast model offers best-effort many-to-many delivery service and offers no guarantees. One-to-many and few-to-few services may become more important in the future. Reliable multicast transports add delivery guarantees to the group-delivery model of multicast. Examples of multicast applications that could use reliable bulk multicast transfer include collaborative tools, distributed virtual reality, and software upgrade services. However, the expected reliability is not necessarily like that of reliable unicast TCP. Indeed, some multicast applications such as videoconferencing and video-on-demand do need a full reliability as that provided by TCP but, in the contrast, they can accept a maximum loss rate which depends on the nature of the application. Several classes of RMT (Reliable Multicast Transport) techniques have been

proposed to the IETF for standardization such as MFTP (Multicast File Transport Protocol) [87].

Finally, multicast routing protocols are largely different from unicast routing protocols in the sense that they try to build a multicast delivery tree serving more that one receiver.

In the following section, we review the main existing multicast routing protocols which serve as background for the second part of this dissertation addressing several issues related to the multicast routing in GEO satellite-based networks.

2.1.3 Existing IP Multicast Routing Protocols

The IP multicast routing protocols can themselves be divided into two basic sets of groups, depending upon a taxonomy predicated on the basic assumptions made about traffic use in the spanning tree, and on the distribution of multicast group members throughout the network. These are that the multicast tree is either source-based or a shared tree, and that it is either sparse (assumes few group members in the internetwork) or dense (assumes many members in the internetwork). Source-based trees are generally dense, while core-based trees are generally assumed to be sparse.

In the following, we summarize both type of multicast routing protocols.

Source-based trees

Source-based tree multicast protocols are data-driven or source-initiated. Construction of the multicast spanning tree begins top-down from the source outward as it transmits information, and data on the state of the tree is flooded to all routers. Routers on subnetworks with no interested members prune back by requesting the tree no longer reach them[1]. Source-based protocols include:

- **Distance Vector Multicast Routing Protocol (DVMRP)** [118], where every host on the network is initially assumed to be part of the multicast group receiving traffic from the source.The tree is then pruned to an optimal state, connecting only interested networks, via the use of Reverse Path Forwarding (which requires bi-directional links for pruning, although the resulting tree is still unidirectional from source to destinations). The spanning tree effectively begins as an uncontrolled broadcast from the source that is then cut back to a more efficient multicast state. The periodic discovery floods required by this protocol to set up trees would be undesirable in a wide-area network. Each sender in a multi-way multicast would require its own tree to be set up, as the spanning tree is for one-way delivery of communication from a single source. DVMRP also requires that all routers receive and maintain state on every multicast group, and so can be expected to scale badly for large networks where group members are widely separated.
  The initial flooding assumes implicitly that potential group members are densely distributed throughout the internetwork, i.e. that many subnetworks contain at least one group member and will be interested in receiving the communication, making the imposition of state concerning the multicast tree and the resulting network overhead worthwhile. For full group-to-group communication using source-based multicasts, a separate

[1]This results in considerable soft state overhead, as uninterested routers must continually remove newly-received state concerning new multicast trees.

multicast tree must be set up for each source, or (source, group) tuple. This scales badly for large groups and imposes considerable joining and leaving overhead (build a new tree, then destroy it) if we are considering group communication between peers. DVMRP is widely implemented on the MBone. It is derived from and relies on characteristics of the earlier RIP, the Routing Information Protocol, its unicast equivalent [81].

- **Multicast Open Shortest Path First (MOSPF)** [89] is based upon OSPF [88]. OSPF routes messages along a least-cost path, where cost is expressed in terms of routing metrics that can represent such things as the amount of traffic on the link, or the latency involved in using the link. MOSPF relies on OSPF and uses the Dijkstra algorithm to compute a shortest-path tree. However, MOSPF floods group membership information across the routing domain periodically, and so does not scale well, although the requirement that all MOSPF routers have a complete topology map and know all the locations of members is not infeasible in the single autonomous system.

Core-based trees

Core-Based Trees (CBT), with one or more central routers from which the tree branches out in all directions, have been suggested for groups where there are many active senders within the group, allowing multi-way communication over a single tree. An architecture for core-based trees is described in [11], while a related protocol is specified in [10].

Core-based-tree multicast protocols have receiver-initiated multicast spanning trees based on explicit join and prune messages, where a router becomes involved in a branch of a multicast distribution tree only when one of the hosts on its subnetwork explicitly requests membership by issuing a join message. There may be one or more central core routers that receive join and leave messages, and that pass received multicast packets downstream through the tree. The lack of any initial flooding and the assumptions of constrained capacity and fewer interested members, sparsely distributed, mean that these shared trees are more scalable for internetworks than source-based trees. In a source-based tree, every active source is associated with its own tree. This results in a scaling of O(set of sources $\star$ set of members for each source group). In a shared tree, the scaling will be O(all group members) with less state held in the network. However, there is a delay tradeoff in going from shortest-path trees, based directly on the underlying routing protocols, to shared trees, where all multicast traffic must travel via the core. This single shared-tree (*, group) approach differs from the (source, group) pairings of source-based trees such as DVMRP and MOSPF. Any source wishing to send data transmits it to the core, which then multicasts it to all receivers in the group via the tree. Any source that is not already a tree member will encapsulate the multicast packet in a unicast packet addressed to the core.

This prevents the need for all routers in the network to know the location of the core, and decreases the amount of multicast tree state that must be held in the network. Choosing an appropriate position in the network for core routers for the multicast group, or changing the core positions as the receiver set/network topology changes, a non-trivial problem. A good core position decreases the amount of multicast state routing information that needs to be stored and the number of routers involved. Protocol-Independent Multicast Sparse Mode (PIM-SM) [40] constructs multicast tree around a chosen router, called a rendezvous point,

similar to the core CBT[2]. However, PIM-SM allows shortest-path source-based trees as well as shared-group trees. In PIM-SM, a multicast packet is always encapsulated in a unicast packet sent to the rendezvous point address, whether the source of the packet is a group member or not, since PIM-SM state held in routers participating in the tree is downstream only.

Exterior protocols

The sparse or dense, source- or shared-tree protocols that have already been discussed are for use within a single managed network, or domain. A GEO satellite-based network forms one such domain. Exchanging information about a multicast sources with other domains to enable branches of spanning trees crossing multiple administrative domains to be established an entirely different problem, and has resulted in creation of protocols designed to address that problem. These include multi-protocol extensions to the Border Gateway Protocol (MBGP) [13] to exchange multicast RIB (M-RIB) tables, and the Multicast Source Discovery Protocol (MSDP) [86] to discover multicast sources in different domains.

To join shared trees of the same group in different domains together and establish a root domain in which the core of the resulting shared tree is placed, the Border Gateway Multicast Protocol (BGMP) has been proposed [114].

2.2 Satellite Communications - An overview

2.2.1 A Brief History and Description

The field of satellite communication systems is a rich, multidisciplinary field involving several areas of electrical, aeronautical, and mechanical engineering. Several books provide overviews of the field as a whole; among the works in wide use today are those by Maral and Bosquet [83], Gordon and Morgan [62], and Pratt and Bostian [100].

The first idea of using a satellite orbiting at a geostationary altitude (35,780 km above the equator) to provide communication services is attributed to author Arthur C. Clarke in 1945. The first artificial communications satellite (SCORE in 1958) did not follow long after the Sputnik launch in 1957, and the first commercial geostationary satellite (INTELSAT 1, or Early Bird) in 1965 ushered in the era of overseas telephony via satellite. This first INTELSAT satellite had a capacity of 480 telephone channel at an annual cost of $32500 per channel [83].

In the 1970s and 1980s, both the market for satellite communication services and the technology grew rapidly. Besides providing international telephony and data services between large earth stations owned by national carriers, communication satellites were increasingly used for video (television) distribution. The international organization INMARSAT was founded to provide telephony and data services to maritime customers. The first satellite network experiments based on packet switching (the Atlantic Packet Satellite Network, or SATNET) commenced in 1976. Finally, the construction of systems based on Very Small Aperture Terminals (VSATs) for transaction-oriented traffic such as credit card verification and database management was begun in the 1980s.

In the 1990s the growth of alternative, cheaper technologies such as high speed fiber optic networks has gradually eliminated much of the international telephony service for non-mobile customers. However, technological advances enabled the creation of direct-to-home (DTH)

[2]PIM-SM is distinct from Protocol-Independent Multicast Dense Mode, or PIM-DM [1]. PIM-DM is similar to DVMRP, but relies more upon the underlying unicast routing protocol.

satellite television services that are competitive with cable television systems. And because of the explosion of interest in the Internet, in the latter 1990s satellite channels have begun to be used for trucking between international Internet Service Providers (ISP) and the Internet backbone.

The latter half of the 1990s has seen a resurgence of interest in satellite-based data networks. Satellite communication systems have long been one of the hallmarks of advanced communications technology, with their remarkable and distinctive ability to link most of the populated areas of the Earth. Yet, until recently, the satellite communication industry had increasingly begun to look more like a dinosaur, with competition from fiber optic and terrestrial wireless networks steadily "eating" away at the industry most profitable markets.

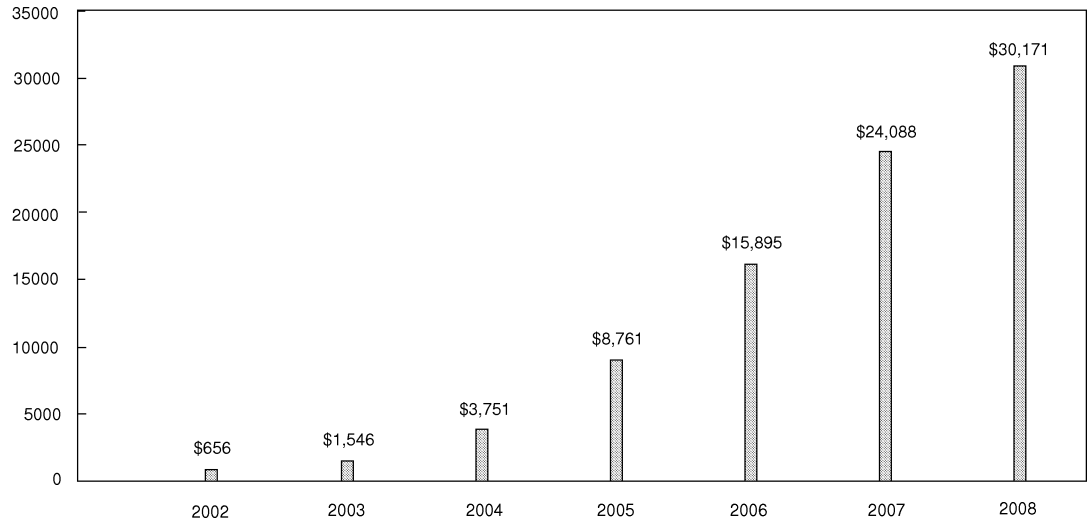


Figure 2.2: Projections of Annual Satellite Broadband Revenue (Source: Merril Lynch and Co.)

As of this writing, however, the satellite industry is poised for rapid growth, with funding in place for the deployment of technically ambitious, multi-billion dollar systems, and growing competition in both service provision and hardware manufacturing. As illustrated in Figure 2.2, some analysts project that the growth in the satellite industry will outpace the growth of the entire communications market over the next six years, as the satellite sectors market share rises from 2.3% today to 6% a six years from now [91]. What has triggered this rapid turnaround? The answer lies in the confluence of two economic and technological trends:

1. **The Internet boom** The 1990s will likely be remembered as the decade during which the Internet came of age. There is presently an incredible (and increasing) demand for faster and cheaper Internet services, and many companies are scrambling to offer these broadband services. Satellite networks provide a fast way to reach customers because they do not rely on buildout of a high-speed terrestrial network, which may take years to accomplish in many areas of the world. Moreover, with the advent of the World Wide Web [12], broadband Internet access tends to be highly asymmetric in traffic usage, with users downloading (consuming) much more information than they generate. As we shall discuss, this type of traffic pattern matches well with satellite networks, where it is much cheaper to receive data at broadband rates than to transmit at such rates.

2. **Advances in satellite technology** The rapid technological progress that has spurred the growth of the Internet has also helped to significantly advance the state-of-the-art in satellite technology. Most notably, miniaturization of electronics has allowed more and more sophisticated satellite and terminal hardware to be economically deployed. Satellites, which once were mainly repeaters in space, have much more on-board processing functions and have the capability to juggle multiple directional spot beams on the Earths surface while also communicating with other satellites via high frequency radio links. Sophisticated constellations of low-earth-orbiting satellites, and handsets and terminals that can track the motion of these satellites, are now being designed and expected to be deployed in the next few years. New frequency bands at 20-30 GHz, the use of which was once precluded by the lack of affordable high-frequency hardware, are now being opened to satellite communications, greatly increasing the available bandwidth of newer satellite systems. Probably the most widely-observable evidence of the impact of advances in electronics on this industry can be seen in the growth of direct-to-home (DTH) satellite television services, one of the most rapidly deployed consumer electronics product in history. DTH services, using small, affordable satellite dishes, have brought satellite services into the mainstream in a way that was not possible using technology of a decade ago.

As mentioned earlier, we focus, in the second part of this thesis (from Chapter 6 to Chapter 8), on supporting IP multicast delivery in data networks that include GEO satellite links. We describe in the next two sub-sections, the two generations of GEO satellites which we consider in our study. Typically, we emphasize on GEO transparent satellites where the satellite has only to retransmit the received signal to the downlink and next-generation of GEO satellites that implement more sophisticated technologies.

2.2.2 GEO Transparent Satellites

Geostationary satellites are fixed with respect to a terrestrial observer and they are on an equatorial circular orbit at about 36,000 km altitude. Theoretically, only three GEO satellites are sufficient to serve all the earth. Unfortunately, this type of coverage is not reliable for areas at North or South latitudes greater than approximately 70°, because the elevation angle falls below the minimum acceptable (considered here to be in the region of 10°) and, therefore, the link is not reliable.

Transparent (or Bent pipe) GEO satellites act as a relay station in space. People use them to bounce messages from one part of the world to another. Signal is amplified and retransmitted at the satellite, but there is no improvement in Signal-to-Noise (S/N) ratio, since there is no demodulation, decoding or other type of processing.

The main aspects that characterize the orbital configuration of GEO transparent satellites are summarized below.

Advantages:

- the simplest space configuration;
- good system modularity (one satellite to cover regional areas, three satellites to provide global coverage);

- the spot-beam footprints on the earth are fixed (due to the stationary of satellites) and they are so wide that we can consider that a Mobile Subscriber does not change the spot-beam during the call lifetime (i.e. no handover procedure);
- simple space control system;
- no tracking system at the earth stations;
- no variation of propagation delay and elevation angle;
- negligible Doppler effects.

Disadvantages:

- problematic links feasibility due to the long satellite-user distance (prohibitive power levels and/or too large on-board antennas could be required if low power hand-held user terminals are considered);
- high propagation delays (higher than 400 ms recommended by CCITT, in case of double hop mobile-to-mobile communications);
- low minimum elevation angles at high latitudes (i.e. polar regions can not be covered).

In Chapter 6, we study the behavior of IP multicast routing protocols that we have discussed in Section 2.1.3 in data networks integrating transparent GEO satellite links.

2.2.3 Next-Generation GEO Satellites

The next-generation of satellite-terrestrial hybrid networks are expected to provide multimedia (e.g. voice, video, data) services to the ground user. A next-generation satellite performs tasks like demodulation and decoding which allow signal recovery before retransmission. Since the signal is available at some points in base band, other activities are also possible, such as routing, switching, etc. Most of such satellites in operation today or planned for deployment in the nearest future are characterized by the support of two interesting technologies:

- **On-Board Switching (OBS):** OBS provides a method of "network" switching on the satellite. It can be used to actively control signal routing. To cope with high network transfer rates, the switching of packets from an uplink to a different downlink spot beam needs to be done at a very high speed. There are two type of satellite switched network implementation: (1) Fully Switched where the satellite does all the processing and there is no or very little ground control. (2) IF or RF switching that involves electronically controlled RF/IF switches, which can be configured on a near-real-time basis via Ground Control.
  Note that our proposed schemes concerning the support of IP multicast for the next-generation of GEO satellites do not depend on the technology used by the satellite to handle the switching.
- **Spot-beams technology:** instead of covering the whole footprint of a satellite by a global beam, the beam is divided into a number of spot beams. The benefits of spot beams are twofold: a) the power requirements of user terminals are reduced, thereby

permitting the use of smaller antennas in the ground segment and reducing cost; and b) the frequency between beams can be re-used, thereby increasing the capacity of the space segment.

Chapter 7 of this dissertation emphasis on the multicast routing in such type of GEO satellites.

2.3 IP Multicast over Heterogeneous Terrestrial Networks

2.3.1 Current Status and Remaining Challenges

During the 1980s and early 1990s, cellular telephone technology had limited reach and use. Fast forward to 2000, and it seems everyone owns a cell phone. IP Multicast today is the same as early cell phone technology - useful and available, but not ubiquitous.

Though it has distinct bandwidth-saving advantages over other modes of transmission, IP Multicast has not taken off like many had predicted when the technology was first introduced in Steve Deering's doctoral dissertation in 1988.

The technology can reduce traffic on the corporate network by eliminating redundant access to the same content. Multicast can also reduce the load placed on network servers.

After a long period of very useful experimentation using the MBone, commercial deployment of multicast services is expected to be generalized in the next few years. The initial design of multicast was motivated by the need to support one-to-many and many-to many applications in a scalable fashion. Such applications cannot be serviced efficiently with unicast delivery. The commercial design of multicast must now include the market requirements of ISPs and their customers. ISPs require a service and a protocol architecture that is easy to deploy, control and manage, and that scales well with the growing Internet. ISP customers expect to be the sole owners of multicast addresses, if only temporarily, to have protection from malicious network attacks and thefts of service and content, and to be able to correct network problems quickly. A deployable architecture should be driven by these concerns. The current multicast service architecture does not consider these concerns well. It lacks simple and scalable mechanisms for supporting:

- Access controls: including group creation and membership. ·
- Security: for protection against attacks to the routing and data integrity of multicast datagrams.
- Address allocation.
- Network management.

Such mechanisms are not well developed at this stage. Many of the mechanisms in the current architecture that address these issues do so too broadly because they consider both the multipeer and single source models. Applications that are most popular today are one-to-many, such as file transfer, streaming media, and information push. Many-to many applications at this point mainly consist of less popular DIS and serverless multiplayer games. Conferencing over the Internet remains few-to-few but is currently better supported by unicast.

2.3.2 Problem Statement and Scope

As mentioned above, despite extensive work on multicast deployment, there are still several challenges problems that require further investigations. Those we consider in this thesis are:

1. **Multicast bandwidth sharing:** one of the open multicast issues that remain without a well accepted solution is the bandwidth sharing between multicast competing flows. In [75], authors proposed and compared a set bandwidth sharing strategies based on the number of downstream receivers in each intermediate router. The same idea was mentioned in [108] as a new criterion for implementing end-to-end multicast congestion control mechanisms. One of the goals of this thesis is to develop an efficient mechanism that shares the bandwidth "fairly" according to a pre-defined inter-multicast fairness function.

2. **Multicast and unicast fairness:** as recommended by the IETF [82], all proposed policies for multicast bandwidth sharing should take into account the TCP-friendliness criterion. Currently, there is no mechanism which is able to provide both an efficient bandwidth sharing among multicast flows, while guaranteeing that each multicast flow gets a bandwidth share at most equal to that of a TCP connection under the same conditions. Our target is to develop a simple scheme that shares the bandwidth "equally" between aggregated unicast and multicast flows, while respecting the TCP-friendliness criterion.

3. **Multicast in DiffServ networks:** another issue that we consider in this thesis is the integration of two key technologies, namely, multicasting and Differentiated Services (DiffServ). Although both are complementary technologies, the integration of the two technologies is a non-trivial task due to architectural conflicts between multicasting and DiffServ. We aim to enhance the integration of both technologies by developing an extension to the DiffServ architecture in order to handle differently the multicast packets, while keeping its original intentions.

In addition to these problems, we also address the problem of counting the number of group members. This membership information represents a key component of the three solutions that we develop for the three above problems, thereby we fixed as a strategic objective the development of an extension of the multicast service model to count the number of downstream members at the senders as well as the intermediate routers in the multicast delivery tree. The information of group size is interesting for other applications such as feedback suppression mechanisms in reliable multicast transport protocols and multicast pricing models.

Note that while the addressed problems are examined separately, their solutions are in fact complementary to each other.

2.4 IP Multicast over Satellites

Before introducing the problems of the area of the deployment of IP multicast over satellites that we address in the dissertation, we first describe the different remaining challenges in integrating satellite links in the terrestrial Internet.

2.4.1 Current Status and Remaining Challenges

The term "Internet" refers to a wide collection of packet switching networks that are tied together through the common use of the Internet Protocol (IP) |97| and its associated routing and addressing conventions. Each "network" can be thought of as a separate autonomous system that takes responsibility for delivering traffic within its own network however it sees fit while conforming to standard protocol mechanisms at exchange points (interfaces) with other participating networks. The most distinguishing characteristic of this network architecture is that it is decentralized and has no single administrator. Another key aspect of the architecture is how the various protocols interrelate.

Satellite communication technology has been developed for nearly 50 years. Over the past few years, the demand to use satellite devices to access the Internet is growing because satellite communication can deliver Internet services to consumers and institutions in remote areas of the world not covered by good terrestrial connectivity. In addition, satellite broadcast system is ideal for multicast service, satellite access method is also suitable for asymmetric Internet data transmission, service providers can use satellite ocean range beam to easily extend their network nationwide and ocean wide without last mile problem. In the mean time, operation expense is not related to the distance. Due to the above reason, satellite technology will be more utilized in Internet transmission. In this thesis, we will explore only multicast problems involving protocols that lie at the network layer.

The IETF Pilc WG studies mainly the implications of link characteristics on the performance of transport protocols by providing general recommendations documents for the Internet community. However, it has not considered the problem of dynamic routing over unidirectional and which has been the main focus of the IETF UDLR Working Group. Since the Internet routing model was designed to work properly only over bidirectional links, the integration of unidirectional satellite links such as satellite (with no feedback channel) and some HFC links in the Internet can not be guaranteed by simply adding them like bidirectional links. Indeed, both unicast and multicast routing protocols forward packets on interfaces from which they have received routing control protocol |81|, |88|, |118| messages. If a node receives a routing information packet on one interface, the routing software often assumes that the source (sender of the packet) is directly reachable via the same interface. This assumption breaks down in a network with unidirectional links (UDLs). In a network with UDLs, a packet containing the routing information might be received via one link but it might not be possible to transmit packets on that link.

Two approaches have been discussed in the IETF UDLR working group |39|. The first approach proposes to add necessary modifications to the current dynamic routing protocols, while the second one proposes to develop a link layer tunneling mechanism (LLTM) which can be used regardless the upper Internet layers. The later mechanism was adopted by the working group since it performs well and it captures the dependence of network layer on the bidirectionnality of the communication link. Although the Generic Encapsulation Encapsulation (GRE) |51| mechanism is the recommended method to use as an encapsulation method, it is worth noting that the idea of link layer encapsulation can be used in conjunction of other encapsulation mechanisms.

2.4.2 Problem Statement and Scope

While the use of satellite networks as a part of the Internet backbone dates back almost thirty years, the use of satellites to provide high-speed network access is relatively new. The success of new satellite networks in delivering high-speed access hinges on the ability of the underlying protocols to function correctly and efficiently in the satellite environment, an environment characterized by (for traditional geostationary (GEO) satellites) much longer propagation delays than are found in terrestrial networks, and (for newer low-earth-orbiting (LEO) satellites) a rapidly time-varying network topology.

In the previous subsection, we presented a taxonomy for Internet Support over satellite and addressed many related work in many areas of Internet Protocols over satellite that are more relevant to this dissertation, in particular:

- Challenges in routing over unidirectional satellite links
- Multicast routing over satellite
- Multicast over the new generation of satellites

Previous research efforts that focused in these problems present a rich spectrum of work in this area, none of these contributions presented a comprehensive solution for delivery of multicast applications over satellite networks in an efficient fashion.

In this dissertation, we focus on the use of GEO satellite systems to provide multicast delivery to end users, and we address in particular on two problems relevant to Internet data networking over these broadband satellites i) enhancing the IP encapsulation scheme, and ii) improving the IP multicast routing.

We propose a new encapsulation scheme that provides an efficient segmentation of IP packets into MPEG2-TS segments and allows the satellite to switch receiving packets to the appropriate spot beams in the on-board satellite. The satellite maintains a switching table containing the list of outgoing spot beams for each active session.

Another area that we investigate is the adaptation of PIM-SM multicast protocol for satellite networks that use the DVB-S standard [41]. We develop a new protocol called SMRP (Satellite Multicast Routing Protocol), which is an adaptation of PIM-SM protocol for satellite networks. Thanks to SMRP, the entities can manage multicast sessions and the switching of IP multicast packets in the on-board satellite processor (OBP). SMRP is effective not only for satellite networks, but also for hybrid networks combining ground and the satellite communication links.

2.5 Chapter Summary

We presented in this chapter an overview of IP multicast over heterogeneous terrestrial and satellites networks. A survey of the different challenges that face the large deployment of multicast in the Internet have been given. We mainly focused on explaining the problems that we evoke in this dissertation.

In fact, in this thesis we propose a set of contributions in the area of multicast deployment in the Internet. We consider two research problems: the first problem is related to the network resource allocation for multicast flows, while the second one concerns the support of multicast in transparent GEO satellites and the next-generation GEO satellite systems integrating the on-board switching and the multiple spot beams technologies.

Chapter 3

Multicast Bandwidth Sharing

3.1 Introduction

One of the major barriers for the wide-range deployment of multicast is the lack of an effective mechanism which enables multicast traffic to share the network resource reasonably fairly with TCP unicast connections. The basic problem can be defined as the following: Consider the transport protocols of different multicast connections with one sender and multiple receivers over the Internet, the network has to share the bandwidth between different multicast connections.

A key component of the solution of this problem is the development of a multicast fairness function to share "fairly" the bandwidth between multicast competing flows. Currently there is no consensus on the fairness issue between multicast flows, let alone useful quantitative definition. Should a multicast session be treated as a single session which deserves no more bandwidth than a single TCP session when they share network resource? Or, should the multicast session be given more bandwidth than TCP connections because it is intended to serve more receivers? If the later argument is creditable, how much more bandwidth should be given to the multicast session and how do we share the bandwidth between multicast flows that have different number of receivers?

It was pointed out in the RMRG (Reliable Multicast Research Group) meetings[1], that we may just have to leave the slow responsiveness of multicast flow and only try to achieve fairness with TCP in a long run. Yet, it is not clear how the network and other flows will sustain during this period of overload and how long the period should be. In addition, the IETF orchestrated a very strong guideline for developing a TCP-friendly multicast congestion control scheme [82] regardless of the number of group members. It is well-known that multiplicative decrease/linear increase congestion control mechanisms, and in particular TCP, lead to proportional fairness [74]. However, in [25] the author has proven that if we treat the multicast flow as if it were a TCP flow (as the TFMCC single-rate multicast congestion control protocol does [119] based on the TFRC equation-based unicast congestion control proposed in [57]), then the application of Kelley, Maullo and Tan's model [74] shows that the larger the multicast group the smaller its share of the proportional bandwidth would be. Thus, we believe that the definition of the multicast fairness function should take into account the number of receivers.

We study the general problem of the coexistence of unicast and multicast applications in

[1] `http://www.east.isi.edu/rm/newindex.htm`

the Internet from two aspects. Indeed, in this chapter, we focus on the bandwidth sharing among multicast competing flows and in the next chapter we examine the issue of network resource sharing between unicast and multicast flows.

To fairly distribute AF (Audio Frequency) resources between multirate traffic generate by audio/video codecs[2], the max-min fairness definition [14] could be used since its formal definition is a well accepted criterion for fairness and its multicast definition [115] was extended to include multirate sessions [103]. However, Rubenstein et al. [103] show that max-min fairness can not be provided in the presence of discrete set of rates, as is the case of multirate sources.

The maximal fairness definition presented by Sankar et al. [106] could be applied in the presence of a discrete set of rates, but it does not consider the number of receivers in each session. Therefore, maximal fairness cannot maximize network resource utilization and at the same time maximize the number of receivers with good quality level.

Li et al. present [76] another proposal to improve inter-session fairness based upon the maxi-min fairness definition. Besides max-min fairness limitation with discrete multirate sessions, this proposal only considers one shared link and does not consider the number of receivers and layers importance of a session.

We argue that the definition of the inter-multicast fairness (fairness between multicast flows) should take into account the number of competing groups, the number of flows per group, and the number of receivers per flow [108]. In [75], W. Biersack et al. have defined three different bandwidth allocation strategies for multicast flows as well as criteria to compare these strategies. They showed that the LogRD policy[3] always leads to the best tradeoff between receiver satisfaction and inter-multicast fairness. To implement their proposal in real networks, they recommended to introduce their allocation scheme to the GPS scheduler [96] by configuring the weights according to the expected share provided by the fluid model algorithm. The goal can also be met by reserving the bandwidth in the network for either individual connections or group of connections, and explicitly allocating network bandwidth on a packet-by-packet basis by scheduling packets across network links. However, the two methods are complex because the former requires the use of Fair Queuing mechanisms [36, 96] in each router and the later the use of RSVP-like bandwidth reservation signaling protocols which needs a close coordination and integration between all routers (and hence all network providers) along the path from sender to receiver.

In this chapter, we investigate an alternate approach based on Active Queue Management (AQM) rather than packet scheduling or explicit bandwidth reservation. We develop a new active queue management mechanism for routers that (1) achieves the expected bandwidth allocation between multicast flows, (2) adapts to the change in multicast group sizes, in the number of active flows, and in the bandwidth allocation strategy used, (3) increases the link utilization ratio. Since social, economic, and technical issues lead the ISPs to implement different fairness policies, considering their business strategy, we made the choice that our mechanism will be independent of the fairness function used. Indeed, the multicast bandwidth allocation module may implement either a multicast fairness function as those described in

[2]There are several experimental multirate codecs, such as the Scalable Arithmetic Video Codec from the University of Berkely developed by D. Taubman [113], or the Scalable Video Conferencing project from the Framkom Research Corporation [73].

[3]Assume n active multicast flows and denote by n_i the number of downstream receivers of flow i. The receiver-independent (RI), linear (LIN), logarithmic (LOG or LogRD) bandwidth sharing policies consist to give to flow i a bandwidth share equal to $\frac{1}{n}$, $\frac{n_i}{\sum_j n_j}$, and $\frac{1+\log n_i}{\sum_j (1+\log n_j)}$, respectively.

[75] or a multicast pricing model [24, 32, 65, 66].

We call our AQM mechanism, Multicast Fair Queuing (MFQ). MFQ belongs to the class of per-flow dropping mechanisms like FRED (Flow Random Early Drop) [78]. The operations done by MFQ are not as complex as manipulation of several queues like FQ (Fair Queuing) [36], given that they only consist of dropping or queuing the packet.

It is important to note that we usually associate the flow-based mechanisms support with complexity and scalability problems since they require connection specific information and processing. These concerns are justifiable only in point-to-point connections, for which routing tables do not maintain connection-specific state. In multicasting, routing tables keep connection specific state in routers anyway; namely, the multicast group address refers to a connection. Thus, adding multicast flow specific information should slightly increase the routing state. We demonstrate that a modest amount of state and computation at network routers can yield significant performance gains for multicast applications.

For each flow is assigned a dynamic weight computed using an inter-multicast fairness function that may implement either a bandwidth allocation strategy or a multicast pricing policy. MFQ achieves the expected share of the link bandwidth among all the competing flows using a single FIFO queue. In fact, it interacts with an external multicast bandwidth allocation module which implements a pre-defined inter-multicast fairness function. To guarantee a fine-grained packet queuing/dropping, MFQ uses a novel bandwidth sharing notion, called Multicast Allocation Layer (MAL). Based on this notion, MFQ classifies multicast packets into layers and adjusts their weights in order to provide a bandwidth sharing being as close as possible to that given by the fluid model algorithm.

We use simulation to evaluate the effectiveness and performance of MFQ for different communication scenarios. In absence of similar approaches, MFQ performance is compared to the theoretical expected results computed using the fluid model algorithm. We first consider non-responsive multicast source; i.e., sources that do not modify their behavior when a packet loss occurs. Then, we examine the case when the sources implement the Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL) [20] as a multicast congestion control mechanism. Finally, we experiment MFQ for heterogeneous multicast flows where there are both responsive and non-responsive competing sources. For the three cases, we will show that the bandwidth share obtained using MFQ is very close to the "expected allocation" and we will demonstrate that for layered multicast transmission, layers with lower number of members (lower priority layers) will see a loss rate higher than those with higher number of members (higher priority layers).

Without loss of generality, we validate MFQ for both linear and logarithm inter-multicast fairness functions. In addition, we demonstrate that our mechanism still achieves the expected share when there is a big dynamic change in the multicast group size due to join and leave events.

Simulation results demonstrate that MFQ achieves the expected allocation for both responsive and non-responsive multicast flows. We validate our findings by analyzing the impact of network and groups dynamics on the expected bandwidth allocation and comparing it to that obtained by MFQ. Furthermore, we show that MFQ converges very fast to a stable state and easily adapts itself to the dynamic change of the flows weights which affects the expected share of the bandwidth.

The remainder of this chapter is organized as follows. The fluid model algorithm of fair bandwidth sharing among multicast flows is presented in Section 3.2. Section 3.3 details the

main components of MFQ mechanism. We discuss some complexity and implementation issues in Section 3.4. In Section 3.5, we analyze how MFQ could be incrementally deployed. The interaction between MFQ and layered multicast transmission is discussed in Section 3.6. In Section 3.7, we provide some directions for using MFQ mechanism to develop network-based multicast congestion control mechanisms. We show the simulation results for both responsive and non-responsive multicast sources and for single and multiple congested links in Section 3.8. Section 3.9 concludes this chapter by summarizing our main results.

3.2 Fluid Model Algorithm

Before embarking into the mechanism details, let us present the formal network model used to examine our active queue management mechanism.

We consider a network as a set of links $\Im$ where each link l_j has a capacity $C_j > 0$. We assume a known number of multicast flows with different number of receivers. These flows compete for access to communication links. A multicast session S_i is a tuple $(X_i, \{r_{i,1}, ..., r_{i,k}\})$ of session members: X_i is the session sender that transmits data within a network; each $r_{i,p}$ is a receiver that receives data from X_i. Each session contains exactly one sender and at least one receiver. We denote G_j the multicast group number j to be a multicast group to which belong one or more multicast sessions. We write $r_{i,p} \in S_i$ to indicate that receiver $r_{i,p}$ is a member of the multicast session S_i and $S_i \in G_j$ to indicate that the multicast session S_i belongs to group G_j.

We define $N_{i,j}$ to be the number of receivers in session S_i whose path toward the source includes link l_j and define N_j to be the number of all receivers whose multicast delivery tree includes link l_j, i.e., $N_j = \sum_i N_{i,j}$.

Considering a single link from the set $\Im$ with a capacity equal to C. We assume n active multicast flows, and that the source number i sends at the instantaneous rate $\alpha_i(t)$, in bits per second.

Usually for unicast flows, we use the unicast max-min fairness [14] to develop the fluid model. As we will discuss below, currently there is no well-accepted fairness definition for multicast flows. To be independent of the bandwidth allocation strategy used, we assume a vector $\lambda(t) = (\lambda_1(t), \lambda_2(t), ..., \lambda_n(t))$ that defines the expected allocation at the instantaneous time t at the output queue of the router connected to a given link. The value of $\lambda_i(t)$ may be either a function of the number of competing multicast groups, the number of flows per group, and the number of receivers per flow, or even a set value. We call this vector of fair sharing, the *multicast allocation vector*. If the fair share is achieved, the multicast flow number i receives service at a rate given by $min(\alpha_i(t), \lambda_i(t))$. Let $A(t)$ denote the total arrival rate at the considered link: $A(t) = \sum_{i=1}^{i=n} \alpha_i(t)$. If $A(t) > C$ the congestion phenomena holds and then the multicast allocation vector $\lambda(t)$ is the unique solution to

$$C = \sum_{i=1}^{i=m} min(\alpha_i(t), \lambda_i(t)). \tag{3.1}$$

If $\alpha_i(t) > \lambda_i(t)$, then the fraction of bits $\frac{\alpha_i(t)-\lambda_i(t)}{\alpha_i(t)}$ will be dropped, and the multicast flow i will have an output rate of exactly $\lambda_i(t)$. The arrival rate to the next hop is given by $min(\alpha_i(t), \lambda_i(t))$.

The number of downstream receivers in each path of the multicast delivery tree does not remain constant because some receivers may be reached via interfaces other than that belongs to this path. Therefore, the *multicast allocation vector* $\lambda(t)$ may be different in the tree branches even when there is no more competing multicast flows.

In the next section we explore the MFQ components that extend the fluid model algorithm to real networks where transmission is packetized.

3.3 MFQ Architecture

Two modules compose our mechanism:

- The multicast bandwidth allocation module which computes the expected fair share for each active multicast flow[4].
- The buffer management module which uses a single FIFO queue and interacts with the first module to decide the drop preference in order to achieve the expected bandwidth sharing.

In Figure 3.1, we show a simplified architecture of MFQ. In the following sections, we explore and discuss the two modules.

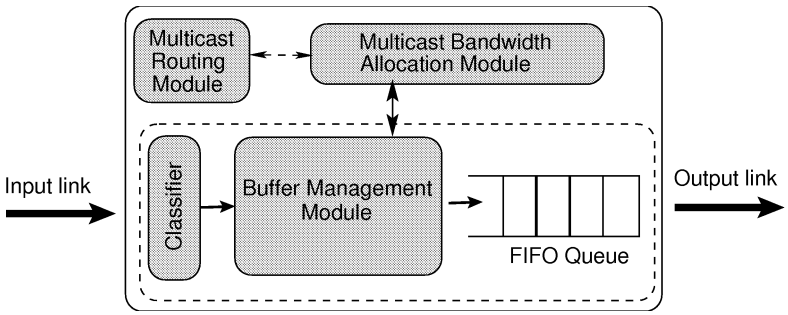


Figure 3.1: A simplified MFQ architecture

3.3.1 Multicast Bandwidth Allocation Module

We first present a general framework for the multicast fairness notion. Then, we enumerate some inter-multicast fairness candidates functions that could be implemented by the ISPs. We consider the two existing multicast service models: the ASM (Any Source Multicast) [34] model and the SSM (Single Source Multicast) model [67].

3.3.1.1 General Framework

The multicast bandwidth allocation module determines the link capacity fraction[5] that should be allocated to the flow to which the incoming packet belongs. We develop hereafter a general

[4] A multicast flow is considered instantaneously active if it has at least one packet in the queue.

[5] Throughout this dissertation we use the terms "link capacity fraction", "flow weights", and "flow bandwidth allocation", interchangeably.

framework of the multicast bandwidth allocation module. As we have pointed out earlier, the multicast fairness function may depend on the number of groups, the number of flows per group, and the number of receivers per flow. We assume that each ISP has a single and clearly defined multicast bandwidth sharing policy. This policy can be configured in all or some routers inside its network.

Using the network model introduced in Section 3.2, we define the bandwidth γ_{ij}, in bits per second, allocated to group G_i, in link l_j as follows:

$$\gamma_{ij} = F_1(G_i) * C_j \tag{3.2}$$

where $F_1(.)$ is the ***inter-group multicast fairness function*** that implements the bandwidth sharing policy among active multicast groups. In other words, $F_1(G_i) * C_j$ is the maximum bandwidth, in bits per second, that should be allocated to group G_i in link l_j.

In the ASM (Any Source Multicast) service model [33], a multicast group may have one or more sessions (flows) from different sources that share the same communication link. The bandwidth allocated to flow k of the multicast session $S_k \in G_i$ in link l_j is given by the following expression:

$$\lambda_{kj} = F_2(S_p / S_p \in G_i) * \gamma_{ij} \tag{3.3}$$

where γ_{ij} is computed using Eq. 3.2. The function $F_2(.)$ determines the fraction of the bandwidth which has already been allocated to group G_i and that should be given to the session S_k. We call this function the ***intra-group multicast fairness function***. Both F_1 and F_2 depend on the number of downstream receivers of each multicast session that belongs to the same group. The functions $F_1(.)$ and $F_2(.)$ must satisfy the following properties:

- for each multicast group G_i: $0 < F_1(G_i) < 1$, and $0 < F_2(S_p) < 1$ for each $S_p \in G_i$.
- in each link l_j: $\sum_i F_1(G_i) = 1$, and $\sum_{pS_p \in G_i} F_2(S_p) = 1$ for each group G_i.

Our main focus in this chapter is not to find the "optimal" functions $F_1(.)$ and $F_2(.)$, however we will show that MFQ can adapt itself according to the bandwidth allocation function used, the number of receivers per flow, the number of active flows, and the number of active multicast groups.

3.3.1.2 Examples of inter-multicast fairness functions

In the ASM service model [33], it is possible to have two or more different sources sending to the same multicast group G_i. Therefore, it makes sense to use the function F_1 to share the bandwidth between multicast competing groups. Let's assume that the capacity C_j of link l_j is **equitably** shared between them, then the group G_i gets a bandwidth share γ_{ij} equals to $F_1(G_i) * C_j = \frac{1}{g} * C_j$, where g is the total number of distinct groups. If F_2 is a logarithm function of the number of receivers, the session $S_k \in G_i$ will get a bandwidth share equal to $\lambda_{kj} = \frac{1+\log n_k}{\sum_{p=1}^{p=m_i}(1+\log n_p)} * \frac{1}{g} * C_j$, where m_i is the number of sessions that belong to group G_i, and n_p is the number of receivers of session S_p.

In the SSM service model [67], each sender may use a different group address and the multicast session is identified by the couple (sender address, group address). Thus, there is only one source per multicast group and there is therefore no need of using function F_1. In

the case of using a logarithm bandwidth sharing function, the session S_k will get a bandwidth share equal to $\lambda_{kj} = \frac{1+\log n_k}{\sum_{p=1}^{p=n}(1+\log n_p)} * C_j$, where n is the total number of competing sessions.

3.3.2 Buffer Management Module

3.3.2.1 Module Description

The role of the buffer management module is to make the queuing/dropping decision of each incoming multicast packet. In the MFQ design phase, we have taken some key decisions in order to have a suitable mechanism independent of the network and sources characteristics and especially the variation of flows weights[6], source behavior when a packet is lost, and sources rates.

We use a single FIFO queue with a pre-configured maximum size in packets. For each multicast active flow, we maintain a flow state containing:

- the number of packets belonging to this flow which are waiting to be served,
- the current flow weights which is provided by the bandwidth allocation module.

If the flow is new or if there is a change on the number of receivers, we get the new flow weight from the bandwidth allocation module. We assume that we know the number of downstream receivers for each active multicast flow and in each router belonging to the multicast delivery tree. In Chapter 5, we emphasize on this issue and we propose an extension to the multicast service model to count the number of downstream members at the senders as well as at the intermediate routers in the multicast delivery tree.

To prevent the queue from being monopolized by high-rate or bursty multicast sources, we use a pre-configured threshold variable ***thrsh***. If the mean queue size[7] is less than *thrsh* the packet will be accepted only if the number of waiting packets belonging to the flow does not exceed the allowed number (its MAL value or its MAL value plus one more packet depending on the generated number u as explained above).

If the mean queue size is more than the threshold *thrsh* or the queue is full, we accept the packet only if it belongs to an inactive flow. If the queue is full, we drop the incoming packet if its flow is active, otherwise we drop randomly a packet from the queue and we queue the incoming packet. By this way we allow a new multicast flow to become active and we remove the bias against bursty sources. If the packet was accepted, we update the flow state.

3.3.2.2 The Multicast Allocation Layer (MAL) Scheme

In order to achieve a fine-grained queuing/dropping, we introduce a new scheme, called Multicast Allocation Layer (MAL) which is a key component of the MFQ's buffer management module. We define a MAL as follows:

Definition: *A MAL is a set of flows that may have the same or different expected allocation in term of the link capacity fraction (the bit-level fairness), but they have the same allocation*

[6] In the case of using a fairness function which depends on the number of downstream receivers, all flows weights change when at least one receiver joins or leaves one of the multicast active sessions.

[7] We use the same method as RED (Random Early Drop) [58] to estimate the mean queue size $qlen$. The formula for calculating the average queue length $qlen$ is $qlen = (1 - W_q) * qlen + W_q$, $0 \leq W_q \leq 1$. The weighted moving average formula, with weight W_q is used to filter out transient congestion. The value of W_q is set to 0.002 in all simulations.

in term of the maximum number of packets (the packet-level fairness) allowed to be present at the same time in the queue.

We assume that at the time t, there are n active multicast flows in the queue and that the flow f_i has a weight equal to w_i which is provided by the bandwidth allocation module. We define the MAL mapping function F_{MAL} as follows:

$$\begin{array}{ccc} \{f_1, f_2, ..., f_j, ..., f_n\} & \longrightarrow & \{0, 1, 2, ..., qlim\} \\ f_j & \longmapsto & F_{MAL}(f_j) = \lfloor w_j * qlim \rfloor \end{array}$$

where $qlim$ is the queue size in packets. Two flows f_i and f_j belong to the same MAL number k only if $F_{MAL}(f_i) = F_{MAL}(f_j) = k$.

Given that the number of active flows change, the flows weights and the set of flows per MAL are dynamic. A MAL which has a non-empty set of flows is considered active. We can have at most $qlim$ active MALs in a queue of a size equal to $qlim$ and in this case each MAL contains only one flow.

Let us explain the MAL scheme through the example given in Figure 3.2. In the x-axis, we show the distribution of 21 active flows in the different MALs. As we can see, there are 5 active MALs with various size in term of the number of active flows that contains each MAL. The flows belonging to the same MAL have different weights drawn in the y-axis by vertical arrows. For example, flows f_{20} and f_3 belong to the MAL number 4 which has 5 active flows $\{f_4, f_{20}, f_{19}, f_3, f_{16}\}$.

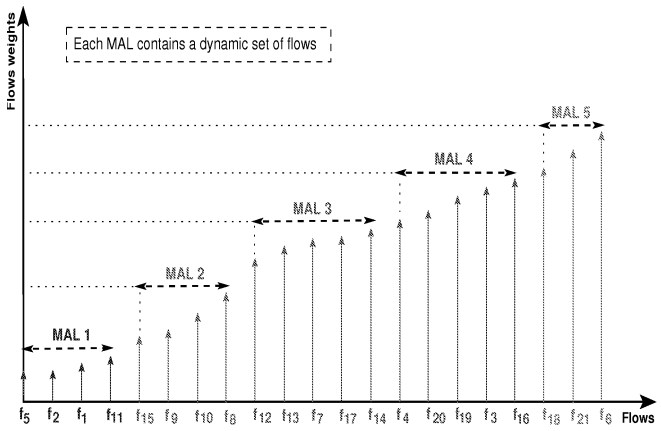


Figure 3.2: An illustration of the Multicast Allocation Layer (MAL) scheme

To do a fine grain dropping, MFQ maintains for each active MAL the identity of the flow which has the highest weight among other flows in the same MAL. As we will detail in the next section, MFQ discriminates between flows belonging to the same MAL in order to achieve the expected fair share.

One can make the observation that when the flows weights are equal, the bandwidth is equitably distributed among flows given that there is only one active MAL including all active flows. We believe that the use of the MAL scheme will be also helpful even for unicast flows in differentiated networks by providing a fine-grained dropping for unicast packets belonging to the same DiffServ class when their corresponding flows have different weights[8].

[8]The details about how to use the MAL scheme in DiffServ networks for unicast flows is, of course, out of

Given that we do queuing using per-packet manner and not per-bit manner, MFQ tries to guarantee that the maximum number of packets allocated to each active flow i remains always less than the integer value of its expected allocation **in term of the capacity fraction** w_i multiplied by $qlim$, the maximum queue size in packets, i.e.; $\lfloor w_i * qlim \rfloor$. However, two flows that have different weights may have the same maximum number of packets allowed to be queued. In the example given in Figure 3.2, the maximum number of allowed packets of flow number 3 and flow number 20 is equal to 4 because $\lfloor w_3 * qlim \rfloor = \lfloor w_{20} * qlim \rfloor = 4$ where $w_3 = 0.07$, $w_{20} = 0.065$ and $qlim = 64$. To guarantee a more fine-grained queuing, we enhance the buffer management module by using the MAL scheme that we have described above.

For every arriving packet, the router starts by identifying the multicast flow and the MAL to which it belongs. Let flow number i be the flow to which belong the arriving packet and j be the number of its MAL. We generate a random value $u \in [0, 1]$ and we allow the flow i gets **one more packet** than its MAL value if $u < w_i / MAL[j].maxAllocation$, where $MAL[j].maxAllocation$ is the maximum weight of flows belonging to the MAL number j. As consequence, we ensure that each two flows that belong to the same MAL will get **randomly and proportionally** to their fair share one more packet than their MAL values and we ensure a much more fine-grained bandwidth sharing. If the packet was accepted, we update its MAL state.

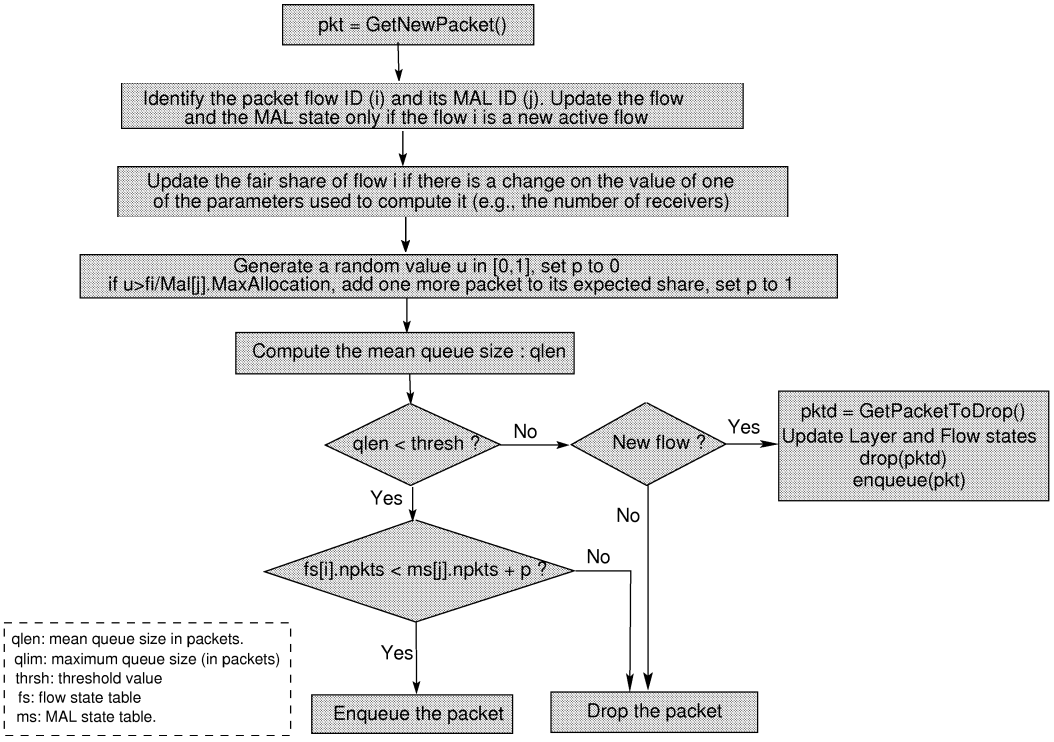


Figure 3.3: The MFQ algorithm flowchart

For completeness, we give the flowchart of MFQ algorithm in Figure 3.3 and the pseudo-code of the algorithm in Algorithm 3.1.

the scope of this work on multicast bandwidth sharing and it will be investigated in future works.

Algorithm 3.1 : The pseudo-code of MFQ mechanism

Variables:

```
fs_ :  Flow State {receivers, active, qlen, fairness}
ls_ :  Layer State {maxFlowID, maxAllocation }
```

Algorithm:

```
/* we first determine the multicast flow ID */
flowID ← classify(pkt)
/* we update the flow state if needed*/
if (flowID is not active) {
  fs_[flowID].receivers ← getReceivers(p)
  fs_[flowID].count++
  newflow ← 1
} else (if the number of receivers has changed)
  update the number of receivers and the total number
/* we get the expected allocation */
weight ← getAllocation(flowID])
/* we determine the MAL to which belongs the current flow */
int mal_ ← max(1,(int)(weight*thrsh))
/* we compute the number of packets to add to the expected allocation {0 or 1} */
double u ← Random::uniform();
if (u > f / ls_[mal_].maxAllocation )
  p ← 0;
else p ← 1;
/* we update the mean queue size */
update the mean queue size(qlen)
/* the MFQ dropping decision as explained in Section 3.3.2 */
if (qlen >= qlim)
  if (!  newflow ) drop(pkt)
  else{
    pktd ← getPacketToDrop()
    update_mal_state(pktd)
    update_flow_state(pktd)
    drop(pktd)
    enqeue(pkt)
  }
else {
if (fs[flowID].qlen < mal_ + p)
  if ( !  newflow && qlen > thrsh ) drop(pkt)
  else enqeue(pkt)
else
  drop(pkt)
}
/* we update the flow and the MAL states if the pkt was not dropped */
if (pkt was not dropped) {
  update_mal_state(pkt)
  update_flow_state(pkt)
}
```

3.4 Complexity and Implementation Issues

At each "MFQ router", we need to maintain a state per active multicast flow. Upon a packet arrival, the router needs to (1) determine the flow and the MAL number to which the arriving packet belongs, (2) update the flow state and the MAL state parameters such as the number of packets of the corresponding flow and the allocation of this flow provided by the multicast bandwidth sharing module. As shown in [110], these operations and even the packet classification could be efficiently implemented because it only consists of reading the flow ID for IPv6 or the pair (source IP address, multicast destination address) for IPv4.

At branch points in the multicast tree, MFQ routers must record additional information needed by the multicast bandwidth allocation module such as the number of downstream receivers. The processing complexity at MFQ routers is increased in two (minor) ways. First, during the lookup-operation for each packet arrival, useful information must also be retrieved from the routing table entry. Second, before accepting the datagram, MFQ should verify that the flow is authorized to get more packets in the queue. While our mechanism would benefit from better bandwidth allocation functions, it is explicitly designed to be robust to coarse implementation of the inter-multicast fairness function using the MAL scheme described in Section 3.3.2.2.

It should also be noted that the source address and the destination group address are not only needed by the MFQ mechanism but also by the multicast routing lookup module which has to determine the list of outgoing interface(s) for each incoming multicast packet.

3.5 Incremental Deployment

MFQ does not escape to the rule that any network service must be able to be deployed in an incremental fashion on the Internet, due to the scale and inherent heterogeneity of the network. It allows to enhance performance even if it is supported only by specific few routers. In addition, there is no need to use MFQ in routers that have a low multicast traffic. Then, MFQ can be implemented/activated in some routers that the ISP administrative authority consider to handle an important amount of multicast traffic load.

As a first step in MFQ deployment, it could be implemented/activated in the multicast border routers that implement the inter-domain multicast routing MBGP (Multicast Border Gateway Protocol) [13]. Using MFQ, the border routers can provide a diffserv-like service for arriving multicast flows from directly-connected domains based on a predefined bandwidth sharing function.

3.6 MFQ and Layered Multicast

When evaluating new network control mechanisms one must evaluate the impact of the proposed mechanisms on application performance. We discuss in this section the MFQ impact on layered multicast application performance.

When the multicast session using a layered transmission scheme, the data is split into layers and each layer sends to a different group address. Depending on the loss rate seen by the receivers, they join and leave layers to adapt to the network situations. We demonstrate how MFQ can achieve a priority dropping without explicitly assigning priorities to the transmission layers.

Assuming a multicast source decodes data into n transmission layers and that there are R_i receivers subscribed to the layer l_i (l_0 is the base layer). Given that receivers who join the layer l_i should join all lower layers $l_0...l_{i-1}$, we can easily write the following inequality: $R_{n-1} \leq ... \leq R_j \leq R_{j-1} \leq ... \leq R_1 \leq R_0$.

Without loss of generality, we assume the use of the LogRD function to allocate the bandwidth fairly between multicast flows. The weight of the transmission layer number j is $w_j = \frac{1+\ln R_j}{\sum_{p=0}^{p=n}(1+\ln R_p)}$. We can easily write the following inequality:

$$w_{n-1} \leq ... \leq w_j \leq w_{j-1} \leq ... \leq w_1 \leq w_0. \tag{3.4}$$

We can then write:

$$F_{MAL}(f_{n-1}) \leq ... \leq F_{MAL}(f_j) \leq F_{MAL}(f_{j-1}) \leq ... \leq F_{MAL}(f_1) \leq F_{MAL}(f_0), \tag{3.5}$$

where $F_{MAL}(f_i)$ is the MAL to which belongs the flow associated to the transmission layer number i. Thus, it is clear that layers with lower number of receivers (lower priority layers) will see a loss rate higher than those with higher number of receivers and in particular the base layer l_0 (highest priority layer).

Similar approaches that need a network-support including priority dropping [9] schemes require that the network support as many loss priority levels as layers. In addition, to ensure a fair allocation of network resource, they also require that each session uses the same set of priorities than others. Furthermore, priority dropping provides no incentives for receivers to lower their subscription level.

The development of a multicast-congestion control mechanism that uses MFQ to ensure a priority dropping between flows corresponding to the transmission layers is one of our future works that we will investigate.

3.7 Towards a Network-Based Multicast Congestion Control

There is a big debate going on in today's the network research community on whether "end-to-end"argument is still the golden rule, or just dead [26]. The current Internet is built on top of the "end-to-end" philosophy. End systems are responsible for constructing better than best-effort service, therefore the network is often drawn as a big cloud with end system (PCs, etc) attached to it, with its inner workings hidden from the end users. However, this is not always true in today's Internet, where servers and proxies for specific applications (the web, mainly) are installed inside the network in attempt to improve the application performance. Therefore many people believe the "end-to-end" service model is dead, since lots of services can be done better inside the network than by the end system.

In a multicast network, it is very important that the bandwidth is allocated fairly among the multicast sessions. An appealing approach that accommodates heterogeneous receivers and adapts to congestion is to encode the data onto multiple layers and transmit each layer on its own multicast group.

Layered multicast protocols typically use a receiver-driven approach in which the end-systems decides which layers should be delivered [20, 116].

An alternative approach is to use a MFQ scheme where the network, rather than the receiver, decides which layers (flows) should be delivered. When congestion arises, the network

drop the least important packet (e.g; that has the less number of downstream receivers) first. An important benefit of MFQ is stable and fair allocation of bandwidth. Because packet losses are concentrated at the highest layer(s) (given their lower number of receivers), the highest layers absorb the majority of transient losses caused by short term congestion.

As shown in Section 3.6, during periods of congestion, an "MFQ router" drops lower priority packets before higher priority packets. By assigning the highest weight to the base layer and successively lower weights to additional layers, losses during short term congestion are confined to the enhancement layers without affecting basic layers. Consequently, MFQ is very effective at reacting to transient congestion. Furthermore, one could also integrate in MFQ an explicit congestion notification to the source, as RED [58] does.

We do not claim that such type of mechanism can be easily deployed in the today's Internet, however minimizing the complexity of developing multicast congestion control mechanisms between the end hosts and the network can encourage the development of added-value multicast applications. Indeed, MFQ allows the ISPs to differentiate between multicast flows according to a pre-defined multicast bandwidth allocation function.

3.8 Simulation Methodology and Results

We have examined the behavior of MFQ under a variety of conditions. We use an assortment of traffic sources and topologies. All simulations were performed in ns-2 [84], which provides accurate packet-level implementation for various network protocols.

3.8.1 Non-Responsive Multicast Flows

We start by validating MFQ for a simple topology consisting of a single congested link connecting two routers n_1 and n_2 and having a capacity C equal to 10 Mbps and a propagation delay D equal to 1 ms. As shown in Figure 4.3, the multicast sources are connected to router $n1$ and receivers are downstream to router $n2$.

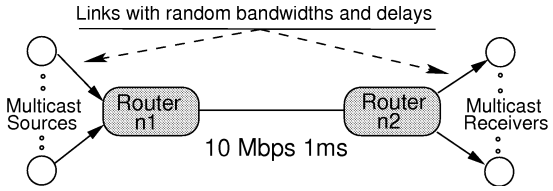


Figure 3.4: A single congested link.

We compare the expected bandwidth with that obtained by MFQ for 32 multicast flows. The maximum buffer size *qlim* used in all our simulations is set to 64 packets. We index the multicast flows from 1 to 32 and we compute the bandwidth share of each flow. We use linear and logarithm bandwidth allocation function. When we use a linear allocation function, the fair share of flow i is equal to $\frac{i}{\sum_{j=1}^{j=32} j} * C$ and it is equal to $\frac{1+\lg i}{\sum_{j=1}^{j=32}(1+\lg j)} * C$ in the case of using a logarithm allocation function.

In this section, we assume that each multicast source is a non-responsive CBR source. For this case, we use a CBR generator simulating a non-adaptive audio application. We assume that the source i sends data at a rate equal to i more that its expected fairness rate provided

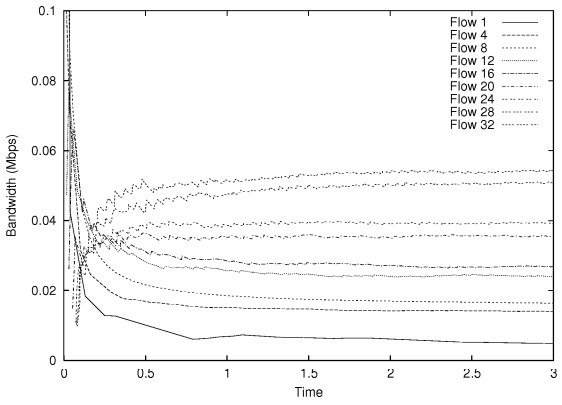


Figure 3.5: Convergence of MFQ

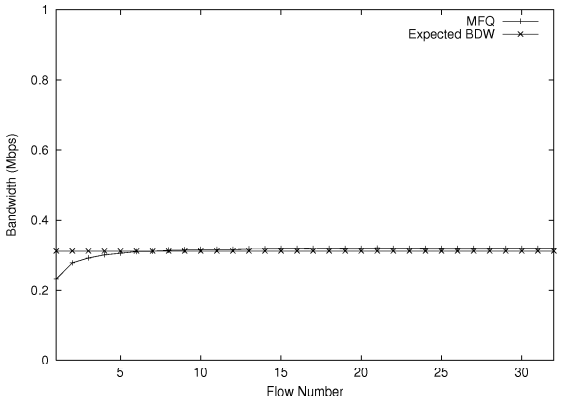


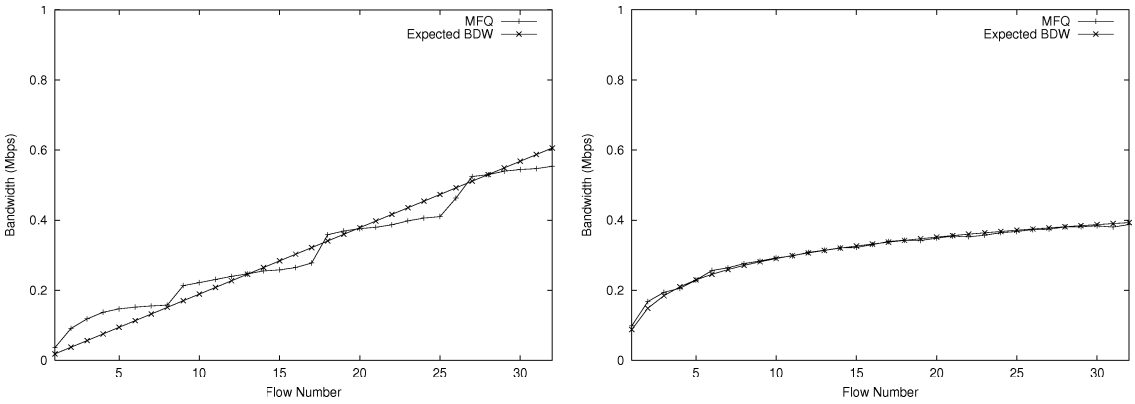
Figure 3.6: The bandwidth share provided by MFQ when each multicast flow has exactly one receiver

by the fairness function. For example, there is a total amount of $\sum_{i=1}^{i=32} \frac{i}{32} * C = 16.4 * C$ kbps arriving at the bottleneck link when we use a receiver-independent multicast fairness function. Thus, the flow 1 sends 0.3125 *Mbps*, and flow 2 sends 0.625 *Mbps*, and so on.

Unless otherwise specified, each simulation lasts 30 seconds, the source number i starts sending data $i * 0.001$ sec after the simulation starting time and the packet size is assumed to be equal to 1000 bytes. To help the understanding of our result plots and without loss of generality we assume that the flow number i has i downstream receivers.

In a first experiment, we focus on the convergence phase of MFQ. We use a linear bandwidth allocation function and we plot in Figure 3.5, the variation of the bandwidth share of some flows in function of the simulation time. We can see that for all flows MFQ reaches a steady sate after approximately one second of simulation. In addition, as expected the flow number i gets more bandwidth share than flows $1, ..., (i-1)$ giving that we use a receiver-dependent fairness function.

In all the following simulations, we start the measurements one second after the simulation starting time so that, the current and the average queue size have already reached a stable state.



(a) Using a linear multicast allocation function

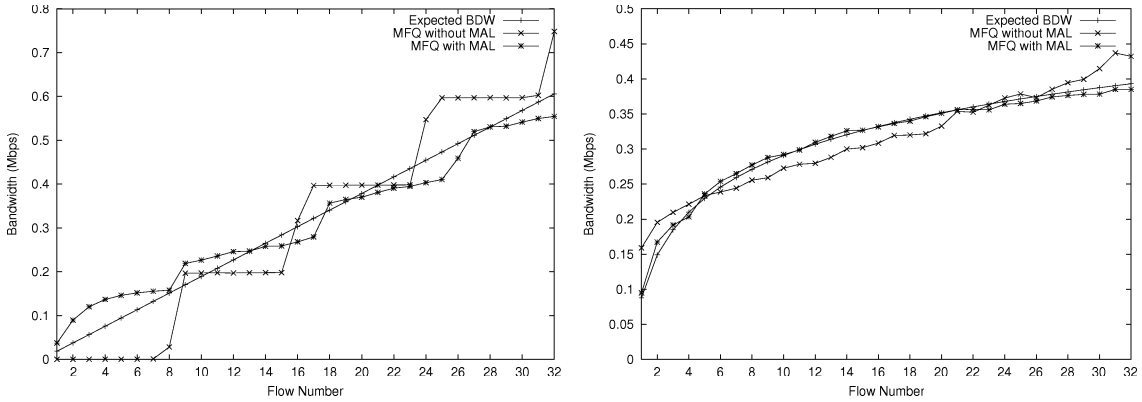
(b) Using a logarithm multicast allocation function

Figure 3.7: 32 CBR multicast flows. The flow number i has i downstream receivers.

In a second experiment, we validate our mechanism for a simple case where each multicast flow has only one receiver which corresponds also to the case when using a receiver independent bandwidth allocation function. Figure 3.6 shows the bandwidth share obtained by MFQ. Regardless the multicast bandwidth allocation function used, all the flows get the same bandwidth share $\frac{1}{32} * C = 0.3125\ Mbps$. Thus, MFQ achieves a good precise degree of fairness between multicast flows.

In Figure 3.7(a) and Figure 3.7(b) , we plot the bandwidth share of each multicast flow when using a linear and a logarithm allocation function, respectively. We can see that the bandwidth allocation provided by MFQ is close to the expected fairness for both cases.

The third experiment aims to demonstrate how MFQ performance can be improved by the use of the MAL scheme. In Figure 3.8(a), and Figure 3.8(b), we plot the bandwidth share obtained by MFQ with and without the MAL scheme. According to the plots, our MAL



(a) Using a linear multicast allocation function

(b) Using a logarithm multicast allocation function

Figure 3.8: 32 CBR multicast flows used to evaluate the added value of the MAL scheme.

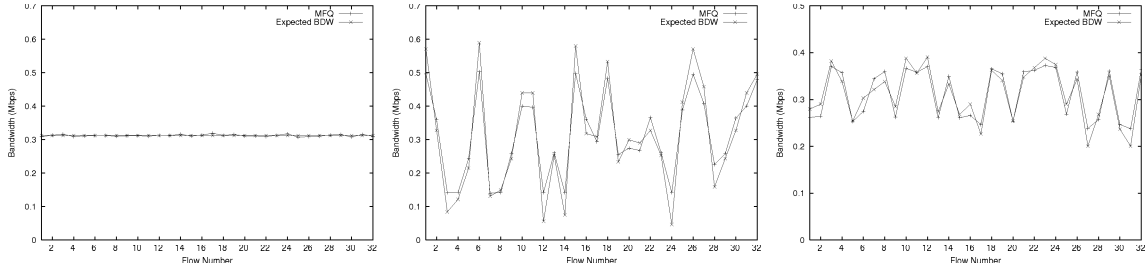
scheme can achieve a good fine-grained bandwidth allocation. In addition, we can easily see the distribution of flows in MALs when we do not use the MAL scheme. For example, in the case of linear allocation function (Figure 3.8(a)), flows from 9 to 15 belong to the same MAL number 2, flows from 16 to 24 belong to the MAL number 3, and so on.

Without using the MAL scheme, flows 9 and 15 (belong to the MAL number 2) will get the same bandwidth share, which is not fair. The use of MAL allows these flows to share the bandwidth fairly. Indeed, flow 15 gets more bandwidth share than flow 9.

In the previous experiments we assumed that the number of receivers for each flow is equal to the flow index. Now, we assume that the number of receivers of each flow is randomly generated between 1 and 64. For all other parameters, we use the same values as the second experiment. We show in Figure 3.9(a), Figure 3.9(b), and Figure 3.9(c) the obtained bandwidth for each flow when we use a receiver-independent allocation function, a linear allocation function, and a logarithm allocation function, respectively. These results confirm that our mechanism is independent of the number of receivers per group and it is independent of the multicast fairness policy used.

3.8.2 Responsive Multicast Flows

In this section, we examine the behavior of MFQ in presence of responsive multicast sources where the senders use the layered transmission scheme which was first proposed for the RLC (Receiver-driven Layered Congestion) congestion control protocol [116]. To address some of the deficiencies of RLC, authors of |80| propose Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL) protocol [20]. The protocol uses a "Digital Fountain" [21] mechanism at the source. A digital fountain allows any number of heterogeneous clients to acquire bulk data with optimal efficiency at times of their choosing. Moreover, no feedback channels are needed to ensure reliable delivery, even in the face of high loss rates. FLID-DL introduces the concept of Dynamic Layering to reduce the join and leave latencies associated with adding or dropping a layer. With Dynamic Layering, the bandwidth consumed by a layer decreases



(a) RI fairness function (b) LIN fairness function (c) LOG fairness function

Figure 3.9: 32 CBR multicast flows. The number of receivers of each flow is randomly generated.

over time. Thus a receiver has to periodically join additional layers to maintain its receive rate. The receive rate is reduced simply by not joining additional layers, whereas rate increase requires joining multiple layers.

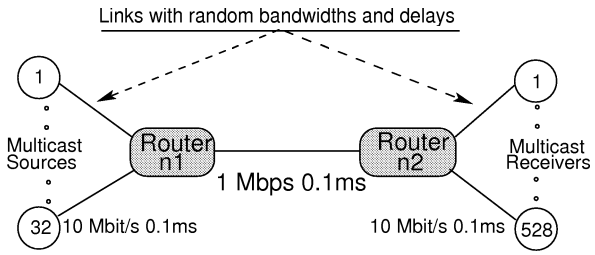


Figure 3.10: A single congested link

In this experiment, we assume that we have 32 multicast competing sources that use the FLID-DL congestion control protocol. We assume that the source number i has exactly i receivers which result on 528 different receivers. We use the network configuration given in Figure 3.10.

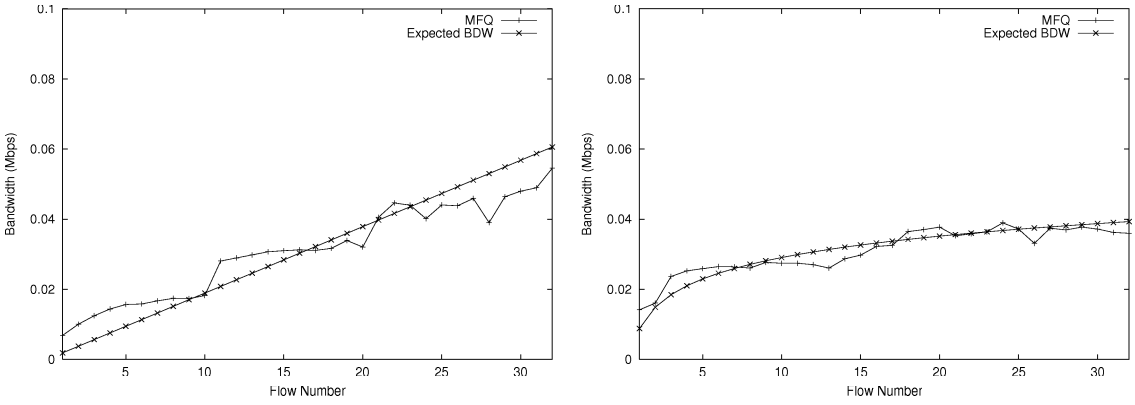
Once a receiver has obtained the transmission session description, which includes the information about the groups associated with a session that is needed in order to join those groups, it may join one of the groups in the session. The receivers must join and leave groups within a session as described in detail in [20].

Each source uses 17 layers encoding, each layer is modeled by a UDP traffic source. In Figure 3.11, we give the values of FLID-DL parameters set in the FLID-DL ns simulation.

The performance of MFQ in presence of FLID-DL sources is shown in Figure 3.12(a) and Figure 3.12(b) for both linear and logarithm bandwidth allocation function, respectively. As can be seen from the plots, MFQ enforces the required fairness very well. The obtained results are slightly different from the case of CBR non-responsive sources because the FLID receivers join and leave layers according to the loss rate observed. That's why we see an oscillation in the obtained plot around the expected plot.

Parameter	Description	Value
c_mult_	the multiplicative factor	1.3
slot_time_	the slot time	0.5
number_of_layers_	the number of layers	17
simulated_rtt_	for tcp rate value calculation	0.1
rng_speed_	the random generator's speed	0
packet_payload_	the number of bytes in packet	1000

Figure 3.11: FLID-DL parameters used in ns-2 simulation



(a) Using a linear bandwidth allocation function

(b) Using a logarithm bandwidth allocation function

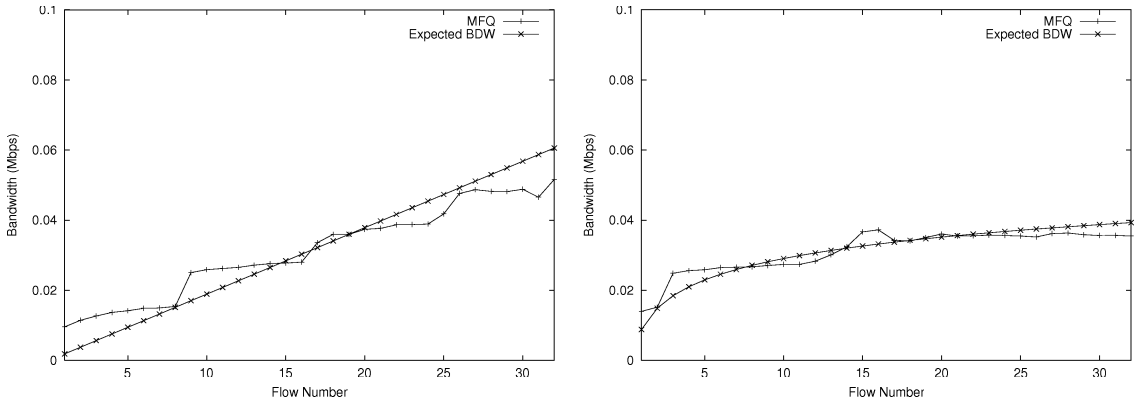
Figure 3.12: 32 FLID-DL sources

3.8.3 Heterogeneous Multicast Flows

We have conducted simulation to evaluate the performance of MFQ when we have heterogeneous multicast sources: responsive and non-responsive. To this end, we use 32 multicast sources indexed from 1 to 32 where flows from 1 to 16 are generated by FLID-DL sources and flows from 17 to 32 correspond to CBR sources.

We use the network configuration of Figure 3.10, and we suppose again that the flow i has exactly i receivers. The CBR sources are similar to those of the first experiment of Section 3.8.1, and the FLID-DL parameters are similar to those of Section 3.8.2.

We plot in Figure 3.13(a) and Figure 3.13(b) the obtained and the expected bandwidth sharing for linear and logarithm bandwidth allocation policy, respectively. As show in these figures, MFQ matches closely the fair share for both policies despite the heterogeneity of the multicast sources.



(a) Using a linear allocation function

(b) Using a logarithm allocation function

Figure 3.13: 32 multicast flows. Flows form 1 to 16 are FLID-DL sources and those from 17 to 32 are CBR sources

3.8.4 Responsiveness to Group Size Dynamics

An important concern in the design of active queue management mechanisms is their responsiveness to changes in flow weights. To illustrate how MFQ adapts to that change, we assume the use of LogRD fairness function so that the change on the number of receivers affects the expected fair share of all active flows. When receivers join and leave the multicast session, it is important that MFQ reacts sufficiently fast should a change of *multicast allocation vector* be required. This behavior is investigated by randomly generating join and leave events. We measure how long MFQ takes to adapt to the variation of the flow weights.

We consider the single link of Figure 4.3 and we use the same configuration as the first experiment of Section 3.8.1. After 10 seconds of simulation we increase the number of receivers of the flow number 5 from 5 to 32 to emulate join events arriving towards the source from 27 $(32 - 5)$ new receivers.

As shown in Figure 3.14, MFQ mechanism adapts to the change on the number of receivers of flow number 5. Indeed, we see that after one second of the arriving of the new join events,

the flow number 5 gets exactly the same bandwidth as the flow number 32 which has already 32 receivers. We can also see that the flow number 10 has not been affected by this variation only for one second. As expected, each flow gets a slightly less bandwidth share after increasing the number of receivers of flow number 5. This experiment demonstrates that MFQ reacts rapidly to the change in flow weights.

The same simulation configuration can be used to investigate responsiveness to changes in sizes of many multicast groups. The results are similar to those above, since all flows get their fair share fastly.

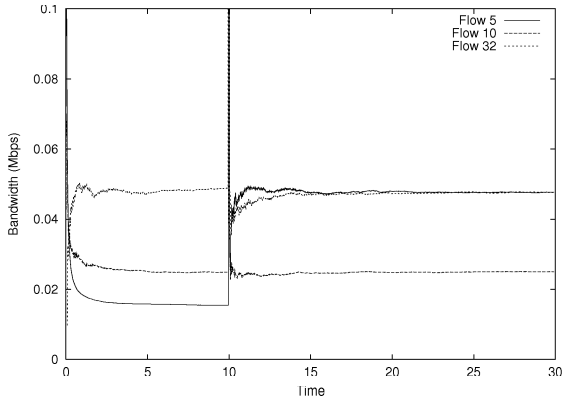


Figure 3.14: 32 multicast flows using a linear allocation function. At 10 seconds of simulation 27 new receivers join the multicast session number 5.

3.8.5 Different Starting Times

We consider the single link of Figure 4.3 and we use the same configuration as the first experiment of Section 3.8.1. In particular, the flow number i has i downstream receivers, and a logarithm bandwidth allocation function is used to share the bandwidth between competing flows.

The goal of this experiment is to determine how MFQ reacts when the multicast sources have different starting time. Again we consider the single link of Figure 4.3 and we use the same configuration as the first experiment of Section 3.8.1. In particular, the flow number i has i downstream receivers, and a logarithm fairness function is used to share the bandwidth between competing flows. We assume that the flow number i starts sending 2 seconds after the flow number $i-1$. The flow number 1 starts at 0.001 *sec.*

In Figure 3.15, we plot the bandwidth share variation of flows 1, 10, 20, and 32 in function of the simulation time. Two main observations can be derived from the obtained plots. Firstly, when the flow number i starts, the bandwidth share of all flows number $1...(i-1)$ decreases to reach a new stable state. Secondly, the flow number i gets more bandwidth share than all its lower indexed flows (flows from 1 to $i-1$). MFQ achieves the bandwidth sharing according to the logarithm multicast fairness function. Indeed, we can observe that the bandwidth share increases logarithmically with the number of receivers.

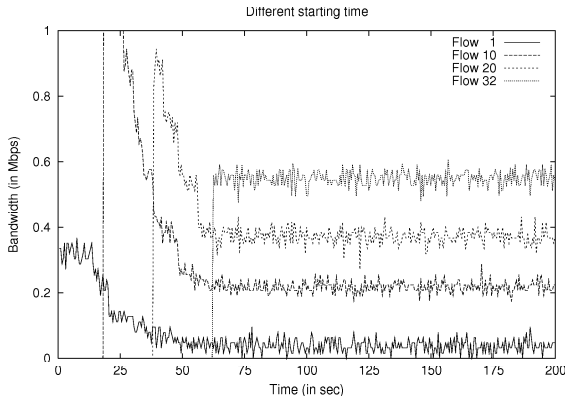


Figure 3.15: 32 CBR multicast sources using a logarithm allocation function. The flow number i starts 2 seconds after the flow number $i-1$.

3.8.6 Multiple Congested Links

In this sub-section, we analyze how the results obtained above are affected when the multicast flows traverse L congested link. We performed two experiments based on the topology of Figure 3.16. We index the links from 1 to L and the capacity C_1 of the link number 1 is set to 1 Mbps. A link L_j, $2 \leq j \leq 10$, is ensured to be always congested by setting its capacity C_j to $C_{j-1} - 50$ Kbps.

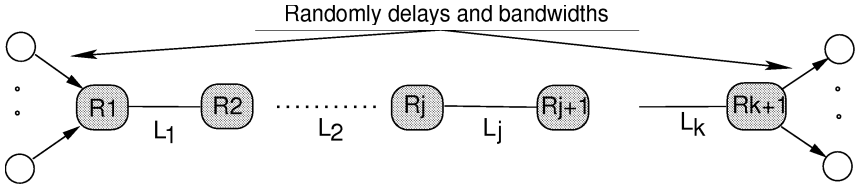


Figure 3.16: Topology for analyzing the effects of multiple congested links on the performance of MFQ

We extended the unicast fairness index introduced by Jain in |72| to multicast traffic. Instead of the expected unicast fair rate, which is always the same for unicast connections, we use that of multicast flows which may differ from one flow to another depending on the fairness function implemented in the bandwidth allocation module. The **multicast fairness index** is then computed as follows:

$$1 - \frac{1}{n} \sum_{i=1}^{i=n} \left| \frac{t_i - \overline{t_i}}{\overline{t_i}} \right| \tag{3.6}$$

where n is the number of active multicast flows, $\overline{t_i}$ and t_i are the expected and the obtained bandwidth share of the multicast flow i, respectively. We plot in Figure 3.17, the variation of the multicast fairness index as a function of the number of congested links. As we can see, the index value remains close to 1 even when the number of congested links increases for RI, LIN, and LOG fairness functions. As expected in |75|, the LOG fairness function has better

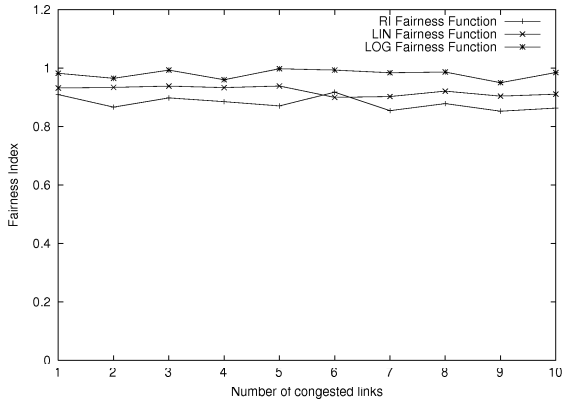


Figure 3.17: MFQ performance in multiple bottleneck link

fairness index variation than the two other functions.

3.8.7 Optimal Threshold Determination

As explained in Section 3.3.2, we use a pre-configured threshold, to bias against bursty sources. In fact, the threshold is the only tuning parameter of our mechanism comparing to other AQM mechanisms such as RED [58] that uses three main parameters which could not be efficiently tuned in heterogeneous networks [23]. By analyzing the simulation results, we argument, in this section, the necessity of using a threshold and we try to find for each type of multicast source traffic the "optimal" threshold value.

To determine the optimal threshold to use for each type of traffic, we use the multicast fairness index that we have introduced in Section 3.8.6.

Recall that our goal is to find the threshold value that maximizes the multicast fairness index. Again, we examine the three representative cases: all non-responsive, all responsive, and heterogeneous flows. We run several simulations for the three cases by varying the value of the threshold. We plot the threshold value along the x-axis and the multicast fairness index value along the y-axis in Figure 3.18, Figure 3.19, and Figure 3.20, for the three traffic types, and for both linear and logarithm bandwidth allocation functions.

We can observe from the three figures that when we do not use a threshold (threshold=100% of the buffer size), the multicast fairness index reaches approximately its lower value for all cases. As we pointed out in Section 3.3.2, this is due to high-rate sources which monopolize the queue, thereby the choice of MFQ threshold value requires a considerable attention in order to improve the fairness between multicast flows. Indeed, from the three figures, we can conclude that the use of a threshold has an effect on the fairness result especially when there are CBR high-rate sources (Figure 3.18 and Figure 3.20). We see no major effect of the threshold value when there are only FLID-DL sources (Figure 3.19) because the sources have similar sending rate in each layer.

Giving that the general case is that of heterogeneous traffic, we should configure the threshold dynamically depending on the traffic type. From the three figures, we can conclude that the optimal threshold value depends also on the bandwidth allocation scheme used.

For the CBR sources case (Figure 3.18), the use of a threshold has a big effect on the fairness result. For this case, we obtain a value equal to 95 and 66 when we use a linear

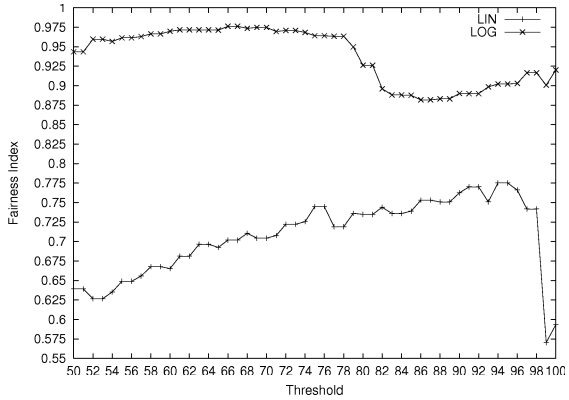


Figure 3.18: 32 CBR multicast source with a linear and a logarithm bandwidth allocation policies. We vary the value of the threshold from 50% to 100% of the maximum buffer size which is set to 64 packets.

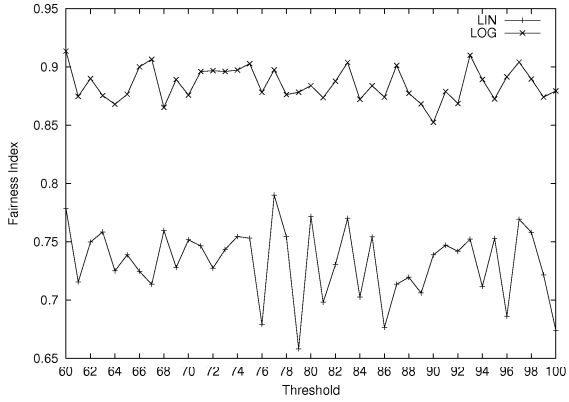


Figure 3.19: 32 FLID-DL multicast sources with a linear and a logarithm bandwidth allocation policies. We vary the value of the threshold from 60% to 100% of the maximum buffer size which is set to 64 packets.

Fairness Function / Traffic type	Linear		Logarithm	
	Threshold	Fairness Index	Threshold	Fairness Index
CBR Sources	95	0.7753	66	0.9716
FLID–DL Sources	77	0.7912	93	0.9151
Heterogeneous Sources	91	0.8651	69	0.7931

Table 3.1: Optimal threshold and their multicast fairness index values for three cases: all responsive, all non-responsive, and heterogeneous sources

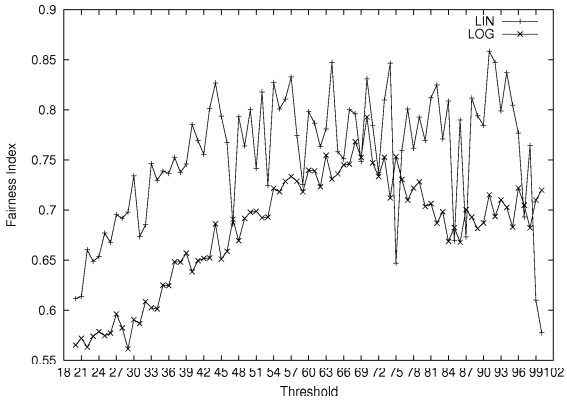


Figure 3.20: 16 FLID-DL multicast sources and 16 CBR sources with a linear and a logarithm bandwidth allocation policies. We vary the value of the threshold from 20% to 100% of the maximum buffer size

multicast allocation function and a logarithm allocation function, respectively.

We summarize in Table 3.1, the optimal threshold and the multicast fairness index values obtained for the three studied cases.

An interesting result from this analysis concerns the "best" multicast allocation function. In fact, we get the same conclusions given in [75], where the authors have shown that the logarithm policy achieves a "best" compromise between receiver satisfaction and fairness. Indeed, when using a logarithm allocation function, the multicast fairness index reaches its higher values: 0.96 and 0.92 for CBR and FLID-DL sources, respectively.

3.9 Chapter Summary

In this chapter, we have presented MFQ, a multicast active queue management mechanism that achieves the expected multicast bandwidth sharing using a single FIFO queue. MFQ interacts with a fairness module which implements the multicast bandwidth allocation policy. The queuing and the dropping decisions are designed in a manner to provide a bandwidth sharing being as close as possible to that achieved by the fluid model algorithm. MFQ uses a threshold to penalize high rate multicast sources and accept packets from new flows to be queued.

Without loss of generality, MFQ is evaluated for both linear and logarithmic bandwidth allocation functions. The scheme is also applied in the presence of both responsive and non-responsive flows. We showed that MFQ performs well even when there is a dynamic change in the number of receivers, since it converges very fast to the new expected bandwidth fair share.

We focused, in this chapter, on bandwidth sharing among multicast flows without taking into account the presence of unicast flows. In the next chapter, we address the challenge problem of network resource sharing between unicast and multicast flows in the Internet. In fact, this problem is directly related to the issue of developing multicast congestion control mechanisms.

Chapter 4

Bandwidth Sharing Between Unicast and Multicast Flows

4.1 Introduction

It is widely accepted that one of the several factors inhibiting the usage of the IP multicast is the lack of good and well-tested multicast congestion control mechanisms.

The precise requirements for multicast congestion control are perhaps open to discussion given the efficiency savings of multicast, but it is well known that a multicast flow is "acceptable" if it achieves no greater medium-term throughput to any receiver in the multicast group than would be achieved by a TCP flow between the multicast sender and that receiver. Such requirement can be satisfied either by a single multicast group if the sender transmits at a rate dictated by the lowest receiver in the group, or by a layered multicast scheme that allows different receivers to receive different number of layers at different rates.

In the previous chapter we focused on the bandwidth sharing between multicast competing flows. In this chapter, we develop a novel approach that helps multicast congestion control mechanisms to fairly exist with TCP protocol. Our approach is based on a new fairness notion, *the inter-service fairness*, which is used to share the bandwidth fairly between unicast and multicast services. Our fairness definition requires that the aggregated multicast traffic remain *globally TCP-friendly* in each communication link. In other words, the aggregated multicast average rate should not exceed the sum of their TCP-friendly rates. This approach allows the ISPs to define their own intra-multicast bandwidth sharing strategy which may implement either a multicast pricing policy |65| or an intra-multicast bandwidth allocation strategy |75|.

To implement our approach, we propose a two classes CBQ/WRR-like scheduler |59|: one for the unicast flows and the other one for multicast flows. We call our scheduler Service-Based Queuing (SBQ) because it distinguishes between two different transfer services: unicast and multicast services. SBQ integrates a method to dynamically vary the weights of the queues in order to match the expected bandwidth share between unicast and multicast flows. Upon a packet arrival, the router has to classify and redirect it to the appropriate queue.

It is important to bear in mind when reading this chapter that SBQ is only one, though arguably key, component of router-level congestion control support. In fact, the aim of this scheduler in the global architecture is to achieve the bandwidth sharing between unicast and multicast classes. In order to share the multicast bandwidth fairly among all competing multicast flows, we have proposed and validated in the previous chapter a new active queue

management mechanism for multicast flows called MFQ (Multicast Fair Queuing) which represents another key component. MFQ uses a single FIFO queue to share the bandwidth according to a pre-configured bandwidth allocation scheme.

To the best of our knowledge there is no prior work on unicast and multicast bandwidth sharing using a scheduling mechanism in the open literature. Furthermore, we consider our scheduler a promising avenue in developing congestion control mechanisms for multicast applications, and so an additional motivation for this work is to lay a sound basis for further development of multicast congestion control with a small but efficient help from the network.

In a first step, we consider best-effort networks. We use simulation to evaluate the effectiveness and performance of our scheme for various sources including not only TCP, UDP, and multicast CBR sources, but also multicast sources that implement the recently proposed layered multicast congestion control scheme FLID-DL [80]. Simulations are done for heterogeneous network and link characteristics and with different starting time, finish time, packet size, rate, and number of receivers in the multicast sessions.

The simulation results obtained for very heterogeneous sources and links characteristics suggest that, on the one hand, our scheduler achieves the expected aggregated bandwidth sharing among unicast and multicast service, and on the other hand the multicast flows remain TCP-friendly.

In a second step, we focus on using SBQ in networks supporting service differentiation. In fact, increasingly, the Internet is becoming a part of our day-by-day life, which yields into an exponential growth of IP based traffic, being this evolution particularly true for multicast realtime traffic which imposes on the underlying communication infrastructure the additional burden of QoS-sensitivity. IP-based Multiparty Videoconference services represent a typical example of multiparty, multimedia conversational services requesting for QoS-sensitive connectivity.

Recently, there has been a push from business and user communities for next generation applications demanding Quality of Service (QoS). However, the Internet in its current form does not support the notion of Quality of Service (QoS). Rather, the Internet follows the same-service-to-all paradigm in which all packets receive the same QoS. This best-effort service model is inadequate in meeting the growing demands of the next generation applications, most of which demand QoS assurances for effective data delivery and presentations.

There are two schools of thought for providing QoS to users across the Internet. The first school of thought is to increase the bandwidth available to users such that the extra capacity of the network allows all users to meet their appropriate QoS. By providing capacity beyond the needs of the users on the network, the network will not become congested and thus all users will be able to meet their QoS. In contrast, the second school of thought is that bandwidth can never be considered cheap and therefore the limited bandwidth should be appropriately prioritized among users. Whereas the first method provides QoS by default (i.e. without a complex scheduling model), the second method provides QoS through prioritization and network resource allocation. Both schools of thought provide valid concerns which introduce the two concepts considered in the second part of this chapter, namely multicast and the Differentiated Services (DiffServ) model [15].

One non-trivial research challenge is the task to integrate two different but complementary technologies like multicasting and DiffServ architecture. We propose a simple way to fairly share the available bandwidth among unicast and multicast flows inside each DiffServ class using the SBQ scheduler. We also extend the DiffServ architecture to enable the re-marking of multicast packets. The re-marking procedure assigns different priority to flows according to

the number of receivers downstream to each outgoing interface without modifying the DiffServ class. We evaluate the performance of our proposals through simulation.

The body of this chapter is organized as follows. In Section 4.2, we present the inter-service fairness notion. The fluid model algorithm is presented in Section 4.3. We explore and discuss many issues related to our scheduler from Section 4.5 to Section 4.8. The performance evaluation of SBQ in best-effort networks is detailed in Section 4.9. In Section 4.10, we turn our attention to the support of SBQ in DiffServ-enabled networks. We describe and evaluate an extension to DiffServ architecture to enhance the bandwidth sharing between unicast and multicast flows belonging to the same Diffserv class using our SBQ scheduler. Section 4.11 summarizes our main findings.

4.2 Inter-Service Fairness

In the informational IETF standard [82], the authors recommended that each end-to-end multicast congestion control should ensure that, for **each** source-receiver pair, the multicast flow must be TCP-friendly. We believe that this recommendation has been done because there is no network support to guarantee the TCP-friendliness and that it was an anticipated requirement which aims to encourage the fast deployment of multicast in the Internet. In addition, the multicast congestion control mechanisms that tried to achieve the TCP-fairness criterion are not always fair with TCP [116, 20] especially under variable network conditions.

We propose a new notion of unicast and multicast fairness called: the *inter-service fairness*. This notion is defined as follows.

Definition 4.2.1 *The Inter-service fairness: The multicast flows must remain globally TCP-friendly and not individually TCP-friendly for each flow. In other words, we should ensure that the sum of multicast flows rate does not exceed the sum of their TCP-friendly rates.*

The TCP throughput rate R_{TCP}, in units of **packets per second**, can be approximated by the formula in [60]:

$$R_{TCP} = \frac{1}{RTT\sqrt{q}(\sqrt{\frac{2}{3}} + 6\sqrt{\frac{3}{2}}q(1+32q^2))} \tag{4.1}$$

where R_{TCP} is a function of the packet loss rate q, the TCP round trip time RTT, and the round trip time out value RTO, where we have set $RTO = 4RTT$ according to [95]. Since the multicast analogue of RTT is not well defined, a target value of RTT can be fixed in advance to generate a "target rate" R_{TCP}.

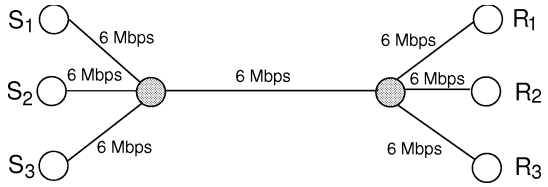


Figure 4.1: A topology used to illustrate the inter-service fairness definition. All links have the same Round Trip Time (RTT).

We illustrate the inter-service fairness definition using the topology shown in Figure 4.1. We consider two multicast sessions, one from source S_1 sending to R_1 and the other one from source S_2 sending to S_3 and one unicast session from source S_3 sending to R_3. Let r_1, r_2, and r_3, be the *inter-service fair share* of source S_1, S_2, and S_3, respectively. The TCP-friendly rate is equal to $\frac{6}{3} = 2$ Mbps given that all links have the same RTT. When applying the inter-service fairness definition to share the bandwidth between the three sources, S_1 and S_2 will get together an aggregated fair share equal to 4 Mbps and source S_3 will get alone a fair share r_3 equal to 2 Mbps. Our definition of the inter-service fairness does not specify the way how the bandwidth should be shared among multicast competing flows. Therefore, each vector (r_1,r_2) where $r_1 + r_2 = 4$ Mbps is considered as a feasible intra-multicast fair share solution. As explained in Section 3.3.1, in MFQ the pair (r_1,r_2) is given by an external module that implements a pre-defined bandwidth sharing function.

4.3 Fluid Model Algorithm

We consider a single link with a capacity equal to C. We assume n active TCP flows and m active multicast flows arriving to the link. The TCP source number i sends at the instantaneous rate $\alpha_i(t)$ and the multicast source number i sends at the instantaneous rate $\beta_i(t)$, in bits per second.

Unicast max-min fair bandwidth allocations [14] are characterized by the fact that all TCP flows that are bottlenecked (i.e., have packets dropped) by a router have the same output rate. We call this rate the *unicast allocation rate* of the server; let $\lambda(t)$ be the unicast allocation rate at time t. **In general, if max-min bandwidth allocations are achieved, each unicast flow i receives service at a rate given by $min(\alpha_i(t), \lambda(t))$.**

For multicast flows, the nature of the expected allocation is not yet well defined. It can be either determined by a fairness function scheme depending on the group membership [75] or by a multicast pricing model [65]. To be independent of the allocation strategy used, we define a vector $\gamma(t) = (\gamma_1(t), \gamma_2(t), ..., \gamma_n(t))$ giving the expected allocation at the instantaneous time t. As explained in Section 3.3.1, the value of $\gamma_i(t)$ may be either a set value or a function of the number of competing multicast groups, the number of flows per group, and the number of receivers per flow. We call this vector of fair, the *multicast allocation vector*. If the fair share is achieved, the multicast flow number i receives service at a rate given by $min(r_i(t), \gamma_i(t))$.

Let $A(t)$ be the total arrival rate: $A(t) = \sum_{i=1}^{i=n} \alpha_i(t) + \sum_{i=1}^{i=m} \beta_i(t)$. If $A(t) > C$ the congestion phenomena holds and then the unicast allocation rate $\lambda(t)$ and the multicast allocation vector $\gamma(t)$ which corresponds to the inter-service fairness definition given in Section 4.2 are the unique solution to

$$\sum_{i=1}^{i=n} min(\alpha_i(t), \lambda(t)) + \sum_{i=1}^{i=m} min(\beta_i(t), \gamma_i(t)) = C \tag{4.2}$$

subject to:

$$\sum_{i=1}^{i=m} \gamma_i(t) \leq m\lambda(t) \tag{4.3}$$

This constraint is added in order to guarantee that the aggregated multicast bandwidth share rate does not exceed that of m equivalent unicast flows. In other words, each multicast flow will get an average rate equal to $\lambda(t)$.

If $\alpha_i(t) > \lambda(t)$, then the fraction of bits $\frac{\alpha_i(t)-\lambda(t)}{\alpha_i(t)}$ will be dropped, and the unicast flow i will have an output rate of exactly $\lambda(t)$. The arrival rate to the next hop is given by $min(\alpha_i(t), \lambda(t))$.

If $\beta_i(t) > \gamma_i(t)$, then the fraction of bits $\frac{\beta_i(t)-\gamma_i(t)}{\beta_i(t)}$ will be dropped, and the multicast flow i will have an output rate of exactly $\gamma_i(t)$. The arrival rate to the next hop is given by $min(\beta_i(t), \gamma_i(t))$.

As mentioned above, the multicast allocation vector can be computed using a multicast fairness function that depends on the number of receivers which are distributed in the multicast delivery tree. Therefore, the number of downstream receivers in each router of the multicast delivery tree does not remain constant because some receivers may be reached via different interfaces. Thus, $\gamma(t)$ may be different in the tree branches even when there is no more competing multicast flow. To illustrate this, we consider two successive routers r_1 and r_2 composing a branch of the multicast tree of flow i and denote by R_1 and R_2 the number of receivers downstream to r_1 and r_2, respectively. One can easily write the following inequality $R_2 \leq R_1$ because router r_1 is close to the multicast source than router r_2. Hence, an immediate inequality between the flow weights in the two routers holds: $\gamma_i^2(t) \leq \gamma_i^1(t)$ which means that the flow weight in router r_2 ($\gamma_i^2(t)$) is less than or equal to that in router r_1 ($\gamma_i^1(t)$).

4.4 Principals and Architecture

In order to implement the fluid model algorithm that integrates our inter-service fairness definition given in Section 4.2, we propose to use a CBR/WRR-like scheduler [59] but with only two queues, one for each class as shown in Figure 4.2. We call our scheduler SBQ (Service-Based Queuing) because it differentiates between the packets according to their transfer service: unicast or multicast.

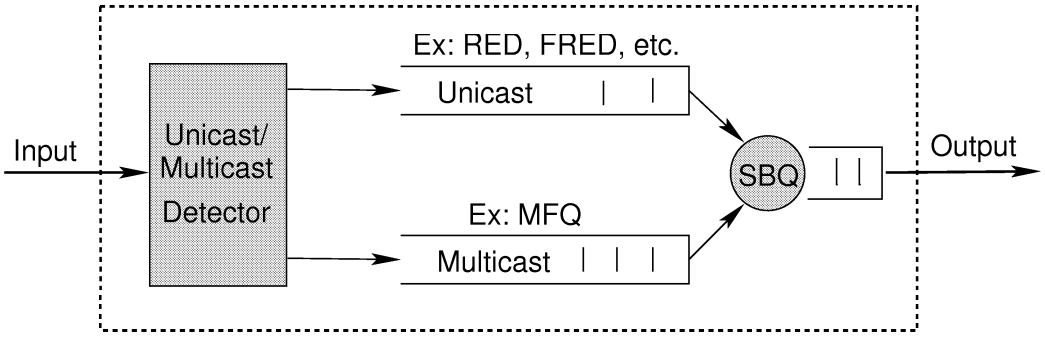


Figure 4.2: SBQ scheduler architecture

Before being queued, unicast and multicast packets are classified into two separated classes. We use a modified version of the Weighted Round Robin (WRR) algorithm which is the most widely implemented scheduling algorithm as of date [59]. This is due to its low level of complexity and its ease of implementation which follows hence. In this scheduler, packets receive service according to the service quantum[1] of flows to which they belong. Each queue

[1]The service quantum of a given flow is expressed in term of packets belonging to this flow which could be served in a service round.

has an associated weight. In every round of service, the number of packets served from a queue is proportional to its associated weight and the mean packet size.

As buffer management schemes: we use MFQ mechanism that we have proposed in the previous chapter for the multicast queue and another queue management mechanism (RED, FRED, WRED, etc.) for the unicast queue.

The SBQ scheduler's operation is illustrated by the following example: consider a WRR server that uses two queues: U (for Unicast) and M (for Multicast) with weights 0.6 and 0.4, respectively. Let the mean packet size for U and M be 1000 Bytes and 500 Bytes, respectively. The weights are first normalized by the mean packet size to get 0.6 for class U and 0.8 for class M. The normalized weights are then multiplied by a constant to get the lowest possible integer. In this case we get 6 and 8 for U and M, respectively. This means that in every round of service 6 packets are served from queue U and 8 packets are served from queue M.

4.5 Scheduler Configuration

When configuring WRR to share the link bandwidth between the two classes. At time t we configure the multicast class (M queue) with a weight equal to $X(t)$ and the unicast class (U queue) with a weight equal to $(1 - X(t))$.

To implement our inter-service fairness notion defined in Section 4.2 and to ensure that the bandwidth sharing is as close as possible to the fluid model algorithm developed in Section 4.3, we propose to update the weight $X(t)$ at time t as follows:

$$X(t) = \min\left(\frac{\sum_{i=1}^{i=m(t)} R_{TCP_i} * S_i}{C}, \frac{m(t)}{u(t) + m(t)}\right) \tag{4.4}$$

where:

- R_{TCP_i} is the TCP-friendly throughput rate of the multicast flow i estimated using Eq. 4.1,
- S_i is the average packet size in **bytes**,
- C is the link capacity in **bytes per second**,
- $u(t)$ is the number of active unicast flows at time t,
- $m(t)$ is the number of active multicast flows at time t.

To be **globally fair** against TCP-connections, the sum of the rates of $m(t)$ active multicast flows must not exceed the sum of that of their $m(t)$ single TCP flows rate over large time scales. This corresponds to the first term of Eq. 4.4. The second term of this equation allocates instantaneous bandwidth fairly between unicast and multicast flows by sharing it proportionally to the number of flows in the two queues. Using this simplistic and efficient formula of $X(t)$, we can guarantee both short-term and long-term fairness. Indeed, the two terms used in $X(t)$ configuration allow us to ensure both short-term max-min fairness between active flows based on a local information about the number of flows and a long-term TCP-friendliness between active sessions based on a global information concerning the rates of multicast sources. The TCP-friendliness term (the first one) is useful when there is another bottleneck in the multicast delivery tree.

A possible way to extend the configuration of $X(t)$ is to add a third term referring to the maximum portion of the link capacity that should not be exceeded by multicast flows. This value can be tuned by the ISP depending on a chosen policy used to handle multicast connections crossing its network. In this case, the configuration of the weight $X(t)$ of the multicast queue will be done as follows:

$$X(t) = \min\left(\max\left(\frac{\sum_{i=1}^{i=m(t)} R_{TCP_i} * S_i}{C}, minBDW_{mcast}\right), \frac{m(t)}{u(t)+m(t)}\right), \qquad (4.5)$$

where $minBDW_{mcast}$ is the minimum capacity fraction that should be given to multicast flows[2].

Upon each change on the number of active flows in the unicast or multicast queue, the weights of both queues are updated to match the new fairness values. Both queues priority are set to 1.

To compute the value of $X(t)$, each SBQ router has to know the TCP-friendly rate of active multicast flows and maintain only the aggregated rate to be used in the first term of Eq. 4.4.

The TCP-friendly rate of a multicast session corresponds to the sending rate of the source. In single rate multicast transmission protocols, such as TFMCC [119], every receiver periodically estimates its TCP-friendly reception rate using for example the formula 4.1 and reports this rate to the source. The source determines the lowest rate among all receivers rates and uses it to send data downstream to all receivers. In multi-rate multicast transmission such as RLC |116|, WEBRC |79| and FLID-DL |20|, the source sends data using several layers. For each layer, the source uses a specific sending rate depending on the data encoding scheme. The receivers join and leave the layers according to their reception TCP-friendly reception rates computed using Eq. 4.1.

For both cases, the source TCP-friendly throughput rate can be included in the IP packet header by the multicast source or the source's Designated Router (DR). This technique is largely used by many other mechanisms such as CSFQ |111|, TUF |29| for different purposes. Thus, a SBQ intermediate router gets the rates from the IP multicast packet headers and computes their aggregated value according to Eq. 4.4 or Eq. 4.5.

4.6 Weights updating time

In this sub-section, we answer the question: How often we update the weights? In other words, what is the time-scale on which we should look at the bandwidth allocation. There is a tradeoff between complexity and efficiency when choosing the time-scale value. Indeed, larger time-scale are not suitable for short-lived TCP connections which are the most of TCP connections currently in the Internet[3]. On the other hand, what happens on shorter time-scales if we consider fairness on longer time-scale. We do not claim that there is an optimal value of the time-scale which can be applied for each type of traffic and which leads to both less complexity and good efficiency.

The time-scale is designed to allow the update of weights to take effect before their values are updated again. It should be therefore longer than the average round trip time (RTT)

[2]The weight of the unicast queue is therefore equal to $1 - X(t)$.

[3]Internet traffic archive: `http://www.cs.columbia.edu/~hgs/internet/traffic.html`

among the connections passing through the router. In the current Internet, it can be set in the range from 10 ms up to 1 sec, so this property can guarantee that the unfairness can't increase very quickly and make the queue parameters stable. But this parameter can't be set too big so that the scheduler weight can't be adaptive enough to the network dynamics.

We will show in the simulation section that the use of a time-scale equal to 1 sec, which we believe an accepted value, can provide a good tradeoff between efficiency and complexity.

4.7 Counting Unicast Connections

To update the value of the weight used in our scheduler computed using Eq. 4.4, we need to know the number of multicast and unicast flows. While the former is provided by MFQ, the latter could be obtained through the use of a flow-based unicast active queue management mechanism such as FRED [78]. However, unicast flow-based AQM are not available everywhere and most of current routers use a FIFO or RED [58] schemes which don't provide the number of unicast connections. That's why, we propose hereafter a simple method (which we use) to estimate the number of unicast connections in the unicast queue.

SBQ counts active unicast connections as follows. It maintains a bit vector called v of fixed length. When a packet arrives, SBQ hashes the packets connection identifiers (IP addresses and port numbers) and sets the corresponding bit in v. SBQ clears randomly chosen bits from v at a rate calculated to clear all of them every few seconds. The count of set bits in v approximates the number of connections active in the last few seconds. The SBQ simulations in Section 4.9 use a 5000-bit v and a clearing interval (t_{clear}) of 10 seconds. The code shown in Algorithm 4.1 (hereafter) may under-estimate the number of connections due to hash collisions. This error will be small if v has significantly more bits than there are connections. This method of counting connections has two good qualities. First, it requires no explicit cooperation from unicast flows. Second, requires very little state: on the order of one bit per connection.

Algorithm 4.1 : The algorithm for counting unicast connections

Connection count(packet p):

```
h ← H(p)
if v(h) ← 0
  v(h) ← 1
  N ← N + 1
t ← currentTime
n_clear ← v_max (t − t_last) / t_clear
if n_clear > 0
  t_last ← t
   for i = 1 to n_clear - 1
    r ← random(0...v_max - 1)
    if v(r) = 1
       v(r) ← 0
       N ← N + 1
return (N)
```

Variables:

$v(i)$	Vector of v_{max} bits. $v(i)$ indicates if a packet from a connection with hash i has arrived in the last t_{clear} seconds.
N	Count of one bits in v.
t_{last}	Time at which bits in v were last cleared.
r	Randomly selected index of a bit to clear in v. Constants: v_{max} Size of v in bits; should be larger than the number of expected connections.
t_{clear}	Interval in seconds over which to clear all of v.
$H(p)$	Hashes a packet connection identifying fields to a value between 0 and v_{max}.

4.8 Complexity and Deployment Issues

The deployment of WRR-like algorithms in the Internet may raise some open questions for large deployment. The scalability issue is the main barrier of their large deployment in the Internet. We mean by the scalability, the ability of the mechanism to process a very large number of flows with different characteristics at the same time.

We believe that our scheduler can be deployed in large networks thanks to two mainly key points:

- it uses only two queues, so we need to classify only two types of service: unicast and multicast flows. This task has already been done in part by the routing lookup module before the packet being queued.
- all unicast flows are queued in the same queue.

It is important to note that we usually associate the flow-based mechanisms support with complexity and scalability problems since they require connection specific information. These concerns are justifiable only in point-to-point connections, for which routing tables do not maintain connection-specific state. In multicasting, routing tables keep connection specific state in routers anyway; namely, the multicast group address refers to a connection. Thus, adding multicast flow specific information is straightforward and increases the routing state only by a fraction.

Comparing to CBQ/WRR, our mechanism is less complex to be deployed in the Internet. It should be noted that even CBQ is now supported by a large number of routers and many research works such as [110] demonstrate its deployment feasibility.

One major advantage of our approach is that it minimizes the complexity of designing multicast congestion control schemes. Indeed, the network guarantees that the multicast flows will share fairly the bandwidth with competing unicast flows. Moreover, our scheme provides to the ISPs a flexible way to define and implement their own intra-multicast fairness strategy. The simulation results presented in the next section will confirm our claims.

4.9 Performance Evaluation of SBQ in Best-Effort Networks

We have implemented the SBQ scheduler in the ns-2 network simulator [84] and we conducted several experiments to evaluate its performance for different traffic characteristics.

4.9.1 Single Bottleneck Link

We validate our scheme for a topology consisting of a single congested link connecting two routers n_1 and n_2 and having a capacity C equal to 1 Mbps and a propagation delay D equal to 1 ms. As shown in Figure 4.3, sources are connected to router n_1 and all destinations are downstream to router n_2.

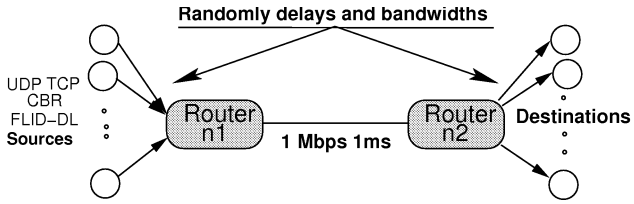


Figure 4.3: A single congested link simulation topology. The congested link has a capacity of 1 Mbps and 1ms propagation delay.

We configure our scheduler in the bottleneck link from router n_1 to router n_2. The maximum buffer size $qlim$ of both queues is set to 64 packets. Other links are tail-drop and they have sufficient bandwidth to avoid packets loss. We consider responsive and non-responsive sources and heterogeneous links with different delay and bandwidth.

We assume 32 multicast sources and 32 unicast sources that compete to share the link capacity. We index the flows from 1 to 64. The 32 multicast sources are divided as follows:

- Flows from 1 to 16: CBR sources. These sources implement no type of congestion control mechanism (neither application-level nor transport-level).

- Flows from 17 to 32: FLID-DL (Fair Layered Increase-Decrease with Dynamic Layering) [20] sources. As we have outlined earlier, the protocol FLID-DL uses a Digital Fountain [21] at the source in which the sender encodes the original data and redundancy information such that receivers can decode the original data once they have received a fixed number of arbitrary but distinct packets. The FLID-DL simulation parameters are the same as those recommended in [20][4].

Each source uses 17 layers encoding, each layer is modeled by a UDP traffic source. In Tab. 4.1, we give the values of different parameters used.

As outlined earlier, we use MFQ as the active queue management in the multicast queue. Without loss of generality, we utilize a receiver-dependent logarithm policy (the LogRD policy) proposed in [75] to share the bandwidth between multicast flows. This policy consists in giving to the multicast flow number i a bandwidth fraction equal to $\frac{1+\log n_i}{\sum_j (1+\log n_j)}$, where n_j is the number of receivers of flow j.

The 32 unicast sources are composed as follows:

- Flows from 33 to 48: UDP sources. These unicast sources transmit packets at different constant bit rates (CBR unicast sources).

[4]These parameters are used in the ns implementation of FLID which is available at `http://dfountain.com/technology/library/flid/`.

Parameter	Description	Value
c_mult_	the multiplicative factor	1.3
slot_time_	the slot time	0.5
number_of_layers_	the number of layers	17
simulated_rtt_	for tcp rate value calculation	0.1
rng_speed_	the random generator's speed	0
packet_payload_	the number of bytes in packet	1000

Table 4.1: FLID-DL parameters used in simulation

- Flows from 49 to 64: TCP sources. Our TCP connections use the standard TCP Reno implementation provided with ns-2 network simulator.

Unless otherwise specified, each simulation lasts 100 seconds and the weight updating period is set to 2 sec. Other parameters are chosen as following:

- packet size: the packet size of each flow is randomly generated between 500 and 1000 bytes.
- starting time: the starting time of each flow is randomly generated between 0 and 20 seconds before the end of the simulation.
- finish time: the finish time of each flow is randomly generated between 0 and five seconds before the end of the simulation.
- rate: the rate of both unicast and multicast UDP flows is randomly generated between 10 Kbps and 100 Kbps.
- number of receivers: the number of downstream receivers of the 32 multicast sessions is randomly generated between 1 and 64.

The four first unicast UDP and multicast CBR flows are kept along the simulation time (start time = 0 sec, and finish time = 100 sec) to be sure that the link will be always congested. Each one of these flows is sending at a rate equal to $\frac{1 \text{ Mbps}}{8} = 125$ Kbps. Initially (at $t = 0$ sec), the weight X of the SBQ scheduler (see Eq. 4.4) is set to 0.5.

In Figure 4.4, we plot the variation of unicast and multicast queue weights in function of the simulation time. As we can easily see the value of weights change during the simulation because they depend on the number of active unicast and multicast flows in the two queues of the SBQ scheduler. The multicast weight increases when the unicast weight decreases and vice versa.

We look at the inter-service fairness defined in Section 4.2 which is the main performance metric of our scheme. To this end, we compare the average unicast and multicast rates over

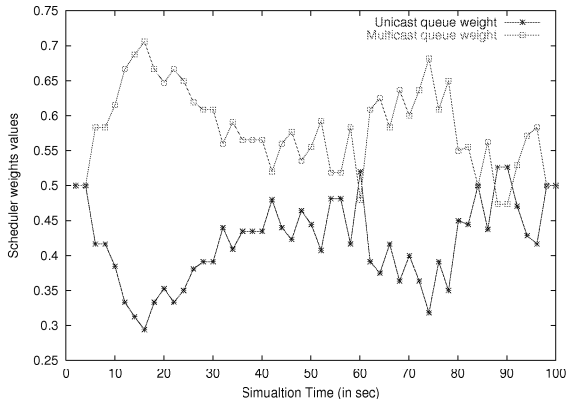


Figure 4.4: Scheduler weights variation in function of the simulation time

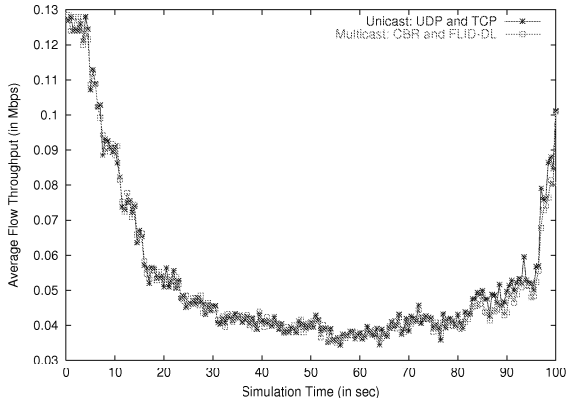


Figure 4.5: The variation of the average unicast and multicast average rate in function of the simulation time over 500 msec

500 $msec$ of simulations. We show in Figure 4.5, the variation of the unicast and the multicast average rate in function of the simulation time.

As we can observe, the results match exactly what we expect. Indeed, two main observations can be derived from the plots of Figure 4.5. Firstly, there is a fluctuation on the average aggregated rate for unicast and multicast flows which due to the random start and finish time of all flows. Secondly, the multicast average rate is very close to the unicast average rate. This demonstrates the ability of SBQ to share the bandwidth fairly between unicast and multicast flows according to our inter-service fairness notion defined in Section 4.2.

In order to evaluate the impact of the time-scale value on the performance of SBQ, we measure the normalized aggregated rate (average unicast rate/average multicast rate) obtained for various values of the time-scale. In Figure 4.6, we show the variation of this metric in function of the time-scale value. We can conclude that using a 1 sec time-scale allows to reach 93 % of the performance of SBQ. In other words, the use of an updating period equal to 1 sec leads to a good tradeoff between complexity and efficiency.

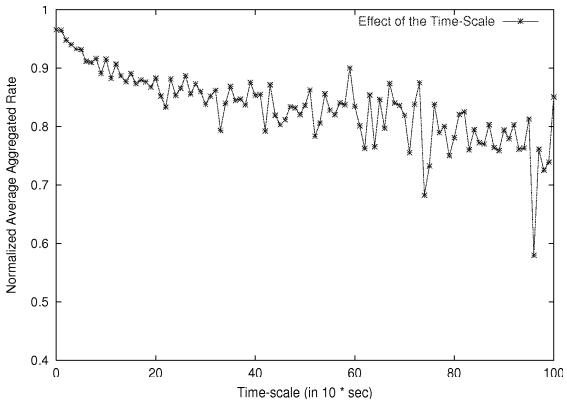


Figure 4.6: Sensitivity of SBQ performance on the time-scale value

We claimed earlier that our scheduler is flexible in the sense that its performance is independent of the intra-multicast fairness function. To argument this claim, we compare in Figure 4.7 the normalized rate over 500 $msec$ of simulation time for linear (LIN), logarithm (LOG), and receiver-independent (RI) bandwidth sharing policies which are defined in [75][5]. As we can see the normalized average rate is always varying around 1 (between 0.82 and 1.18) for the three cases.

We study the bandwidth shared among multicast flows in the multicast queue of the SBQ scheduler provided by the MFQ buffer management mechanism that we have proposed in Chapter 3. The multicast flow number i is assumed to have exactly i downstream receivers (members). We use a logarithm multicast bandwidth allocation scheme and we vary the number of UDP unicast flows from 1 to 16 flows. In Figure 4.8, we show the bandwidth fair rate obtained for the 32 multicast flows. It is very clear from the plots that the shape of flows rate curves follows a logarithm function of the number of downstream receivers given that it was assumed to be equal to the flow index.

[5]Recall that assume n active multicast flows and denote by n_i the number of downstream receivers of flow i. The RI, LIN, LOG bandwidth sharing policies consist to give to flow i a bandwidth share equal to $\frac{1}{n}$, $\frac{n_i}{\sum_j n_j}$, and $\frac{1+\log n_i}{\sum_j (1+\log n_j)}$, respectively.

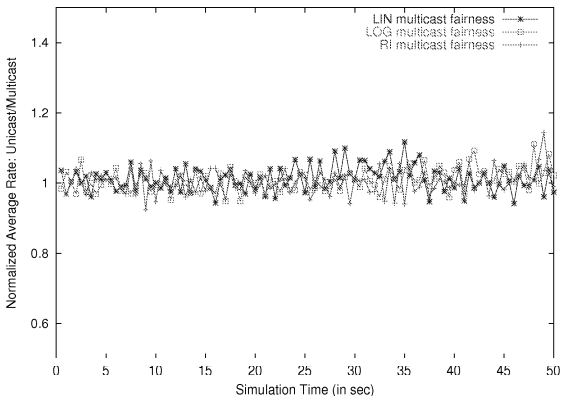


Figure 4.7: The normalized rate when modifying the intra-multicast fairness function over 500 msec of simulation time

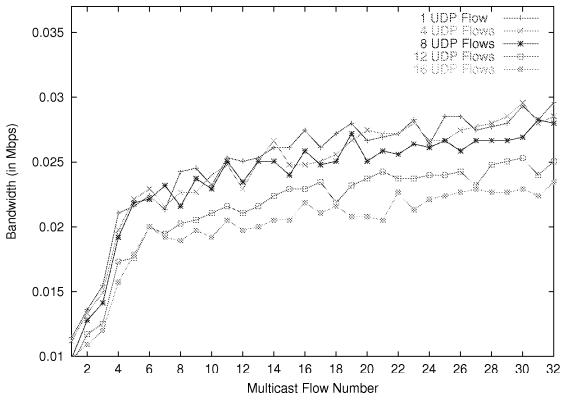


Figure 4.8: Rates of multicast flows when using a logarithm multicast bandwidth sharing function

4.9.2 Multiple Bottleneck Links

In this sub-section, we analyze how the throughput of multicast and unicast flows is influenced when the flow traverses L congested link. We performed two experiments based on the topology of Figure 4.9. We index the links from 1 to k.

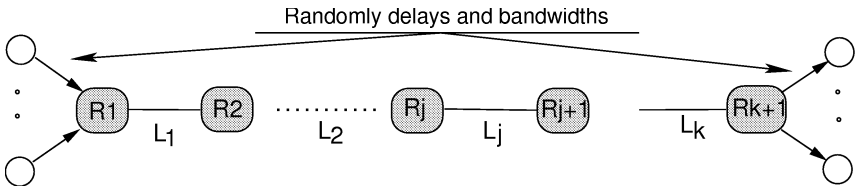


Figure 4.9: Topology for analyzing the effects of multiple congested links on the performance of SBQ

We use exactly the same traffic parameters as the case of single bottleneck described above and we measure the aggregated bandwidth received by each service type (unicast or multicast) in function of the number of congested links. Again, a link L_j, $2 \leq j \leq 10$, is kept congested by setting its capacity C_j to $C_{j-1} - 50$ Kbps. The capacity C_1 of the link number 1 is set to 1 Mbps.

We plot in Figure 4.10, the variation of the normalized average rate as a function of the number of congested links. As we can see, the normalized average rate remains close to 1 even when the number of congested increases.

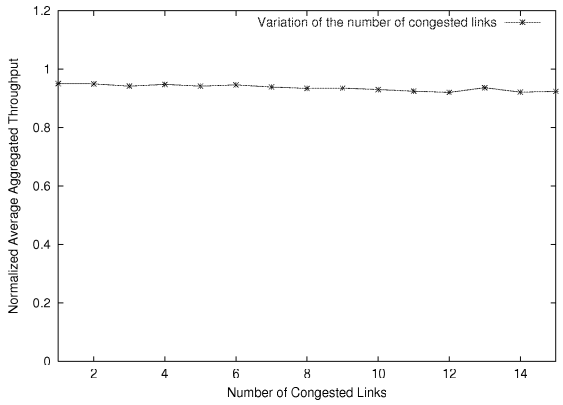


Figure 4.10: The normalized rate as a function of the number of congested links

4.10 Integrating SBQ in DiffServ Architecture

Till now, we only considered the coexistence problem of unicast and multicast flows in best-effort networks. In this section, we emphasize on the issue of integrating SBQ in DiffServ architecture in order to enhance the multicast deployment in diffserv-enabled networks.

4.10.1 Problem Statement

As the available bandwidth to end users increases, new applications are continually being developed which erode gains in network capacity. Thus, for the foreseeable future, some form of resource provisioning is necessary to provide QoS across the Internet. One of the more promising models for providing QoS across the Internet is the Differentiated Services (DiffServ) model [15]. The DiffServ model attempts to provide scalable QoS across the Internet while allowing for gradual deployment across the Internet.

Integration of multicasting support in the DiffServ domain is useful in three aspects. First, DiffServ provides a method for service differentiation in the Internet while multicasting provides a method for conservation of network bandwidth. The integration of DiffServ and multicasting provides a data delivering model that can provide service differentiation (thanks of the use of DiffServ) with conservation of network bandwidth (which is the main advantage of using multicast service). Second, it is likely that some form of DiffServ will be implemented in the next generation Internet. Therefore, multicasting support in the DiffServ domains will be useful from an implementation and performance standpoint. Third, several evolving continuous media applications have a variety of QoS requirements and are predominantly group-oriented. In addition, these applications consume a large amount of network bandwidth. Thus, the support of multicasting in the DiffServ domain will be able to meet these goals for the evolving classes of applications.

Thus, from an initial glance, it would appear that multicast and DiffServ are complementary technologies. Whereas multicast attempts to conserve network bandwidth, DiffServ attempts to provision the bandwidth in an appropriate fashion to users.

The integration of multicast and DiffServ is a non-trivial task due to fundamental architectural conflicts between the two technologies. One of the fundamental differences between DiffServ and multicast lies within the structure of the multicast tree. With multicast-aware routers (traditional IP multicast), the tree structure is maintained in the routing table. Packets are appropriately replicated onto links based on entries inside the routing table. However, under DiffServ (DS), all core routers are assumed to be simple routers maintaining no per-flow state information across the DS domain. Each core router is assumed to be independent of the other core routers and must react to the flows according to a PHB (Per-Hop Behavior) as identified by the DSCP in the packet. Information for the PHB of the packets is maintained on a per-class basis and that information is maintained for each individual core router only.

Given that the per-flow state in the multicast routing table is necessary for the forwarding of incoming multicast packets to the outgoing interfaces (see Section 4.8 for more details), we propose to take advantage from this state to reach two goals:

- enhance the bandwidth sharing between unicast and multicast flows, and
- improve the network resource sharing among multicast flows in DiffServ-enabled networks.

The first goal is achieved through the integration of SBQ in the DiffServ architecture, while the second one is guaranteed with the development of re-marking methods for multicast flows that take into account the number of downstream members. In the following sub-sections, we explore our proposed solutions to realize both goals.

4.10.2 Enhancing Unicast and Multicast Bandwidth Sharing

According to the DiffServ specifications [15], there are fourteen different "classes" of service: one EF (Expected Forwarding) class, four AF (Assured Forwarding) classes and one BE (Best Effort) class. Within each AF_x class, it is possible to specify three drop precedence values. Thus, if there is congestion in a DS-node on a specific link, and packets of a particular AF_x class (say AF_1) need to be dropped, packets in AF_{xy} will be dropped such that the $dP(AF_{x1}) \leq dP(AF_{x2}) \leq dP(AF_{x3})$, where $dP(AF_{xy})$ is the probability that packets of the AF_{xy} class will be dropped.

We propose to use SBQ in order to share the bandwidth fairly between unicast and multicast flows in BE and AF DiffServ classes as shown in the core router architecture of Figure 4.11. SBQ shares the bandwidth between both "services" (or transfer mode) according to the inter-service fairness criterion that we have introduced in Section 4.2. As explained in Section 4.2, each ISP may also explicitly specify the portion of the bandwidth that should be given to each "service" (unicast or multicast transfer mode) and for each class of service (AF or BE). As we can observe from the figure, we do not modify the behavior of the EF class because the EF PHB (Per-Hop Behavior) is the key ingredient in DiffServ for providing a low-loss, low-latency, low-jitter, and guaranteed bandwidth service. Applications such as voice over IP (VoIP), video, and online trading programs require such a robust network-treatment.

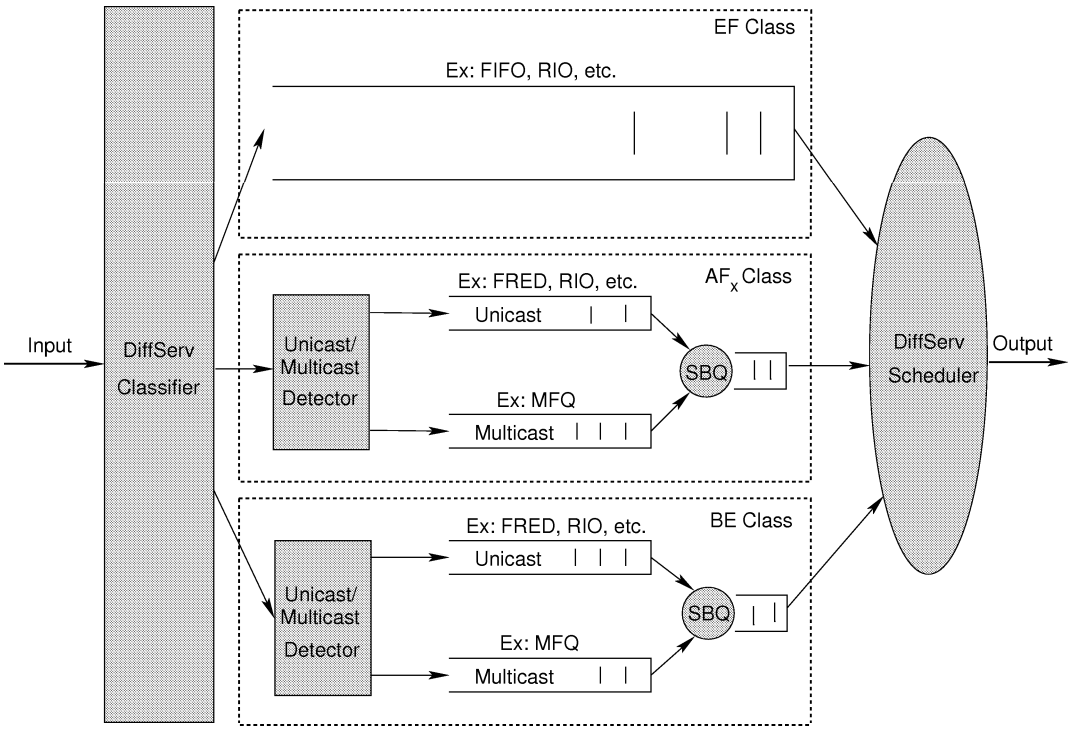


Figure 4.11: Integrating SBQ in DiffServ Architecture

Recall that SBQ provides two queues: one for unicast flows and one for multicast flows. Each virtual queue is managed by a specific buffer management mechanism in order to provide fairness among competing flows. The multicast buffer management mechanism (MFQ) maintains a per-flow state table only for active flows.

The DiffServ multi-field classifier (MFC) has the task to redirect each incoming packet to the appropriate diffserv class block. If the class of service is BE or AF, the packet enters the corresponding SBQ scheduler and the unicast/multicast detector forwards this packet to the corresponding queue. The diffserv scheduler decides which packet to dequeue from the EF queue and the EF and BE SBQ scheduler instances.

4.10.3 Re-marking Multicast Packets

As we have outlined earlier, DiffServ (DS) routers use the DSCP field to determine the service that should be attributed to each incoming packet. Given that DS core routers do not modify the value of the DSCP, each packet is expected to get the same service along the path between the source and the destination(s). One motivation behind the no modification of the DSCP value inside a DiffServ (DS) network is to minimize the complexity of DS routers especially for unicast connections for those there is no per-flow state maintained at the router. Or, for multicast flows there is already a state maintained by the multicast routing protocol for each multicast active session. The multicast packets are replicated to the outgoing interfaces where there is at least a downstream member.

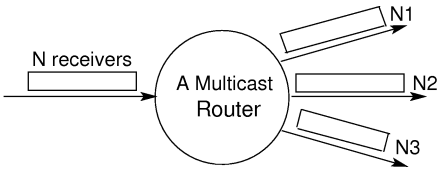


Figure 4.12: Multicast packets replication in a multicast router. There are N_i receivers which are downstream to the outgoing interface i so that the total number of members N is equal to $\sum_i N_i$.

As shown in Figure 4.12, IP Multicast packet replication usually takes place when the packet is handled by the multicast router, i.e., when it is forwarded according to the multicast routing table. Each incoming multicast packet is replicated to every interface belonging to the list of the outgoing interfaces of the corresponding multicast session. Thus, a DiffServ capable router would also copy the content of the DS field |15| into the IP packet header of every replicate. Consequently, replicated packets get exactly the same DSCP as the original packet, and, therefore experience the same forwarding treatment as the incoming packets of this multicast group.

We believe that a multicast packet should get distinct services along the multicast delivery tree given that the number of receivers change down to the leaf members. Therefore, keeping the same DSCP along the delivery tree may not be efficient for multicast flows. We argue that it is possible to re-mark the multicast packets with new DSCP values according, for example, to their number of downstream members in each outgoing interface. This is very useful for ISPs that want to differentiate between multicast flows according to the number of downstream receivers using a specific bandwidth allocation function like those proposed in |75|.

Let's take and example to illustrate the advantages of re-marking multicast packets. Assume a multicast flow (S_1, G_1) that serves ninety receivers and a multicast flow (S_2, G_2) serving ten receivers and that both flows belonging to the same DiffServ class. The DiffServ routers do

not differentiate between the two flows and they try to give them the same service. However, if a packet belongs to the (S_1, G_1) flow was lost, only 90% of the total number of users will be affected. In the case when we attribute a "priority" to the flow (S_1, G_1) higher than that of flow (S_2, G_2) (lower drop precedence) without modifying the class of service, the the number of receivers affected by the packet loss will decrease, which could be intuitively accepted given that the first flow serves more receivers than the second flow.

We propose to dynamically adapt the DSCP value of each incoming multicast packet by taking into account the number of downstream receivers which it is expected to serve. Before being queued to the corresponding DiffServ class (AF_{x1}, AF_{x2} , AF_{x3}, or BE), we require that the DSCP of the incoming multicast packet should be updated. **The goal is then to develop an efficient and well-accepted method to modify the DSCP of multicast packets in DiffServ core routers in each outgoing interface according to the number of downstream receivers.** Note that, we do not change the DiffServ class of the multicast packet, we only change the drop precedence associated to it.

In the following, we propose and discuss three main mapping methods based on the number of members.

We maintain for each outgoing interface and for each AF or BE DiffServ class, the following two parameters[6]:

- MAX: the maximum number of downstream members among all competing multicast flows belonging to the same class of service.
- MIN: the minimum number of downstream members among all competing multicast flows belonging to the same class of service.

The values of the two above variables depend on the flows to which belong the packets waiting in the multicast queue and they are maintained and updated by the buffer management mechanism MFQ described in the previous chapter.

The idea is to divide the space between the two values MIN and MAX to three subspaces. Each one corresponds to a specific drop precedence from those defined in the DiffServ standard. The multicast flow which has the highest number of members will get the highest priority, i.e. the lowest drop precedence and vice-versa.

We follow the same methods used to share the bandwidth between multicast competing flows in best effort networks that we have described in Section 3.3.1 and which will be implemented in the bandwidth allocation module of MFQ (see Section 3.3.2). Thus, we define three ways to map the number of members of a flow to a drop precedence (and so to a DSCP value) among the three available values (1, 2, and 3).

The first way consists on dividing linearly the space between the maximum and the minimum value of the number of members. Therefore, three ranges can be obtained as shown in Figure 4.13. We call this kind of mapping, the **LIN mapping**.

Let us take a simple example to explain how this scheme works. Suppose that the minimum number of receivers in the AF_x queue is 2 which implies that $MIN = 2$ and the maximum number of receivers in the same AF_x queue is 11, i.e. $MAX = 11$. Let N_R be the number of downstream receivers of the incoming packet. When applying the LIN mapping method, the new DSCP value which will be attributed to the multicast packet depends on the value of N_R as follows:

[6]As an example we consider only AF classes, the same methods and discussions are applied also to the BE class.

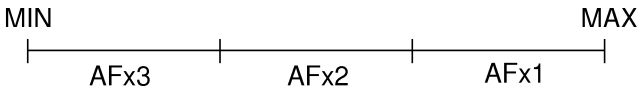
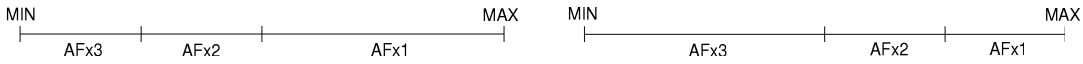


Figure 4.13: The LIN mapping method

- $2 \leq \mathrm{N_R} \leq 5$: the packet will be re-marked with the DSCP of AF_{x3} class: low priority packets.
- $5 < \mathrm{N_R} \leq 8$: the packet will be re-marked with the DSCP of AF_{x2} class: medium priority packets.
- $8 < \mathrm{N_R} \leq 11$: the packet will be re-marked with the DSCP of AF_{x1} class: high priority packets.

The second way to do the mapping between the number of receivers and the drop precedence is to divide logarithmically the receivers number space. We explain, in Figure 4.14, the partitioning of receivers number space using the Log Mapping method. We distinguish two log-based mapping schemes: the **Log top mapping** scheme (Algorithm 4.2) and the **Log bottom mapping** scheme (Algorithm 4.3)



(a) Log top mapping

(b) Log bottom mapping

Figure 4.14: Log-based mapping schemes

Algorithm 4.2 : The pseudo-code of the Log top mapping algorithm

$interval \longleftarrow \frac{MAX-MIN}{4}$,
$MAX' \longleftarrow MAX - (2 * interval)$.
$MAX'' \longleftarrow MAX' - interval$.
`if` $N_R \in [MAX',\ MAX[$ `then`
    `class` $\longleftarrow AF_{x1}$
`else if` $N_R \in [MAX'',\ MAX'[$ `then`
    `class` $\longleftarrow AF_{x2}$
`else if` $N_R \in [MIN,\ MAX'']$ `then`
    `class` $\longleftarrow AF_{x2}$

Algorithm 4.3 : The pseudo-code of the Log bottom mapping algorithm

$interval \longleftarrow \frac{MAX-MIN}{4}$,
$MAX' \longleftarrow MAX - interval$.
$MAX'' \longleftarrow MAX' - interval$.
`if` $N_R \in [MAX',\ MAX[$ `then`
    `class` $\longleftarrow AF_{x1}$
`else if` $N_R \in [MAX'',\ MAX'[$ `then`

```
    class ⟵ AF_x2
else if N_R ∈ [MIN, MAX''] then
    class ⟵ AF_x2
```

Again let us take an example to explain the log-based mapping schemes. Suppose that the minimum number of receivers among all the flows in the AF_x queue is 2, which implies $MIN = 2$ and the maximum number of receivers in the same AF_x queue is 10, i.e. $MAX = 10$. When applying the *LOG top mapping* scheme to this configuration, the new DSCP value of the incoming multicast packet having a number of members equal to N_R will be determined as follows:

- $2 \leq N_R \leq 4$: the packet will be re-marked with the DSCP of AF_{x3} class;
- $4 < N_R \leq 6$: the packet will be re-marked with the DSCP of AF_{x2} class;
- $6 < N_R \leq 10$: the packet will be re-marked with the DSCP of AF_{x1} class.

On the other hand, when applying the *LOG bottom mapping* scheme, the new DSCP value is determined as follows:

- $2 \leq N_R \leq 6$: the packet will be re-marked with the DSCP of AF_{x3} class ;
- $6 < N_R \leq 8$: the packet will be re-marked with the DSCP of AF_{x2} class;
- $8 < N_R \leq 10$: the packet will be re-marked with the DSCP of AF_{x1}class.

The LOG bottom mapping scheme allows large multicast groups to gain more bandwidth, while the LOG top mapping do a slight discrimination between groups having large size given that the space reserved for the AF_{x1} class is larger than that of AF_{x2} and AF_{x3} (see Figure 4.14).

Note that it is possible to make a static re-marking of the DSCP values of the multicast packets according to an economic criterion. If we consider an AF class, noted AF_{xy}, x could be fixed according to an economic criterion and y could be set dynamically according to one of the mapping schemes described above.

4.10.4 Incremental Deployment

We believe that our proposals, concerning the integration of the SBQ scheduler in the DiffServ architecture and the re-marking of the multicast packets, can be easily and incrementally deployed in the Internet given that they do not require to be supported by all the routers belonging to the diffserv-enabled network. Indeed, a router that implements our schemes does not need to exchange messages with other routers to obtain the expected behavior but it simply relies on the local information concerning the number of downstream members for each active multicast session. We propose in the next chapter an extension to the multicast service model to explicitly count the number of members in the intermediate routers and we show that this extension is itself incrementally deployed in the Internet.

4.10.5 Complexity and Scalability Issues

As we have discussed in Section 4.8, SBQ is a simple scheduler that do not need complex processing given that it looks only to the high prefix of the IP destination address in the received packet to detect whether it is an unicast or a multicast packet. The use of SBQ in each queue does not affect the characteristics of DiffServ architecture, originally designed to be simple and scalable comparing to the IntServ architecture [16]. Indeed, our proposal does not require a per-flow state for unicast and the multicast per-flow states are already maintained by the multicast routing module to handle the forwarding of incoming multicast packets.

In addition, our re-marking scheme is not complex (cf. Algorithm **??** and **??**) and it does not need to store a large amount of data. We only need to include the number of receivers downstream to each outgoing interface in corresponding entry in the multicast routing table, something which adds only a very slight fraction of complexity both in terms of size and processing.

Moreover, our proposal is scalable in the sense that the re-marking relies only on the number of members of the active flows that have at least one packet in the queue and not on the number of all active sessions which are declared in the multicast routing table. Thus, the maximum number of states is equal to the size of the buffer in packets (for example in CISCO routers, the buffer size is configured by default to 64). Furthermore, the buffer management mechanism (MFQ) maintains a per-active-flow state table, and it also does not store all active sessions.

4.10.6 Simulations and Results

We integrated SBQ scheduler that we have described in the first part of this chapter in the DiffServ code available in the ns-2 network simulator [84]. We also implemented the re-marking schemes that we have proposed above. The number of downstream members is provided by our extension of the multicast service model that we will describe in the next chapter and which has also been implemented in the ns-2 simulator.

The objective of the simulation experiments described in this section is to evaluate the performance of the two aspects of our architecture which we have described earlier namely: the bandwidth sharing between unicast and multicast flows and the multicast re-marking schemes.

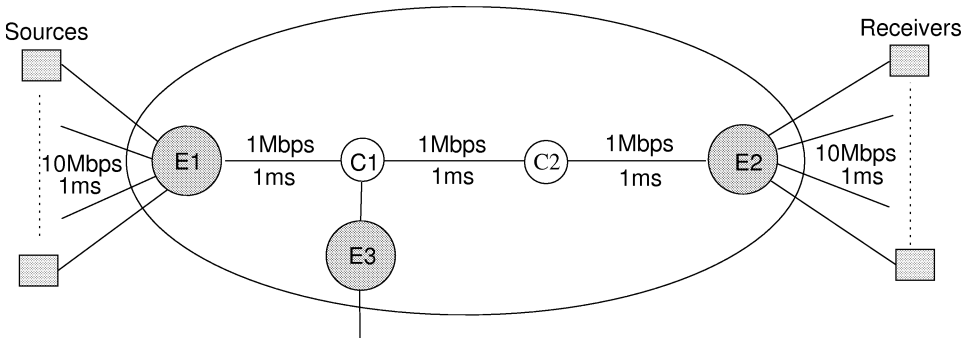


Figure 4.15: The network topology used to evaluate the performance of SBQ in DiffServ networks

We use the network topology shown in Figure 4.15. We analyze the bandwidth sharing in the bottleneck link between the two diffserv core routers: C_1 and C_2.

The simulation lasts 50 seconds and all sources send packets during the whole simulation. Other parameters are set as follows:

- 12 multicast flows:
  - 2 flows (f_1, f_2): each one has 6 receivers;
  - 2 flows (f_3, f_4): each one has 5 receivers;
  - 2 flows (f_5, f_6): each one has 4 receivers;
  - 2 flows (f_7, f_8): each one has 3 receivers;
  - 2 flows (f_9, f_{10}): each one has 2 receivers;
  - 2 flows (f_{11}, f_{12}): each one has 1 receivers;
- 12 unicast sources: flows f_{13} to f_{24}.
- the packet size value is 1000 bytes for each flows in order to better understand the result plots.

The (a) figures (4.16 to 4.21), showing the average bandwidth allocated for an unicast and a multicast flow, allow us to evaluate the performance of the bandwidth sharing between unicast and multicast flows which represents the first aspect of our proposal. And, the (b) figures (4.16 to 4.21), showing the aggregated bandwidth allocated for each drop precedence belonging to the same CoS (Class of Service), permit to evaluate the proposed multicast re-marking schemes constituting the second aspect of our proposal. We summarize in Table **??** the repartition of flows in diffserv classes for each re-marking scheme: Lin mapping, Log-top mapping, and Log-bottom mapping scheme.The multicast flows belonging to the same CoS are partitioned into three drop precedences.

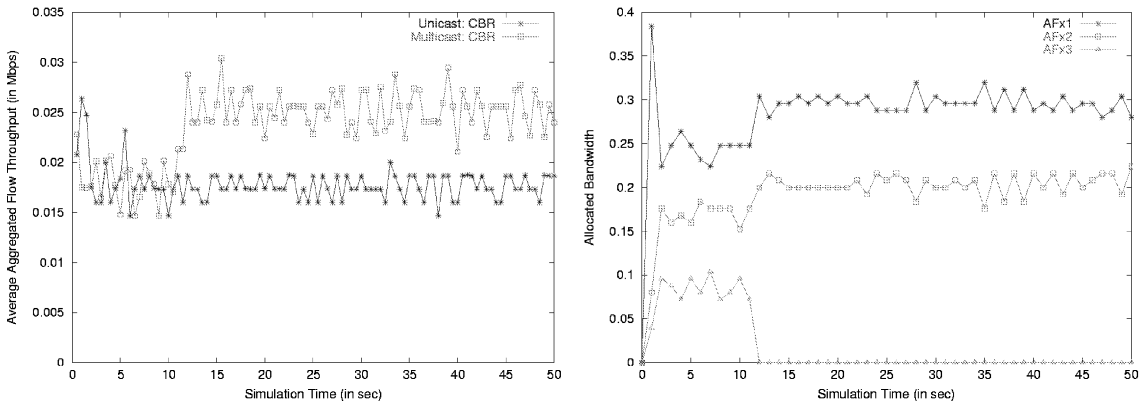
	LIN mapping	**Log top mapping**	**Log bottom mapping**
$\mathbf{AF}_{x1}$	$f_1,\ f_2,\ f_3,\ f_4$	$f_1,\ f_2,\ f_3,\ f_4,\ f_5,\ f_6$	$f_1,\ f_2,\ f_3,\ f_4$
$\mathbf{AF}_{x2}$	$f_5,\ f_6,\ f_7,\ f_8$	$f_7,\ f_8$	$f_5,\ f_6$
$\mathbf{AF}_{x3}$	$f_9,\ f_{10},\ f_{11},\ f_{12}$	$f_9,\ f_{10},\ f_{11},\ f_{12}$	$f_7,\ f_8,\ f_9,\ f_{10},\ f_{11},\ f_{12}$

Table 4.2: The repartition of the flows in DiffServ AF classes of service for each multicast re-marking method

The objective of the first set of simulations is to evaluate the LIN mapping scheme for re-marking multicast packets. When applying this mapping method to the twelve multicast flows listed above, we obtain four flows in the AF_{x1} class, four flows in the AF_{x2} class, and four flows in the AF_{x3} class. As the bottleneck link is congested, AF_{x3} class is the class from which packets will be dropped at first. Figure 4.16(a) and Figure 4.17(a) show that the bandwidth is approximatively equitably shared between unicast and multicast flows until 12 sec and 8 sec, respectively: the same bandwidth is allocated to unicast and multicast flow, for unicast CBR sources and TCP unicast sources, respectively. SBQ shares the available bandwidth between unicast and multicast flows, so that each one of them gain 500 Kbps from the 1 Mbps bottleneck link capacity.

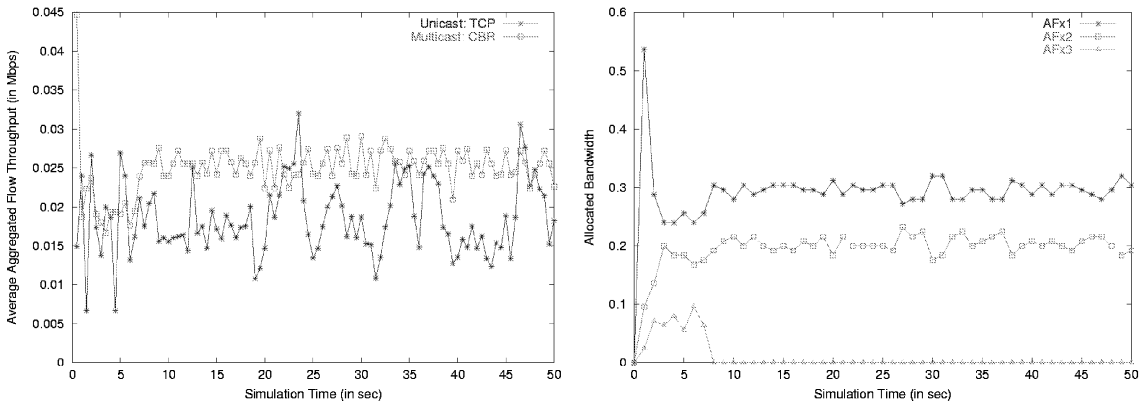
As shown in Figure 4.16(b), for unicast CBR flows, and Figure 4.17(b), when using TCP unicast flows, after a short time (about 12 sec for CBR unicast flows and 8 sec for multicast

flows), the class AF_{x3} does not have any bandwidth because the DiffServ router drops packets with the higher drop precedence (lower priority) to preserve bandwidth to lower precedence (higher priority) classes. We can also see that among the 500 Kbps allocated to the multicast flows, at the end of the simulation the four AF_{x1} flows obtain about 300 Kbps, and the four AF_{x2} flows are sharing about 200 Kbps. The mean allocated bandwidth per flow for the AF_{x1} class is higher than that for AF_{x2}, which corresponds to the expected results in the sense that we aim to give more bandwidth to flows with higher number of members.



(a) (b)

Figure 4.16: Using LIN mapping and unicast CBR sources



(a) (b)

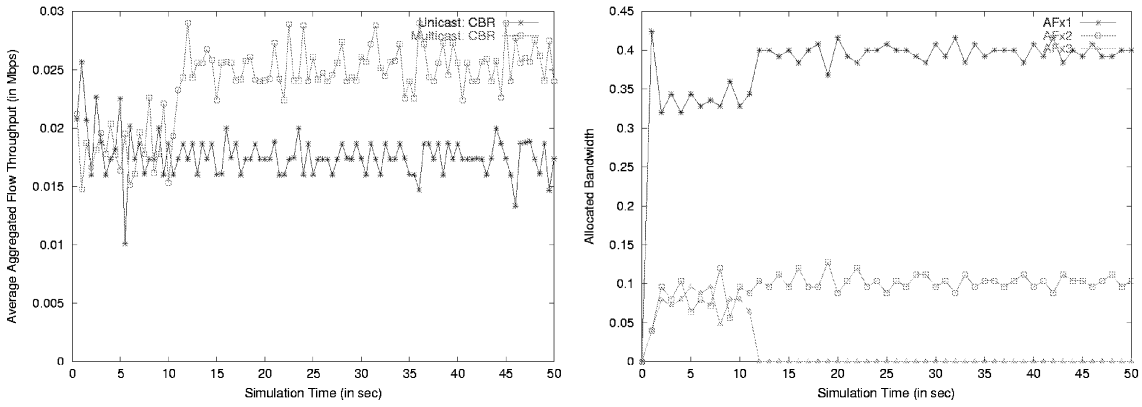
Figure 4.17: Using LIN mapping and unicast TCP sources

We use exactly the same simulation scenario as before to evaluate the performance of the LOG top re-marking scheme. Figure 4.18(a) and Figure 4.19(a) also show that bandwidth is

fairly shared between unicast and multicast flows given that the per flow aggregated allocated average rate is the same.

Now, if we apply the re-marking LOG top scheme to the considered flows, we obtain that there are six flows belonging to the AF_{x1} class, two flows belonging to the AF_{x2} class, and four flows belonging to the AF_{x2} class. Like the previous simulation experiment, the congestion is too important to allow AF_{x3} class to gain bandwidth. The bandwidth allocated to flows belonging to AF_{x1} and AF_{x2} classes is shared among eight flows. This explains the results shown in figures 4.18(b) and 4.19(b) where the bandwidth allocated to one multicast flow is higher than that of unicast flow since the AF_{x3} class stops transmitting data at about ten seconds.

There we can see that the six AF_{x1} flows are sharing about 400 $kbps$, and the two AF_{x2} groups are sharing about 100Kbps. The allocated bandwidth per flow for the AF_{x1} class is higher than that of flows of the AF_{x2} class which also corresponds to the expected result.

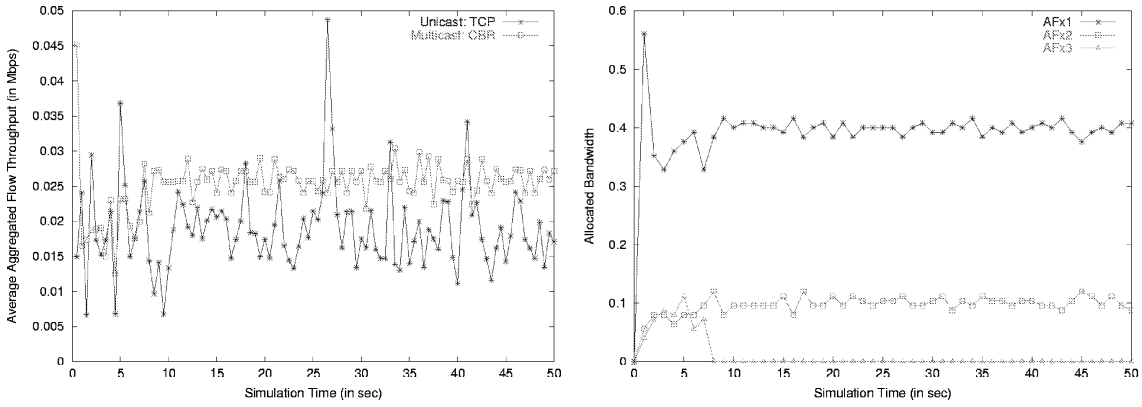


(a) (b)

Figure 4.18: Using LOG top mapping and unicast CBR sources

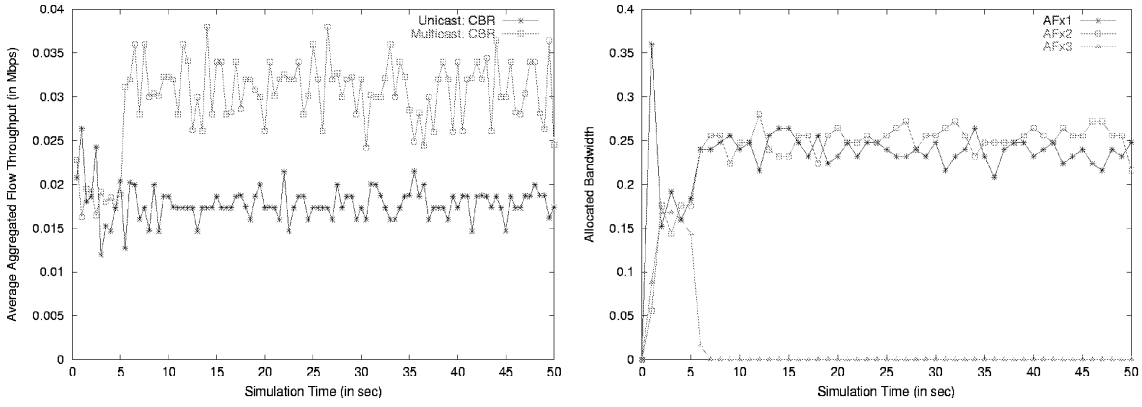
We focus now on evaluating the performance of the LOG bottom mapping re-marking scheme. There are four flows belonging to the AF_{x1} class, two flows belonging to the AF_{x2} class, and six flows belonging to the AF_{x3} class. Figure 4.20(a) and Figure 4.21(a) show that after 6 sec the AF_{x1} class shares 250 $kbps$ among the 500 $kbps$ allocated to the multicast service and the AF_{x2} class shares 250 $kbps$. We can also observe the average allocated bandwidth in Figure 4.20(b) and Figure 4.21(b). In fact, the average allocated bandwidth of multicast flows is higher than the average unicast one. Indeed, AF_{x3} class does not have any allocated bandwidth, so there are only four flows to share the multicast bandwidth.

These re-marking schemes can really preserve bandwidth for groups according to the local multicast state. However, the three different re-marking functions can not be used for the same purposes because their characteristics are too much different. A future work could establish heuristics to dynamically change the re-marking scheme in order to improve the satisfaction rate of clients and to encourage the use of multicast service.



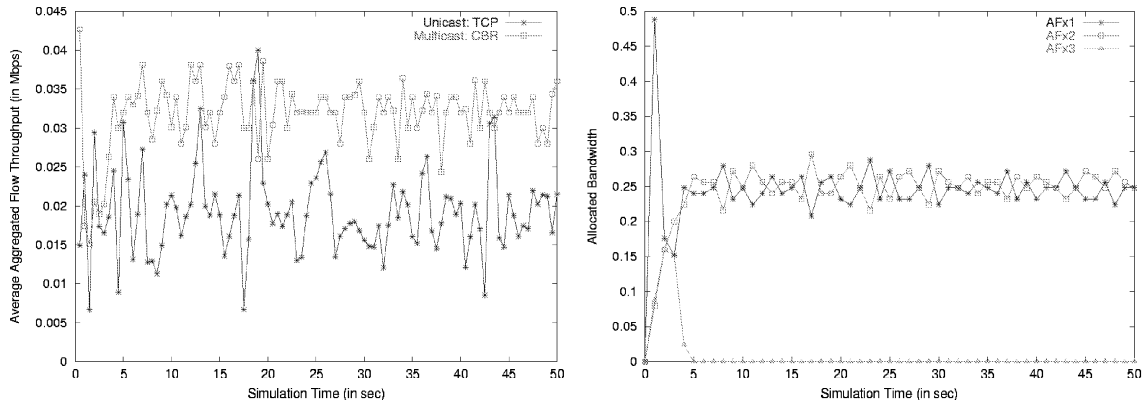
(a) (b)

Figure 4.19: Using LOG top mapping and unicast TCP sources



(a) (b)

Figure 4.20: Using LOG bottom mapping and unicast CBR sources



(a) (b)

Figure 4.21: Using LOG bottom mapping and unicast TCP sources

4.11 Chapter Summary

In the first part of this chapter, we have presented a WRR-like scheduler for bandwidth sharing between unicast and multicast flows. The scheduler uses two queues: one for unicast and the other one for multicast. We used a simplistic and efficient dynamic configuration of the WRR scheduler to achieve the expected sharing based on a new fairness notion called *the inter-service fairness*.

The buffer management mechanism used in the multicast queue was MFQ, a new scheme that we have proposed in the previous chapter and which provides the expected multicast bandwidth sharing between multicast flows using a single FIFO queue.

To validate the SBQ scheme, we simulated a very heterogeneous environment with different types of sources, starting times, sending rates, delays, and packets size. We demonstrated that SBQ achieves the expected results in the sense that the bandwidth is shared fairly between unicast and multicast flows according to the inter-service fairness criterion.

In the second part of this chapter we have proposed a modification to the DiffServ architecture in order to enhance the network resource sharing between unicast and multicast flows, in each DiffServ class. Our proposal is based on replacing the single queue of each class of service by the simple SBQ scheduler.

In addition, we have proposed and evaluated three methods (the LIN mapping, the LOG top mapping, and the LOG bottom mapping) based on the number of downstream members on each outgoing interface to re-mark multicast packets in order to better share network resource among competing multicast flows. Each method uses a specific way to map the number of members of a multicast group to a new a drop precedence, and therefore to a new DSCP value.

We believe that our proposals may help in better deployment of multicast services in best-effort networks as well as DiffServ-enabled networks. Our contribution will encourage the ISPs to deploy multicast in their networks because they could be able to deploy their multicast pricing models which are based on the number of members. Indeed, the re-marking concept

that we have proposed could be coupled with a pricing strategy to select the appropriate method to be used.

Note that, the current IP multicast facility provides no indication of the group size, making it infeasible for an ISP to charge based on this value. In conventional media, a ten million subscriber base is much more valuable than a ten thousand one, and similar economics arise in the Internet. However, IP multicast does not currently help the ISP to distinguish between even very broad ranges of group size. We will investigate the issue of counting the number of members in the next chapter.

Chapter 5

Counting Group Members in Multicast Communications

5.1 Introduction

As we have pointed out earlier, the knowledge of the number of downstream receivers in senders and intermediate routers for each active multicast session can help to resolve some of the issues related to the multicast deployment in the terrestrial networks and it will be then a main key of the widespread deployment of IP multicast. Indeed, it will be possible to solve the feedback implosion problem that occurs when we deal with NACKs messages or IGMP [34] in some kind of physical link support such as satellite link. In addition, the ISPs will be able to establish a valid business model for multicast that take into account the number of group members [65]. In some other protocols such as RTP [107] and SRM [56], the information of the number of group members can be used for feedback suppression in order to minimize the number of reports sent by the receivers.

In the standard multicast service model [34], neither the source nor the intermediate routers can know the number of downstream receivers. Previous work on estimation of group size using probing techniques and analytic models includes [6, 17, 61, 77, 92, 93]. These proposals have three main drawbacks. First, the analytic model used is always based on some tuning parameters whose optimal values largely depend on the network and members conditions. Second, they allow only the source to estimate the group size but not intermediate routers which may be useful for supporting some mechanisms such as bandwidth sharing strategies [75] and multicast pricing policies [65, 66]. Third, these methods are not scalable because of the use of periodic signaling messages from members to senders even when there is no change on the group membership.

The work described in this chapter is a complementary component of our proposals for bandwidth sharing between unicast and multicast flows that we have described in the two previous chapters. Indeed, these proposals use the membership information to decide how to handle the network resource sharing among competing flows. In particular, in Chapter 3, we have described an active queue management mechanism that shares the available bandwidth between competing multicast flows based on an inter-multicast fairness function that depends on the number of members in each intermediate router. At the end of Chapter 4, we have proposed three re-marking schemes of the multicast replicated packets in a DiffServ and multicast-enabled router that takes into account the number of downstream members in each

outgoing interface. Note also that while this work was motivated by our previous described schemes, the service of counting the number of members is useful for many applications that we review in Section 5.6.

We are investigating an alternate approach based on explicit counting rather than probing techniques. The aim of our proposal is to count the number of downstream receivers[1], for all active multicast sessions, in every router belonging to the multicast delivery tree.

The main idea behind our scheme is to extend the used multicast routing protocol to carry the number of members for a multicast group in addition to other routing information. Each intermediate router consolidates the number that it receives from its children and passes it up the multicast tree until the root of the multicast tree receives the total number of members in the corresponding multicast group.

If each host uses IGMPv3 [22] or MLDv2 [117], no protocol change is required, since it sends join and leave messages explicitly to its Designated Router (DR). From this basis, our protocol allows every DR to know the number of directly-attached hosts which are members of each active multicast session. In addition, it allows senders and intermediate routers to count the number of downstream members in each outgoing interface based on a hop-by-hop approach. Our proposal integrates a soft-state mechanism in order to correctly count and update the membership information by overcoming the problem of the loss of count updating messages. A router sends only the difference between the previous number of downstream members and the new one to its upstream router.

We describe the necessary extensions to existing protocols for implementing the scheme. Although the scheme itself does not depend on the multicast routing protocol, we consider the use of PIM-SM intra-domain multicast routing protocol which is probably the most widely used multicast routing protocol today.

Another advantage of our extension is its ability to count the multicast members in each domain. Indeed, by using MSDP protocol [86], the RPs of each domain can exchange the number of receivers in their domain using specific MSDP messages. Each ISP multicast border router that belongs to the multicast delivery tree is able to know the number of members of the sessions having the sender in another ISP and the number of subscribers in other ISPs which are members of local multicast sessions.

The body of this chapter is organized as follows. We give a full description of the host and the router sides of our protocol in Section 5.2. Section 5.3 explores some implementation and deployment issues. We describe in Section 5.4 an example that illustrates the operations of our proposal in a simple network topology. The results of the performance evaluation of our protocol are detailed in Section 5.5. We enumerate the research works related to our work in Section 5.6. Section 5.7 concludes this chapter.

5.2 Extension Description

In this section, we detail the host and the router sides of our proposal. The host side may be included in IGMPv3 or MLDv2 protocols, while the router side concerns multicast routing protocols such as PIM-SM [40].

[1]Throughout this chapter, we use the terms "receivers" and "members" of a multicast session to refer to the end-hosts which joined this session. In a Local Area Network (LAN), we use the terms "local members" and "local receivers".

5.2.1 Host Side

Let us consider a DR connected to a LAN where there are several hosts, and assume that at least one of them wants to receive data from a multicast sender. Before starting receiving the data, this end-host sends an explicit IGMPv3/MLDv2 report to its DR. The DR sends a join message either towards the RP (Rendezvous Point) in case of using PIM-SM and the session is identified by $(*, G)$ or directly toward the source in case of joining a (S, G) session. This message is forwarded by intermediate routers until it reaches a router which is already connected to the multicast delivery tree. This end-host is then considered as a new member of the multicast session.

In this case, our protocol requires that the DR creates a new entry in a new specific table called **Multicast Counting Table** (MCT) in which it keeps and updates the number of members of the requested sessions. Based on IPv4 environment, for each active multicast session, which may be either a (*,G) or a (S,G) entry, the DR maintains the number of local members and updates it when receiving IGMPv3/MLDv2 join or leave messages[2] [22]. Given that IGMPv3 protocol uses a Robustness Variable which specifies the number of duplicated join or leave messages that should be sent by an end-host to be sure that the DR receives the message, each DR should also maintain the end-hosts addresses in order to recognize the exact number of local receivers.

When the number of local members for an active entry changes, the DR sends a message called "counting message" to the upstream router to inform the number of new members that joined or left the session. The message contains only the number of the differences between the new and the old counts and is forwarded hop-by-hop toward the sender's DR.

It makes sense that when the sender's DR receives a count update message for a specific multicast session, it may announce to the sender(s) in its local network the up-to-date group size. This may be done also upon senders demand, which means, when a sender needs to get the current number of members, it sends a specific request to its DR. To this end, we need to introduce a new IGMPv3/MLDv2 message type or a new MSNIP [54] message.

For completeness, we show in Figure 5.1 the flowchart of the algorithm operations at the DR of the receivers.

5.2.2 Router Side

When an intermediate router receives a counting message from an outgoing interface, before forwarding this packet to the upstream router, each router updates the number of receivers by taking into account other counting messages received from other interfaces. The result will be sent to its upstream router with a negative count value if the number decreases and a positive count value, otherwise. If the calculated number is equal to 0, the router does not send any counting message to the upstream router. Using simple formula, when the number of downstream members of a multicast session (S,G) at an intermediate router is $n(t_1)$ at time t_1 and $n(t_2)$ at time t_2 where $n(t_1) \neq n(t_2)$, the counting message which would be sent by this router to the upstream router at time t_2 is then $(S,\ G,\ sign(n(t_2) - n(t_1)),\ n(t_2) - n(t_1))$[3].

When a router receives a prune message from a downstream router, it simply deletes the router's entry and sends a negative counting message to the upstream router including the

[2]For both protocols there is no leave message, but in this chapter we use the term, "leave" which means IGMPV3 CHANGE-TO-INCLUDE with NULL source list.

[3]The function $sign(x)$ return "+" if $x > 0$ and "−" otherwise.

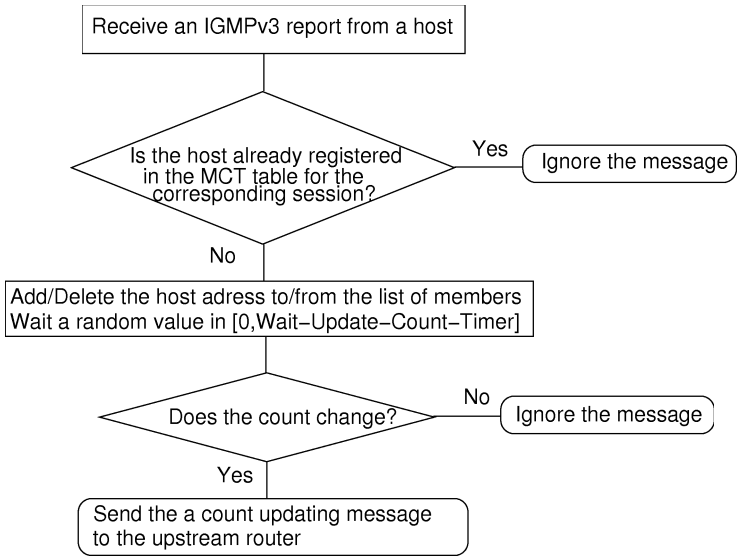


Figure 5.1: The counting algorithm flowchart at the Designated Router of receivers

total number of members. For example, when the current entry of the outgoing interface leading to the pruning router is $(S,\ G,\ n)$, the router sends a message $(S,\ G, -,\ n)$ toward the source when receiving a prune message from this interface.

If an upstream neighbor router is changed because of routing path (MRIB) change, the intermediate router sends new $(S\ , G)$ or $(*, G)$ join message to the new upstream router. In this case, a counting message is also sent to the same router with specifying total number of senders of the join. Old upstream router just clears the number of receivers since it receives a $(S\ , G)$ or $(*, G)$ prune message.

In order to reduce the overhead of control messages, each intermediate router should wait for a random value, which must be less than **[Wait-Update-Count-Timer]** seconds[4], before sending a counting message to the upstream router because in this period it may receive other updating messages from other routers and it aggregates them in a single counting message.

Given that the members of a multicast session may belong to different domains, the RPs within different domains can exchange the total number of members in their domains using the peer-to-peer connection established by MSDP protocol [86].

For completeness, we show in Figure 5.2 the flowchart of our scheme at intermediate router in the multicast delivery tree.

5.2.3 Routing-Depending Configurations

Some functionalities of our protocol may depend on the type of the intra-domain multicast routing protocol used: sparse-mode or dense-mode. We discuss in this section some specific configurations of our scheme for both types of protocols.

[4]The "optimal" value of this timer is out of the scope of this work

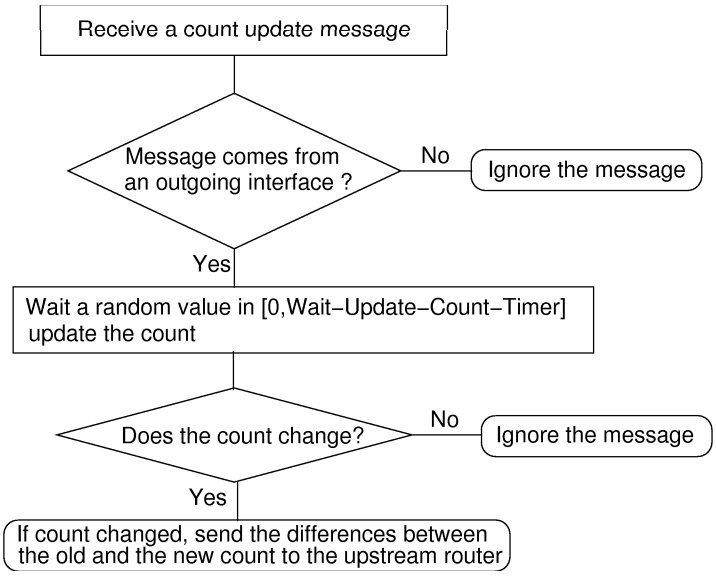


Figure 5.2: The counting algorithm flowchart at an intermediate router in the multicast delivery tree

5.2.3.1 The case of Sparse-Mode Protocols

RP-based-tree multicast protocols such as PIM-SM [40] have receiver-initiated multicast spanning trees, where a router becomes involved in a branch of a multicast distribution tree only when one of the hosts on its subnetwork requests membership by issuing a join message. There may be one or more central core routers that receive join and leave messages, and that pass received multicast packets downstream through the tree.

In PIM-SM, when the RP receives a counting message for an (∗, G) entry, it sends a new counting message toward all senders DRs for group G. At this time, the number of receivers in the new counting message will be copied from the original counting message for (∗, G) entry and inputted to each sender's entry.

5.2.3.2 The case of Dense-Mode Protocols

The concept behind our proposal is valid for dense-mode protocols such as DVMRP [118] or PIM-DM [1] in the sense that it is designed to be independent of the intra-domain multicast routing protocol. In dense mode protocols, there is no rendez-vous point given that each source sending to the multicast group has each own delivery tree.

Source-based tree multicast protocols are data-driven or source-initiated. Construction of the multicast spanning tree begins top-down from the source outward as it transmits information, and data on the state of the tree is flooded to all routers.

For dense-mode protocols, there is no specific configuration to add to the general configuration of our proposal that we have described above.

5.2.4 Scalability Considerations

The scalability of our proposed extension is related to the scalability of the multicast routing protocol itself. As we will show below, the format of the count update messages is similar to that of soft-state based multicast routing protocols such as PIM-SM and their period and timers are comparable.

In a local network with a large number of connected end-hosts such as a satellite broadcast network where the number of satellite terminals may reach several hundreds, many join/leave IGMPv3 reports may arrive to the DR in a short period. In this kind of environment, our protocol maintains the scalability criterion. Indeed, as described above each router waits a random value less than a fixed threshold (Wait-Update-Count-Timer msec) before sending the count message to its upstream router. This method allows us to avoid that the router generates a high load of counting messages to its upstream router. In fact, this method can be also useful even in a small local networks where the connected hosts join and leave the multicast groups frequently.

5.2.5 Packet Loss

In this section, we examine the issue of the lost of count update messages. We introduce a "loose" scheme to avoid the problem of the receivers synchronization and to have always a computed count which is as close as possible to the exact count. We believe that having an "exact" number at every time is not very critical, but at least proposing and then describing a mechanism to approximate the exact number of group members. To this end, we include in our extension two key functionalities:

- the first one consists on using a soft-state mechanism at the routers to update the count value of each outgoing interface. Each router periodically send copies of each up-to-date count update message. This period is called **Send-Update-Count-Period.** This functionality needs to add a sequence number to the count update message header in order to allow the upstream router to distinguish between periodic count update messages (having the same sequence number) and new update messages. This requires that the intermediate routers keep the sequence number of the update messages in addition to the number of members downstream to each outgoing interface.

- the second one consists on periodically sending the total number of members by each router to its upstream router. This period is called **Send-Total-Count-Period**.

Both periods should be configured in advance depending on the expected estimation accuracy and link characteristics. Note also that the **Send-Total-Count-Period** value should of course be higher than the **Send-Update-Count-Period** (for example 10 times higher).

Note that, to solve the problem of packet lost while keeping the same packet format as PIM-SM, it is possible to use only the second functionality but with a shorter period of periodically sending the total number of members.

5.2.6 Inter-domain Counting

The members of a multicast session can be distributed in different domains belonging to distinct ISPs. It may be useful to the multicast senders to know the number of members in each domain. This information may also be useful for the ISPs to which is connected the

sender in order to implement some inter-domain multicast billing and charging protocols or to establish a "multicast connectivity contract" with other ISPs based on the number of members reached through each domain to allow and to control the forwarding of multicast packets to them.

The Multicast Source Discovery Protocol, MSDP [86][5], describes a mechanism to connect multiple PIM-SM domains together. Each PIM-SM domain uses its own independent RP(s) and does not have to depend on RPs in other domains.

A border router, that belongs to a multicast delivery tree of a multicast session (S, G) where the source S belongs to the same domain, is able to know the number of subscribers in neighbor domains which are members to this session. On other hand, if the source S belongs to another domain, this router is able to count the number of local subscribers which are members to the session (S, G).

5.2.7 Incremental Deployment

In this section, we examine how we can deploy our protocol in different configuration scenarios. To this end we consider all possible structures that the multicast delivery tree may have.

We take as example the case when the routers implement IGMPv2 [53] as an example of DRs that do not support the counting and standard PIM-SM routers that do not implement the counting as an example of routers that do not support the counting.

There are two trivial scenarios:

1. *all routers and DRs support the counting*: In this case our protocol is expected to obtain always an estimation of the number of members very close to the exact number.

2. *all routers and DRs do not support the counting*: In this case, of course our extension can not be supported.

In the following, we focus on other cases where our scheme may not provide a good estimation of the number of members given that some of the multicast delivery tree elements do not support the counting service. These cases are:

1. *all routers support the counting but some of DRs do not support it*: We mean by a DR that does not support the counting that it uses IGMP version 2 but its PIM-SM module supports the counting. The problem here is that the DRs can not inform upstream routers about the number of local members although they are able to count them.

2. *some of routers do not support the counting and all the DRs support it*: The problem in this scenario is that when an intermediate router does not receive an updating message from a downstream router and this interface belong to the list of outgoing (`oif`) interfaces, it considers that there is at least one member downstream to this interface.

3. *some of routers and DRs do not support the counting*: when a router does not support the counting, it may implement the standard version of PIM-SM. Even if its downstream routers sends to it the count update messages, it can not interpret these messages and so it is not able to inform its upstream router about the up-to-date count.

[5]Note that MSDP may be used with protocols other than PIM-SM, but such usage was not specified in [86].

We describe a simple method to allow the deployment of our protocol in the three above scenarios where not all the routers support the counting extension.

The idea consists in sending the count message not to the upstream router but toward the RP when it concerns a (S, G) session or toward the source when it concerns a (S, G) session until it reaches a router that supports the counting. This method allows each router to inform the first upstream router towards the RP or toward the source that supports the counting about the number of members and so to pass through the intermediate routers that do not support the counting service.

The number of routers that do not support the counting service between two routers which implement it could be computed based on the TTL (Time To Leave) values. In fact, the router sending the count message sets the TTL in the IP header to the same value used in Join messages and keeps this value in the count message header. The router that receives the count update message has only to compute the difference between the current value of the TTL and the initial value to know the exact number of routers that do not support the counting between both routers. The number of routers that do not support the counting service which are located between two routers that support this service could be used to estimate the number of members that may exist between these two routers[6]. Moreover, this information could also be used to discover the multicast delivery tree.

When all routers in the network support the counting, there is no need to use such scheme. It is so possible, in this case, to send the count message to the upstream router with a TTL equal to 1.

5.3 Implementation Issues

Regularly a count update message is sent only if there is a change on the number of end hosts. To count the number of members, in the host side of our protocol, an DR does not need to send a request to its directly connected hosts because it can do that by simply supervising the hosts reports. It should just create and update the new table shown in Figure 5.3. For each active multicast session, every DR maintains the addresses of end-hosts which are members of that session. Therefore, the number of members is simply equal to the number of these hosts.

Multicast Sessions	(S1,G1)	(S2,G2)	(*,G3)	...	(Sn,Gn)
Hosts addresses	H1, H2	H2	H3, H4	...	H1, H3, H4

Figure 5.3: The multicast counting table (MCT) at the DR

In addition, hosts also do not need to send any specific message. Therefore, there is no modification in IGMPv3/MLDv2 receivers implementation.

The multicast source will be able to get the total number of downstream members by sending an IGMPv3/MLD2 or a MSNIP request to its DR. When receiving this request, the DR sends a report to the source including the current number of members for the requested session.

[6]The method to use to estimate this number of members is left for future work.

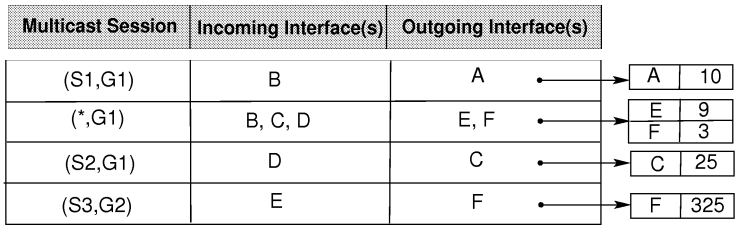


Figure 5.4: The new format of the multicast information table

The router side of our protocol aims to allow intermediate routers to keep the number of downstream members. To this end, we propose to extend the multicast information table of each router by adding a new specific field for each outgoing interface. In Figure 5.4, we show the new multicast information table format. For each multicast session and for each outgoing interface, the router maintains the number of downstream members.

The router side of our protocol can be efficiently integrated in PIM-SM [40] by adding only one new message type. Indeed, currently PIM-SM uses eight message types from sixteen available values. We propose, for example, to use the message type number 9 for the counting messages. The counting message has the format of Figure 5.5. It includes the following fields:

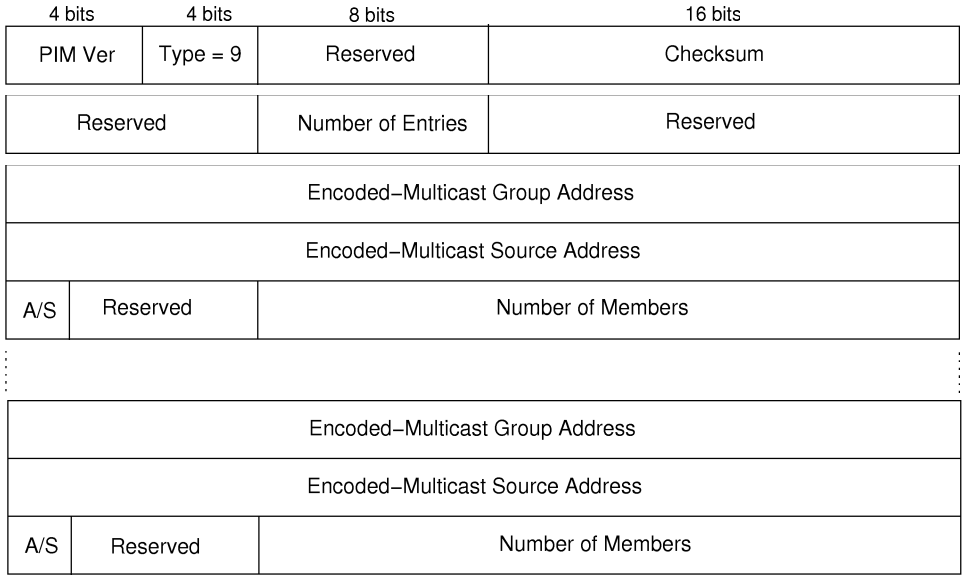


Figure 5.5: The counting message format

- Number of entries (8 bits): is the number of entries announced in the message. An entry is a couple of source and group addresses. In case of using ASM model (group address out of range of the SSM reserved range), the source address can be set to INADDR_ANY (all 0) to refer to a $(*,\ G)$ entry.
- Add/Subtract field (2 bits): it specifies whether the router receiving this message should add or subtract the announced number of members. A value of "1" shows "Add", a value

of "2" shows "Subtract", "0" shows "Total number", and "3" may be used for another case, e.g, to indicate an "Error"

- Number of receivers (24 bits): is the number of members to add or to subtract. This has a maximum value 2^{24}. So, if the number of receivers goes over this value, the protocol may return an error.

The counting message is sent by each intermediate router to its upstream router.

As we have outlined above, a source may request at any moment the number of members from each DR. To this end, we can extend the IGMP/MLD protocol with a new message. One possible format of this message in given in Figure 5.6.

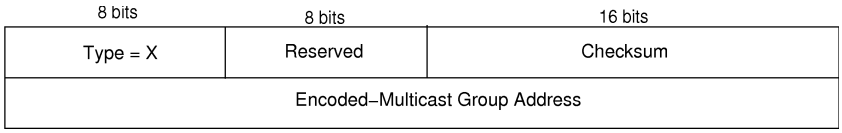


Figure 5.6: The source's request message format

The fields of this message are:

- Group address: the multicast group address
- Source address: the source unicast address

When the source's DR receives a count request from one of its local LANs, it sends a new message to the source containing the requested size of the multicast group if it exists in the MCT table, and an error code, otherwise. This message could have the format of Figure 5.7.

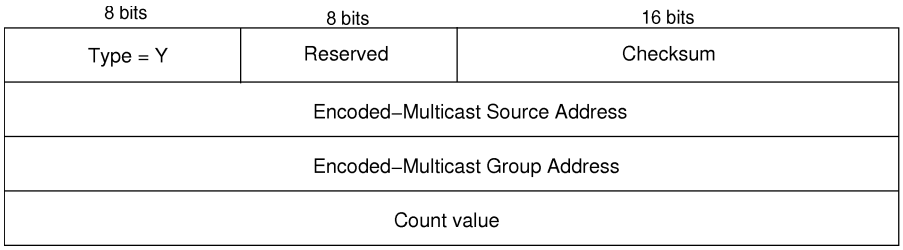


Figure 5.7: The format of the source's DR response message when the source requests the group size value

5.4 Illustration Example

We consider the topology of Figure 5.8, where there is a multicast source S_1 sending to a multicast group G_1. Without loss of generality, we assume the use of IGMPv3 as the group management protocol and PIM-SM as the multicast routing protocol.

We assume that initially there is only one receiver R_1 who joined the source and started receiving the data. When the Designated Router of R_1 (router Rt_1) receives an IGMP v3

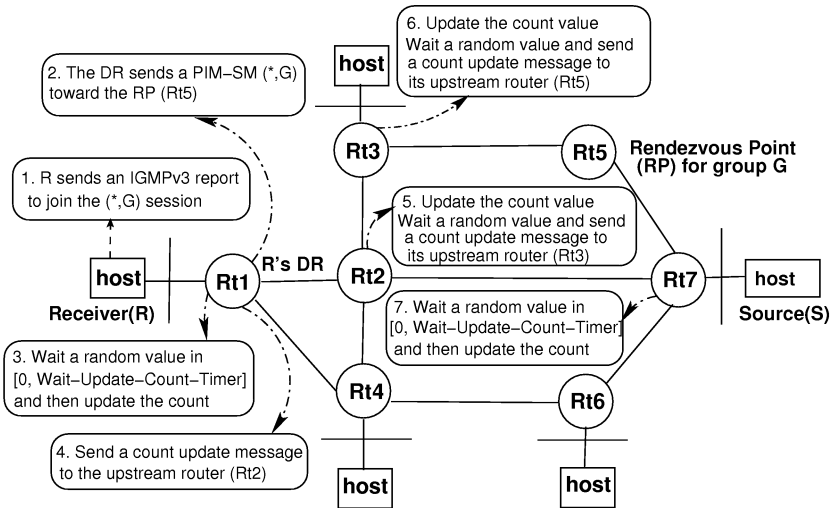


Figure 5.8: An illustration example of the proposed mechanism

Join message, it creates a new entry in the Members Counting Table (MCT) indexed by the requested session identifier (*,G) and set the number of members to 1. After that, Rt_1 sends a PIM-SM join message to the upstream router (router Rt_2). Then, it waits for a random value in [0, Wait-Count-Update-Timer] and sends a count update message to its upstream router (router Rt_2). On the reception of this message, this router sends in its turn a count update message to its upstream router (router Rt_3) after waiting a random value and so on until a count update message reaches the source's DR (router Rt_7). At this time, each router in the multicast delivery tree (composed with routers Rt_1, Rt_2, Rt_3, Rt_5 and Rt_7) knows the number of downstream members.

After that all the members join the group, the sender's DR gets the total number of members in the multicast group.

When this delivery tree is switched to SPT (Shortest Path Tree), which is triggered by Rt_1, the Rt_1 translates $(*, G)$ entry to (S, G) entry in its MCT table. For Rt_2, it copies the current count value from $(*, G)$ entry to (S, G) entry and removes original $(*, G)$ entry, since the incoming interface will be changed. For Rt_3 and Rt_5, when they receive $(*, G)$ Prune, they just remove the MCT as well as each group entry. For Rt_7, since the outgoing interface will be changed, like Rt_2, it recreates a new entry in the MCT table for the session (S, G).

5.5 Performance Evaluation

We implemented our proposal in the ns-2 simulator [84] and we evaluate its performance using several metrics.

We generated 100 network topologies of 1000 nodes with different connectivities values using the Brite tool [85]. We varied the group size from 5 to 200. For each multicast group, we used an uniform distribution to generate the location of the different receivers in the network and we consider only one sender per group. As recommended in [6], we used a Poisson process to generate the join times and a Zipf distribution for the joining period. Each group was assigned a single Rendez-vous Point (RP) which is randomly selected from a set of

some centrally located nodes by a manner to approximately equilibrate the number of groups served by each RP. For each scenario, we conducted 100 simulation instances and we computed the average performance metric value. The simulation time is set to 100 s.

There were three main objectives for our simulations:

- Quantify the accuracy of the counting scheme,
- Quantify the effect and overhead of the count update messages, and
- Understand how message loss affects the accuracy of our proposal.

In a first experiment, we measure the updating latency which consists on the time that takes the algorithm to inform the sender's DR about the total number of members in the multicast group. In Figure 5.9, we plot the average count updating time at the sender's DR as a function of group size. As we can see the updating latency decreases when the group size increases and it reaches a stable value around 1 ms when the group size is more than 40.

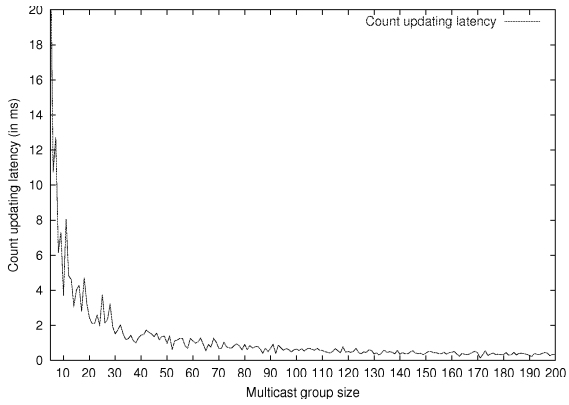


Figure 5.9: Variation of the count updating latency in function of the multicast group size

In a second experiment, we analyze the overhead of our scheme in term of the number of count updating message sent by routers in the multicast delivery tree. In Figure 5.10, we show the number of the count messages sent as a function of the number of group members. We can see that our protocol is expected to maintain a less overhead even for a huge group sizes. Indeed, the number of messages is about 160 for a group size equal to 200. In addition, the overhead of our proposal scale logarithmically with the group size. This is clearly desirable scaling behavior.

In a third experiment, we evaluate the accuracy of our scheme. We compute the estimation error as follows:

$$EstimError = \left| \frac{EstimCount - ExactCount}{ExactCount} \right|,$$

where $EstimCount$ and $ExactCount$ are the estimated count value obtained by our scheme and the exact count value computed theorically, respectively.

In Figure 5.11, we plot the estimation error in function of the group size. As we can see the estimation error increases when the group size increases, however this increase is not linearly. The estimation error is about 1.1 % when the group size is equal to 200, which we believe acceptable for our proposal.

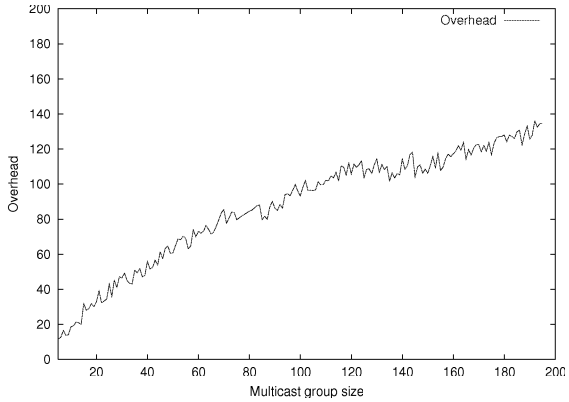


Figure 5.10: The overhead of packets

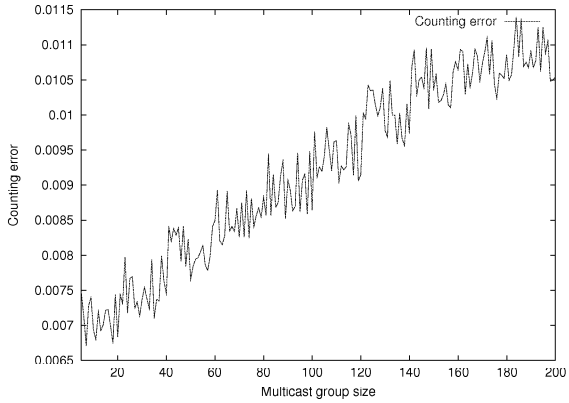


Figure 5.11: The estimation error in function of group size

5.6 Related Work

We identify three main topics related to our work on counting the number of members in multicast communication.

The first topic concerns the group size estimation using probing techniques. Estimating the size of a multicast session can be quite useful to many applications. Bolot, Turletti, and Wakeman [17] used membership information to further estimate the congested receivers as needed in the videoconference system IVS. Future Internet radios and TVs will need to characterize their audience performances and to follow the fluctuations of the audience size.

The second topic is about the accounting and billing in multicast communication. The problem of designing efficient algorithms for sharing the cost of multicasting has recently seen considerable attention [24, 32, 65, 66]. Multicast transmission offers tremendous savings in network bandwidth over unicast transmission for applications that deliver the same content to multiple customers by allowing these customers to "share" the transmission on common access links. However, this sharing of link bandwidth significantly complicates the issue of pricing. Charging all receivers equally is not an adequate solution, since some receivers might be charged more than they would be willing to pay. Each receiver should share the price with other receivers receiving the same data with the same quality. To this end, it is necessary that we know at the sender as well as at each intermediate router the number of downstream members of each multicast session in every outgoing interface.

The third topic, related to our work on counting the number of group members, is the bandwidth sharing and multicast fairness definition. As we have pointed out in the previous two chapters, in [75], W. Biersack et al. have defined three different bandwidth allocation strategies for multicast flows as well as criteria to compare these strategies. They showed that the LogRD policy always leads to the best tradeoff between receiver satisfaction and inter-multicast fairness.

There are other topics to which our proposal can provide an added-value such as layered multicast transmission and multicast congestion control. For example, in layered transmission the sender can use a scheme organizing the data into layers based on the number of members in each layer and assign system/network resource to parallel sessions according to the audience size.

5.7 Chapter Summary

The key contributions of this chapter are: (a) to note that it is possible to count the number of group members based on a hop-by-hop approach instead of probing techniques, (b) to propose an extension to include the service of counting the number of group members in the IP multicast model.

More precisely, we have proposed an extension to the multicast service model[7] which allows senders and intermediate routers to count the number of members downstream to each intermediate router in the multicast delivery tree and this for each outgoing interface. Our protocol has a host side that can be easily added to IGMPv3/MLDv2 and a router side which can be integrated in multicast routing protocols such as PIM-SM.

[7] Our extension can be applied for the primary ASM [33] multicast service model as well as the promising SSM [67] model.

Contrary to probing techniques used in other proposals to estimate the audience size at the sender, our scheme is able to count explicitly this number of group members in each intermediate router at a reasonable cost which is comparable to the cost of the signaling procedures in the multicast protocols themselves.

We demonstrated through simulation that our proposal performed well. Indeed, the overhead of control messages and the group number updating time is reasonable.

We believe that our extension, namely counting the number of members for a multicast group is very useful for several applications such as multicast billing and charging, bandwidth sharing between unicast and multicast flows, multicast congestion control, etc.

Chapter 6

Enhancing Multicast Routing Protocols over GEO Transparent Satellites

6.1 Introduction

Multicasting allows us to send data packet to multiple sites at the same time. The key idea here is the ability to send one message to one or more nodes in a single operation. This provides a tremendous amount of savings in bandwidth when compared to traditional unicast transmission which sends messages to multiple nodes through replication of the message to each node. Besides the performance improvement over unicast transmission, multicast allows the construction of truly distributed applications. Therefore, multicast-based applications and services will play an important role in the future of the Internet as continued multicast deployment encourages their use and development.

However, as we have pointed out in Section 2.3.1, several remaining challenges face the large deployment of multicast in the terrestrial Internet. We have tried in the previous three chapters to contribute to this vast research area and in particular we studied the problem of network resource sharing between unicast and multicast flows. In this part of the dissertation, we focus on the deployment of IP multicast in networks that integrate different transmission media. Indeed, in addition to the evolution of the Internet services, the Internet infrastructure is integrating several types of wired and wireless communication links. Satellite links constitute one of the major components of this infrastructure [101]. In fact, even in the past three decades, satellites have played a pivotal role in global telecommunications. It is anticipated that they will play a complementary role in the so-called information superhighway, a term referring to an infrastructure consisting of networks linking homes, business, government, and institutions to a wide range of interactive multimedia services. Potential applications include teleconferencing, tele-learning, high resolution image transfer, home banking and shopping, video on demand, TV/radio/newspaper/data broadcasting. The nature of these services require the adoption of broadband transmission at the T1 rate and beyond.

Since the Internet protocols, such as routing protocols, have been designed without taking into account the inherent characteristics of the physical support, the harmonious and the efficient integration of GEO satellite links in the Internet requires the study and the adaptation of these protocols. The dynamic unicast routing problem over unidirectional links, and in par-

ticular satellite links, has been investigated in the IETF UDLR working group. In fact, it has proposed the LLTM (Link Layer Tunneling Mechanism) [39] mechanism which performs well by capturing the dependence of the network layer on the bidirectionnality of the communication link. It uses a Dynamic Tunnel Configuration Protocol (DTCP) that provides a means for satellite receivers to dynamically discover the presence of feeds (satellite uplink stations)[1] and to maintain a list of operational tunnel end-points[2]. Feeds periodically announce their tunnel end-point addresses over the unidirectional link. Receivers listen to these announcements and maintain a list of tunnel end-points.

Although the dynamic unicast routing over unidirectional links has been solved, there are several other open issues related to the integration of satellite links in the Internet that remain without solutions or even have not been already addressed such as the scalability of LLTM mechanism, the multicast routing, and the reliable multicast transfer over satellites[3]

This chapter deals with multicast routing protocols over GEO transparent satellites that we have already introduced in Section 2.2.2. We examine flood-and-prune-based protocols such as DVMRP [118], and PIM-DM [1] and from those using explicit join/prune messages to build the multicast delivery tree we focus on PIM-SM [40]. For each multicast routing protocols, we identify its possible undesirable behavior in satellite environment and we develop some suitable tuning methods. For DVMRP and PIM-DM, we describe some configurations where the satellite receivers can receive duplicated packets and we propose a method to overcome this problem. For PIM-SM, we propose a configuration policy that has two main benefits. First, it builds a delivery tree where members receive data from the broadcast satellite downlink either using a RP-rooted tree or a shortest path tree. Second, it minimizes the multicast traffic load in the terrestrial network so that the terrestrial bandwidth could be used by unicast applications or, in general, by applications having requirements (such as the delay) that can not be guaranteed when using the satellite links.

The remainder of this chapter is organized as follows. We study the behavior some multicast routing protocols namely DVMRP, PIM-DM, and PIM-SM in Section 6.2, Section 6.3, and Section 6.4, respectively. A case study of the support of IP multicast over transparent satellites will be detailed in Section 6.5. We conclude this chapter in Section 6.6.

6.2 DVMRP over GEO Satellite Networks

The first protocol developed to support multicast routing is called the Distance Vector Multicast Routing Protocol (DVMRP) [118]. It has been widely used on the MBONE.

As we have mentioned in Section 2.1.3, DVMRP uses a different distribution tree for each source and its destination host group. Each distribution tree is a minimum spanning tree from the multicast source as the root of the tree to all the multicast receivers as leaves of the tree. The distribution tree provides a shortest path between the source and each multicast receiver

[1] We use the terms **feed** and **ST Rx/Tx** interchangeably, to mean the satellite stations having the capability to send and receive the data from the satellite links. In the other hand, the terms **ST Rx only** and **receivers** are used to refer to the satellite stations which have not the capability to send packets to the satellite.

[2] The LLTM mechanism uses GRE (Generic Routing Encapsulation) [51] to tunnel packets from the satellite receivers to feeds via the terrestrial tunnels configured using the DTCP protocol.

[3] The issues concerning the reliable multicast data delivery in satellite environment where a packet may be lost in the satellite segment due to the data corruption and not to the congestion and where the multicast source has to determine whether the receivers have correctly received the packets sent are out of the scope of our work on multicast over satellite addressed in this dissertation.

in the group, based on the number of hops in the path, which is the DVMRP metric. A tree is constructed on demand, using a "broadcast and prune" technique, when a source begins to transmit messages to a multicast group.

The approach used by DVMRP is to assume initially that every host on the network is part of the multicast group. The designated router on the source subnet, i.e., the router that has been selected to handle routing for all hosts on its subnet, begins by transmitting a multicast message to all adjacent routers. Each of these routers then selectively forwards the message to downstream routers, until the message is eventually passed to all multicast group members.

The prune part of the protocol eliminates branches of the tree that don't lead to any multicast group members. The Internet Group Management Protocol (IGMP), running between hosts and their immediately neighboring multicast routers, is used to maintain group-membership data in the routers. When a router determines that no hosts beyond it belong to the multicast group, it sends a prune message to its upstream router. Of course, routers must update (source, destination group) state information in their tables to reflect which branches have been pruned from the tree. This process continues until all superfluous branches are eliminated from the tree, resulting in a minimum spanning tree.

Construction of a DVMRP spanning tree in a hybrid satellite-terrestrial network is illustrated in Figure 6.1. A tree is constructed on demand, using a "broadcast and prune" technique, when a source begins to transmit messages to a multicast group. The designated router on the source subnet (R_1 in Figure 6.1), i.e., the router that has been selected to handle routing for all hosts on its subnet, begins by transmitting a multicast message to all adjacent routers.

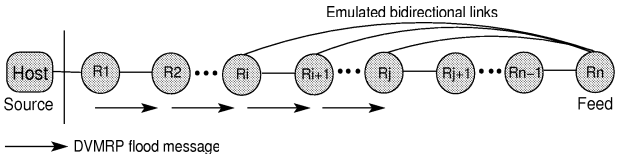


Figure 6.1: DVMRP behavior over hybrid satellite-terrestrial networks. The source sends DVMRP messages towards the feed. The satellite receivers in this path send the message to all their interfaces including the UDLR Tunnel to the feed except that from which the message was received.

Refer to a path from the source that contains satellite feeds or receivers and denote by R_i the first satellite receiver that belongs to this path and R_n the first router after R_i that has the capability to send data to the unidirectional satellite link.

The shortest path from a router R_k to the designated router R_1 is:

- R_k, ..., R_{k-1}, ..., R_2, R_1 if $k \leq j$ where $j = [\frac{i+n}{2} + 1]$
- R_k, R_{k+1}, ..., R_n, R_i, R_{i-1}, ..., R_2, R_1 otherwise.

Therefore, routers from R_1 to R_j forward the message to their downstream routers, since they receive the message from their interface used to send the data toward the source. In particular, router R_i sends the message also to the feed R_n via the UDLR tunnel so that the routers from R_{j+1} to R_n receive the message from R_n. In fact, on the reception of a message from R_j, the router R_{j+1} sends a prune message via its terrestrial interface toward R_j.

The undesirable DVMRP behavior here is that the routers from R_i to R_j who have a satellite-receiving capability will receive **duplicated copies** of multicast packets. One copy from the terrestrial interface toward the source (it is the shortest path) and another copy from the satellite link since the feed should send data to receivers from R_{j+1} to R_n. It is should be noticed that the multicast copy received from the satellite link will not be delivered to the network layer and so it will not disturb the multicast application. However, we argue that it is more efficient to use the satellite copy instead of the terrestrial one especially to the bandwidth-sensitive multicast applications. To this end, we propose to proceed as follows:

- When the satellite link layer of the routers between R_i and R_j detects that multicast packets have been received via the satellite link which belongs to a group to which the router is already a member on the terrestrial interface, it triggers a DVMRP prune message towards R_i.

- Routers between R_i and R_j forward each message received via the satellite interface to the terrestrial interface. By this way each multicast packet received from the satellite link will be considered by the network layer as that it was received from the terrestrial interface.

Modifications should be added only to the LLTM mechanism to execute the two above tasks and we do not need to modify the implementation of DVMRP protocol.

6.3 PIM-DM over GEO Satellite Networks

Protocol Independent Multicast-Dense Mode (PIM-DM) [1] is similar to DVMRP. Both protocols employ Reverse Path Multicasting (RPM) to construct source-rooted distribution trees. The difference between DVMRP and PIM-DM is that in DVMRP, prior to forwarding to a certain interface, DVMRP makes sure that the interface leads to a node that will recognize the local node as a node that is in the shortest path between it and the source (poison-reversed route). PIM-DM accepts additional overhead in order to simplify the forwarding algorithm. Apart from this, the protocol is very similar to DVMRP and thus, all that has been stated for DVMRP applies to PIM-DM also.

The use of PIM-DM over satellite has the same problem as that of DVMRP explained in Section 6.2 and so it has the same solution described above. Besides this problem, the additional overhead added by PIM-DM in a satellite network can have a big impact of the performance of this protocol. We illustrate this problem via the configuration given in Figure 6.2.

According to PIM-DM specification, when the feed receives multicast data from the source via one of its terrestrial interfaces, it automatically forwards the packets received by all other interfaces and in particular to the satellite link even when there is no member which is reached via the satellite. Or, some of the receivers may receive unicast data from the source via one of their terrestrial interfaces. The bandwidth consumed by the useless PIM-DM messages can be high and even have an impact on network resource required by other established connections.

As we will explain further, PIM-SM has a similar problem to that of PIM-DM presented above. We will present the solution to this problem (useless multicast traffic) for the concrete GEO transparent system that we will study in Section 6.5.

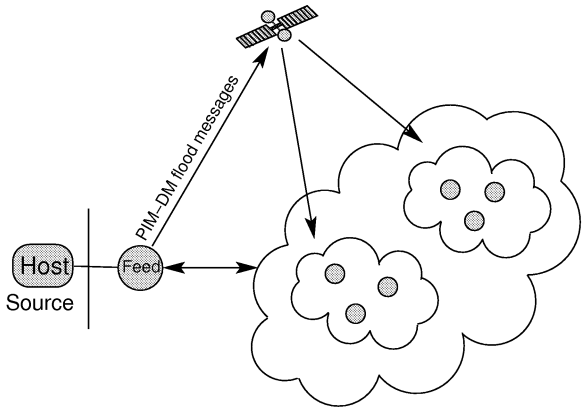


Figure 6.2: The feed forwards the flood message to all spot beams without checking if the receivers use the satellite link to reach the source or not.

6.4 PIM-SM over GEO Satellite Networks

6.4.1 Overview

Similar to the CBT protocol [10], PIM-Sparse Mode (PIM-SM) is designed to restrict multicast traffic to only those routers interested in receiving it. PIM-SM constructs a multicast distribution tree around a router called a rendezvous point (RP). This rendezvous point plays the same role as the core in the CBT protocol; receivers "meet" new sources at this rendezvous point. However, PIM-SM is a more flexible protocol than CBT. While CBT with trees are always group-shared trees, with PIM-SM an individual receiver may choose either to connect to a group-shared tree or to switch to a shortest-path tree based on the source.

There are advantages to each type of distribution tree. The shared tree is relatively easy to construct, and it reduces the amount of state information that must be stored in the routers. Accordingly, a shared tree would conserve network resource if the multicast group consists of a large number of low-data-rate sources. However, shared trees cause a concentration of traffic around the core or the rendezvous point, a phenomenon that can result in performance degradation if there is a large volume of multicast traffic. Another disadvantage of shared trees is that traffic often does not traverse the shortest path from source to destination. If low latency is a critical application requirement, it would be preferable for traffic to be routed along a shortest path. PIM-SM architecture supports both types of distribution trees.

The PIM-SM protocol initially constructs a group-shared tree to support a multicast group. The tree is formed by the senders and receivers both connecting to the rendezvous point, just as a shared tree is constructed around the core with the CBT protocol. After the tree is constructed, a receiver (actually the router closest to this receiver) can opt to change its connection to a particular source to a shortest-path tree. This is accomplished by having this router send a PIM join message to the source. Once the shortest path from source to receiver is created, the extraneous branches through the RP are pruned. This procedure is illustrated in Figure 6.3. Note that different types of trees can be selected for different sources within a single multicast group.

The PIM protocol specifies soft-state mechanisms to periodically refresh the system state, adapt to topological changes in the network, and adapt to changes in group membership. While

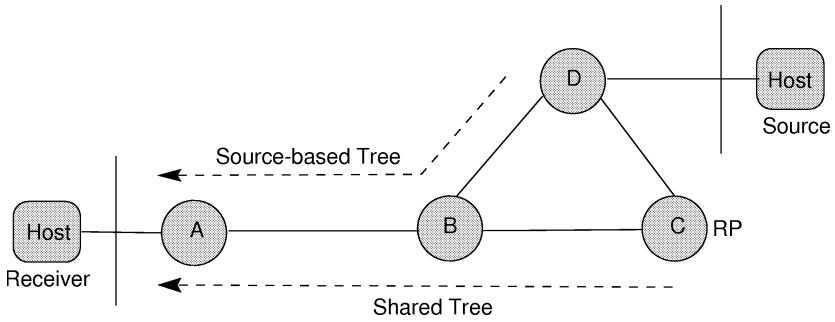


Figure 6.3: The receiver first uses the shared tree to receive data from the source then it can decide to switch to the shortest path tree

PIM relies on unicast routing tables to adapt to network topology changes, it is independent of the particular unicast routing protocol that is used to construct those tables. Other features of PIM, such as using multiple rendezvous points to eliminate the problem of having a single failure point, are too numerous to describe in this chapter.

Both DVMRP and PIM-DM protocols are in fact "historic" protocols and becoming less used in the current multicast infrastructure. Currently, the most promising multicast routing protocol is PIM-SM [40] which integrates both spare and dense modes. We develop, in the next section, a configuration policy for PIM-SM in GEO transparent satellite-based networks.

6.4.2 PIM-SM Configuration Policy

In considering a routing protocol to be used for multicasting in terrestrial-satellite hybrid networks, one has to carefully look at the issues unique to this type of network and make use of the broadcast nature of GEO satellites.

PIM-SM requires routers that are directly attached to downstream members to join a distribution tree by transmitting explicit join messages to the group's primary Rendezvous Point (RP) which acts the root of the tree. PIM-SM creates a shared, RP-rooted distribution tree that reaches all group members. PIM-SM also provides a mechanism to switch from a RP-rooted tree to a shortest path tree (SPT).

Using PIM-SM effectively in hybrid networks depends on our capacity to: (1) manage the choice of the RP of each multicast group, and (2) configure the policy used by group members to switch from a RP-rooted tree to a SPT.

6.4.2.1 RP Placement

The *bootstrap mechanism* used in PIM-SM employs an algorithmic mapping of multicast group to rendezvous point address, based on a set of available RPs distributed throughout the network by the dynamically-elected BootStrap Router (BSR) from a list of Candidate-BSRs [40]. Routers belonging to the set of Candidate-BSRs or Candidate-RPs should be manually configured in the network [40]. For hybrid networks, we assume that all the feeds and send-only feeds are configured as Candidate-BSRs and Candidate-RPs[4].

[4][40] recommends that C-BSRs should be equal to C-RPs.

In order to profit from satellite link broadcast nature, the RP placement policy that we recommend in such type of network depends on satellite uplinks (feeds) positions in the terrestrial network.

For each multicast group we process as follows to select the RP:

- If there is a feed or a send-only feed which is a multicast source of the group, it is chosen as the RP of this group. If there is more than one feed belonging to the group, the closest feed to the source is chosen as the DR.
- If there is no feed which is a source of the group, it is recommended to choose the closest feed to the multicast sources as the RP of this group. If there is more than one feed in the hybrid network, the feed which has the highest priority for this group will be elected.

This policy of RP placement in a hybrid network can be satisfied by effectively choosing the priority values of each RP in the Candidate-RP message sent to the BSR.

6.4.2.2 Switching from a RP-rooted tree (RPT) to a SPT

The PIM-SM specification |40| does not specify a fixed policy to switch from the RP-rooted tree to the SPT, but it recommends that the router monitors data packets from sources for which it has no source-specific multicast route entry and initiates such an entry when the data rate exceeds the configured threshold. Let us apply this method to a hybrid network. If at least one satellite receiver that is a member of the multicast group decides to switch to the SPT and when the SPT contains a feed, all satellite receivers will receive multicast packets sent by the source. Multicast receivers that are still using the RP-rooted tree will receive a **duplicated packet**: a copy from the terrestrial interface belonging to the RP-rooted tree and another copy from the satellite interface.

We propose a switching policy that can be used effectively in a terrestrial-satellite hybrid network. This policy is as follows:

- If the source is a feed, it makes sense for all satellite receiver members to join source-specific tree and prune the source's packets off the shared RP-centered tree since it forwards data to members via the satellite link. Or, the RP triggers Register-Stop messages in response to Register messages sent by the source only if the RP has no downstream receivers for the group (or for that particular source), or if the RP has already joined the (S,G) tree and it is receiving the data packets natively. Then we recommend that all satellite receiver members directly join the source. This can be done by properly configuring the threshold maintained by the router. For example, we can attribute for each (S,G) a threshold value close to zero to guarantee that each satellite receiver member switch to the SPT when the source is a feed.
- if the source is not a feed, it is not desired to switch from the RP-rooted tree to the SPT especially when the RP is a feed. Multicast packets will be sent by the source to the RP via the UDLR tunnel [39]. The threshold maintained by each member for the (S,G) entry should be set to a value higher than the link capacity.

A major advantage of the use of PIM-SM in hybrid networks is that the option provides routers to switch from an RP-shared tree to a Shortest-Path-Tree (SPT) as soon as they start receiving data packets directly from the source.

We believe that PIM-SM, used with UDLR, grants an efficient use of communication link resources considering that multicast packets will be sent via the satellite downlink. Terrestrial links will then be used effectively by applications that need resources that cannot be offered by the satellite connectivity and mainly delay-sensitive applications such as Distributed Internet Simulations (DIS).

6.5 Case Study: DIPCAST Transparent System

6.5.1 System Architecture

6.5.1.1 Overview

As we show in Figure 6.4, the DIPCAST satellite system[5] that we consider in this study takes place in the edge network. This satellite belongs to the generation of transparent satellites which we have described in Section 2.2.2. The satellite Gateway is connected to the terrestrial Internet. It has three or more terrestrial interfaces, each one is connected to a different ISP. The satellite terminals (STs) are located in the ISP's access networks. Some of them, referred as **ST Rx only**, have not the capability to send packets to the satellite. Others having the capability to both send and receive from the satellite are called **ST Rx/Tx**.

This system gives the possibility to the ISPs to supply their PoPs via the Gateway with up-to-date data and so to bring the content to the subscribers in the access networks. This system also allows the communication between the users belonging to the different access networks but with the constraint that they should always use the Gateway as an intermediary. The applications that will be deployed are mainly multicast applications but it will also be possible to have unicast connections.

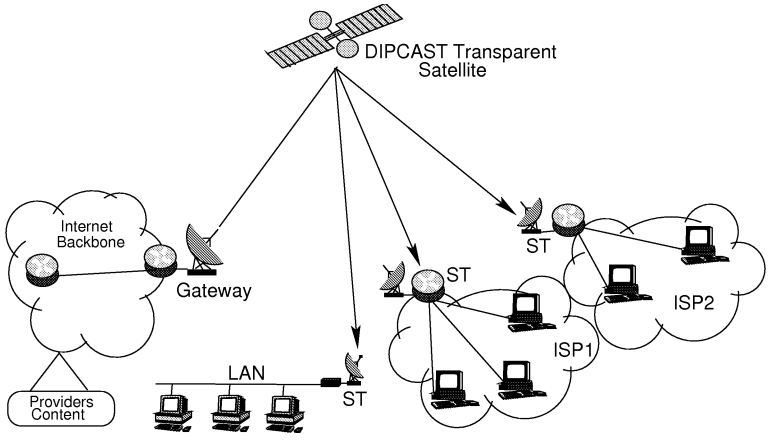


Figure 6.4: The architecture of the DIPCAST Transparent System

[5]DIPCAST (DVB as a support of IP multiCAst via SaTellite) is a French government-funded research project. More information about this project can be found in `http://www.dipcast-satellite.com`. In this dissertation, we use the term DIPCAST to refer to the GEO satellite system which is under development by the partners in this research work.

6.5.1.2 Multicast Services

The user services that will be provided by the DIPCAST system are mainly:

- Efficient content distribution in the CDNs (Content Delivery Networks)
- Content transfer to the ISP's PoPs which can then be directly distributed to end users, or kept in local caches waiting to be requested,
- Interactive services between end users.

The multicast applications which are expected to profit from these services are summarized in Table 6.1:

Profile	Application
Reliable transfer 1 to n	caching services
	multimedia on demand
	interactive TV
Interactive applications n to m	tele-teaching and tele-engineering

Table 6.1: Classification of multicast applications

Our study of the adaptation of PIM-SM protocol to the DIPCAST transparent system will take into account the inherent characteristics of these applications. In particular, we are able to make the following significant observation: ***the source of 1 to n applications will be located behind the Gateway whereas the sources of n to m applications will be located behind the satellite terminals (STs).***

6.5.1.3 Main Assumptions and Constraints

We enumerate in this sub-section, a certain number of assumptions that affect directly or indirectly the adaptations of PIM-SM that we will propose.

- We assume that all the STs belong to the same domain and so there is no need to consider the inter-domain multicast protocols such as MSDP [86].
- The intra-domain multicast routing protocol that will be used is the PIM-SM protocol. It will be implemented in the Gateway as well as in the STs. The behavior of this protocol should respect the standard PIM-SM specifications in the terrestrial interfaces.
- Terrestrial routers communicating with the DIPCAST entities are assumed to implement the PIM-SM standard version.
- Some STs will not be able to send packets to the satellite interface (receiving only terminals, ST Rx only). These terminals should use the LLTM mechanism [39] to send data packets to the Gateway via the terrestrial network.

Recall that to send or receive data (unicast or multicast), the DVB standard specifies that an ST should have made the connection procedure (the log-on procedure) to the NCC successfully. It can then send the multicast packets when the log-on procedure ends and a PID (Packet

Identifier) is allocated to the corresponding multicast session. This identifier is used to identify the MPEG-2 data segments which will be sent by the ST[6].

6.5.1.4 Multicast communication scenarios

As we have outlined above, the multicast communication scenarios that will be used in the DIPCAST system respect the following assumptions:

- The source of a multicast group may be behind the gateway. Packets will be multicasted by the Gateway to the satellite terminals.
- The multicast source may be behind an ST. In this case, the ST sends the multicast packets in unicast to the Gateway which then broadcasts them to the members via the satellite interface.
- For both cases the Gateway or the ST should start sending the packets to the satellite segment only if there is at least one ST which is member to the multicast session.

Two scenarios are possible: the source is behind the Gateway or the source is behind an ST.

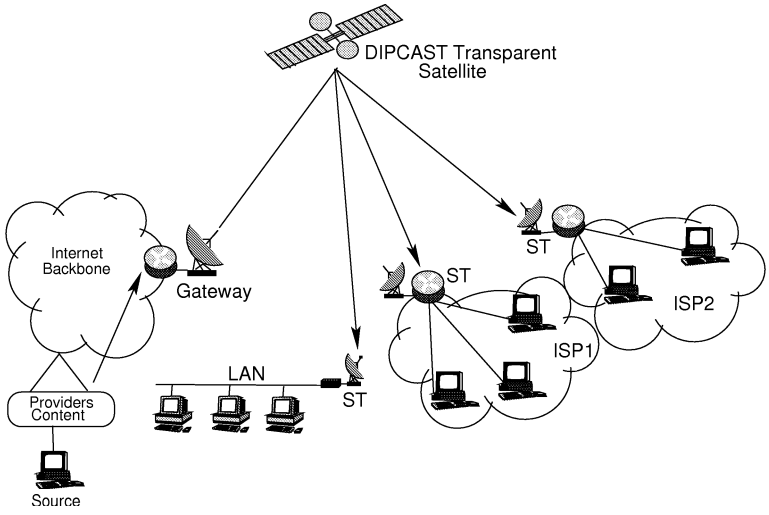


Figure 6.5: The multicast source is behind the Gateway

The multicast source is behind the Gateway As shown in Figure 6.5, when the multicast source is behind the Gateway, a one-to-multipoint connection is established between the Gateway and the STs which are members of this multicast session.

The source is behind an ST In this case the STs establish a point-to-point link-level connection with the Gateway. The Gateway sends the encapsulated packet received from the STs to the satellite segment in native multicast. Figure 6.6 shows the path of the multicast packets sent by a source behind an ST.

[6]In Section 7.2, we detail the format of the MPEG-2 data segments and we show how the PID identifier is used.

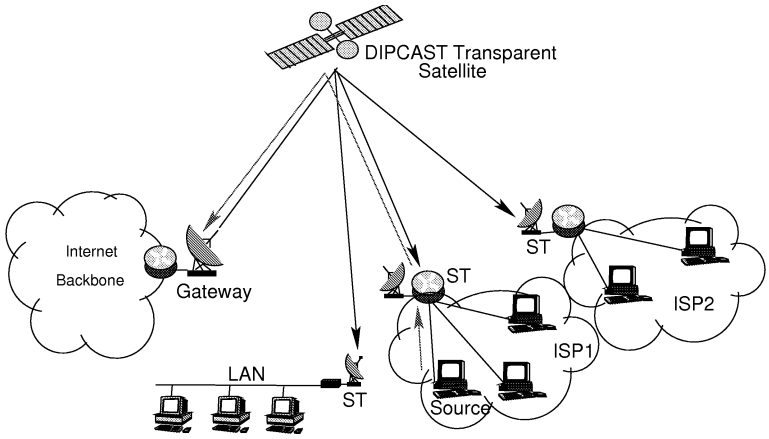


Figure 6.6: The multicast source is behind an ST

In the following, we describe the different configurations of PIM-SM in order to guarantee an efficient multicast transfer in the DIPCAST system and to respect the different constraints imposed by the system that we have described above.

6.5.2 PIM-SM Configuration

The Gateway and the satellite terminals in the DIPCAST system are considered as routers and implement a modified version of PIM-SM which is adapted to this system.

In this section, we describe how PIM-SM should be configured in the studied system. We analyze two aspects of PIM-SM configurations that we have already examined for generic satellite transparent systems in Section 6.4.2:

- Rendez-vous Points (RPs) configuration.
- Switching between RPT tree and SPT tree modes.

Note that we have already discussed these two points in Section 6.4.2, but here we give some specific recommendations that take into account the system architecture and constraints described above.

6.5.2.1 RPs Configurations

The ISPs have a direct link with the BAS (Broadband Access Server) present in the Gateway of the DIPCAST system. The BAS communicates with a PIM-SM router of the ISP network. As there is a virtual PIM-SM router for each ISP, it seams like that there is a PIM-SM router in the Gateway which takes part of the ISP network.

A multicast session is identified by the pair (source, group) addresses. The dynamic mechanism that could be used in PIM-SM to determine the RP of each multicast session is the BootStrap (BSR) Mechanism [55]. This mechanism is better suitable for a network which is very dynamic and especially terrestrial networks which is not the case of DIPCAST.

In the studied system, the ISPs should statically configure the RPs of different multicast groups in the routers traversed by the multicast packets. This requires of course the knowledge

of the group addresses which will be used by the applications that will be deployed in the system.

Recall that, as shown in Figure 6.4, the ISP network is composed of:

- An upstream part which contains the servers of content connected to PIM-SM terrestrial routers. This part can communicate with the terrestrial Internet.
- A downstream part which is close to ISP's access networks. This part contains the content servers and it may contain PIM-SM terrestrial routers.

In the following, we describe the configuration of the RP for each multicast communication scenario.

Case 1: Multicast source behind the Gateway

In this case, the RP will be located in the upstream network. There are two main possibilities:

- In the gateway: in this case the RP do not send the multicast data to terrestrial interfaces even if there are terrestrial members to respect the hard constraint in the DIPCAST system.
- Or in a router inside the upstream network: in this case the DIPCAST network is considered as a branch of the RPT tree.

The first option raises a problem of traffic concentration around the Gateway, because the sources of all the ISPs to which the Gateway is connected will continue sending their data to the RP even if there are no interested members in the satellite segment. Moreover, the terrestrial members will not be able to receive the data from the RPT because this is not authorized in the DIPCAST system.

As conclusion, for this scenario, a router inside the upstream network (for example the border router connecting the ISP to the Gateway) will be configured as the RP in order to allow the multicast sessions to efficiently use the satellite segment. This option guarantees that the Gateway will receive multicast packets to forward only if there is at least one ST which is member of the session. The ISPs are free to decide which router they configure as RP for their multicast sessions that will use the satellite network.

It should be noted that the current deployment of PIM-SM in the terrestrial networks is based on static configuration of the RPs and not on a dynamic configuration using the BSR mechanism.

Note also that it is possible to configure the RPs only in the DRs of multicast sources when other routers in the multicast tree do not have a RP configured for the multicast group address. In this case, according to the PIM-SM specification [55], all intermediate routers update the IP address of the RP when they receive a PIM-SM Join message to be forwarded to the RP.

Case 2 : The sources are behind the STs

In this case, the sources send the multicast data only to other users which are reached via the DIPCAST system. For this scenario, the RP might be localized:

- in the downstream ISPs part: in a router inside the ISP's access network,
- or in the Gateway.

Let us assume that the RP will be configured using the first method and take as example the case when it is the ST to which the multicast source is connected. In the case of n to m applications there is more than one source located in the ISP's access networks sending to the same multicast group address. As it is not possible in PIM-SM to have different RPs for the same multicast group belonging to the same domain, only one ST among the STs for all the sources could be used as the RP. Other sources which are not directly behind the STs should send their packets first encapsulated in PIM-SM Register packet to the RP which is one of the STs. Or, in any way one of the constraints of the system is that this packet has to be sent by the RP to the Gateway then to the members. In consequence, this solution introduces an additional satellite round trip time to the transfer delay of multicast packets.

It is therefore more judicious and efficient to choose the second option which consists of configuring the Gateway as the RP of all multicast groups for n to m applications when the sources are located behind the STs.

In order to reduce the control and data traffic in the satellite segment, we will develop in Section 6.5.4 some specific mechanisms for the DIPCAST system which will be included in the PIM-SM protocol implemented in the Gateway and in the STs.

6.5.2.2 Switching from RPT to SPT

One of the main PIM-SM configuration issues that we have to consider in the DIPCAST system is the switching between the two tree modes (RPT and SPT). In fact, PIM-SM gives the possibility to members to switch to the source-based tree (shortest path tree) by sending a PIM-SM (S,G) Join message directly toward the source after obtaining the unicast address via the reception of multicast packets from the RPT tree.

In order to always receive the multicast packets via the satellite segment, the switching to the SPT will not be authorized in both multicast communication scenarios described above. This can be realized via the filtering of all PIM-SM (S,G) Join messages at the Gateway and at the STs. The members which belong to the same network as the source will have the possibility to switch to the source because their PIM-SM (S,G) Join messages will not go neither through the Gateway nor through an ST.

6.5.3 PIM-SM over DIPCAST Satellite System

The multicast scenarios of the DIPCAST system that we have described in Section 6.5.1 depend on the localization of multicast sources and receivers in the system. In the following, we examine for each scenario the multicast delivery tree obtained using the PIM-SM standard protocol and then we derive the main problems caused by this protocol as well as some undesirable multicast delivery ways. We propose a set of protocol adaptations in order to solve these problems.

6.5.3.1 The case of a multicast source behind the Gateway

This case corresponds to one to many (1 to n) applications. In DIPCAST, the target 1 to n applications are:

- caching services,
- multimedia on demand, and
- interactive TV.

As we have outlined in Section 6.5.2.1, the RP is configured in this case as a router that belongs to the ISP, for example the border router that connects the ISP to the Gateway.

The Gateway sends multicast packets to the satellite segment as soon as it receives a PIM-SM Join message from the first member that joined the multicast session. The STs having members behind them will forward the received data to the terrestrial interface.

The PIM-SM behavior in the terrestrial network of the ISP will not be modified. In particular, the members in the ISP networks can subscribe to the multicast sessions if no restriction has been enforced.

Building RPT tree

At the beginning of the multicast session, the source sends PIM-SM Register messages towards the RP (for example the edge router of the ISP). The RP discards the received packets if there are no members which are joined through its interface connected to the Gateway.

The first user belonging to an access network which wants to join a multicast session, sends an IGMP (*,G) Join to its Designated Router (DR) which in turn sends a PIM-SM Join message towards the RP, for example the edge router of the ISP to which the source of content belongs. This message will be received and treated by the ST and then sent toward the RP via the Gateway. The ST, the Gateway, and the edge router create an (*,G) entry in the multicast routing table maintained by PIM-SM which has as list of incoming interfaces (`iif` list) the interface used to join the RP and as list of outgoing interface (`oif` list) the interface from which the message was arrived.

Other users who want to subscribe to the same multicast group make similar procedure as the first user except that particular adaptations, that we will detail in the following section in order to minimize the traffic of PIM-SM control messages on the satellite segment, will be applied.

After these steps, there is an establishment of the RPT tree, insofar as all the multicast packets are sent by the multicast source towards the RP then from the RP towards the Gateway to be then transmitted to the satellite segment. When receiving multicast packets, the ST having members in the access network to which it is attached, transmits the packets towards the access router and which forwards them to the end-users.

As specified in the PIM-SM standard, on receiving the first join message, the RP will send a PIM-SM (S,G) Join message to the source followed by a PIM-SM Register-Stop message as soon as it receives a new data packets from the source. This allows to build an SPT tree between the RP and the source; i.e., in the terrestrial networks of the ISP if the source has already started sending multicast packets.

We illustrate in Figure 6.7, the sequence of messages sent in order to build the RPT tree in the case when the source is behind the Gateway.

Switching to the SPT

The members which belong to the upstream network of the ISPs will have the possibility to switch to the SPT tree in order to receive the multicast packets directly from the multicast

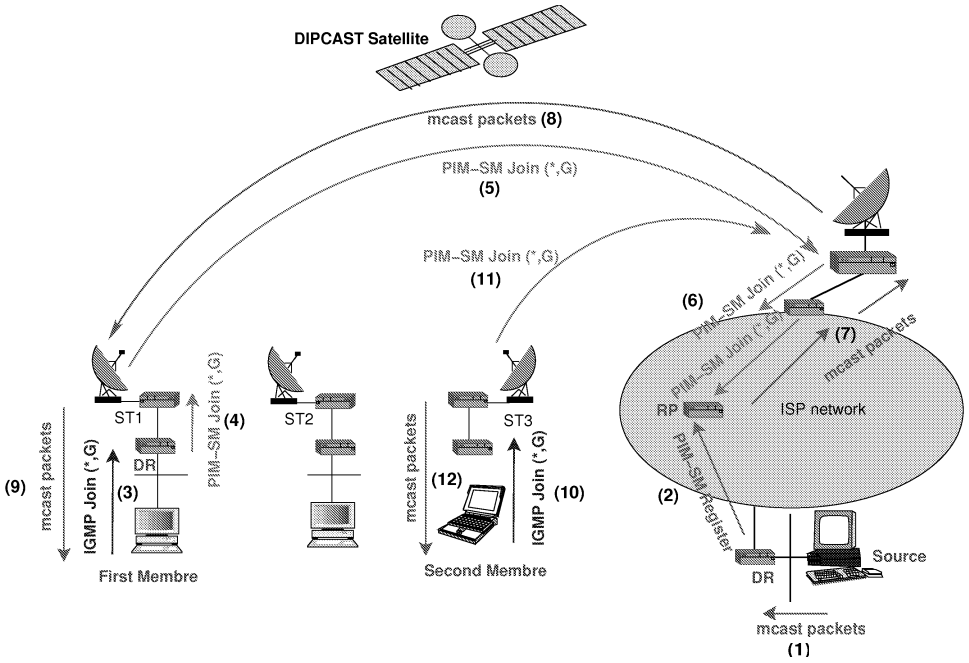


Figure 6.7: PIM-SM behavior when the source is behind the Gateway

sources via the terrestrial network. Their DRs send a PIM-SM Join (S,G) message directly toward the multicast source as soon as they receive the first packet from the RPT tree.

In order to avoid the reception of multicast data by the ground network when the terrestrial path is shorter than the satellite one, the switching to the SPT tree by the DRs of the members that receive the data by the satellite segment should not be authorized.

Most of PIM-SM routers are configured, when they play the role of an DR, to switch towards the SPT mode as soon as they receive the first multicast data packet from the RPT. In the DIPCAST context, the access routers should not allow the members behind them to switch to the SPT. To this end, these STs have to ignore all PIM-SM Join (S,G) messages sent by access routers to which they are connected. This solution has the advantage of not modifying the PIM-SM behavior in the access routers given that it is not required to modify the PIM-SM protocol implementation in these routers.

The switching from the RPT tree to the SPT tree will be of course authorized for the terrestrial members in the upstream networks (access networks) of ISPs to which the multicast source belongs. This gives the possibility to these members to use the shortest path towards the source to receive multicast packets.

We can easily see in Figure 6.8 that the users in the upstream network and those in the downstream network can send PIM-SM Join (S,G) messages. The STs blocks (they do not generate in their turn a PIM-SM (S,G) Join message as required in the standard) the PIM-SM Join (S, G) messages coming from the members which are attached to them in order to avoid the switching to the SPT tree mode.

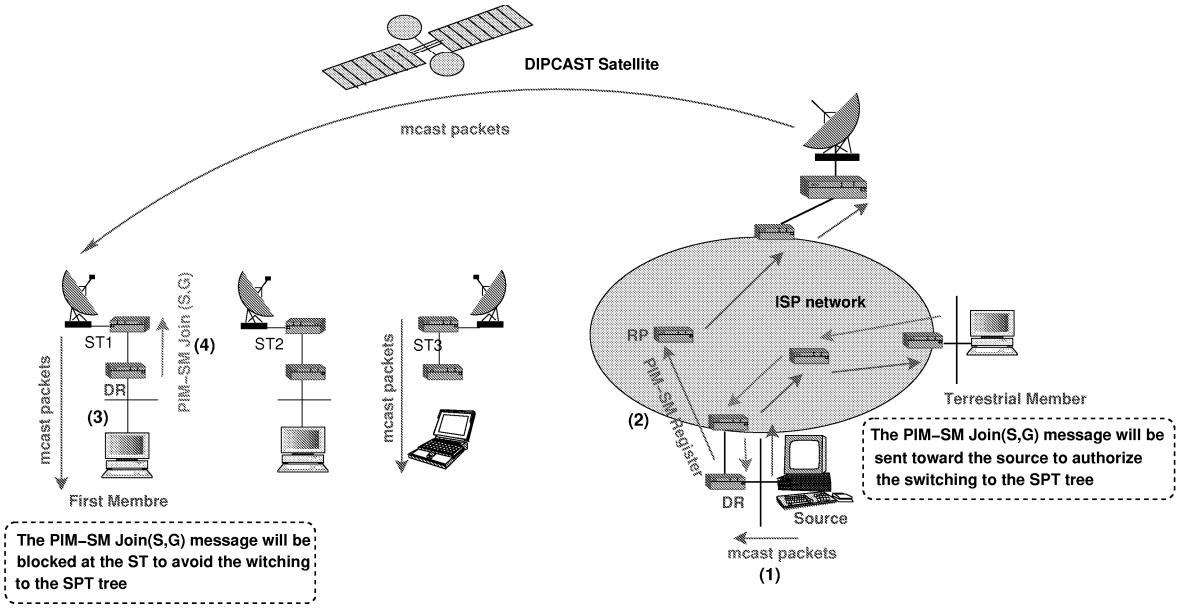


Figure 6.8: Switching from the RPT tree mode to the SPT tree mode in the case when the source is behind the Gateway

6.5.3.2 The case of a multicast source behind an ST

We study in this section the case when the multicast source is behind the ST. Within the framework of DIPCAST system, the multicast applications which will use this scenario are mainly the n to m applications:

- Tele-teaching
- Tele-engineering

In this scenario, the Gateway plays the role of the RP for all multicast groups. However, in order to reduce the useless traffic sent by the STs, to which the multicast sources are connected, towards the Gateway, they transmit the data only when they receive an implicit request from the Gateway and they stop sending the data on its request.

Building RPT tree

Let us take the case of a multicast session with two multicast sources behind two distinct STs, and thus they belong to two different access networks but, of course, to the same ISP. Indeed, in the Gateway there is a virtual router for each ISP. Thus, the PIM-SM router of the ST belonging to an ISP communicates with the PIM-SM virtual router of this ISP in the Gateway. Note that it is not possible to have multicast sessions between two STs which belong to different ISPs.

When the source starts sending multicast packets to the RP (which is the Gateway), the data will be encapsulated into PIM-SM Register messages which are then transmitted in unicast (@IP of dest = @IP of RP). So, the RPT tree is built and all the packets sent by the

sources will pass through the RP (the Gateway) before being transmitted toward the satellite segment.

The members will send their PIM-SM Join messages to the Gateway via the ST to which they are connected.

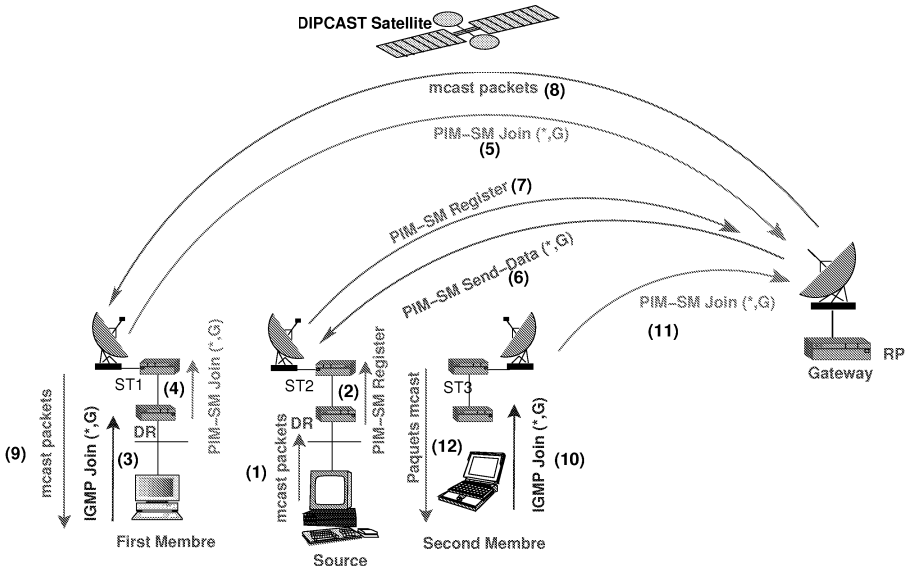


Figure 6.9: PIM-SM behavior when the source is behind an ST

We illustrate in Figure 6.9, the sequence of messages exchanged between the system entities to build the RPT tree in the case when the source is behind the Gateway.

Switching to the SPT Tree

We will not authorize the switching to the SPT tree because in any way the STs should receive the data from the Gateway. To this end, the STs should forward received multicast data only on explicit request from the Gateway. This issue will be investigated in more details in the next sub-section.

6.5.4 PIM-SM Adaptations

In this section, we detail the adaptations to be brought to PIM-SM in the STs and in the Gateway in order to optimize as well as make possible the multicast transfer in the DIPCAST system.

In fact, the PIM-SM standard protocol can operate correctly for both communication scenarios. However, it would be better to optimize its behavior by taking account of the various constraints and assumptions considered within the DIPCAST system.

We identify two main problems resulting in using the PIM-SM standard protocol in the DIPCAST system, which are:

- the useless signaling messages in the satellite segment, and
- the useless data traffic sent by the multicast source to the satellite segment.

We propose in the two following sub-sections a set of schemes and techniques that allow us to avoid these undesirable behaviors in our studied system.

6.5.4.1 Reducing PIM-SM Join/Prune Periodic messages (for both scenarios)

According to the standard PIM-SM, each ST must periodically send a PIM-SM (*,G) Join message to the RP in order to announce the list of multicast groups for which there are still members in its access network. This allows us to keep the state of the list of outgoing interfaces for these groups maintained by the Gateway up-to-date. The period of sending these messages is called JT (Upstream Join/Prune Timer) and it is set as default to 60 s.

The approach based on installing and periodically refreshing the states by network nodes is known as the *Soft-State approach* in contrast of the *Hard-State approach* which consists in explicitly and reliably setup the states using acknowledgments and retransmissions when routers exchange signaling messages (for example the Join and Prune requests).

The application of this PIM-SM operation in DIPCAST system leads to a high load of useless control messages. Indeed, the STs having members which are connected to them will periodically send PIM-SM Join messages to the satellite interface. Given the significant number of these terminals, the control traffic load generated by these messages is considerable.

We propose in this section a new mechanism which will be applied in the satellite segment in order to reduce the load of PIM-SM control messages.

We suggest not to use the Soft-State approach for the entries where the satellite interface is one of the incoming/outgoing interfaces in the STs and the Gateway. This implies that the STs will never send PIM-SM Join messages to the satellite interface in order to refresh the states in other STs or in the Gateway, but they still have the PIM-SM standard operations in their terrestrial interface(s). Indeed, the Soft-State approach is used in the ground network to overcome the problem of packet loss because of congestion in the network. The loss rate in the satellite segment is, in general, low than that of terrestrial links, thanks to the use of powerful error correction mechanisms at the link level.

Our goal now is to find a method to detect the last member that has left the multicast group in order to remove the satellite interface from the list of outgoing interfaces in the corresponding entry of the multicast routing table maintained by the Gateway. This method should be applied for the two multicast communication scenarios presented above.

We propose a mechanism based on the election, for each multicast session, a specific ST among all the STs members called the **"ST master"** which will periodically send PIM-SM Join messages to the RP. This ST could be for example the one having the lowest IP address among all the STs members of the multicast session present at a given moment[7]. Our proposed mechanism consists on the following operations:

- When an ST receives the first join from its access network for a specific multicast group, it sends this message towards the RP (the Gateway or a router of the ISP depending on the multicast communication scenario) even if the session is already announced in the MMT table[8]. However, the STs will not send the PIM-SM Join messages periodically. Note that the PIM-SM (*,G) Join messages sent by an ST to the Gateway will have

[7]Other methods could be of course applied to select the "ST master".

[8]The MMT table is a specific table which is periodically broadcasted by the Gateway and which gives the correspondence between the PID and the multicast group address. This table will be detailed in Section 6.2.

as IP source address the address of the satellite interface of the ST and not that of the member's DR.

- The Gateway maintains a list of the IPs addresses of the STs which are members of each session multicast. This list is updated on each reception of a PIM-SM Join message of a new ST member[9] or a PIM-SM Prune message from an ST which is already a member of this session. The Gateway attributes then the role of "ST master" to a specific ST among the list of STs member and it sends to it a new PIM-SM message called **PIM-SM Join-Send** to ask it to periodically send the PIM-SM Join messages.

- When an ST member receives a PIM-SM Prune message from its terrestrial interface it forwards it directly towards the RP.

- When the Gateway receives a PIM-SM Prune message from an ST, it removes this ST from the list of STs members of the corresponding group. If this ST which has left the group is the current "ST master", the Gateway sends a PIM-SM Join-Send message to the new "ST master".

- We use the soft-state approach at the Gateway only for the periodic Join messages which are sent by the "ST master". For each "ST master", a timer is maintained by the Gateway and it is set to its maximum value **(ST-Master-Timer msec)** when receiving a new periodic join message from this ST. When the timer expires, the Gateway assumes that the "ST master" has left the group or that it is not able to send the PIM-SM Join messages, removes this ST from the list of STs members of the multicast session, and attributes the role of "ST master" to a new ST.

- If all the STs have left the multicast group or that the last ST has not sent any PIM-SM Join message during ST-Master-Timer msec, the gateway removes the corresponding entry from the multicast routing table. In addition,

  - if it is the first multicast communication scenario, the Gateway sends a PIM-SM Prune message towards the RP,
  - if it is the second multicast communication scenario, the Gateway sends a PIM-SM Date-Stop message towards the ST so that it stops forwarding multicast packets towards the Gateway.

It should be noted that the above proposed mechanism is a centralized algorithm given that only the Gateway that maintains the list of all STs members for all active sessions and decides which ST plays the role of the "ST master" for each active session. For a significant number of STs members, this solution could become complex to manage at the Gateway because of the expected large size of data to save and handle. However, in the DIPCAST system it is assumed that the number of STs will be at most equal to 30.

Anyway, a distributed approach based on the same idea could be adopted. In this distributed election algorithm of the "ST master", each ST member maintains the list of ST members and decides to take the role of "ST master" or not. In this case, the Gateway must broadcasts all received PIM-SM Join/Prune messages towards the satellite segment to allow

[9]We mean by an "ST member" of a multicast session, an ST having at least one host in its access network which is a member to this session.

the ST members to maintain and update the list of members. It should be noted that this solution may cause some security problems which are out of the scope of this work but it is an alternative scheme when the number of STs increases in future extensions of the system.

6.5.4.2 Reducing useless multicast data in the satellite segment (source behind an ST)

In the second multicast communication scenario, it is desirable that each ST forwards the packets sent by the sources of its access network towards the Gateway (which is the RP for this scenario) encapsulated in PIM-SM Register messages only if at least a member is already subscribed to the multicast session.

To allow this, we suggest that the ST block all PIM-SM Register messages waiting for an explicit sending request from the Gateway using a new proposed PIM-SM message type called **PIM-SM Data-Send**. Indeed, on the reception of the first join of a multicast group, the Gateway sends a PIM-SM Data-Send message specifying the list of requested multicast groups. As other PIM-SM messages, this message has as destination address the address reserved to PIM-SM routers and the field Router Alert activated. When receiving a PIM-SM Data-Send message from the Gateway, the STs having the sources behind them start forwarding the received packets (PIM-SM Register packets or native multicast packets) towards the Gateway (which is the RP) which sends them towards the ST members via the satellite segment. An ST which receives a multicast packet from the satellite, forwards this packet towards the terrestrial interface if there are users subscribed to the corresponding session.

The Gateway destroys the (*,G) entry present in its multicast routing table in the following cases:

- it receives a PIM-SM Join message from the last "ST master",
- or that the last "ST master" has not send the periodic PIM-SM Join messages for ST-Master-Timer msec.

In this case, the Gateway sends a **PIM-SM Data-Stop** message towards the ST sources so that they stop transmitting both PIM-SM Register messages and native IP multicast packets towards the Gateway.

This new proposed PIM-SM message type like other new PIM-SM message types that we have detailed earlier is used only in the satellite segment to overcome some undesirable behaviors of PIM-SM in GEO transparent satellite systems.

6.5.5 The Multicast Mapping Table

The MMT table is a signaling DVB table which gives the correspondence between the IP address of each multicast group and the PID used at the MPEG level to send MPEG data segments belonging to this group. This table is periodically broadcasted by the Gateway to the satellite terminals. It looks like that of Table 6.2.

In our system, we assume the use of dynamic MMT table in the sense that it will contain only the multicast sessions which are really active in the satellite segment. In other words, it contains only the list of sessions to which the Gateway has received at least a PIM-SM join message from the satellite interface.

The events that affect the content of the MMT table are the following:

PID	Multicast Sessions
X	(*, G1), (*, G4)
Y	(*, G2), (*, G5), (*, G8)
Z	(*, G10)

Table 6.2: The format of the MMT table

- The first PIM-SM (*,G) Join message is received by the Gateway: in this case a new (*,G) entry is created in the MMT table.
- The last PIM-SM Prune (*,G) is received by the gateway, the corresponding entry is removed from the MMT table.

The sending period of this table is about ten seconds and it is possible that between two sending periods, the Gateway sends the MMT table if a change has happened on one of its entries.

6.5.6 Case of Terrestrial Reverse Path

If an ST does not have the sending capability (ST Rx only) via the satellite segment, it must implement the LLTM (Link Layer Tunneling Mechanism) [39] (RFC 3077) mechanism to be able to send the data towards the Gateway through the terrestrial network. The standard LLTM Mechanism aims to emulate the bidirectionnality of unidirectional links and, in particular, satellite links.

The default feed as defined in [39] has to be configured in all ST Rx only as the Gateway. The Gateway should implement all the functionalities of the feed as described in [39].

Let us take the case where the multicast source is behind an ST Rx only. According to the LLTM mechanism, when receiving a multicast packet to be sent to the Gateway (which is RP), this ST encapsulates this packet using the GRE (Generic Routing Encapsulation) encapsulation standard |51| in an unicast packet and sends it via a terrestrial tunnel to the Gateway. Then the Gateway multicasts it toward the satellite segment if there is at least a member of the multicast group which could be joined by one of the STs.

Case of ST to ST communications

If the DIPCAST system includes at least one ST Rx only and that it is necessary to authorize the communications of an ST with other STs, the STs Tx/Rx must implement the "feed part" of the LLTM mechanism.

6.6 Chapter Summary

In this chapter, we have discussed the behavior of the IP multicast standard model in satellite-terrestrial transparent networks. We have examined many problems related to the use of the IP-level multicast protocols in such type of networks.

We have addressed the problem of dynamic multicast routing protocols over satellite links. For DVMRP and PIM-DM, we have identified some network configurations where the satellite receivers can receive duplicated packets and we have proposed a method to overcome this

problem. We have developed a configuration policy of PIM-SM in hybrid networks concerning the choice of the list of Rendezvous Point (RPs) and the switching from the RP-rooted tree to the shortest path tree.

We have presented a case study concerning a concrete system (DIPCAST system) that uses a GEO transparent satellite to provide the multicast service for end users. We have proposed and described a set of configurations and adaptations of PIM-SM protocol in order to optimize the multicast transfer in this system.

Despite its large benefits, the deployment of IP multicast over transparent GEO satellites has the inconvenience that the multicast packets will be sent to the continental coverage even if the group members are located in a limited region in the continent. Next-generation of GEO satellite systems are able to redirect received packets to specific spot beams rather than providing a global coverage. In the next chapter, we provide a set schemes and techniques to enable the efficient deployment of IP multicast in these data communication systems.

Chapter 7

Supporting IP Multicast in the Next Generation of GEO Satellite

7.1 Introduction

GEO transparent satellite systems have been an important element of telecommunications networks for many years serving, in particular, long distance telephony and television broadcasting. The involvement of satellite in IP networks is a direct result of new trends in global telecommunications where Internet traffic will hold a dominant share in the total network traffic. The large geographical coverage of the satellite footprint and its unique broadcasting capabilities as well as its high-capacity channel combined with readily available Ka Band spectrum will retain satellite systems as an irreplaceable part of communications systems, despite the high cost and long development and launching cycle of a satellite system.

To support high bandwidth applications, it is anticipated that the next-generation satellite communications systems will differ from the traditional systems by including intelligent functions in the on-board satellite, the use of ka-band and V-band, and the use of the spot beams technology. All theses aspects may handle different question in the manner to use the Internet protocols over satellite.

Future generation satellites are expected to offer services that are not limited to existing mainstream C- or Ku- Band transparent satellites. As outlined in Section 2.2.3, the next generation broadband systems are expected to employ regenerative satellites rather than the current transparent satellites. Such satellites will also operate at higher frequencies (e.g. the 20/30 GHz, Ka-Band) and may be expected to employ spot beams, rather than providing continental coverage. On-board processing (switching) will direct packets to each appropriate downlink spot beams. This will enable multiple uplink terminals (at different locations) to serve as feeds to the multicast receivers, thus providing (effectively) a space-bourne multicast overlay over the existing terrestrial Internet. This approach is well suited for unicast transmission, which emulates the use of Internet switching in LANs. However, in order to support efficient multicast, the on-board switch needs to be multicast-enabled.

In this chapter[1], we study the issue of the encapsulation and the efficient segmentation of IP multicast packets into MPEG2-TS data segments in order to allow the on-board satellite

[1]The work described in this chapter is the result of a research cooperation with Alcatel Space Industries (DIPCAST RNRT project). The proposed mechanisms could be integrated in the next-generation of GEO satellites in the next few years.

processor to switch all received data segments to the appropriate spot beams based for example on a switching table containing the list of outgoing spot beams for each active session.

We also address the problem of adapting the PIM-SM multicast routing protocol [55] that we have presented in Section 2.1.3 for the next-generation of GEO satellite networks which use the DVB-RCS standard [41].

The remainder of this chapter is organized as follows. The next section presents the architecture of the next-generation satellite-terrestrial hybrid networks and their major characteristics. We review, in Section 7.2, the MPEG-2 standard and the main advances in IP data transmission over the DVB (Digital Video Broadcasting) technology. In Section 7.4, we describe the MPE encapsulation scheme used in GEO transparent satellite systems and we outline its main limitations in the case of the next-generation GEO satellite systems. We detail in Section, 7.5, our proposed IP-Optimized encapsulation technique that could be used for two different on-board switching approaches namely: the self-routing and the label-switching which we describe in the same section. Section 7.6 presents a protocol for enabling the IP multicast delivery over the next-generation of satellite systems using the DVB-RCS (DVB-Return Channel via Satellite) standard [41]. This protocol is called SMRP (Satellite Multicast Routing Protocol) and it is in fact an adaptation of the PIM-SM protocol. We conclude this chapter in Section 7.7.

7.2 MPEG-2 and IP over DVB

The DVB system specified by the European Broadcast Union (EBU) is based on the cell-oriented packet transmission system defined by ISO/IEC 13818-1 MPEG-2 Systems Standard [47]. MPEG-2 Systems Standards provide the mean of multiplexing several types of multimedia information into one Transport Stream (TS) that can be transmitted over a variety of transmission media [63, 90].

Traditionally an MPEG-2 TS contains packets of compressed video and audio data. The compression causes a variable data rate of each TV program because scenes with a lot of motion in the picture are encoded with a higher bit rate then scenes with less motion.

Within MPEG-2 TS, it is also possible to carry defined data containers in addition to the audio and video [90]. These data containers can be used to realize new data services or to carry IP datagrams.

Compressed data from a single source (i.e., audio, video, data, etc.) plus ancillary data needed for synchronization, identification, and characterization of the source information build up Elementary Streams (ES). Elementary Streams are packetized into either constant-length or variable-length packets to form Packetized Elementary Streams (PES). Each PES packet consists of a header followed by stream data called the payload. PES packets from various elementary streams are combined to form a Program.

Several Programs combine to form the Transport Stream together with other descriptive data called Program Specific Information (PSI). PSI defines the program and its constituent parts.

As shown in Figure 7.1, TS packets are 188 byte fixed-sized. Each TS consists of a TS Header, followed optionally by ancillary data called the Adaptation Field, followed typically by some or all of the data from one PES packet. The TS Header consists of the following fields:

- The header starts with the well-known synchronization byte which has the bit pattern

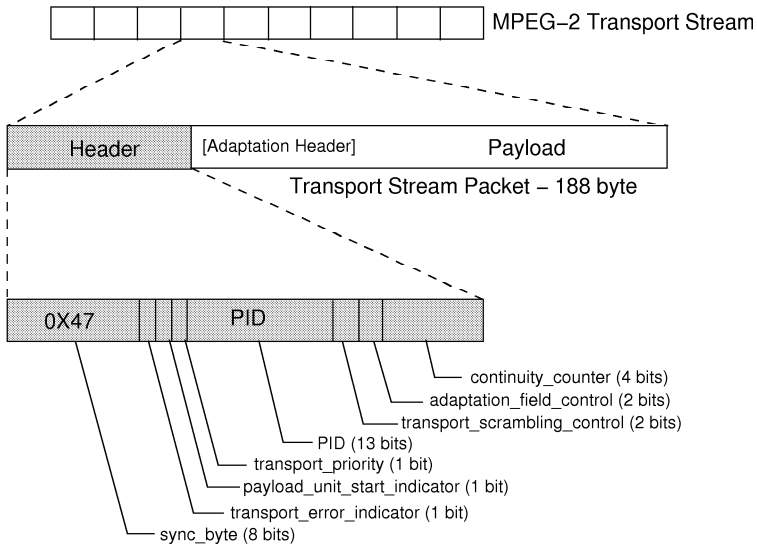


Figure 7.1: Transport Stream packet and Header structure |27|

0x47 (01000111).

- A set of three bits are used to indicate how the payload should be processed:
  - the first flag indicates a transport error (transport_error_indicator),
  - the second flag indicates the start of a payload (payload_unit_start_indicator) (PUSI), and
  - the third flag indicates transport priority bit (transport_priority).
- The flags are followed by a 13 bit Packet Identifier (PID). This is used to uniquely identify the stream to which the packet belongs (e.g. PES packets corresponding to an ES generated by the multiplexer). The PID allows the receiver to differentiate the stream to which each received packet belongs. Some PID values are predefined and are used to indicate various streams of control information. A packet with an unknown PID, or one with a PID which is not required by the receiver, is silently discarded. The particular PID value of 0x1FFF is reserved to indicate that the packet is a null packet (and is to be ignore by the receiver),
- The two scrambling control bits (transport_scrambling_control) are used by conditional access procedures to encrypted the payload of some TS packets.
- Two adaptation field control bits (adaptation_field_control) which may take four values:
  1. 01 - no adaptation field, payload only
  2. 10 - adaptation field only, no payload
  3. 11 - adaptation field followed by payload
  4. 00 - RESERVED for future use

- Finally there is a half byte Continuity Counter (4 bits) (continuity_counter).

There are mainly five types of PSI streams [43, 46]: Program Association Table (PAT), Program Map Table (PMT), Network Information Table (NIT), Conditional Access Table (CAT), and Digital Storage Medium Command and Control (DSM-CC). The decoder selects a desired program by extracting blocks, which have the required PID (the PIDs are described in the PAT and PMT blocks embedded in the transport stream). NIT specifies physical network parameters, while CAT carries information for scrambling. DSM-CC provides protocol and application program interface for user-to-network and user-to-user communications.

The DVB specification for data broadcasting [42] defines three different ways of inserting data into MPEG-2 transport stream:

1. Data packets can be encapsulated and carried inside the PES packets intended for video and audio streams. This method is referred as *Data Streaming* [42].

2. Data packets can be carried inside the section packets defined for system internal tables in DSM-CC. This method is called *Multiprotocol Encapsulation* (MPE) [42].

3. An adaptation layer protocol can segment data packets directly into a sequence of cells. This method is called *Data Piping* [42].

All three methods involve a certain overhead, which stems from the header fields and the fact that IP datagrams usually do not come in multiples of 184 bytes. The total overhead for a transmission depends on the packet length distribution and the encapsulation method selected. The observed overhead for MPE is typically between 13 and 15 per cent [27].

7.3 Next-Generation Satellite-Terrestrial Hybrid Networks Architecture

7.3.1 Architecture Description

Prior to studying the behavior of the IP multicast standard model over the next-generation of GEO satellite networks, a general network architecture of this type of networks must be specified. As we have described in Section 2.2.3, a next-generation GEO satellite is characterized by the support of *On-Board Switching* (OBS) capability offering a high bandwidth shared by satellite uplinks and a broadcast downlink toward terrestrial receivers.

There are several ways of designing a return channel from the receivers to the satellite sender for multicast services, and many people believe that terrestrial return channels are the most cost-effective and practical. Commonly-proposed terrestrial return channels include PSTN, ISDN and GSM. However, there is huge worldwide interest in defining a return channel via satellite, and there are several reasons for this. Firstly, the ordinary consumer does not want to be bothered by technical set-ups which require interconnections between the TV, PC and telephone. A solution where all the technical equipment is concentrated within one box, and without having to worry about blocked telephone lines etc., will certainly be appealing to many people. Another reason for choosing satellite services to provide interactivity is the increasing traffic on the terrestrial networks, which often results in blocking or reduced quality of service. With efficient resource allocation for example, the instantly-available capacity on a satellite link can be set as high as 2 Mbps. At this bit-rate, a 100 Mbyte file will need

just 7 minutes to transfer over a satellite circuit, whereas the time required over a 64 kbps terrestrial line will be about 3 $\frac{1}{2}$ hours. Finally, there is an advantage, both for the users and the operators, that the forward and return channels are carried on the same medium. This enables better control of the QoS and the network management: terrestrial infrastructures are not always controlled by the same operator as for satellite, particularly when national borders are crossed.

Due to the recognized need for a specification in this area, the DVB Technical Module (DVB-TM) created an ad hoc group in early 1999, called DVB-RCS (DVB Return Channel via Satellite). As shown in Figure 7.2, a next-generation GEO satellite-terrestrial network has two main entities [41]:

- **The RCSTs**[2]: Return Channel via Satellite Terminal, this is a feeder, with bidirectional Link capability. We mean by **RCST Sender** and **RCST Receiver** the RCST which sends or receives the multicast packets depending on the writing context, respectively. An RCST may have both roles for different multicast sessions.
- **The NCC**: Network Control Center, this is the core of the satellite network.

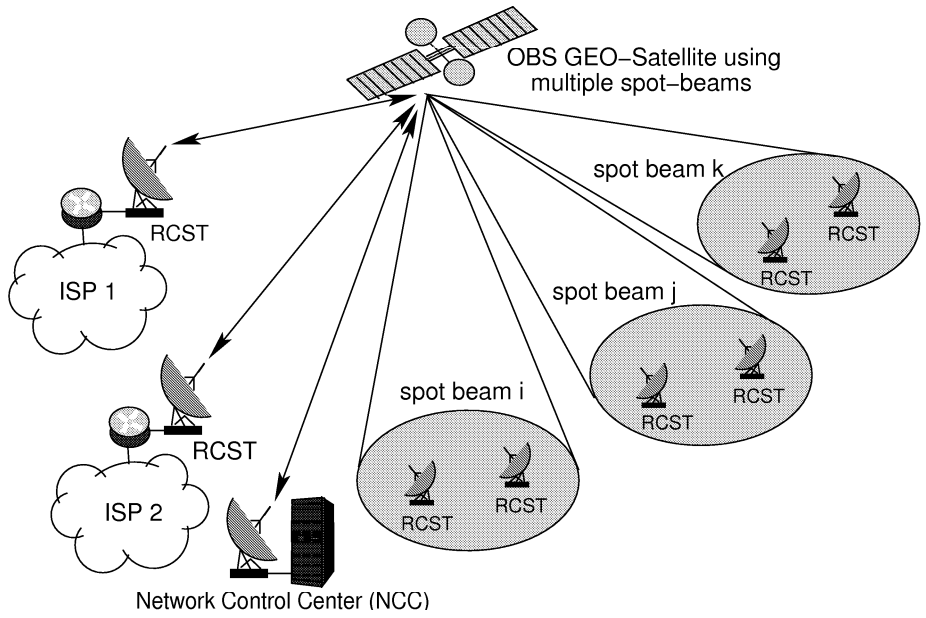


Figure 7.2: Satellite-terrestrial hybrid network architecture characterized by the support of spot-beams and on-board switching technologies

The NCC is in charge of the network control, which will include several RCSTs, but perhaps also several satellites, feeders, gateways and even several networks. The RCST network to be managed is a multipoint-to-point structure, far more complex to administrate than the opposite, the point-to-multipoint structure. The NCC is thus in charge of the control of every

[2]In the DVB-S terminology, the satellite terminal is known as the ST (Satellite Terminal), a term that we have used in the previous chapter, and in the DVB-RCS [41] terminology, is referred as the RCST (Return Channel via Satellite Terminal).

RCST in the network, as well as the network as a whole. A terminal will log on after having received general information by listening to the forward link. The information given here is on the status of the network but, most importantly, the forward link provides the Network Clock Reference (NCR). When the RCST has obtained synchronization with the NCR, it will use one of the designated slots (indicated in the forward channel) to issue a log-on request, in a slotted-aloha manner. If the terminal is successful with this request, the NCC will forward various tables containing general network and terminal-specific information. The specific information is about the necessary frequency, timing and power-level corrections to be performed by the terminal before the transmission starts. These tables will also indicate the resources allocated for the terminal, and it is possible to ask for different services or increased capacity during transmission. The NCC has the possibility, with certain intervals, to correct the transmission parameters of the RCST and, if something goes wrong during transmission, the NCC shall also have the possibility of forcing log-off from the RCST. The continuous signaling from the NCC is provided according to MPEG-2 SI |63|.

Using the CSC (Common Signaling Channel) message which is defined in the DVB-RCS standard, the RCSTs acquire all information needed for the data transmission on the satellite segment. This information includes the PID used for sending data as well as the PID reserved for control messages (CTRL_MNGM_PID) between the RCSTs and the NCC.

In the satellite-terrestrial hybrid network architecture shown in Figure 7.2, we assume that the satellite networks are capable of providing multimedia (e.g. voice, video, data) services to the ground user. As we have explained in Section 2.2.3, most of such networks in operation today or planned for deployment in the nearest future are characterized by the support of *on-board switching* and *spot-beams technology*.

7.3.2 Protocol Stack

In Figure 7.3, we show the protocol stack at the RCST and the satellite. We assume that the system integrates two existing satellite transmission standards: DVB-RCS [41] and DVB-S [42] (Digital Video Broadcasting Satellite). Note that both of which are also used in transparent satellites without any regeneration on board. The system combines these two standards into a single regenerative multi-spot satellite system allowing for full cross-connectivity between the different uplink and downlink beams.

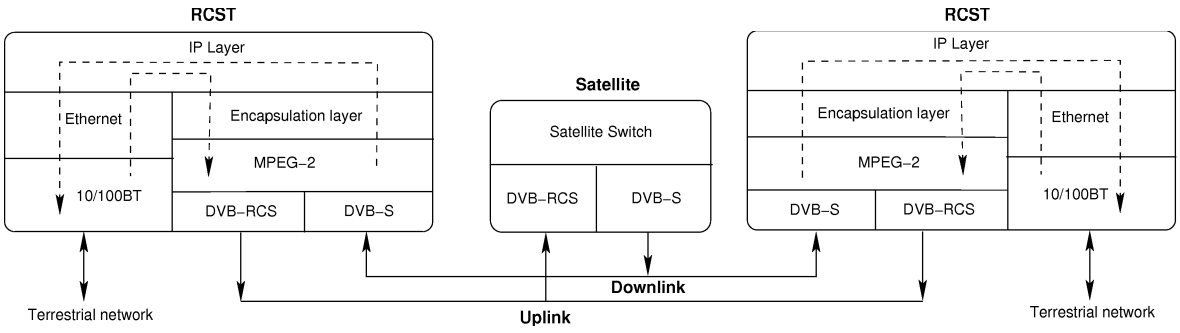


Figure 7.3: Protocol stack

As shown in Figure 7.3, the Uplink is assumed to be compliant with the DVB-RCS standard. This fact will allow users to use standard RCST (Return Channel Satellite Terminal)

stations, which will be widespread and relatively inexpensive in the future thanks to the standardization effort of terminal manufacturers and broadcast satellite operators. Individual users and broadcasters will be able to access the satellite on any of several uplink coverage footprints illuminated by the satellite, using multiple frequencies, within a TDMA frame, and at several transmission rates (multiple-rate MF-TDMA or Multiple Frequency Time Division Multiple Access). The Downlink will be fully compliant with the DVB-S Standard, including all the possible convolutional rates. This will allow users to take advantage of the economies of scale and the performance of standard commercial DVB-S receivers, which are widespread across Europe today. A key feature of the system will be the capacity to route data on any of the uplink coverage footprints on to any combination of downlink coverage footprints; the system will implement full cross-connectivity between uplink and downlink footprints. In order to accomplish all this, contributions from all DVB-RCS compliant uplink users must be demultiplexed, demodulated, and decoded and then switched and re-multiplexed into the DVB-S compliant downlink data streams as required by users. On board switching and multiplexing will take place in accordance with a dynamic multiplexer table. Each downlink has associated with it a multiplexing table. It will be possible to reconfigure this table very quickly through a signaling channel allowing very fast circuit switching at packet level onboard. In case of emergency the standard Telecommand (TM/TC) channel will be used to configure the payload.

As we can see from the figure above, the encapsulation layer takes place between the MPEG layer and the IP layer. Its role at an RCST sender is to add a specific header to the IP packet received from the upper IP layer, and send the new packet to the lower layer (MPEG-2 layer). At an RCST receiver, this layer removes the added header from the received packet and sends the resulting IP packet to the IP layer.

The satellite protocol stack consists only in two layers: the link layer which is responsible of the processing of incoming and outgoing data segments and the DVB layer (DVB-S and DVB-RCS) which represents the physical layer. The on-board processor switches the received data segments to the destination ports (spot beams).

7.4 The MPE Encapsulation Scheme

As we have outlined in Section 7.2, the MPE (Multi Protocol Encapsulation) scheme is one among the three methods which are described by the DVB specification for data broadcasting [42] to carry voice, audio, and data in MPEG-2 transport segments. The data containers (called datagram sections) of the MPE method are optimized for carrying IP datagrams.

In this section, we first describe the MPE encapsulation scheme and then we outline its main limitations to be applied in the next-generation of GEO satellite systems.

7.4.1 Scheme Description

The MPE specification of DVB uses private sections for the transport of IP datagrams and uses an encapsulation, which is closely tailored after the IEEE LAN/MAN standards. Data packets are encapsulated in datagram_sections, which are compliant with the DSM-CC section for private data [42]. This encapsulation makes use of a medium access control (MAC) level device address. The address format conforms to the ISO/IEEE standards for LAN/MAN.

We illustrate, in Figure 7.4, how IP datagrams are encapsulated within DVB-MPE Datagram Sections and are then segmented into MPEG transport packets. IP datagrams must be

fragmented at the IP layer such that they do not exceed the specified Maximum Transfer Unit (MTU) for the payload portion of the DVB-MPE Datagram Section.

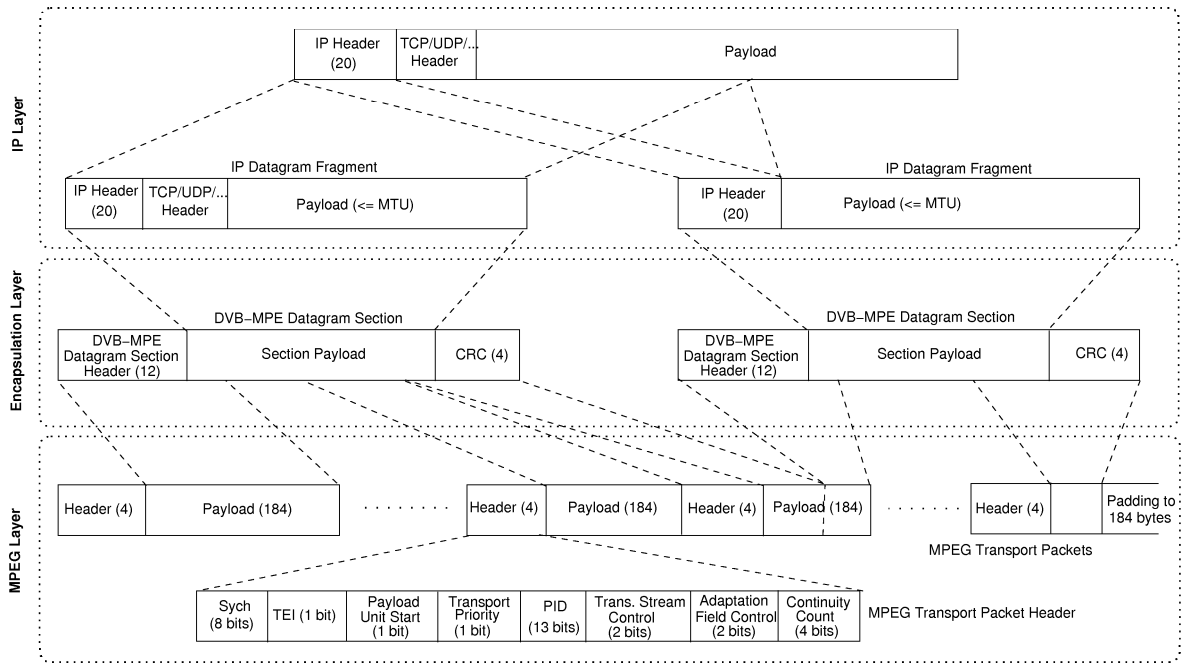


Figure 7.4: Carrying IP packets into MPEG segments using the MPE encapsulation approach

Devices delivering IP datagrams within DVB MPE Datagram Sections must segment datagrams such that they do not exceed a specific size called, the Maximum Transfer Unit (MTU) of the network. IP datagrams can be as large 65536 bytes[3] in length. Therefore, the source device must use IP layer fragmentation to breakup the large IP datagrams no larger than the MTU before passing them to the data link layer, where they will be encapsulated within DVB-MPE datagram sections. Likewise, when the DVB-MPE datagram sections are received by the decoder, the IP fragments will be passed to the IP layer software for re-assembly into a complete IP datagram.

7.4.2 Reasons To Not Use MPE

We identify three reasons whey we cannot use the MPE encapsulation method for the next-generation of GEO satellites systems. In fact, the MPE encapsulation scheme as presented above cannot fulfill the requirement of our system for three major reasons. First, the MPE scheme is not optimized for transporting IP packets given that not all the header fields added to each IP packet header are required for IP packets delivery. Second, the on-board satellite processor is not able to determine the session to which belongs the received MPEG data segment and then it is not able to forward it to the correct list of outgoing ports. Third, the filtering in the MPE layer is based only on the MAC destination address which is a mapping of the IP destination address. Or, a multicast session may be identified by the pair (source address, group address), for example when using the SPT mode of PIM-SM or for the SSM

[3]Decoders may limit the maximum IP datagram size due to internal buffering constraints.

service model. It is therefore impossible to handle the filtering at the lower layer (link layer) which makes the upper layer (IP layer) more complex and forces it to filter more packets.

Note that the IETF proposed "IP over DVB" working group[4] is designing a new encapsulation scheme for transmitting IP packets over DVB medium |28, 50|, however our work focus particularly on the IP multicast support over the new generation of GEO satellite systems.

To allow the RCSTs to filter incoming packets and to help the on-board satellite to switch incoming MPEG segments to the appropriate spots, we propose a new encapsulation layer, called *IP-Optimized Encapsulation Scheme* (see Figure 7.3), which will replace the MPE Layer used in the MPE encapsulation method. This layer will be specially implemented for satellite link use, however, it could be adapted in the future for the terrestrial network.

7.5 The IP-Optimized Encapsulation Scheme

The next-generation of GEO satellites are expected to use an on-board processor (OBP) with multiple spot beams. We study the IP packet delivery in this type of satellites and especially for multicast communication. As we have outlined earlier, the satellite receives MPEG data segments that should be processed on board and switched to the correct outgoing ports. Therefore, each data segment should contain an information that helps the on-board processor to handle the switching. The MPE encapsulation scheme detailed in Section 7.4 does not provide this kind of information and so it is not adequate for the new generation of GEO satellite systems.

Our goal is to design a new encapsulation scheme (instead of using the MPE method) between the IP level, and the MPEG2 level that allows the switching in the on-board satellite processor as well as the delivery of multicast packets. The design of this scheme should take into account the required tasks sharing between the satellite and the ground stations on handling IP multicast delivery over the next-generation of satellite-terrestrial networks. We describe two main approaches that could be applied to solve this problem:

- the **self routing** approach, is the fact of switching the MPEG data segments on the satellite based on an on-board switching table.
- the **label switching** approach, is the fact of switching the MPEG data segments on board the satellite based on a label included in the header of each data segment.

Each approach has to provide three main functionalities:

- **the mapping**: this functionality deals with the mapping of each IP packet at the RCST sender to a set of MPEG data segments and the information to include in order to allow the switching on-board the satellite. This function specifies also how the data segments should be reassembled at the RCST receiver.
- **the switching**: this is a functionality of the on-board satellite processor which describes the procedure used to switch the incoming MPEG segments to the appropriate outgoing ports (spot beams). The switching function should not depend on the technology used by the satellite to handle the switching: fully switched or IF/RF switching that we have described in Section 2.2.3.

[4] `http://www.erg.abdn.ac.uk/users/gorry/ip-dvb/`

- **the filtering**: this functionality deals with the filtering procedure of the incoming MPEG segments to a RCST Receiver. The lower layer (link layer) should forward to the upper layer (network layer) only the segments that belong to an IP packet which should be received by the terminal.

The mapping procedure should provide the required information to the switching and filtering functions. We detail in the following section the three functionalities of each approach.

7.5.1 The Self Routing Approach

The self routing consists on the use of a switching table maintained by the on-board satellite to forward the incoming data segments to the correct list of spot beams. This type of routing corresponds to the method used in the classical IP terrestrial networks. Indeed, routers use routing tables to switch the received IP packets to the correct outgoing interface(s). For these reasons, we call this approach the "self-routing" approach.

7.5.1.1 Mapping

The switching on-board the satellite based on the self routing approach requires that the identifier of the packet should be included by the RCST sender on each data segment. This could be possible through the direct mapping between IP addresses and PID identifiers in the MPEG2-TP data segment header [69] which allows to use the existing DVB cards without any modification or improvement. However, the PID is coded in 13 bits allowing only 2^{13} simultaneous connections to be mapped, which is not enough comparing with the available IP address growing. Moreover, different IP addresses will be mapped to the same PID which does not provide an efficient switching in the on-board satellite processor and makes impossible the efficient filtering at the MPEG layer. On the other hand, current RCSTs can only filter, with a hardware implementation, at most 32 simultaneous PIDs. This problem may be avoided using a software implementation of the data segments filtering module.

To overcome these problems, we propose a new encapsulation scheme that can be used instead of the DSM-CC (Digital Storage Media Command and Control) [70] scheme used in MPE. We suggest an encapsulation dedicated for satellite with on-board processor and multi-spots satellite. As shown in Figure 7.5, a new header is added to each IP packet. This header consists on four fields:

- *session_identifier*: the identifier of a multicast session.
- *type*: is not the Type of Ethernet packet, but a new Type field.
- *length*: contains on the length of the IP packet.
- *FEC*: included in order to guarantee the good transmission of the various fields of the new header.

The new packet, is segmented to several payload data which size is equal to 184 bytes. A standard MPEG header of 4 bytes is added at the MPEG layer to each data payload to compose a MPEG2-TP. The segments will be then sent successively to the satellite interface.

Given that the *session_identifier* field is coded in 3 bytes, this encapsulation allows the mapping of 2^{24} IP addresses simultaneously.

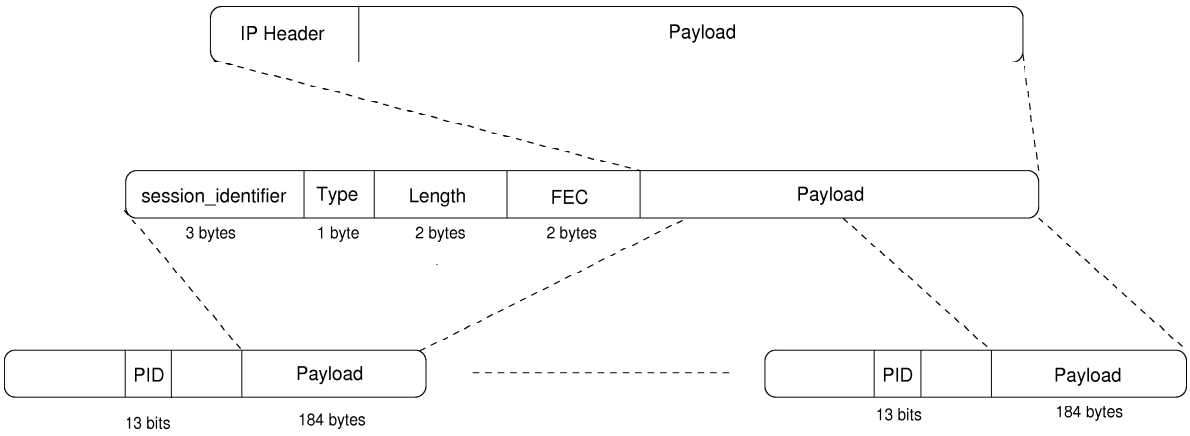


Figure 7.5: An IP-Optimized Encapsulation Method

Each RCST sender maintains a **mapping table** that gives the mapping between the IP packet identification and the *session_identifier* field. For example, Table 7.1 shows the mapping between IP multicast packets and the session_identifier. In this case, the identifier of the IP multicast packet is the pair (*@source*, *@Destination*).

(@source, @Destination)	**session_identifier**
$@s_1$, $@d_1$	ox000001
$@s_1$, $@d_2$	ox000002
$@s_3$, $@d_1$	ox000003
...	...
$@s_3$,$@d_2$	oxFFFFFE

Table 7.1: The mapping table on the RCST sender for the self routing approach

When receiving an IP packet from the upper layer to send through the satellite interface, the encapsulation layer of the RCST sender determines the corresponding session_identifier value from the mapping table to be included in the header.

7.5.1.2 Switching

When receiving a data segment from an RCST sender, the satellite should determine the list of spot beams to which it has to forward the received segment.

In order to handle the switching on board the satellite, it is necessary that the on-board processor keeps and updates two main switching tables. The first one is the Switching Table shown in Table 7.2 which gives for each *session_identifier* the list of destination spots. This table is updated by the NCC which periodically sends a message to the satellite that specifies for each session_identifier of an active session the list of destination spots. The same message is also broadcasted to all the RCSTs in order to update their own tables. The proposed protocol to handle these messages will be described in Section 7.6.

The second table, shown in Table 7.3, gives for each input port the PID and the list of destination spot beams of the current received data segment. This table is updated by the

session_identifier	Spots
ox000001	1, 2
ox0000E2	4, 5, 9
..	...
ox0FE002	11, 14, 18, 20

Table 7.2: Permanent switching table on the satellite

satellite when it receives a data segment that belongs to a new IP packet from an input port. Form this first data segment, it determines from the payload (the first 3 bytes) the session-identifier value of the packet and the PID and copies the corresponding list of destination spots from the second column of the permanent switching table (Table 7.2) to the third column of the temporary switching table (Table 7.3). The flowchart of the algorithm used to update this table is shown in Figure 7.6.

It can be noted that the first table is quite permanent, while the second one may vary on each reception of a data segment that belongs to a new IP packet.

Input Spot	PID	Output Spots
1	pid 2	1, 2
2	pid 7	4, 5, 9
...	...	...
32	pid 20	11, 14, 18, 20

Table 7.3: Temporary switching table on the satellite

7.5.1.3 Filtering

Each RCST Receiver maintains a table that we call, the **subscription table** as shown in Figure 7.4. This table consists in three columns. For each session_identifier value, given in the first column, it indicates the IP identification (for example (@IP source, @IP destination) pair for IP multicast connections) of the corresponding IP packet in the second column. As we have outlined earlier, this mapping is done by the NCC for all active sessions and it is periodically broadcasted to all the RCSTs. The third column of the subscription table specifies the session state, which is very useful to reduce the number of data segments sent from the MPEG layer to the encapsulation layer at the receivers and so the number of IP packets sent to the network layer. This column is set by the RCST receiver itself when receiving PIM-SM Join and Prune messages from its downstream members.

The field "State" could have the following values:

- ***Standby***, informs that this session is accessible (i.e.; there is at least one source sending to the corresponding multicast group) but there is no subscriber for this session downstream to **all RCSTs**. This state is the initial state of each new entry created by the RCST when receiving an updated information from the NCC.
- ***Available***, informs that this session is active but all the members belong to spot-beams other than that to which belong the RCST.

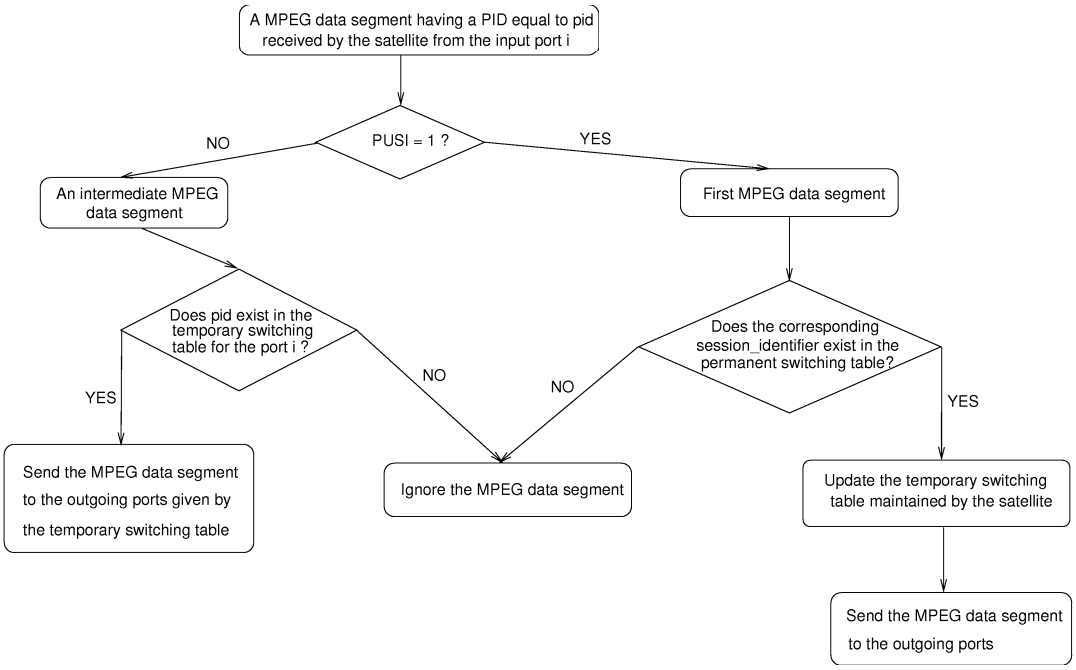


Figure 7.6: Algorithm of generation of the switching table in the on-board satellite processor

session_identifier	**(@source,@IP Destination)**	**State**
ox000002	$@s_1$, $@d_1$	Standby
oxFFFFFE	$@s_1$, $@d_2$	Available
ox000001	$@s_2$, $@d_1$	Active
...	...	...
ox000004	$@s_3$, $@d_2$	Online

Table 7.4: Subscription table on the RCST receiver

- ***Active***, means that this session is accessible via the satellite network in the spot-beam of the RCST. In other words, this RCST is receiving data from this session but there is no member downstream to this RCST.

- ***Online***, informs that this session is requested and the received data segments should be forwarded to upper layers in order to be re-assembled and sent to the terrestrial interfaces where there are downstream members.

The subscription table is maintained by the link level layer (MPEG layer). A specific simple protocol between the network and the link layer could be designed to update the third column of this table by the network layer. This protocol is out of the scope of the research described in this chapter.

To allow the filtering of incoming data segments at the MPEG layer, each RCST maintains at the MPEG level the list of current accepted PIDs as shown in Table 7.5.

List of Accepted PIDs
pid_5
pid_20
...
pid_n

Table 7.5: The filtering table of the PIDs at an RCST

The flowchart of the filtering algorithm used to filter the incoming data segments at the MPEG layer and to update the list of accepted PIDs is given in Figure 7.7. As we can see, a data segment will be sent to the upper layer only if it concerns a session which belongs to the set of active sessions with a state "Online" in the **subscription table** of Figure 7.4. In this case, the PID will be added to the filtering table if it has not already been added.

7.5.2 The Label Switching Approach

The label switching approach consists in adding an information to each data segment in order to help the on-board satellite processor to switch it to the correct list of the outgoing destination ports without using an on-board switching table as the self-routing approach does.

7.5.2.1 Mapping

The mapping for the label switching is the same as that of the self routing, except that we add a new field called **switching_label** to each MPEG data segment in order to encode the list of destination ports of this data segment. The size of this field depends on the number of spot beams that the satellite supports. For example, with a filed of 4 bytes, it is possible to support the label switching approach for 32 spot beams. Each bit in the field "switching_label" concerns a single outgoing port on satellite linked to a specific spot beam. This field will be removed by the satellite before forwarding it to the outgoing ports in order to allow the use of current MPEG-2 drivers.

As shown in Figure 7.8, we still using the field session_identifier that we have proposed for the self routing approach detailed in the previous section.

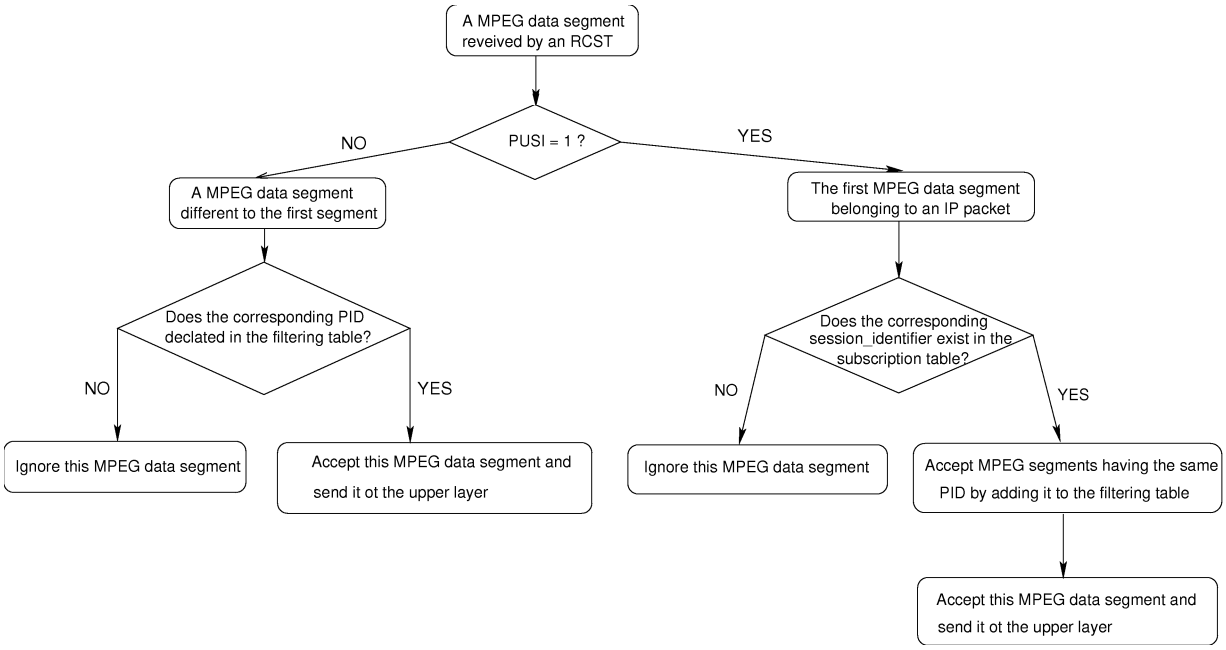


Figure 7.7: The Filtering algorithm at an RCST receiver

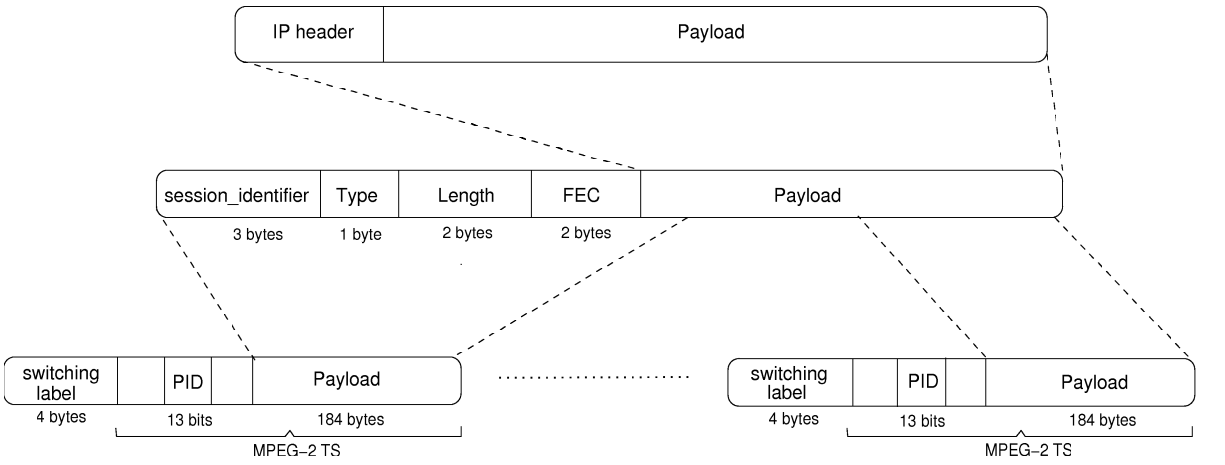


Figure 7.8: The IP-optimized encapsulation for the label-switching approach

Other fields have the same role as when using the self routing approach. As it is presented in Table 7.6, the mapping table on the RCST sender, will be extended by adding the switching_label field.

(@source, @Destination)	session_identifier	switching_label
@s_1, @d_1	ox00001E	ox00403F10
@s_2, @d_1	ox1F00A2	ox00400410
@s_1, @d_3	ox00B0C3	ox90403180
...	...	...
@s_n, @d_m	oxA0F270	ox00010001

Table 7.6: The mapping table on the RCST sender for the label switching approach

Before sending each data segment, the RCST sender should add the corresponding switching_label and other fields. The value of the switching label is determined from the information multicasted by the NCC, which gives the list of destination spots of each active session identified by its session_identifier as for the case of the self routing approach.

7.5.2.2 Switching

The switching on the on-board satellite processor is based on the switching_label field included in each data segment. Therefore, there is no need to keep a switching table as for the case of the self routing approach.

For each incoming data segment, the satellite has to read bit-per-bit the switching label inserted by the RCST sender in the head of each MPEG-2 header. When a bit is set to 1, it means that the satellite should forward the data segment to the corresponding port.

7.5.2.3 Filtering

The filtering at the RCST Receiver is exactly the same as the case of the self routing approach given that the formats of the packet that arrives to the RCST receiver are similar for both approaches (see Section 7.5.1.3). Indeed, as we have pointed out earlier, the satellite removes the switching_label field before forwarding each data segment to the corresponding outgoing ports.

7.5.3 On-Board Switching Approaches Comparison

7.5.3.1 Overhead

Let S be the size of the IP packet. We define the overhead of each approach as the amount of header added both by the encapsulation scheme and the MPEG layer over the size of the IP packet. Hence, the overhead of the two approaches is computed as follows:

- the self-routing approach: there are 8 bytes added by the encapsulation layer to the IP packet and 4 bytes added by the MPE layer for each MPEG-2 payload (184 bytes). The overhead of the self-routing approach is therefore equal to $\frac{\frac{S+8}{184}*4+8}{S}$.

- the label-switching approach: there are 8 bytes added by the encapsulation layer to the IP packet, 4 bytes added by the MPE layer for each MPEG-2 payload (184 bytes), and 4 bytes of the switching label field to each MPEG-2 data segment. The overhead of the label-switching approach is therefore equal to $\frac{\frac{S+8}{184}*8+8}{S}$.

We show in Figure 7.9, the variation of the overhead of the self-routing and the label-switching approaches in function of the IP packet size[5]. As we can see, the overhead of both approaches decreases with IP packet size, while still less than that added by the MPE traditional encapsulation scheme which has an overhead between 13 and 15 per cent [27]. Furthermore, the label-switching approach adds more overhead than the self-routing approach.

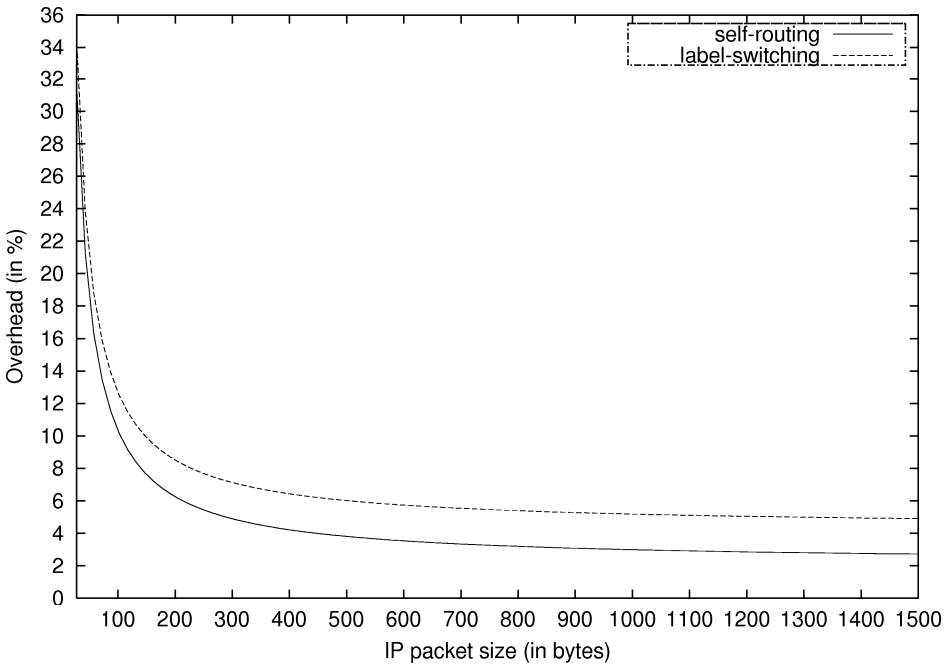


Figure 7.9: Overhead comparison of the label-switching and the self-routing on-board satellite switching approaches

7.5.3.2 Complexity

We show in Table 7.7, the complexity of the self-routing and the label switching approaches. From this comparative table, we can conclude that from the economic point of view, we can say that the self-routing approach is bandwidth-conservative in the satellite uplink than the label-switching approach and from the technical point of view, the label-switching approach is simpler than the self-routing approach. Indeed, in the former case, the design of the GEO satellite becomes more simple and it is much more easier to process incoming data segments given that the check of the switching label and the switching to the outgoing spot beams can be implemented in a remote programmable switch. Note also that both approaches consume

[5]The minimum IP packet size is equal to 28 bytes (20 bytes of the IP header and 8 byte of the LLC header).

	Self Routing	Label Switching
Complexity of the RCST	medium	high
Complexity of the Satellite	high	little low
Expected Terminal Cost	medium	medium
Feasibility	yes	yes
Compatibility with MPE DVB cards	no	no

Table 7.7: Self routing vs. label switching approach

the same bandwidth in the satellite downlink given that the label switching field is removed from the MPEG-2 data segment before being forwarded to the satellite receivers.

7.6 The SMRP Protocol

In this section, we describe SMRP (Satellite Multicast Routing Protocol) that improves the forwarding of the multicast packets on the satellite. First, we describe a new DVB descriptor which will be used by the NCC to announce the multicast active sessions to the RCSTs. Then, we discuss how the RCST members join and leave the multicast sessions. Finally, we detail the processing of the PIM-SM messages received by the RCSTs in order to optimize the delivery of multicast data in the satellite segment.

7.6.1 The Multicast Session Descriptor

Our approaches described earlier namely: the self routing and the label switching use the information broadcasted from the NCC to maintain and update the subscription table on the RCST receiver (see Table 7.4) and the mapping table on the RCST sender (see Table 7.1 and Table 7.6). In the DVB standard systems, the NCC periodically sends signaling tables that manage some functionalities such as the PIDs allocation, the frequency usage, etc. Each information is described in a specific descriptor. In order to provide an efficient support of multicast applications in the new generation of satellite systems, we propose to add a new descriptor to periodically announce the multicast session and to send the information required by the different components of the system (the on-board processor, the NCC, and the RCSTs)

We call our new descriptor *session_descriptor* which is shown in Table 7.8.

As we can see, there are seven fields, where every one can be used either by the satellite, or by the RCSTs, or by both of them. These fields are:

- *session_identifier*, as we have explained above, it is a label which identifies the session concerned by the descriptor.
- *switching_label*, is used by the satellite in case of using self routing approach in order to build the switching table of Table 7.2 and it used by the RCST senders in the case of the label switching approach in order to insert this label in the switching_label field at the head of each MPEG data segment.
- *address_type*, indicates which IP version is used. It is equal to 0 when using IPv4, and to 1 for IPv6.

Syntax	Number of bits
Session_ Descriptor() {	
descriptor_tag	8
descriptor_length	8
session_identifier	24
switching_label	32
address_type	1
source_address	32/128
group_address	32/128
}	

Table 7.8: Session Descriptor format

- *source_ address* and *group_ address*, are the multicast group IP address and the multicast sender unicast address, respectively. They may be encoded in 4 or 16 bytes depending on the IP version given by the *address_ type* field.

The *session_ descriptor* descriptor is included in each BAT (Bouquet Allocation Table) [41, 42, 43] according to the content of DULM messages.

The BAT is an optional table within the DVB standard [41, 43]. We choose this table to enable the use of our approach in the satellite systems because it is the only table in the standard which informs about of the general state of the satellite network.

The NCC keeps all information about the active multicast sessions and periodically informs the RCSTs about the active sessions using the session_descriptor. When there is a change in the state of a given session[6], it immediately broadcasts the corresponding *session_ descriptor* (see Table 7.8), within the BAT [41, 43], with a PID equal to 0x0011 according to the DVB standard [43, 45].

When receiving the BAT table from the NCC, an RCST receiver checks, in addition to the standard fields defined in the DVB-RCS standard [41], the source_address, group_address, and switching_label fields and it updates its subscription table (Table 7.4).

7.6.2 Using DULM Messages

To handle signaling messages between the NCC and the RCSTs, we use two new IE (Information Elements) types [41], which are defined in Table 7.9. The first one (id = 0x0E) will be used by the RCSTs senders, and the second one (id = 0x0F) by the RCSTs receivers. These messages are sent using a PID equal to CTRL_MNGM_PID which is allowed to be used by every RCST in the connection phase.

Following the standard specifications of DULM (Data Unit Labeling Method) messages [41], we propose a specific format for both messages.

In Figure 7.10, we show the format of the DULM New Session message. We added three main new fields to those already defined in the DVB-RCS [41] standard:

- the field *@Type*, defines the type of the IP addresses, equal to 0 for IPv4 and 1 for IPv6.

[6]For example a change on the *switching_ label* value.

Identifier	IE type	Usage
0x0E	New Session	Used by an RCST Sender to announce new session
0x0F	Join / Leave Session	Used by an RCST Receiver to join/leave one or more sessions

Table 7.9: The new types of Information Elements (IE)

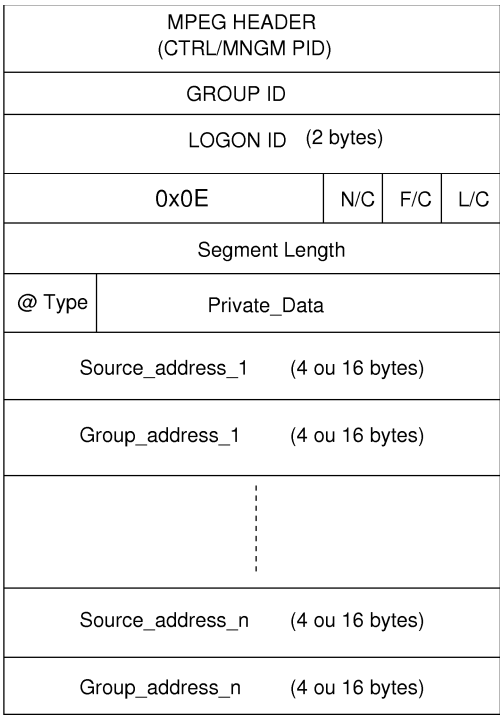


Figure 7.10: The format the DULM New Session message

- the field *Private_ Data*, is a reserved filed for specific utilization for each operator.
- the fields *source_ address_ n* and *group_ address_ n*, define the addresses that determine the new flow of session *n*.

Thanks to this message, every RCST sender informs the NCC about the new active sessions having the source behind him. Periodically, the RCST sender transmits a *DULM New Session* message as shown in Figure 7.10 toward the NCC in order to inform it about the current multicast active sessions. When receiving a *DULM New Session* message, the NCC updates its **multicast sessions table** (MST), which describes the set of all multicast active sessions sending to the satellite network. In addition, the NCC assigns a new identifier *session_ identifier* and adds the *session_ descriptor* (see Table 7.8) to the the BAT to be broadcasted to all the RCSTs.

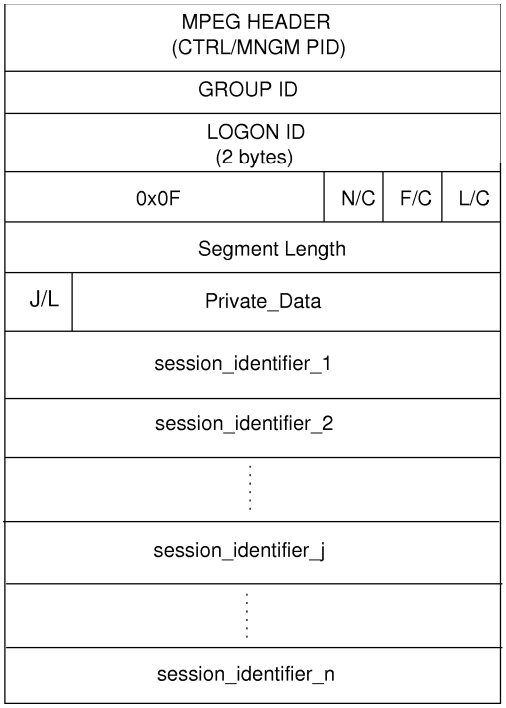


Figure 7.11: The format of the DULM Join Leave Session message

In Figure 7.11, we show the format of the DULM Join/Leave session message. There are three new fields added to those already defined in the DVB-RCS |41| standard:

- the bit *J/L* (Join/Leave) specifies whether the request concerns a joining or a leaving action for the listed sessions.
- the field *session_ identifier_ n* gives the identification of the multicast session (S_i, G_j), defined by a source S_i and group G_j.

This message is sent by the link layer of each RCST that receives a new PIM-SM Join/Prune from one of its terrestrial interfaces to the NCC only if the switching_label of the corre-

sponding session (if it exists) in its switching table does not contain the spot to which it belongs.

7.6.3 Handling PIM-SM messages

In this section, we present the different operations that the system entities have to do when receiving PIM-SM signaling messages. Depending on the routing approach (self routing or label switching) used and other factors, the RCSTs, the satellite, and the NCC, handle differently these messages. In the following sub-sections, we detail the processing of these messages.

7.6.3.1 Sending PIM-SM Join message

When an RCST receives a PIM-SM join message from its terrestrial interface, it first checks whether the corresponding session is active or not by consulting its subscription table. If it is not the case, the RCST simply ignores the message. In contrast, if the session is active, it verifies if the session is already broadcasted to the spot beam to which it belongs and this by checking the switching_label value. If it is the first RCST in its spot beam which joins the session, it informs the NCC, that it is interested by this session. In other words, this RCST receiver sends a *DULM Join/Leave Session* message (see Figure 7.11) with J/L bit set to 0, to the NCC in order to update the switching_label value in its multicast sessions table. As explained in Section 7.6.1, the NCC broadcasts the new session_descriptor of the corresponding session to the RCSTs.

For the self-routing approach, the same descriptor is also sent to the satellite through a specific input port in order to be able to update its switching table (see Table 7.2), which provides for each *session_identifier*, the *switching_label* value.

The RCSTs that receives the broadcasted message from the NCC, should update the entry of the corresponding session in their subscription tables. When an RCST detects that the session being announced is already declared in its mapping table, it concludes that the source is behind him and allows further multicast packets of this session to be sent to the satellite interface. At the same time, it updates the switching_label value in its mapping table when using the switching label approach. Recall that for the self routing approach, in each MPEG2 packet, the RCST specifies only the *session_identifier* identifier, while for the label switching approach, it must also specifies the *switching_label* for each packet.

7.6.3.2 Receiving PIM-SM Join messages

An IGMP Membership Report [22] is sent by an end-host (the receiver) to its DR in order to join one or more multicast sessions. When receiving this message, the receiver's DR, sends a PIM-SM Join including the list of sessions to be joined, to the Rendez-vous Point (RP)[7]. We consider the case when the path toward the RP include the satellite segment. In other words, there is an RCST that will receive the multicast packets from the satellite and forward them to the end-host.

When receiving a PIM-SM Join message, this RCST consults its subscription table (see Table 7.4) and the action to do depends on the value of the state of each session that the downstream end-host wants to join.

[7]The "optimal" configuration of the RPs is such system is out of the scope of this chapter.

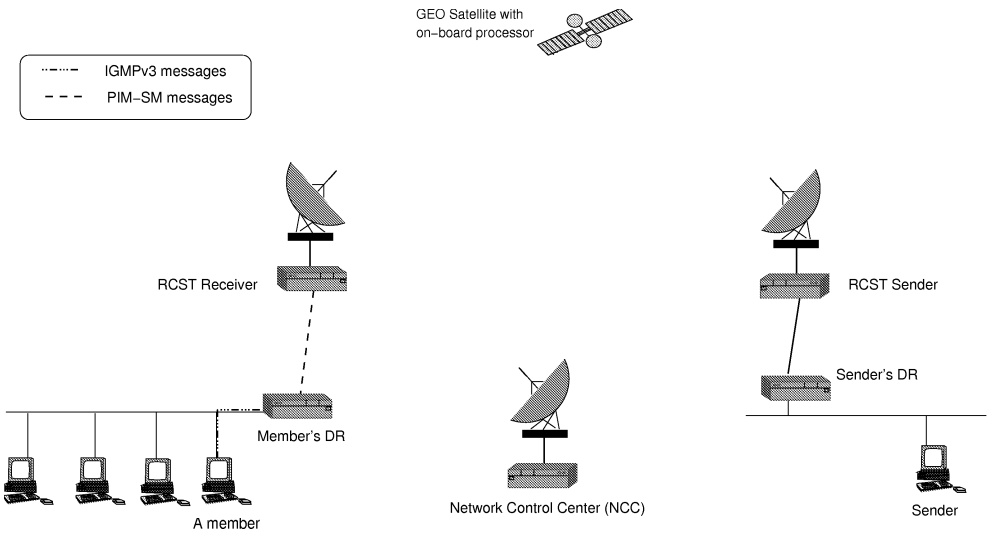


Figure 7.12: Transmission of PIM-SM Join message

If the state is *Active*, then it modifies it to *Online* and it starts to take into account the corresponding session_identifier. In other words, it will not forward the PIM-SM join message to the RP because it is already receiving the session data from the satellite segment given that there is at least one another RCST which belongs to the same spot beam who is subscribed to the same session. It has only to accept the MPEG data segments at the link layer and forward them to the IP-Optimized encapsulation layer.

If the state is *Standby* or *Available*[8], the RCST sends to the NCC a *DULM Join/Leave* message (see Figure 7.11), and as soon as it receives a confirmation of subscription from the NCC (see Table 7.8)[9], it turns the state to *Online*.

In the case when the session does not exist in the subscription table, this means that there is no multicast source sending packets to the requested multicast group using the satellite network because there is no PIM-SM Join message received by the RP from the satellite segment. In this case, we allow the RCST to forward the received PIM-SM Join message toward the RP through the satellite network. The RP will receive this join from another RCST which will forward after the multicast packets to the members through the satellite link. At the same time, this RCST sender (to which is behind the source) sends a *DULM Join/Leave* message (see Figure 7.11) to the NCC which will attribute a *session_identifier* to the new session and creates a new entry in the subscription table. All the bits, except that of the spot beam to which belongs the RCST receiver who has sent the join message, of the *switching_label* value will be set to 0 and the state to *Online*. The NCC broadcasts the corresponding session descriptor (see Table 7.8) to all the RCSTs and the RCST receiver creates a new entry in its subscription table that describes the new multicast session.

[8]Which means that may be other members but they belong to other spot beams.

[9]The NCC will switch the bit of the spot beam of the RCST to 1 in the switching label in order to duplicate the data segments on-board the satellite of this spot beams.

7.6.3.3 Receiving PIM-SM Prune Messages

The RCST receiver who receives a PIM-SM Prune from one of its terrestrial interface, should first check if there are other receivers interested by the session reached from other interfaces[10]. If not, it sends to the NCC a *DULM Join Leave Session* message (see Figure 7.11) and sets the J/L bit to 1 to indicate that it is a Leave message.

When the RCST which has just left the session is the last member in the spot-beam[11], all the system entities should update some information.

The NCC updates the *switching_label* field of the pruned session by setting the corresponding bit to 0, so as the packet will not any more be forwarded to this spot beam.

When using the self-routing approach, the RCST sender will be concerned by this change as long as the *switching_label* is different to 0x0000, and they should stop sending data to the satellite link. In the case of using the Label Switching approach, they must update the switching_label fields of packets corresponding to the session in question (see Table 7.6) in order to avoid that the satellite forwards the multicast to the spot being removed from the list of destination spots.

For the satellite, in the case of the self routing, it must update its switching table (see Table 7.6), while for the Switching Label approach, there is nothing to change.

7.7 Chapter Summary

Two different approaches that provide an efficient multicast delivery in the next generation of satellite systems have been proposed in this first part of this chapter. The self-routing approach is based on maintaining a switching table on board the satellite, while the label switching approach includes from a label in each data segment to allow the on-board satellite processor to switch it to the corresponding outgoing ports (spot beams).

In the second part of this chapter, we described a new protocol called SMRP (Satellite Multicast Routing Protocol) which is an adaptation of PIM-SM for the the new generation of GEO satellite networks.

Within the exciting research area of the multicast delivery over the next-generation of satellite systems, there are several future works that could be investigated as an extension of our work described in this chapter. One possible extension is the comparison of the two switching techniques that we have proposed (self-routing and label-switching approaches) in terms of complexity and efficiency using a real test-bed networks. Another area of research work that could also be examined, is the compatibility issues with the DVB cards that use other kind of encapsulation other than the IP-optimized encapsulation method described in this chapter. These issues will be, in fact, studied in future working packages of the DIPCAST RNRT project.

[10] The list of outgoing interfaces is given the `oif` field of the corresponding session in the corresponding entry in the multicast routing table maintained by PIM-SM.

[11] A mechanism similar to that described in Section 6.5.4.1 for the DIPCAST system could be designed for the new-generation of GEO satellite systems to detect when the last member has left the session.

Chapter 8

An Enhanced PIM-SM Switching Mechanism

8.1 Introduction

Along this dissertation we have mainly considered the PIM-SM protocol as the reference multicast routing protocol given that it is probably the most widely used multicast routing protocol today [35, 40]. Indeed, in Chapter 5, we have described the specifications for integrating the counting of group members in PIM-SM, while in Chapter 6 and Chapter 7, we have studied the support of PIM-SM in GEO transparent satellites and next-generation GEO satellites, respectively.

In this chapter, we emphasize on the issue of switching between the two modes provided in PIM-SM protocol. Indeed, PIM-SM creates a shared, RP-rooted distribution tree that reaches all group members and it authorizes the receivers to switch from a RP (Rendez-vous Point)-rooted tree (RPT) to a shortest path tree (SPT), however it doesn't specify how the switching policy should be done. The recommended policy in PIM-SM specification is to initiate the switch to the SP-tree after receiving a significant number of data packets during a specified time interval from a particular source. This heuristic for determining when to migrate from the RPT to the SPT, and vise versa, is far from being satisfactory. There is no efficient provision in PIM-SM for migrating from these two modes. Individual routers make policy decisions as when to change the routing type for a given source.

In Chapter 6 (Section 6.4.2), we have proposed a switching configuration policy for enhancing the multicast delivery over satellite-terrestrial networks and another one (Section 6.5.3) for the specific DIPCAST GEO transparent satellite system. In this chapter, we propose a switching mechanism which does not depend on the transmission medium used that could be either terrestrial, satellite or another type of meduim.

To the best of our knowledge there is only one prior work describing an enhanced PIM-SM switching mechanism in the open literature. Indeed, the authors of [68] have proposed an extension of PIM-SM called PIM-Switch which is based on the estimation of the density of the multicast group in the network. However, PIM-Switch has three main drawbacks. Firstly, it proposes the exclusive use either of RPT or SPT for all receivers[1]. Secondly, it does not

[1]Throughout this chapter we mean by a receiver, the designated router of at least one host connected to one of its directly-attached LANs and which is a member of a multicast group. We use the terms "receiver" and "designated router", interchangeably.

take into account neither the QoS requirements of receivers nor the network characteristics. Finally, there is no coordination between receivers to decide when and how to switch between the two modes of PIM-SM.

We believe that the use of a mechanism based on the coordination between the concerned receivers to switch between the RPT and the SPT will be more efficient and can fulfill to PIM-SM original intentions and make it more efficient comparing to other routing protocols and especially CBT [10].

The design of an efficient switching mechanism involves three major parts. The first part is the development of a decision algorithm that allows receivers to decide when requesting the switching from the RPT to the SPT and vise versa. When such decision is done, the second part involves the acceptance of the switching request by routers belonging to the path toward the source. This acceptance should take into account not only the receiver QoS requirements but also the network parameters such as traffic concentration and network resource usage. The third part deals with the mechanism specifications for making the switching decision and its integration in PIM-SM protocol.

In this chapter, we explore the three parts. Clearly, we would like to improve the performance parameters that are of particular importance to the network as well as to the receivers.

The PIM-SM switching mechanism proposed in this chapter aims to achieve the receivers requirements without influencing the network conditions. It is a coordination-based mechanism in the sense that all concerned receivers contribute to the switching decision and not only the receiver requesting the switching. Furthermore, the mechanism could use information about the temporary available network resource provided by an underlying QoS-based unicast routing protocol or using periodic switching experiments to decide when and how to switch between the two modes based on the QoS constraints.

We study and compare using simulations the tradeoff relationship between the complexity and the effectiveness of the proposed switching mechanism. Simulation results show that our mechanism achieves the inter-receiver fairness and that its integration in PIM-SM provides efficiently its original intentions.

The remainder of this chapter is structured as follows. We start by giving, in Section 8.2, a preliminary background of this work by giving some details about the operations of PIM-SM protocol and mentioning related work. In Section 8.3, we present a motivation example illustrating the drawbacks of using the switching mechanism described in the standard PIM-SM. Some preliminaries, assumptions and terminology will be presented in Section 8.4. In Section 8.5, we describe network and receiver-related parameters that should be taken into account during the design of our switching mechanism. We provide a short overview of our proposal in Section 8.6. Section 8.7 details our switching mechanism and the new PIM messages used to handle the switching between the RTP and the SPT. To understand the operations of our scheme, we describe in Section 8.8 a concrete example that shows when and how the switching decision is done. The eligibility tests which are done by the intermediate routers will be detailed in Section 8.9. In Section 8.10, we give the necessary specifications of our protocol to be integrated in PIM-SM protocol. The performance evaluation of the mechanism will be the subject of Section 8.11. Finally, Section 8.12 concludes this chapter.

8.2 Background and Related Work

Similar to the CBT protocol [10], PIM-Sparse Mode (PIM-SM) [40] is designed to restrict multicast traffic to only those routers interested in receiving it. PIM-SM constructs a multicast distribution tree based on a router called a rendezvous point (RP). This rendezvous point plays the same role as the core in the CBT protocol; receivers "meet" new sources at this rendezvous point. However, PIM-SM is a more flexible protocol than CBT. While CBT with trees are always group-shared trees, with PIM-SM an individual receiver may choose to construct either a group-shared tree (also called RPT - Rendez-vous Point Tree) or a shortest-path tree (SPT) hence the name Rendez-vous Point.

The PIM-SM protocol initially constructs a group-shared tree to support a multicast group. The tree is formed by the senders and receivers both connecting to the rendezvous point, just as a shared tree is constructed around the core with the CBT protocol. After the tree is constructed, a receiver (actually the router closest to this receiver) can opt to change its connection to a particular source to a shortest-path tree. This is accomplished by having this router send a PIM join message to the source. Once the shortest path from source to receiver is created, the extraneous branches through the RP are pruned. This procedure is illustrated in Figure 8.1. Note that different types of trees can be selected for different sources within a single multicast group.

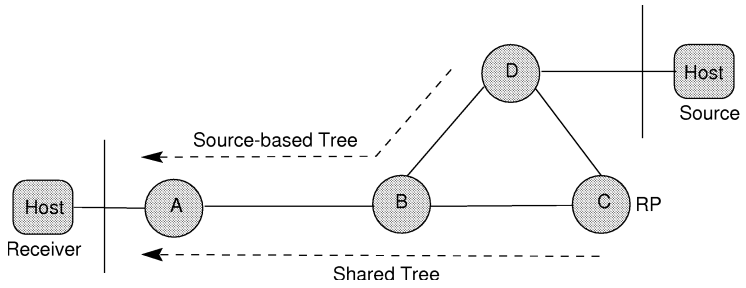


Figure 8.1: Example: Switching from shared tree to shortest path tree

There are advantages to each type of distribution tree. The shared tree is relatively easy to construct, and it reduces the amount of state information that must be stored in the routers. Accordingly, a shared tree would conserve network resource (in term of maintained states in routers) if the multicast group consisted of a large number of low-data-rate sources. However, as indicated above, shared trees cause a concentration of traffic around the core or the rendezvous point, a phenomenon that can result in performance degradation if there is a large volume of multicast traffic. Another disadvantage of shared trees is that traffic often does not traverse the shortest path from source to destination. If low latency is a critical application requirement, it would be preferable for traffic to be routed along a shortest path. PIM-SM architecture supports both types of distribution trees.

Let us explain the switching method recommended in PIM-SM standard. Recall first that a PIM-SM router has at most three entry types in its multicast routing table:

- A (*,*,RP) entry: a special entry type to support interoperability which must be supported by all PIM routers. A data packet will match on a (*,*,RP) entry if there is no more specific entry (such as (S,G) or (*,G)) and the destination group address in

the packet maps to the RP listed in the (*,*,RP) entry. In this sense, a (*,*,RP) entry represents an aggregation of all the groups that hash to that RP.

- A (*,G) entry: a wildcard multicast route entry for the group.
- A (S,G) entry: a multicast route entry that is specific to the source.

The switching policy from the RPT to the SPT that is recommended in the PIM-SM specification [40] is rate-based. The receiver initiates the switching to the SPT after receiving a significant number of data packets during a specified time interval from a particular source. When a (*,G), or corresponding (*,*,RP), entry is created, a data rate counter may be initiated at the last-hop routers. The counter is incremented with every data packet received for directly connected members of an SM group, if the longest match is (*,G) or (*,*,RP). If and when the data rate for the group exceeds a certain configured threshold t_1, the router initiates 'source-specific' data rate counters for the following data packets. Then, each counter for a source, is incremented when packets matching on (*,G), or (*,*,RP), are received from that source. If the data rate from the particular source exceeds a configured threshold t_2, a (S,G) entry is created and a Join/Prune message is sent towards the source. If the RPF interface for (S,G) is not the same as that for (*,G) - or (*,*,RP), then the SPT-bit is cleared in the (S,G) entry.

8.3 Motivation Example

A receiver's DR that wants to switch to the SPT should send a PIM (S,G) Join message toward the source. All receivers in the SP-subtree[2] will receive data from the SPT. The problem is that the QoS received by other receivers in the SP-subtree can be violated when a receiver switches from the RPT to the SPT without coordinating with them.

To establish the limitations of the PIM recommended switching technique, let us consider the topology shown in Figure 8.2, where there is a multicast source S sending to four receivers R_1, R_2, R_3, and R_4 via the RP-rooted tree (RPT). The capacity of the link between the designated router of receiver R_3 toward the RPT is equal to 36 kbps, while that of receiver R_4 is equal to 128 kbps.

We examine the impact of the PIM recommended switching mechanism on the application performance. We distinguish two cases: when the application uses a single-rate congestion control mechanism such as TFMCC protocol described in [119] and when it uses a layered transmission scheme such as WEBRC [79]. We assume that the designated router of receiver R_3 (router Rt_7) decides to switch from the RPT to the SPT.

For single-rate multicast sessions, the source sends at the rate of the slowest receiver which is the receiver R_3 in our example. When switching to the SPT, the bottleneck in the tree will be the link between the source and router Rt_4 which has a capacity equal to 18 kbps. As a result the sending rate will be reduced because of the higher loss rate in this bottleneck. If the receiver R_3 knows in advance the resulting rate, it will not switch to the SPT.

For multi-rate sources, we assume that the source S uses three layers, layer 1 with a rate equals to 9 kbps and sending to multicast group G_1, layer 2 with a rate equal to 18 kbps and

[2]The SP-subtree is the part of the multicast tree from the SP, a specific on-tree router that will be defined in Section 8.4. It is the common part between the SPT and the RPT from the receiver initiating the switching, referred hereafter as the SIR (Switching Initiator Receiver).

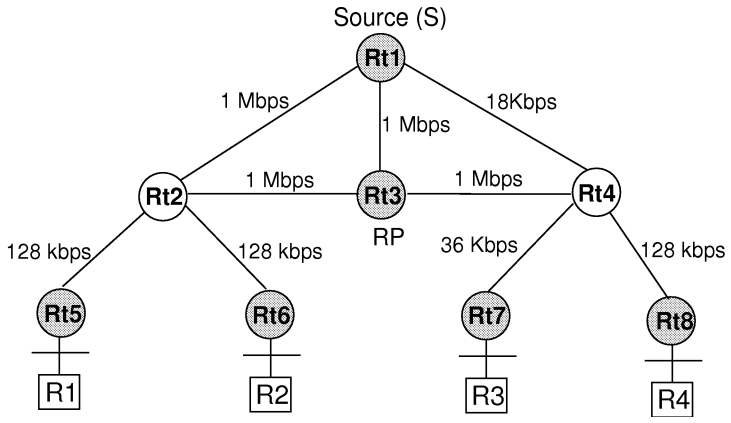


Figure 8.2: Motivation sample. The RP-rooted tree here is is composed of the two sub-trees $\{Rt_2, Rt_5, Rt_6\}$ and $\{Rt_4, Rt_7, Rt_8\}$ which are both connected to the Rendez-vous Point Rt_3.

sending to group G_2, and layer 3 with a rate equal to 36 kbps and sending to group G_3. We suppose that all group addresses are mapped to the same RP, router Rt_3 which is in over-provisioned place. We assume that all receivers except R_3 are already subscribed to the three layers. Their aggregated receiving rate is then equal to 9+18+36 = 63 kbps. Receiver R_3 can subscribe only to the two first layers and it switches to the SPT. As consequence, receiver R_4 will detect a high loss rate and it will be obliged to leave layer 2 and layer 3 which decreases considerably the quality of received data.

8.4 Assumptions and Terminology

We represent a network by a weighted diagraph (V, E), where V denotes the set of network nodes and E the set of communication links that connect the nodes. $|V| = N$ and $|E| = M$ denote the number of nodes and links in the network, respectively. $P(u, v)$ represents the set of routers constituting the path between u and v (including u and v).

Associated with each link are parameters that describe the current status of the link, for example the average link delay, the link cost, the loss rate, and the bandwidth available on the link. We term these parameters *link state*. Similarly, associated with each node are parameters representing the current status of the node, for example routing tables. We term these parameters *node state*. The set of links states and nodes states constitutes the *network state*. We assume that all routers in the network support PIM-SM [40] multicast routing protocol.

During a multicast session, we assume that each receiver controls the quality of reception parameters and have a knowledge of the required values that would be accepted. For example, a receiver R should specify R_R, P_R, D_R referring to the minimum data rate, the maximum loss rate, and the maximum delay, respectively. These parameters values may differ from receiver to another.

Each receiver maintains the values of the data QoS parameters received from the multicast tree, we denote by $R_R(RPT)$, $P_R(RPT)$, and $D_R(RPT)$ the current data rate, loss rate, and delay from the RPT, respectively. Depending on the nature of the multicast application, one

or more QoS parameter may be more or less important measure of the reception quality.

The receiver that decides to switch to the SPT, hereafter called the SWITCHING-INITIATOR RECEIVER (SIR), has to send a switching request toward the source. In its request, the member specify the set of QoS parameters violated and their current and required values.

We use the term SWITCHING-POINT (SP) of a receiver R to refer to the last router in the path between the receiver and a specific source S which belongs both to the RPT and the SPT. This router could also be defined as the first router between the receiver and the source which uses different upstream interfaces to reach the RP and the source.

Therefore, for a given group G, for each receiver R and source S that belongs to G, there is only one switching point. A receiver may have the same SP to two different sources and two receivers may have different switching point to the same source. The SP may be the source itself when the RP belongs to the shortest path between the source and the switching receiver.

For example, for the multicast group shown in Figure 8.2, router Rt_2 is the SP of receivers R_1 and R_2 and router Rt_4 is the SP of receivers R_3 and R_4.

8.5 Switching Parameters

As outlined in |40|, the switching to the SPT can be initiated either by the RP or by designated routers. Different criteria could be applied to trigger the switching from the RPT to the SPT. These criteria may be the QoS requirements and/or the network parameters. In this section, we detail both types of parameters.

8.5.1 QoS Parameters

Certainly that the first requirement of receivers is the quality of service of the data received from multicast sources such as delay, bandwidth, and loss rate.

8.5.1.1 Delay bound

The end-to-end delay is the period of time required for a data packet to be routed through the network from the application where it was created to a destination application.

Aside from those time-sensitive data (e.g. stock prices, and real-time monitoring information), most one-to-many applications have an acceptable tolerance for delay and delay variance (jitter). Most of many-to-many multicast applications are intolerant because they are bidirectional, interactive and request/response dependent. As a result, delays should be minimized, since they can adversely affect the application's usability. This need to minimize delays is most evident in (two-way) conference applications, where users cannot converse effectively if the audio or video is delayed more that 400 milliseconds.

Several real-time multicast applications (e.g; distributed simulation, video-conferences) are delay-sensitive and the delay variation between receivers can degrade the synchronization of informations received and sent by members (receivers and sources).

Let $d(i,j)$ be the delay of the link between node i and node j. Since the delay is an additive parameter, the delay $d(u,v)$ between node u and node v is equal to the sum of individual link delays along the path $P(u,v)$ between u and v:

$$d(u,v) = \sum_{(i,j) \in P(u,v)} d(i,j) \tag{8.1}$$

Each receiver R can estimate and compare the mean delay of data received from the RPT to the required delay. The delay bound inequality is:

$$D_R(RPT) \leq D_R \tag{8.2}$$

where D_R is the minimum delay required by the receiver R and $D_R(RPT)$ is the mean delay from the RPT.

8.5.1.2 Bandwidth bound

Unicast and multicast applications both need to design and to adapt the variability of network conditions and especially in term of the available bandwidth. Multicast bandwidth-sensitive applications such as multicast multimedia file transfer and video on demand, have the bandwidth as the first required network resource and their performance degrade in function of the rate in which the receiver receives data from the source.

Let $b(u, v)$ denote the available bandwidth along the path between node u and node v. The bandwidth is a concave parameter and then $b(u, v)$ is computed as follows:

$$b(u, v) = \min_{(i,j) \in P(u,v)} b(i, j) \tag{8.3}$$

where $b(i, j)$ is the available bandwidth in the link between node i and node j. The rate bound inequality is:

$$R_R(RPT) \geq R_R \tag{8.4}$$

where R_R is the minimum required rate by the receiver R and $R_R(RPT)$ is the actually available rate from the RPT.

8.5.1.3 Loss rate bound

Many of the multicast application such as audio/video distribution have loss-tolerant data content. In other words, the data content itself can remain useful even if some of it is lost. For example, audio might have a short gap or lower fidelity but will remain legible despite some data loss.

We note by $L(i, j)$ the loss probability of the link between node i to node j. The loss rate is a multiplicative parameter, the probability $P(u, v)$ that a packet sent by router i arrives to node j is equal to the product of no loss probability of links between u and v.

$$P(u, v) = \prod_{(i,j) \in P(u,v)} (1 - L(i, j)) \tag{8.5}$$

The loss bound inequality is:

$$P_R(RPT) \leq P_R \tag{8.6}$$

where P_R is the maximum loss rate required by the receiver R and $P_R(RPT)$ is the current loss rate from the RPT.

8.5.2 Network Parameters

The shared tree PIM-SM's mode is expected to concentrate traffic onto the subset of network links that compose the shared trees. In contrast, the source-based tree PIM-SM's mode is expected to distribute the traffic more evenly among all links because it uses a different tree for each sender and each group [40].

One of the most important interest in using PIM-SM is its ability, via the bootstrap mechanism (BSR), to distribute the multicast traffic load between the routers candidate to be RP (also referred as the RP candidates). In fact, the traffic concentration around RP routers when holds may affect the multicast data delivery quality.

The traffic load at RP depends on the number of sources in the multicast group using the RPT and their sending rate. We assume that for each multicast group, the RP maintains the data rate received from all sources and the rate from distinct sources. We note by $R^i_{RP}(G,S)$, the rate of data received from source S belonging to group G on the multicast-enabled interface i of the RP. The total data rate of all sources of group G on interface i is given by:

$$R^i_{RP}(G,*) = \sum_{S \in \{RPT\}_G}^{i} R^i_{RP}(S,G). \tag{8.7}$$

where $\{RPT\}_G$ and $\{SPT\}_G$ are the set of sources of group G that use the shared tree and the source-based tree, respectively. The total rate of the multicast traffic received by the RP on the interface i is computed as follows:

$$R^i_{RP}(*,*) = \sum_{G} R^i_{RP}(G,*) \tag{8.8}$$

We assume that for each interface i the RP maintains a multicast traffic maximum fraction X_i (a multicast traffic threshold) of the link capacity C_i that should not be exceeded. Links where the multicast rate exceed $X_i C_i$ are considered over limit.

8.6 Switching Mechanism Overview

The simplest PIM-SM switching mechanism that we could use is to switch all receivers from the RPT to the SPT if at least one receiver decides to switch to. This is useful when we make the assumption that all the receivers receive data with the same quality of service. In this case each receiver will consider that the SPT is more efficient than the RPT. The advantage of such type of mechanism is that it is easy to implement, however it does not guarantee that all receivers even the SIR will be satisfied after the switching in the general case.

Another switching alternative is to consider only receivers belonging to the SIR's SP-subtree. In such mechanism, we assume that only SP-subtree receivers (and not all receivers) are concerned by the switching decision because is the SP which forwards to them the data received either from the RPT or from the SPT. Currently PIM-SM uses implicitly this kind of mechanism.

We believe that both alternatives are not efficient because they do not take into account the receivers reception quality and their connectivity heterogeneity. Furthermore, there is no coordination between the receivers concerned by the switching.

Our mechanism is based on a coordination between the receivers concerned by the switching initiated by the SIR. Certainly, that such mechanism can be costly in terms of complexity and

switching latency because it requires specific states to be maintained by at least the SP and designated routers but it has the advantage of taking into account the requirements of all concerned receivers by the switching decision. We will evaluate the tradeoff between both parameters in Section 8.11.

Each switching point should maintain a state per multicast entry concerning old PIM switch requests received. For each state a timer is maintained after the expiration of this timer, the SP drops this state and update it whenever it receives new information concerning new PIM[3] Switch requests. This database is called the Switching Data Base (SDB).

The receiver who wants to switch to the SPT should first verify if there is already a switching request under processing that have been sent by him or by another downstream receiver. If it is not the case, it sends immediately a Switch Request message toward the source. Otherwise, it should wait for the expiration of |Switch-Request-Send-Timer| before sending its request.

After receiving the Switch request, the SP router sends a coordination request to other interfaces in the `oif` list of the (*,G) entry. Receivers may accept or refuse the request. When the SP receives a Switch Accept message from all interfaces or the timer |Switch-Coordination-Wait-Timer| expires without receiving all responses, it sends a Switch Ack message to the SIR. If at least one of the receivers refuses the switching (sends a Switch Refuse message to the SP), the SP sends immediately a Switch Nack message to the SIR.

In the next section, we describe in detail our mechanism and the new proposed PIM messages.

8.7 Detailed Switching Mechanism Description

Our mechanism can be integrated in PIM-SM by adding the following new messages:

- PIM Switch Request: this message is sent by the initiator switching receiver toward the source.
- PIM Switch Coordination: this message is sent by the Switching Point (the SIR's SP) to its downstream receivers to ask them to accept or to refuse the SIR's switching request.
- PIM Switch Accept/Refuse message: this message is sent to all SP's downstream receivers toward the SP.
- PIM Switch ACK message: this message is sent by the SP to downstream receivers in order to inform them that they are authorized to receive data directly from the source and then send a PIM Join (S,G) to the source.
- PIM-SM Switch NACK message: this message is sent by the SIR's SP in order to inform it that at least one of them did not accept to switch to the source.

We summarize in Table 8.1, the messages used by our switching mechanism.

In the following sub-sections, we will give more details about these messages: when they are sent? What each router should do when receiving these messages? etc.

[3]Throughout this chapter we use the term PIM to refer to the Sparse Mode (SM) of PIM (PIM-SM).

Message	From –> To	When ?
Switch Request	SIR –> Source	The SIR wants to switch to SPT
Switch Coordination	SP –> Downstream Receivers	The SP receives a Switch Request
Switch Refuse	Receiver –> SP	The receiver refuses the switching
Switch Accept	Receiver –> SP	The receiver accepts the switching
Switch Ack	SP –> SIR	All downstream receivers accept the switching
Switch Nack	SP –> SIR	At least one downstream receiver refuses the switching

Table 8.1: Summary of new PIM-SM Messages

8.7.1 PIM Switch Request Message

A PIM Switch Request message is sent by the receiver that wants to switch to the SPT, so called the SIR (Switching Initiator Receiver), toward the source. This receiver should specify in its request message the QoS parameters needed and their current and requested values. After sending this request, the receiver continues receiving packets coming from the RP. It will send a PIM Join message to the source only after reception of the PIM Switch Ack message from its Switching Point (SP).

A PIM-SM router which receives a PIM Switch Request should first determine if he is the Switching Point (SP) of the receiver sending that message. If so, this router sends to the downstream receivers a PIM Switch Coordination message to inform them that there is one receiver who wants to switch to the SPT and that they may accept or refuse the switching.

Intermediate routers between the SIR and its SP have only to forward the packet toward the source.

When a router receives a PIM Switch request from a new SIR of which he is the SP, while he does not send a PIM switch Ack/Nack to the previous SIR, the router deletes this message.

To prevent from large amount of signaling switching messages, the receiver should send at most one PIM Switch Request each |Switch-Request-Time|. Each receiver maintains a history about switching requests that it has sent toward each active source.

When an SIR does not receive neither a Switch Ack nor a Switch Nack message, it assumes that its message was lost in the path towards the SP and it tries again after the expiration of the timer |Switch-Request-Time|.

8.7.2 PIM Switch Coordination Message

When receiving a PIM Switch Request message, the SP should first verify whether it has local receivers or no. If it is the case it checks whether the switching to the SPT does not violate their QoS requirements. If so, this router sends a PIM-Switch Coordination message to its downstream interfaces belong to the `oif` list of the (*, G) entry except the interface from which it received the switching request. This message contains a copy of the requested parameters given by the SIR in its Switch request message.

A router that receives a PIM switch coordination message, should check whether it has

local receivers or no. If so and when the eligibility tests[4] succeed, it forwards the message to other interfaces that belong to the `oif` list.

If the eligibility test do not succeed, the router should immediately send a PIM Switch Refuse message to the SP. In this case there is no need to forward the coordination message to other interfaces.

8.7.3 PIM Switch Accept/Refuse Message

When receiving a PIM Switch Coordination message, a receiver determines whether the switching to the source-based tree violates its reception quality or no. If so, it sends a PIM Switch Accept message to the SIR's Separation Point, otherwise it sends a PIM Switch Refuse message.

This message should be sent immediately to the separation point which has sent the PIM-SM coordination message. A router who receives a PIM switch Accept message should wait for the response from receivers downstream to other interfaces. It forwards this message only if it receives an accept message from every interface where there is at least one downstream receiver. In other cases, it sends a PIM Switch Refuse message.

The receiver which sends a PIM Switch Nack message may specify the reasons why it refuses the switching to the SPT.

A router which receives the PIM Switch Accept to forward to the SIR's Separation Point and which has already sent a PIM Switch Accept/Refuse of the same SIR before [Switch-Accept-Refuse-Timer] should reject this message. In contrast, when it receives a PIM Switch Refuse message, it should forward it to its destination.

The router can know if it has already sent a PIM Switch Accept/Refuse to the ISP by consulting its Switching Data Base where entries are kept alive for |Switch -SDB-entry-time|.

8.7.4 PIM Switch Ack/Nack Message

After Sending a PIM Switch Coordination message, the SP should wait a maximum time equal to |Switch-Coordination-Timer| for downstream routers answers from all interfaces before sending the Switch Ack/Nack message to the SIR.

When the separation point receives a PIM Switch Accept from all interfaces in `oif` list except that from which it received the switching request, it sends a PIM Switch Ack towards the switching initiator receiver[5]. In the case when there is at least one PIM Switch Refuse message received from one of the downstream interfaces, the SP sends a PIM Switch Nack to the SIR without waiting other PIM switch Accept/Refuse from other interfaces.

For security reasons, upon receiving a PIM Switch Ack/Nack from the SP router, a receiver should first verify if it has sent a PIM Switch Request or no. If not, it deletes the message. Otherwise, it handles the following tasks:

- If the message is a PIM Switch Ack, then the receiver has the permission to switch to the SPT. Therefore, it sends a PIM Join message toward the source.
- If the message is a PIM Switch Nack, the receiver can not switch to the SPT. It may try again after the expiration of |Switch-Request-Timer|.

[4]The eligibility tests will be the subject of Section 8.9.

[5]One could uses another policy to decide when to send the Switch Ack to the SIR. An alternative can be when the SP receives a Switch Accept from the majority of the `oif` list interfaces

8.8 Illustration Example

We illustrate our mechanism using the example shown in Figure 8.3. The receiver R which detects that at least one of the parameter is violated (for example: the delay) sends a PIM Switch message towards the source S. The router Rt1 detects that it is the SP of the SIR. It then does not forward the switching request and it sends a PIM Switch Coordination message to interfaces 2 and 3 because they belongs to the `oif` list of the requested Group. We assume that receivers R_1 and R_2 decide to accept the switching coordination request. As consequence, they send a PIM Switch Accept message toward the SP. After receiving both of messages, the SP sends a PIM Switch Ack to the SIR. The SIR can then switch to the SPT by sending a PIM Join (S, G) toward the source.

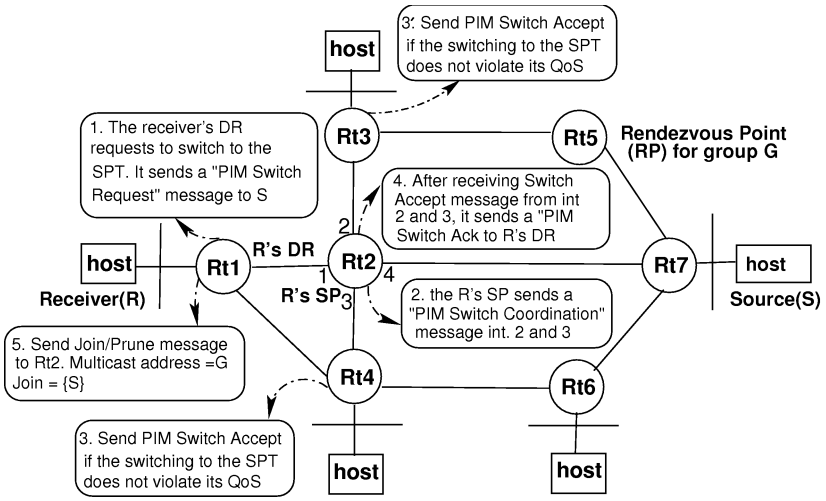


Figure 8.3: Example: Switching from shared tree (RPT) to shortest path tree (SPT). Actions are numbered in the order they occur

Now, assuming that receiver R_3 does not want to accept the switching coordination request, it then sends a PIM-SM Switch Refuse message toward the SP which in return sends a PIM-SM Switch Nack message toward the SIR without waiting for the response from the receiver R_4.

8.9 Eligibility Tests

8.9.1 The Case of Delay Constraint

We first consider the case when the QoS required is expressed in terms of an additive QoS parameter. Without loss of generality, we use the end-to-end delay as example, and impose an upper bound D_R, on the acceptable end-to-end delay perceived by a receiver router in a multicast group.

The receiver R maintains the required delay D_R and the current $D_R(RPT)$ of the multicast session received from the RPT. When detecting that the $D_R(RPT)$ remains superior to D_R for a configured period of time, it decides to switch to the SPT. It sends a switch request towards the source and it continues receiving data from the RPT while waiting for the response for

its switch request. In addition to other information, this request should contain the values of D_R and $D_R(RPT)$. When receiving the switch request from the receiver R, the R's Switching Point (SP)[6] should decide if the switching to the SPT can improve the delay and if it degrades the QoS of other receivers belonging to the SP-based subtree.

Let's now detail how the the SP should process the switching request. The first thing that the SP should do is the estimation of the delay $D_R(SPT)$ denoting the delay if the data was received directly from the SPT. In fact, we have:

$$D_R(SPT) = D_R(RPT) - [D_{SP}(RPT) - D_{SP}(SPT)] \tag{8.9}$$

where $D_{SP}(RPT)$ and $D_{SP}(SPT)$ are the delay viewed by the SP when it receives data from the RPT and from the SPT, respectively. The only missing information to compute $D_R(SPT)$ is the value of $D_{SP}(SPT)$. We propose two manners to evaluate this delay:

- In the case when the routers between the SP and the source support a link-state unicast routing protocol such as OSPF, the SP can determine the delay from the source by cumulating the delays of the links in its path toward the source. In this case, we have:

$$D_{SP}(SPT) = \sum_{(i,j)\in P(SP,S)} d(i,j)$$

- In other cases (i.e., the unicast routing protocol does not provide this information), the SP can join directly the source for a short period of time and compute the delay and leave the source after this period[7]. This approach, which we call the switch-experiment approach, is similar to the join-experiment approach used in layered multicast [116] (of course at the transport level) by the receivers to determine if they are able to join more transmission layers.

After computing the value of $D_{SP}(SPT)$, the SP compares this value to the required value D_R. It considers that the SPT is more effective in term of delay than the RPT if:

$$D_R(SPT) \leq D_R. \tag{8.10}$$

The second eligibility tests that should be done by the SP is the assurance that the QoS requirements receivers in the SP-subtree are not violated due to the switching from the RPT to the SPT. In other words, it should verify that the following equation holds for each concerned receiver X belonging to the SP-subtree:

$$D_X(SPT) \leq D_X + \varepsilon \tag{8.11}$$

where ε is an adjustment parameter estimated or configured by the PIM-SM router and D_X is the delay required by the receiver X. $D_X(SPT)$ is computed as follows:

$$D_X(SPT) = D_X(RPT) - [D_{SP}(RPT) - D_{SP}(SPT)] \tag{8.12}$$

When Eq. 8.10 holds and Eq. 8.11 holds for each concerned receiver, the SP assumes that the switching to the SPT is useful and it forwards the switching request to the source.

[6] On the reception of a switch request, routers compare the outgoing interface to the source and the incoming interface from the RP. In the case when these interfaces are different, the PIM router is considered as the SP of that receiver.

[7] In addition of multicast packets sent to the RP encapsulated in PIM-SM Register Messages, the source unicast multicast packets to the SP during this period of time. A new PIM-SM message can be added to PIM-SM protocol in order to support this scenario.

8.9.2 The Case of Rate Constraint

When the receiver R detects that the equation $R_R(RPT) < R_R$ holds, it sends a switch request toward the source. In its request, the receiver specifies the value of the required rate (R_R) and the current received rate from the RPT ($R_R(RPT)$). When it receives the SIR's switching request, the R's SP compares $R_R(RPT)$ and $R_{SP}(RPT)$ denoting the rate received by the SP from the RPT. If the following equation holds:

$$R_R(RPT) < R_{SP}(RPT), \tag{8.13}$$

the bottleneck link belongs to the path $(SP,\ R)$ and not $(S,\ RP) \cup (RP,\ SP)$[8]. So, there is no need to switch from the RPT to the SPT since the rate will not be increased even if the rate $R_{SP}(SPT)$ is greater then $R_{SP}(RPT)$.

In the case when $R_R(RPT) = R_{SP}(RPT)$, we can conclude that the bottleneck link belongs to the path $(S,\ RP) \cup (RP,\ SP)$ and not $(SP,\ R)$. Then, the switching from the RPT to the SPT may increase the rate received by the receiver.

The missing information needed by the SP to take the switching decision is the value of $R_R(RPT)$ denoting the rate that can be received from the SPT and that of $R_{min}(SP, R)$ corresponding to the minimum available bandwidth in the path $(SP,\ R)$. We suppose that the SP gets these information by sending a request to PIM-routers belonging to the path toward the source (or by joining the source for a small period of time) and the receiver R (or via another manner, i.e., the use of a link-state unicast protocol), respectively.

After getting $R_{SP}(SPT)$ and $R_{min}(SP, R)$[9], the SP judges whether or not the switching to the SPT can improve the rate received by the receiver R. If:

$$R_{SP}(SPT) \geq R_R \text{ and } R_{min}(SP, R) \geq R_R \tag{8.14}$$

the receiver R will receive data from the SPT with the rate $R_R(SPT)$ given by:

$$R_R(SPT) = \min(R_{SP}(SPT),\ R_{min}(SP, R)) \geq R_R \tag{8.15}$$

The second task of the SP is to ensure that the QoS requirements of other switching concerned receivers in the SP-subtree are not violated when the receiver R switches from the RPT to the SPT. To do this, the SP can request the available minimum bandwidth $R_{min}(SP, R)$ in all paths of the SP-subtree toward the receivers and not only the path $(SP,\ R)$ and verify that:

$$R_{SP}(SPT) \geq \max_{R \in SP-subtree} R_R$$
$$\text{and}$$
$$R_{min}(SP, R) \geq \max_{R \in SP-subtree} R_R$$

for each receiver in the SP-subtree.

[8]Considering that the rate is a concave parameter, the rate received by the SIR is the minimum of the available rates of links toward the RP or the Source. If this minimal belong to the path between the SIR and its SP, the rate received by the SIR from the RPT and the SPT is at least the same.

[9]The value of $R_{min}(SP, R)$ is superior to $R_R(RPT)$ because the bottleneck link belongs to the path $(S,\ RP) \cup (RP,\ SP)$ and not to $(SP,\ R)$.

8.10 Implementation Issues

The switching mechanism proposed in this chapter can be easily integrated in PIM-SM protocol. Indeed, currently PIM-SM uses eight message type values among sixteen available values. We propose to add a new message type value (e.g., type number 9) to Switch messages. The sub-messages (Request, Coordination, Accept, Refuse, Ack, Nack) will be included in the switch message as indicated in Figure 8.4 which represents the Switch message format.

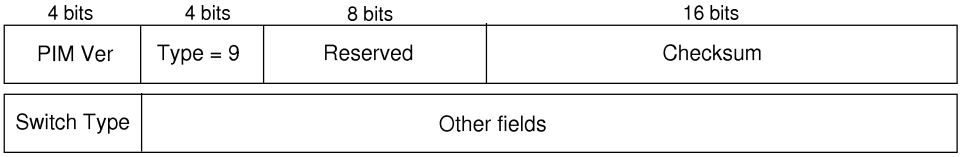


Figure 8.4: Switching message format

In a multicast delivery tree where there are on-tree routers that may implement different PIM-SM versions, i.e, some routers support the switching mechanism and others do not support it, as result we will the default behavior of PIM-SM. Indeed, for example when a receiver's DR does not respond to a PIM Switch Coordination message sent by the SIR's SP, the SP will behave as it receives a PIM Switch Accept. Then, it sends a PIM Switch Ack/Nack based only on the coordination between the downstream receivers that support our switching mechanism.

We choose the following values to the switch sub-types messages: Request message: type number 0, Coordination message: type number 1, Accept message: type number 2, Refuse message: type number 3, Ack message: type number 4, Nack message: type number 5.

8.11 Simulation and Results

In this section, we evaluate the performance of our switching mechanism using simulation.

8.11.1 Simulation Model

We use the Network Simulator [84] to evaluate the performance of our switching mechanism that we implement in NS simulator[10].

Considering that PIM-SM is an intra-domain routing protocol, we assume that the network topology belongs to an unique administrative domain. We generate 100 network topologies of 1000 nodes with different connectivities values using Brite tool [85][11] which integrates the GT-ITM generator.

In our simulations, each group was assigned a single Rendez-vous Point (RP) which is randomly selected from a set of some centrally located nodes by a manner to approximately equilibrate the number of groups served by each RP.

The sources of each multicast group is assumed to generate traffic with a specific characteristics. We attribute to each multicast group a multicast application for which each receiver

[10]The tcl code of the different switching mechanisms is available at `http://www.inria.fr/rodeo/filali/pim-sm`.

[11]BRITE can be downloaded from `http://cs-www.bu.edu/brite/`.

has specific QoS requirements in terms of delay, data rate, and loss probability. The values of these parameters are randomly generated for each receiver.

We distinguished several scenarios depending on the unicast routing protocol used: RIP, OSPF, or Euclidean-distance-based (EUC) and for different QoS requirement: delay, bandwidth, and delay and bandwidth.

8.11.2 Results and Observations

We performed simulations to compare the quality of the delivery tree built by PIM-SM combined with our switching mechanism.

We first consider the case when we don't use a coordination-based switch mechanism. That's mean that a receiver decides to switch to the SPT without collaborating with other receivers.

In a first experiment, we evaluate the fraction of unsatisfied receivers when a receiver belongs to the same delivery sub-tree decides to switch to the SPT. We conducted 500 different simulation scenarios. In each scenario, we randomly choose the receiver initiating the switching.

In Figure 8.5, 8.6, 8.7 we plot this fraction when using RIP, OSPF, and EUC and when the QoS parameter is the delay, the bandwidth, and the delay and the bandwidth, respectively.

As we can see, the fraction of unsatisfied receivers vary from 0% to 100 %. It is close to 100% when the QoS requirement is the bandwidth for most simulation scenarios.

In Figure 8.8, we show the fraction of group members that are interested by the switching request. As we can see, regardless the unicast routing protocol used, this fraction is very variable.

In a second experiment, we consider the case when we use our switching mechanism to manage the switching from the SPT to the RPT. That is, the SIR's switching request will be accepted only when all the SP-subtree downstream receivers accept the switching. We varied the group size from 5 to 50 and for each case we conducted 100 simulation instances. For each multicast group, we use an uniform distribution to generate the location of the different receivers and sources in the network. We computed the average number of accepted switching requests. In Figure 8.9, we plot the fraction of accepted switching requests in function of group size for various unicast routing protocols and when the QoS requirement is only th delay, only the bandwidth, delay and bandwidth.

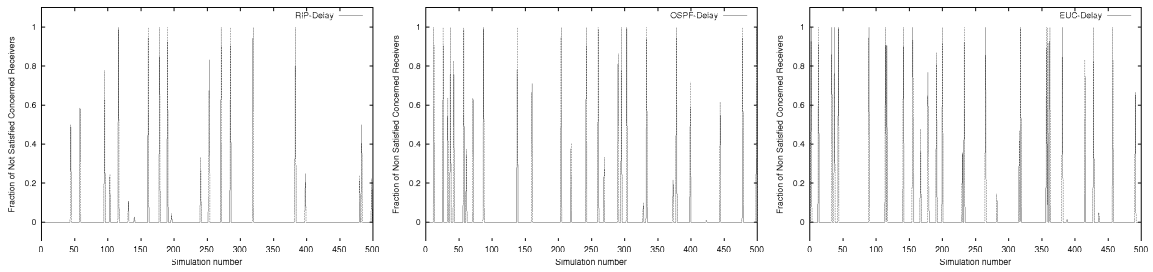


Figure 8.5: Fraction of unsatisfied receivers for delay-based switching

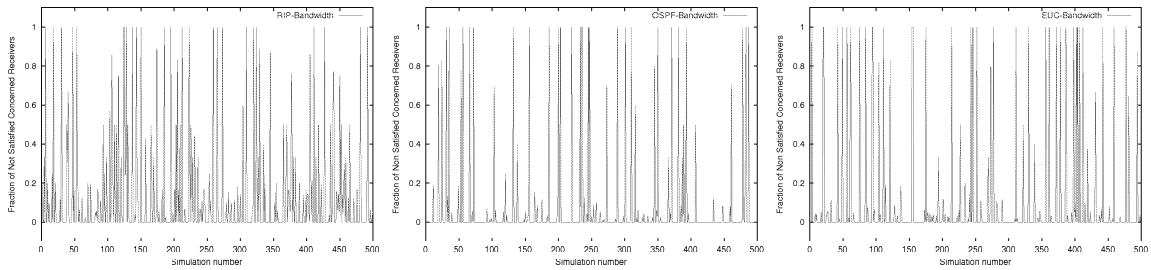


Figure 8.6: Fraction of unsatisfied receivers for bandwidth-based switching

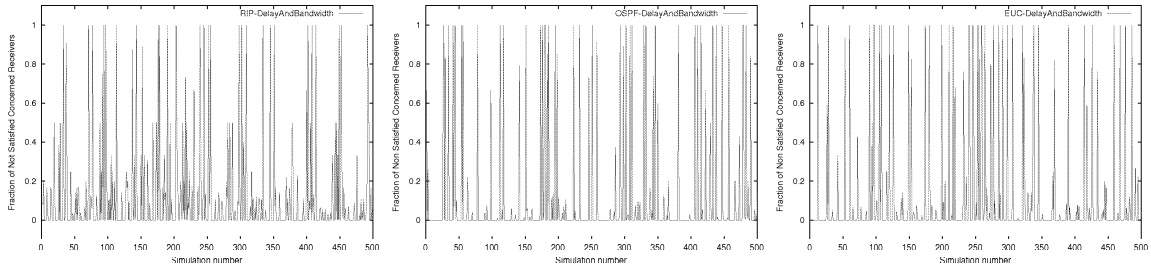


Figure 8.7: Fraction of unsatisfied receivers for delay and bandwidth based switching

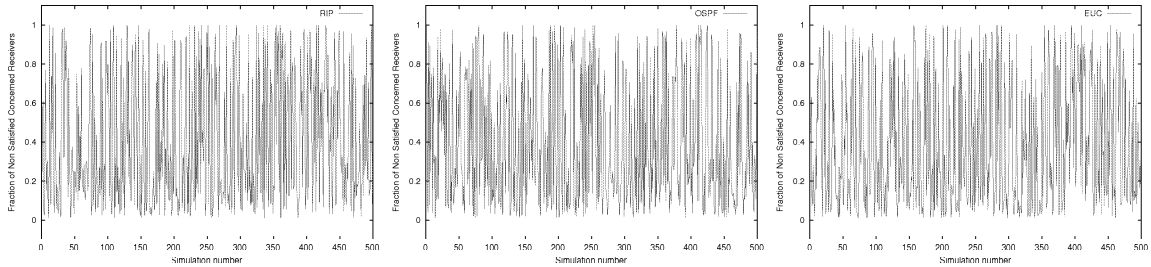


Figure 8.8: Fraction of the group members which are interested by the switching request

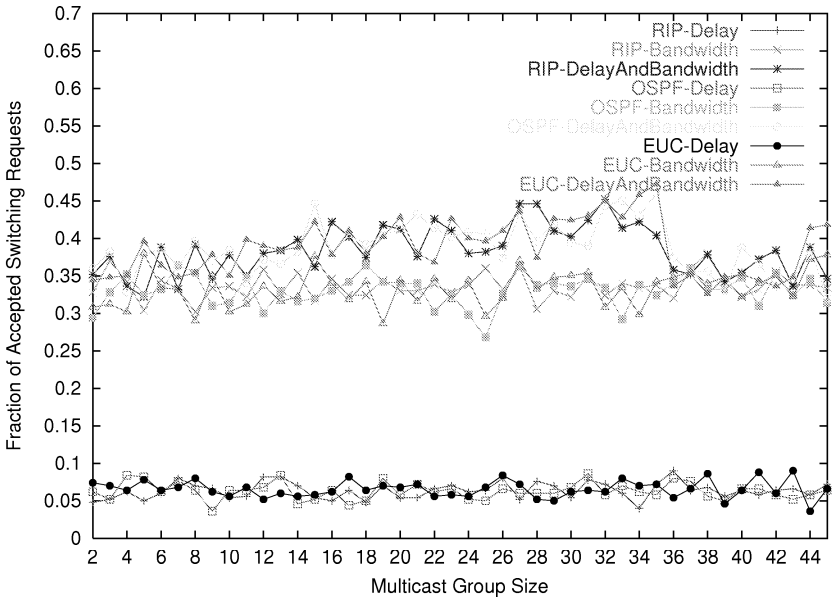


Figure 8.9: Variation of the fraction of accepted switching requests in function of the group size when using a coordination-based switching mechanism

8.11.2.1 Switching Latency

We define the switching latency as the period of time between the time the receiver has sent the join request to the source and the reception of the join acknowledgment. The switching latency is always less than two times the timer |Switch-Coordination-Timer| plus the delay from the SIR to its SP.

We denote by T_l the latency of the switching receiver initiator. T_l can be simply deduced from:

$$T_l = T_{ack/nack} - T_{request}$$

when the switching was initiated by a receiver.

The switching latency time depends on the complexity of the switching mechanism and the processing tasks that must be executed by each router before taking the switching decision.

We plot in Figure 8.10, the variation of the switching latency in function of the group size. We can easily see that the switching latency varies between 4 ms and 16 ms. Regardless the unicast routing protocol used, curves have the same shape.

8.12 Chapter Summary

In this chapter, we have addressed the problem of switching from the shared tree and the source-based tree in PIM-SM intra-domain multicast routing protocol. We demonstrated the necessity of a coordination-based switching mechanism to increase the effectiveness of PIM-SM. The switching can be initiated either a receiver's Designated Router or the RP. We defined a new PIM-SM message (Switch Message) as well as sub-messages (Request, Coordination, Accept, Refuse, Ack, Nack) that can be easily included in the standard PIM-SM. Indeed,

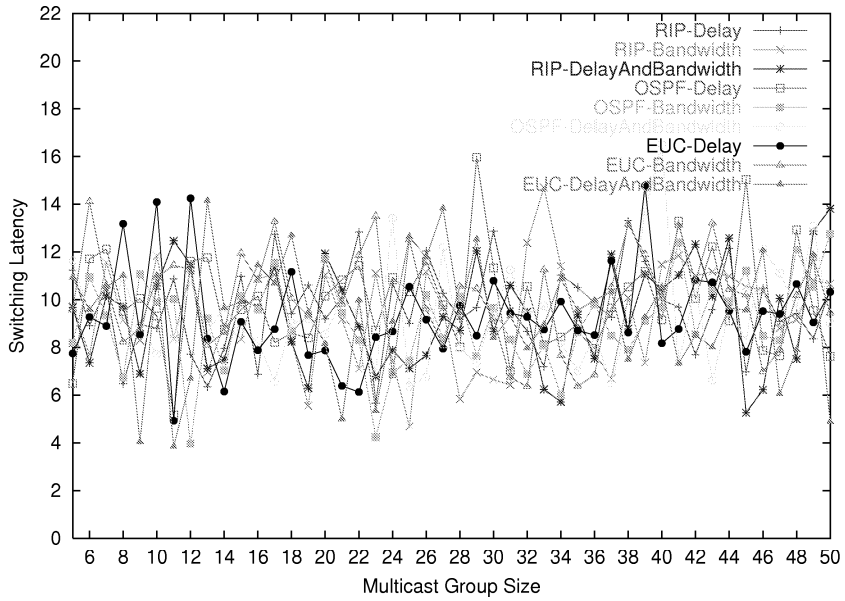


Figure 8.10: Variation of the switching latency

through the use of timers, the default behavior of our mechanism is that recommended by PIM-SM standard.

We used simulation to evaluate our proposed switching mechanism in the case of using RIP as well as OSPF as an the underlying unicast routing protocol. The results show that our coordination-based switching mechanism improves the performance metrics of all receivers concerned by the switching decision. When combined with a QoS-based unicast routing protocol, our mechanism could perform much better then a coordination-less-based one as that described in PIM-SM standard [40].

Chapter 9

Conclusion and Outlook

In this chapter, we first summarize briefly the contributions of this thesis. In Section 9.2, we discuss extensions of this work that will provide interesting challenges for future research. We also give a more long-term vision which we believe research will be able to make significant contributions in the field of multicast deployment in the Internet.

9.1 Summary of Contributions

Recent years have witnessed an enormous increase in the use of the Internet for a large variety of applications such as e-commerce, software distribution, news/stock quote broadcasting, multimedia streaming, video conference, and of course, data communication. The emergence of such applications has created two new phenomena in the Internet: (a) multicast support is becoming critical, since such applications typically involve multiple participants, and (b) the data streams and networks are becoming more and more heterogeneous due to the proliferation of multimedia applications over the net.

In this context, we are interested in answering the following questions: "What are the minimum functionalities to achieve the terrestrial routers in order to ensure the coexistence of unicast and multicast applications?" "How can we design a deployable and scalable multicast architecture for satellite-terrestrial hybrid networks ?"

In the first chapter, we have described the context of the work described in this dissertation. In chapter 2, we have given an overview of the trends and issues in the deployment of IP multicast in heterogeneous environments and we have pointed out the main problems that we have addressed in this thesis.

Chapter 3 has focused on the first component concerning the network resource sharing among multicast competing flows. We proposed MFQ (Multicast Fair Queuing) an active queue management mechanism that aims to share the bandwidth "fairly" among competing multicast flows. One of the main goal when designing MFQ is to be independent of the multicast fairness function used to share the bandwidth among competing flows. The module which computes and updates the flow weights is designed to be implemented separately from the buffer management module. For layered multicast transmission, MFQ effectively addresses the issue of multicast group heterogeneity by ensuring that each receiver receives the highest priority portion of the multimedia streams that its connection quality can sustain. Additionally, MFQ supports packets size heterogeneity, as well as sessions with different number of members.

The second component of our proposal is a scheduling mechanism which has been presented in Chapter 4. The main idea behind this scheduler is to share the bandwidth allocated to the multicast service between multicast competing flows using MFQ mechanism. Our scheduler implements a novel notion of unicast and multicast fairness called inter-service fairness that takes into account the TCP-friendliness criterion. A set of performance results in simulation environment using the *ns* network simulator show that SBQ can closely achieve the expected bandwidth sharing among multicast and unicast active flows while keeping multicast connections globally TCP-friendly. Additionally, it adapts to network dynamics.

Note that in order to speedily deploy our proposed new mechanisms over the Internet, we kept them as simple and scalable as possible while maintaining their original intentions.

Both MFQ and SBQ use the information about the membership to implement the fairness function. In Chapter 5, we addressed the interesting issue concerning the counting of group members in multicast communication. We proposed and evaluated an extension to the multicast service model that enables the multicast senders as well as the intermediate routers in the multicast delivery tree to count the number of members downstream to each outgoing interface. The necessary extensions to the host and the router parts of the multicast service model have been presented.

MFQ and SBQ combined with a well-accepted definition of the inter-multicast fairness provide a significant step towards a complete and scalable congestion control algorithm for multicast applications. We hope that the introduction of MFQ and SBQ will encourage the ISPs to support the multicast in their networks because it provides a flexible way to share the bandwidth between competing multicast flows.

The second part of this dissertation focused on the impact of the transmission medium heterogeneity on the efficiency of an end-to-end multicast delivery over data communication networks that integrate satellite links. Basically, we have considered the problem of building efficient multicast delivery trees using multicast routing protocols designed regardless of the inherent characteristics of the transmission medium which is in fact the general design concept of the Internet protocols.

We were first interested in Chapter 6 in studying the behavior of multicast routing protocols over GEO transparent satellites. We identified some undesirable behaviors for DVMRP, PIM-DM, and PIM-SM and we proposed a set of tuning techniques to overcome these behaviors. We have focused on the support of PIM-SM multicast routing protocol in the DIPCAST system that we considered as a real case study. Recommendations concerning the configuration of PIM-SM in this system have been proposed. These recommendations deal mainly with the choice of the RPs and the switching option from the RPT tree mode to the SPT tree mode for 1 to N and N to M multicast applications. Some techniques have been also proposed to overcome specific problems when using the standard PIM-SM protocol in such system causing a useless signaling and data traffic in the satellite segment.

Then, we concentrated our study in Chapter 7 on the deployment of IP multicast in the next-generation of GEO satellite system characterized by the support of multiple spot beams and the on-board switching capability. We proposed a new encapsulation scheme dedicated to the IP protocol that we have called "the IP-Optimized encapsulation" which could be used instead of the MPE scheme originally designed for GEO transparent satellite systems. We also described two approaches that could be used to enable the switching in the on-board satellite processor. The first approach called the *self routing approach* requires that the on-board satellite maintains a switching table giving the correspondence between the PIDs (MPEG data segments identifier) and the list of outgoing spot beams. The second approach called the

label switching approach needs that a field containing a label to be added by the RCST sender to each MPEG data segment. The satellite uses this field to determine the list of outgoing spot beams. For deployment reasons it removes the label field before forwarding the data segments which also minimizes the overhead in the downlink. For each approach, we have described the required tables and operations to be done in the different entities of the system: the RCST, the NCC, and the satellite. Finally, we presented SMRP (Satellite Multicast Routing Protocol) which is an adaptation of PIM-SM routing protocol for the studied system.

A more generic problem discussed in Chapter 8 concerns the switching between the two modes of PIM-SM: the SPT tree mode and the RPT tree mode. We developed a new switching mechanism that makes a coordination between all members in the sub-tree which will be affected by the switching to decide whether to switch to the second mode of PIM-SM or not. The switching decision also may take into account several metrics related to the members (delay, throughput, loss rate, etc.) as well as to the network (resource usage, overhead, etc.). This switching mechanism has been evaluated using simulations. The results show that this mechanism is able to fulfill the members requirement at a reasonable cost.

Previous attempts to provide new services to applications were not similar to our approach: either the end systems does it all (e.g. use buffering to reduce jitter), or the services must be hardwired in routers. What we are proposing opens up the network (albeit a small part) for end systems to see and use. A router's primary duty is still to store and forward, but they will have additional capability is turned on when needed. Our approach imposes the least intrusive change to routers, yet has the potential to provide wide-ranging benefits, since ISPs can use these additional tools to customize multicast-related services to fit their need.

We believe that the **complexity** of network protocols will be the main challenge for networking in the coming years. Application and equipment developers face an environment of harsh competition and a great diversity of protocols., vendors, service providers, and applications. The researcher's perspective should take this into account by focusing less on global solutions that assume global control over the infrastructure; rather, the design space should be restricted to solutions that are viable in this dynamic, heterogeneous marketplace.

One very important lesson that we have learned from the impressive success of the Internet is the power of its service model simplicity. Offering only connectivity as a largest common denominator of possible services has allowed the Internet to spread rapidly. Monolithic solutions proposing more elaborate service models had more trouble becoming de-facto standards. Research should recognize this by focusing less on complete solutions that are hard to deploy in practice, simply because a global networking infrastructure is not, and probably never will be, a fixed, homogeneous structure under centralized model.

9.2 Future Directions

There are several promising and important extensions to the research problems addressed in this thesis. Indeed, a number of issues remain open in the design of efficient schemes to fairly share the network resource between unicast and multicast competing flows. We will continue our approach of gaining insight using a combination of network and ISP point of view.

First, in the proposed architecture of MFQ, we assumed that the fairness function depends on the number of downstream members of each flow. We plan to develop a generic framework that we have already introduced in Section 3.3.1.1 by taking into account more parameters than the number of group members such as the traffic rate of each flow or to include multicast pricing

models. Future work in this topic could also propose a network-based multicast congestion control that profit from MFQ. An initial investigation on this topic has been already addressed in Section 3.7.

Second, there exist many possible areas for future work in the theme of bandwidth sharing between unicast and multicast flows and still remain many performance aspects to be evaluated. Future work could develop an end-to-end multicast congestion control mechanism that uses a small but efficient help from the scheduler by resolving the issue of fairness between unicast and multicast flows.

Third, the counting service proposed in Chapter 5 could be integrated and evaluated in the SSM implementation described and available in [8]. It is worthwhile to mention that our extension is useful for many other applications such as multicast billing and accounting, multicast congestion control, feedback suppression, etc.

Supporting the IP multicast over satellites is an exciting research area that integrates many open issues. Our work on this topic have focused on the routing issue. Indeed, the field of IP over GEO satellites is ripe for further work, as there are many more interesting issues involving multicast delivery over satellite than we were able to cover. For example, integration of a GEO satellites with a LEO network or wireless networks is an interesting problem. Given a GEO network with multiple spot beams, how can network decide upon which gateway to exit the network ? What kind of load balancing around satellite terminals, is needed ? What are optimal queue sizes for on-board switches ?

The switching mechanism that we have proposed in Chapter 8 could be extended by evaluating other performance metrics such as the bandwidth overhead and the receivers satisfaction. Another area of future study is to investigate how this mechanism can be extended to provide the switch back alternative to the RPT. We believe that the switch back can be integrated by the same way as our mechanism.

In summary, this dissertation provides valuable insights and opens up a number of research directions for multicast deployment in heterogeneous environments. However, the scope of our work is largely limited to the resource sharing between unicast and multicast flows in terrestrial networks and the IP multicast routing over GEO satellite networks.

Research Publications

Conference Papers

1. F. Filali and W. Dabbous, A Simple and Scalable Bandwidth Sharing Mechanism for Multicast Flows, IEEE ICNP 2002, Paris, France, November 2002.
2. F. Filali and W. Dabbous, SBQ: A Simple Scheduler for Bandwidth Sharing Between Unicast and Multicast Flows, QoFIS 2002, Zürich, Switzerland, October 2002.
3. F. Filali, H. Asaeda, and W. Dabbous, Counting the Number of Members in Multicast Communication, NGC'2002, Boston, USA, October 2002.
4. F. Filali and W. Dabbous, A QoS-Aware Switching Mechanism Between the Two Modes of PIM-SM Multicast Routing Protocol, ITC Specialist Seminar on "Internet Traffic Engineering and Traffic Management" (IP2002), Würzburg, Germany, July 2002.
5. F. Filali and W. Dabbous, A new Bandwidth Sharing Scheme for Non-Responsive Multicast Flows, IEEE ICC'2002, New York, April 2002.
6. F. Filali, W. Dabbous, and F. Kamoun, On the Planning of Multi-services GEO Satellite-Terrestrial Hybrid Networks, IEEE Softcom'2001, Split, Dubrovnik (Croatia) and Ancona, Bari (Italy), October 2001.
7. F. Filali and W. Dabbous, Issues on the IP Multicast Service Behavior over the Next-Generation of Satellite-Terrestrial Hybrid Networks, IEEE ISCC'2001, Hammamet, Tunisia, July 2001.
8. F. Filali, W. Dabbous, and F. Kamoun, Efficient Planning of Satellite- Terrestrial Hybrid Networks for Multicast Applications, IEEE ICC'2001, Helsinki, Finland, June 2001.

Papers under preparation

1. F. Filali, L. Fazio, and W. Dabbous, Enhancing the Coexistence of Unicast and Multicast Sessions in DiffServ Architecture, Submitted.
2. F. Filali, G. Aniba, and W. Dabbous, A New Architecture for IP Multicast over the Next Generation of GEO Satellite Networks, Submitted to a special issue of JSAC on Broadband IP Networks via Satellites.
3. F. Filali and W. Dabbous, An Architecture for Bandwidth Sharing between Unicast and Multicast, To be submitted to Computer Communications Magazine.
4. F. Filali and W. Dabbous, Adaptation of PIM-SM to the DIPCAST GEO Satellite System, Under preparation.

Conferences Posters

1. F. Filali and W. Dabbous, Multicast Fairness-Independent and Fine- Grained AQM Mechanism for Multicast Flows, NGC 2001, London, UK, November 2001.

Research Reports

1. F. Filali and W. Dabbous, Optimization of the Deployment of GEO satellite Links in the Internet, INRIA Research Report, Number 3925, April 2000.

Projects Reports

1. F. Filali, IP Multicast Support in the DIPCAST System, DIPCAST RNRT Project, under preparation.
2. F. Filali, Adaptation of PIM-SM to the DIPCAST Transparent System, DIPCAST RNRT Project, August 2002.
3. F. Filali, A Survey of Multicast Routing Protocols Implementation, DIPCAST RNRT Project, October 2001.
4. F. Filali, State-of-the-art Survey of Multicast Routing Protocols, DIPCAST RNRT Project, June 2001.
5. F. Filali, UDLR Implementation in NS Network Simulator, Constellation RNRT Project, June 2001.

Bibliography

[1] A. Adams, J. Nicholas, and W. Siadak, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification* (Revised), IETF, draft-ietf-pim-dm-v2-02.txt, October 2002.

[2] Advanced Television Systems Committee, *ATSC Standard: Delivery of IP Multicast Sessions over ATSC Data Broadcast*, Doc. A/92, January 2002.

[3] AHG-RSAT Final Report, *Harmonisation of Terminals for Regenerative Satellite Multimedia Systems*, `http://telecom.estec.esa.nl/artes/artes1/fileincludes/documentation/ahg-rsat.cfm`, February 2001.

[4] K. C. Almeroth, *The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet Deployment*, IEEE Network, V. 14, pp. 10-20, Jan 2000.

[5] K. C. Almeroth and Y. Zhang, *Using Satellite Links as Delivery Paths in the Multicast Backbone (MBone)*, WOSBIS'98, Dallas, Texas, October 1998.

[6] S. Alouf, E. Altman and P. Nain, *Optimal Size Estimation of a Dynamic Multicast Group*, In Proc. of Infocom'02, June 2002.

[7] F. Ananaso and F. Priscoli, *The role of Satellite in Personal Communication Services*, IEEE J. Selected Areas in Communications, SAC-13, pp. 180-196, Febrary 1995.

[8] H. Asaeda, *IGMPv3 host-side implementation for NetBSD*, `http://www.inria.fr/planete/Hitoshi.Asaeda/igmpv3`

[9] S. Bajaj, L. Breslau, and S. Shenker, *Uniform versus priority dropping for layered video*, In Proc. of SIGCOMM'98, pp. 131-143, September 1998

[10] A. Ballardie, *Core Based Trees (CBT Version 2) Multicast Routing: Protocol Specification*, IETF, RFC 2189, September 1997.

[11] A. Ballardie, *Core Based Trees (CBT) Multicast Routing Architecture*, IETF, RFC 2201, Sptember 1997.

[12] T. Berners-Lee et al, *The World Wide Web*, Communications of the ACM, V. 37, N. 8, August 1994.

[13] T. Bates, et al., *Multiprotocol Extensions for BGP-4*, IETF, RFC 2283, February 1998.

[14] D. Bertsekas and R. Gallagher, *Data Networks*, Englewood Cliffs, NJ, Prentice-Hall, 1992.

[15] S. Blake, et al., *An Architecture for Differentiated Services*, IETF, RFC 2475, December 1998.

[16] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, IETF, RFC 1633, June 1994.

[17] J.-C. Bolot, T. Turletti, and Wakeman I. *Scalable feedback control for multicast video distribution in the Internet.* In Proc. of ACM SIGCOMM'94, London, England, pp. 58-67, September 1994.

[18] R. Braden, et al., *Resource ReSerVation Prtocol (RSVP): Version 1 Functional Specification*, IETF, RFC 2205, September 1997.

[19] R. Brnaudes and S. Zabele, *Requirements for Multicast Protocols*, IETF, RFC 1458, May 1993.

[20] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver, *FLID-DL: Congestion Control for Layered Multicast*, In Proc. of NGC 2000, pp. 71-81, November 2000.

[21] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, *A digital fountain approach to reliable distribution of bulk data transfer,* In Proc. of ACM SIGCOMM'98, September 1998.

[22] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, *Internet Group Management Protocol: version3*, IETF, RFC 3376, October 2002.

[23] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, *Tuning RED for web traffic*, In Proc. of ACM SIGCOMM, pp. 139-150, September 2000.

[24] J. Chuang and M. Sirbu, *Pricing multicast communications: A cost based approach*, In Proc. of INET'98, 1998.

[25] D. M. Chiu., *Some Observations on Fairness of Bandwidth Sharing*, In the Proc. of ISCC'00, Antibes, France, 2000.

[26] D. Clark and B. Marjory, *Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World*, ACM Transactions on Internet Technology (TOIT), Volume 1, Issue 1, pp. 70-109, August 2001.

[27] H. D. Clausen, H. Linder, and B. Collini-Nocker, *Internet over Direct Broadcast Satellites*, IEEE Communications Magazine, pp. 146-151, June 1999.

[28] H. D. Clausen, H. Linder, and G. Fairhurst, *Simple Encapsulation for transmission of IP datagrams over MPEG-2/DVB networks*, IETF, Internet Draft, draft-unisal-ipdvb-enc-00.txt, April 2002.

[29] A. Clerget, and W. Dabbous, *TUF : Tag-based Unified Fairness*, In Proc. of IEEE INFOCOM 2001, April 2001.

[30] "*Cutting-edge Technology for Web Content Delivery*," On-line document, available at `http://www.sightpath.com/technology/.`

[31] W. Dabbous, E. Duros, and T. Ernest, *Dynamic Routing in Networks with Unidirectional Links*, WOSBIS'98.

[32] L A. DaSilva, *Pricing for QoS-Enabled Networks: A Survey*, In the Journal of IEEE Communications Surveys, pp. 2-8, Second Quarter 2000.

[33] S. Deering, *Multicast routing in a datagram internetwork*, Ph.D. thesis, 1991.

[34] S. Deering, *Host Extensions for IP Multicasting*, IETF, RFC 1112, August 1989.

[35] S. Deering and D. Cheriton, *PIM Architecture for wide-area multicast routing*, IEEE/ACM Transactions on Networking, pp. 153-162, April 1996.

[36] A. Demers, S. Keshav, and S. Shenker, *Analysis and Simulation of a fair queueing algorithm*, Internet working: Research and Experience, V. 1, No. 1, pp. 3-26, 1990.

[37] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, *Deployment Issues for the IP Multicast Service and Architecture*, IEEE Network magazine special issue on Multicasting, January/February 2000.

[38] C. Diot, W. Dabbous, and J. Crowcroft, *Group Communication*, IEEE Journal on Selected Area in Communication. Special Issue on Group Communication, May 1997.

[39] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, *A Link Layer Tunneling Mechanism for Unidirectional Links*, RFC 3077, March 2001.

[40] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, *Protocol independent multicast sparce-mode (PIM-SM): Protocol specification*, IETF, RFC 2362, June 1998.

[41] ETSI, DVB-RCS001rev14: *Digital Video Broadcasting(DVB); Interaction Channel for Satellite Distribution Systems*, ETSI, April 2000.

[42] ETSI, EN 301 192 v1.2.1: *Digital Video Broadcasting(DVB); DVB specification for data broadcasting*, ETSI, June 1999.

[43] ETSI, EN 300 468 v1.4.1: *Digital Video Broadcasting (DVB), Specification for service Information(SI) in DVB systems*, ETSI, November 2000.

[44] ETSI, EN 300 421 v1.1.2: *Digital Video Broadcasting(DVB): Framing structure, channel coding and modulation for 11/12 Ghz satellite services*, ETSI, August 1997.

[45] ETSI, ETR 211, *Digital Video Broadcasting(DVB); Guidelines on implementation and usage of Service Information (SI)*, ETSI, August 1997.

[46] ETSI, ETR 162, *Digital broadcasting systems for television, sound and data services; Allocation of Service Information(SI) codes for Digital Video Broadcasting(DVB) systems*, ETSI, October 1995.

[47] ETSI, *Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video an Audio in satellite, cable and terrestrial broadcasting applications*, ETR 154, September 1997.

[48] European Telecommunication Standards Institute (ETSI), *Digital Video Broadcasting specification for data broadcasting*, ETSI.

[49] European Telecommunication Standards Institute (ETSI), *Multiprotocol Encapsulation*, Draft EN 301 192 V1.1.1 (1997-08).

[50] G. Fairhurst, H. D. Clausen, B. Collini-Nocker, and H. Linder, *Requirements for transmission of IP datagrams over DVB networks*, IETF, Internet draft, draft-fair-ipdvb-req-01.txt, May 2002.

[51] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, *Generic Routing Encapsulation (GRE)*, IETF, RFC 2784, March 2000.

[52] *FreeFlow the standard for global Internet content and applications delivery*, On-line document, available at `http://www.akamai.com/service/freeflow.pdf`

[53] B. Fenner, *Internet Group Management Protocol Version 2*, IETF, RFC 2236, November 1997.

[54] B. Fenner, H. Holbrook, and I. Kouvelas, *Multicast Source Notification of Interest Protocol (MSNIP)*, IETF, Internet draft, draft-ietf-magma-msnip-01.txt, November 2002.

[55] B. Fenner, M. Handley, H .Holbrook, and I. Kouvelas, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*, IETF, Internet draft, draft-ietf-pim-sm-v2-new-06.txt, December 2002.

[56] S. Floyd et al., *A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing*, IEEE/ACM Transactions on Networking, V. 5, N. 6, pp. 784-803, December 1997.

[57] S. Floyd, M. Handley, J. Padhye, and J. Widmer, *Equation-Based Congestion Control for Unicast Applications*, In Proc. of SIGCOMM 2000, August 2000.

[58] S. Floyd, V. Jacobson, and V. Random, *Early Detection gateways for Congestion Avoidance,* IEEE/ACM TON, V.1 No.4, pp. 397-413, August 1993.

[59] S. Floyd and V. Jacobson, *Link-sharing and Resource Management Models for Packet Networks*, IEEE/ACM TON, V. 3, No.4, 1995.

[60] S. Floyd, M. Handley, J. Padye, and J. Widmer, *Equation-based congestion control for unicast applications*, In Proc. ACM SIGCOMM'00, August 2000.

[61] T. Friedman and D. Towsley, *Multicast session membership size estimation*, In the Proceedings of IEEE Infocom'99, pp. 965-972, March 1999.

[62] G. Gordon and W. Morgan, *Principles of Communications Satellites*, John Wiley, 1986.

[63] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Vol:1, Chapman & Hall, USA, 1997.

[64] C. Hedrick, *Routing Information Protocol*, IETF, RFC 1058, June 1988.

[65] T. N.H. Henderson and S. N. Bhatti, *Protocol-independent multicast pricing*, In Proc of NOSSDAV'00, June 2000.

[66] S. Herzog, S. Shenker, and D. Estrin, *Sharing the "cost" of multicast trees: an axiomatic analysis*, IEEE/ACM TON, V. 5, No. 6, pp. 847-860, 1997.

[67] H. Holbrook and B. Cain, *Source-Specific Multicast for IP*, IETF, Internet draft, draft-ietf-ssm-arch-01.txt, November 2002.

[68] J. Holt and W. Peng, *Improving the PIM Routing Protocol with Adaptive Switching Mechanism between its Two Sparse Sub-Modes*, In the proceedings of the International Conference on Computer Communication and Networks, pp. 768-773, October 1998.

[69] ISO/IEC 13818-1: *Information Technology - Generic Coding of Moving Pictures and Associated Audio: Systems*, ISO/IEC, November 1994.

[70] ISO/IEC 13818-6: *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Part6, extension for Digital Storage Media Command and Control*, ISO/IEC, 1998.

[71] Information Technology, *Generic Coding of Moving Pictures and Associated Audio Information*, Part 6: Extensions for DSM-CC, ISO/IEC 13818-6: 1998/Amd, 1:1999 (E).

[72] R. Jain, *The art of computer systems performance analysis*, John Wiley and sons QA76.9.E94J32, 1991.

[73] M. Johanson, *Scalable Video Conferencing Using Subband Transfor Coding and Layered Multicast Transmission*, International Conference on Signal Processing Applications and Technonlogy, November 1999.

[74] F. Kelly, A. Maulloo and D. Tan, *Rate control in communication networks: shadow prices, proportional fairness and stability*, Journal of the Operational Research Society 49, pp. 237-252, 1998.

[75] A. Legout, J. Nonnenmacher, and E. W. Biersack, *Bandwidth Allocation Policies for Unicast and Multicast Flows,* IEEE/ACM TON, V.9 No.4, August 2001.

[76] X. Li, S. Paul, and M. Ammar, *Multi-Session Rate Control for Layered Video Multicast*, In Proc. of Multimedia Computing and Networking, January 1999.

[77] C. Liu and J. Nonnenmacher. *Broadcast audience estimation.* In Proc. of IEEE INFOCOM'00, V. 2, pp. 952-960, March 2000.

[78] D. Lin and R. Morris, *Dynamics of Random Early Detection*, In Proc. of ACM SIGCOMM'97, September 1997.

[79] M. Luby, V. Goyal, and S. Skaria, *Wave and Equation Based Rate building block*, IETF, Internet draft, draft-ietf-rmt-bb-webrc-04.txt, December 2002.

[80] M. Luby, L. Vicisano, and A. Haken, *Reliable Multicast Transport Building Block: Layered Congestion Control*, IETF, Internet draft, draft-ietf-rmt-bb-lcc-00.txt, November 2000.

[81] G. Malkin, *RIP version 2 - Carrying Additional Information*, IETF, RFC 1723, November 1994.

[82] A. Mankin, et al., *IETF Criteria for evaluating Reliable Multicast Transport and Applications Protocols*, IETF RFC 2357, June 1998.

[83] G. Maral and M. Bousquet, *Satellite Communications Systems*, John Wiley, 1993.

[84] S. McCanne and S. Floyd, *Ucb/lbnl/vint network simulator (ns) version 2.1b6,* `http://www-mash.cs.berkeley.edu/ns/`, June 2000.

[85] A. Medina, A. Lakhina, I. Matta, and J. Byers, *BRITE: An Approach to Universal Topology Generation*, In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems - MASCOTS '01, Cincinnati, Ohio, August 2001 `http://cs-www.bu.edu/brite/`.

[86] D. Meyer and B. Fenner, *Multicast Source Discovery Protocol (MSDP)*, IETF, draft-ietf-msdp-spec-13.txt, work in progress, November 2001.

[87] K. Miller, et. al., *StarBurst Multicast File Transfer Protocol (MFTP) Specification*, Work in Progress.

[88] J. Moy, *Open Shortest Path First - OSPF version 2*, IETF, RFC 1583, March 1994.

[89] J. Moy, *Multicast Extensions to OSPF*, IETF, RFC 1583, March 1994.

[90] MPEG, *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information. Part 1: Systems, ISO/IEC 13818-1*, November 1994.

[91] J. Musey. Presentation, *29th Annual Banc of America Securities Investment Conference*, San Francisco, September 1999.

[92] J. Nonnenmacher and E. Biersack. *Optimal multicast feedback.* In Proc. of IEEE INFOCOM'98, San Francisco, CA USA, V. 3, pp. 964-971, March 1998.

[93] J. Nonnenmacher and E. W. Biersack, *Scalable Feedback for Large Groups*, IEEE/ACM Transactions on Networking, Vol. 7, Issue 3, pp. 375-386, 1999.

[94] K. Obraczka, *Multicast Transport Protocols: A Survey and Taxonomy*, IEEE Communications Magazine, V. 36, N. 1, pp. 94-102, January 1998.

[95] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, *Modeling TCP throughput: a simple model and its empirical validation*, In Proc. of ACM SIGCOMM, Vancouver, Canada, September 1998.

[96] A. Parekh and R. G. Gallager, *A generalized processor sharing approach to flow control - the single node case*, In Proc. of IEEE INFOCOM'92, v. 2, pp. 915-924, May 1992.

[97] J. Postel, *Internet Protocol - Protocol Specification*, IETF, RFC 791, September 1981.

[98] J. Postel, *Transmission Control Protocol*, STD 7, IETF, RFC 793, September 1981.

[99] C. Perkins, *IP encapsulation within IP*, IETF, RFC 2003, October 1996.

[100] T. Pratt and C. Bostian, *Satellite Communications*, John Wiley, 1986.

[101] G. Pujolle, *Mobile and Satellite Networks: New Trends*, In the Proceedings of the 5^{th} Conference on Computer Communications AFRICOM-CCDC'98, Tunisia 18-21 October 1998.

[102] L. Rizzo, *pgmcc: a TCP-friendly single-rate multicast congestion control scheme*, In Proc. of ACM SIGCOMM'00, pp. 17-26, August 2000.

[103] D. Rubenstein, J. Kuros, and D. Towsly, *The Impact of Multicast Layering on Network Fairness*, In Proc. of ACM SIGCOMM'99, September 1999.

[104] N. K. G. Samaraweera, *Return Link Optimization for Internet Service Provision Using DVB-S Networks*, SIGCOMM Computer Communication Review, 29:(3), July 1999.

[105] N. Samaraweera and G. Fairhurst, *High Speed Internet Access Using Satellite-based DVB Networks*, International Network Conference (ICN'98), Plymouth, UK, 1998.

[106] S. Sankar and L. Tassiulas, *Fair Allocation of Discrete Bandwidth Layers in Multicast Networks*, In Proc. of IEEE Infocom'2000, March 2000.

[107] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, IETF, RFC 1889, January 1996.

[108] J. Shapiro, D. Towsley, and J. Kurose, *Optimization-Based Congestion Control for Multicast Communications*, In Proc of IEEE INFOCOM' 2000, March 2000.

[109] E.C. Sheck, S. K. Dao, Y. Zhang, and D. V. Buer, *Dynamic Multicast Information Dissemination in Hybrid Satellite-Wireless Networks*, in the Proceedings of the ACM international workshop on Data engineering for wireless and mobile access, pp. 30-35, Seattle WA USA, August 1999.

[110] D. C. Stephens, J.C.R. Bennet, and H. Zhang, *Implementing scheduling algorithms in high speed networks*, IEEE JSAC, V. 17, No. 6, pp. 1145-1159, June 1999.

[111] I. Stoica, S. Shenker, and H. Zhang, *Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks*, In Proc. of SIGCOMM'98.

[112] Society of Cable Telecommunications Engineers, Engeneering Committee, Digital Video Subcommittee, *IP Multicast for Digital MPEG Networks*, DVS 311, 2001.

[113] D. Taubman and A. Zakhor, *Multirate 3-D Subband Coding of Video*, In Proc. of IEEE Transactions on Image Processing, Vol. 3, No. 5., September 1994.

[114] D. Thaler, *Border Gateway Multicast Protocol (BGMP): Protocol Specification*, Internet draft, draft-ietf-bgmp-spec-03.txt, June 2002.

[115] H. Tzeng and K. Siu, *On Max-Min Fair Congestion Control for Multicast ABR Service in ATM*, IEEE JSAC, Vol. 15, April 1997.

[116] L. Vicisano, L. Rizzo, and J. Crowcroft, *TCP-like congestion control for layered multicast datda transfer*, In Proc. of IEEE INFOCOM'98, San Francisco, CA, March 1998.

[117] R. Vida et al., *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, IETF, Internet Draft, draft-vida-mld-v2-02.txt, work in progress, January 2002.

[118] D. Waitzman, C. Partridge, and S. Deering, *Distance vector multicast routing protocol (DVMRP)*, IETF, RFC 1075, November 1988.

[119] J. Widmer and M. Handley, *TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification*, IETF, Internet draft, draft-ietf-rmt-bb-tfmcc-00.txt, November 2001.

[120] Y. Zhang, D. De Lucia, B. Ryu, and S. K. Dao, *Satellite Communications in the Global Internet: Issues, Pitfalls, and Potential*, INET'97, 1997.

[121] L. Zhang, S. Michel, K. Nguyen, A. Rosenstein, S. Floyd, and V. Jacobson, *Adaptive Web Caching: Towards a New Global Caching Architecture*, in Proc. of 3^{rd} International WWW Caching Workshop, June 1998.

List of Abbreviations

It's not what you say; it's what they hear.

ADSL Asymmetric Digital Subscriber Line

AQM Active Queue Management

ATM Asynchronous Transfer Mode

BER Bit Error Rate

BSR BootStRap mechanism

CAT Conditional Access Table

CBQ Class Based Queuing

CBR Constant Bit Rate

CBT Core Based Tree

CDN Content Delivery Network

CoS Class of Service

CRC Cyclic Redundancy Check

CSFQ Core-Stateless Fair Queuing

CSC Common Signaling Channel

DiffServ Differentiated Service

DSCP DiffServ Code Point

DTCP Dynamic Tunneling Configuration Protocol

DTH Direct To Home

DULM Data Unit Labeling Method

DVB Digital Video Broadcasting

DVB-RCS DVB-Return Channel via Satellite

DVB-S DVB Satellite

DVB-TM DVB Technical Module

DVMRP Distance Vector Multicast Routing Protocol

EBU European Broadcast Union

FLID-DL Fair Layered Increase/Decrease with Dynamic Layering

FRED Flow RED

FTP File Transfer Protocol

GEO Geosynchronous Earth Orbit

GRE Generic Routing Encapsulation

HDLC High Level Data Link Control

HDSL High-data-rate Digital Subscriber Line-

HFC Hybrid Fiber/Coaxial

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

IP Internet Protocol

ISDN Integrated Services Digital Network

ISM Internet Standard Multicast

ISP Internet Service Provider

ITV Interactive TV

LAN Local Area Network

LEO Low Earth Orbit

LLTM Link Layer Tunneling Mechanism

MAL Multicast Allocation Layer

MBGP Multicast Border Gateway Protocol

Mbone Multicast backbone

MCT Multicast Counting Table

MEO Medium Earth Orbit

MFQ Multicast Fair Queuing

MFTP Multicast File Transfer Protocol

MMT Multicast Mapping Table

MPE Multi-Protocol Encapsulation

MPEG Moving Picture Expert Group

MOSPF Multicast Open Shortest Path First

MTU Maximum Transfer Unit

MSDP Multicast Source Discovery Protocol

NCC Network Control Center

NCR Network Clock Reference

NIT Network Information Table

PAT Program Association Table

PES Packetized Elementary Streams

PHB Per-Hop Behavior

PID Packet Identifier

PIM Protocol Independent Multicast

PIM-DM Protocol Independent Multicast-Dense Mode

PIM-SM Protocol Independent Multicast-Sparse Mode

PMT Program Map Table

POP Point Of Presence

POTS Plain Old Telephone Service

PPP Point to Point Protocol

PSI Program Specific Information

PUSI Payload Unit Start Indicator

QoS Quality of Service

OBP On-Board Processing

OBS On-Board Switching

OSPF Open Shortest Path First

RED Random Early Detection

RCST Return Channel via Satellite Terminal

RIP Routing Information Protocol

RLC Receiver-driven Layered Congestion control

RMT Reliable Multicast Transport

RMRG Reliable Multicast Research Group

RNRT National Network of Telecommunication Research

RP Rendez-vous Point

RPT RP Tree

RSVP Resource Reservation Setup Protocol

RTCP Real time Transmission Control Protocol

RTP Real time Transmission Protocol

RTT Round Trip Time

SBQ Service-Based Queuing

SCPC Single Channel Per Carrier

SIR Switching Initiator Receiver

SMRP Satellite Multicast Routing Protocol

SPT Shortest Path Tree

SRM Scalable Reliable Multicast

SSM Single Source Multicast

ST Satellite Terminal

TCP Transmission Control Protocol

TFMCC TCP-Friendly Multicast Congestion Control

TFRC TCP-Friendly Rate Control

TS Transport Stream

TTL Time To Leave

TUF Tag-based Unified Fairness

UDL UniDirectional Link

UDLR UniDirectional Link Routing

UDP User Datagram Protocol

VDSL Very-high-rate Digital Subscriber Line

VOD Video On Demand

VSAT Very Small Aperture Terminals

WDM Wavelength Division Multiplexing

WG Working Group

WRED Weighted RED

WRR Weighted Round Robin

WWW World Wide Web

Index

Abstract

The first part of this dissertation focuses on bandwidth sharing among unicast and multicast flows. We first propose MFQ (Multicast Fair Queuing), a simple and scalable active queue management mechanism that aims to share the bandwidth fairly between competing multicast flows using a pre-defined inter-multicast fairness function. Then, we describe SBQ (Service Based Queuing), a simple scheduler which uses only two queues in order to enhance the network resource sharing between unicast and multicast flows. A novel method for re-marking multicast packets in DiffServ networks based on the number of downstream members is designed and analyzed. The performance of all these schemes is evaluated under a variety of realistic configurations and traffic patterns. Results indicate that MFQ achieves the expected bandwidth sharing between multicast competing flows, while SBQ guarantees a fair allocation of the bandwidth between unicast and multicast flows. Finally, we describe an extension to the multicast service model that enables the counting of group members at the senders as well as at the intermediate routers in the multicast delivery tree.

The second part of this dissertation addresses the issue of supporting IP multicast in networks including heterogeneous transmission media. In a first step, we consider the GEO transparent satellites and we determine the adaptations that should be added to the different routing protocols in order to take into account the inherent characteristics of these systems and those of the multicast applications that will profit from them. In a second step, we focus on the deployment of IP multicast over the next-generation of GEO satellites supporting multiple spot-beams and on-board switching technologies. We propose a new encapsulation scheme optimized for IP multicast and we develop two alternative approaches that could be used to enable switching in the on-board satellite processor: the self-routing approach and the label switching. We present SMRP (Satellite Multicast Routing Protocol) that process the PIM-SM messages received by the system entities from terrestrial nodes. At the end of this dissertation, we present and demonstrate the advantages of a novel switching mechanism between the two modes of PIM-SM protocol. This mechanism uses a coordination between the members concerned by the switching to decide whether it is better to switch or not. Furthermore, it could take into account the network and members constraints.

Keywords: Multicast bandwidth sharing, Multicast in DiffServ networks, Multicast over transparent GEO satellites, Multicast over the next-generation of GEO satellites, PIM-SM protocol.

Résumé

La première partie de cette thèse est consacrée au problème de partage de ressources réseaux entre les flux multicast. Tout d'abord, nous proposons MFQ (Multicast Fair Queuing), un mécanisme simple de gestion active de files d'attente qui passe à l'échelle et permettant le partage équitable de la bande passante entre les flux multicast et ceci en utilisant une fonction d'équité configurée à l'avance. Ensuite, nous décrivons SBQ (Service Based Queuing), un ordonnanceur simple utilisant deux files d'attentes afin d'améliorer le partage de ressources entre les flux unicast et multicast. Une nouvelle méthode de re-marquage de paquets multicast dans un réseau DiffServ en se basant sur le nombre de membres est également proposée. Les performances de tous ces mécanismes sont évalués pour plusieurs configurations et trafics réels. Les résultats montrent que MFQ permet d'obtenir le partage de la bande passante entre les flux multicast attendu et que SBQ garantie une allocation équitable de ressources réseaux entre les flux unicast et multicast. Enfin, nous décrivons une extension du service multicast permettant le comptage de nombre de membres dans un groupe multicast aussi bien au niveau de la source multicast qu'au niveau des routeurs intermédiaires.

La deuxième partie de cette thèse examine le problème de support de l'IP multicast dans les réseaux intégrant des supports de transmission hétérogènes. Dans une première étape, nous considérons les satellites GEO transparent et nous déterminons les adaptations qui doivent être ajoutées aux différents protocoles de routage multicast. Dans une deuxième étape, nous nous focalisons sur le déploiement de l'IP multicast dans la nouvelle génération de satellites GEO supportant les technologies des multiples sopt-beams et de la commutation à bord. Nous proposons une nouvelle méthode d'encapsulation optimisée pour l'IP multicast et nous développons deux approches permettant la commutation des paquets à bord du satellite: l'approche self-routing et l'approche label switching. Nous présentons le protocole SMRP (Satellite Multicast Routing Protocol) qui traite les messages PIM-SM messages reçus par le entités du système. À la fin de cette thèse, nous présentons et nous montrons les avantages d'un nouveau mécanisme de commutation entre les deux modes du protocole PIM-SM. Ce mécanisme utilise une coordination entre les membres concernés par la commutation. De plus, il prend en compte les contraintes réseaux et les besoins de membres.

Mots-clés: Partage de la bande passante multicast, Multicast dans les réseaux DiffServ, Multicast sur les satellites GEO transparents, Multicast sur les satellites GEO de nouvelle génération, Protocole PIM-SM.