UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES
U.F.R. DE SCIENCES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

Discipline : **INFORMATIQUE**

présentée et soutenue publiquement

par

Marcelo DIAS DE AMORIM

le 15 juin 2001

Titre :

DISTRIBUTION DE FLUX EN COUCHES DANS LES RÉSEAUX MULTICAST HÉTÉROGÈNES

Directeurs de thèse :

Guy Pujolle et Otto Carlos Muniz Bandeira Duarte

JURY

Rapporteur	M.	Ken Chen	Professeur à l'Université Paris XIII
Rapporteur	M.	Roland Schutz	Directeur à EolRing
Examinateur	Mme.	Marie-Claude Gaudel	Professeur à l'Université Paris XI
Examinateur	M.	Serge Fdida	Professeur à l'Université Paris VI
Examinateur	M.	Jean-Pierre Claudé	Professeur à l'Université de Versailles
Examinateur	M.	Otto Carlos M. B. Duarte	Professeur à l'UFRJ
Examinateur	M.	Guy Pujolle	Professeur à l'Université Paris VI

À ma famille, fiancée et amis.

Remerciements

À mes parents Yêda et Paulo Roberto qui m'ont offert le cadeau d'une bonne éducation, le respect aux autres et la volonté de toujours y arriver. À mes soeurs, nièce et toute ma famille.

À ma très chère Pascale, pour son amour dont j'ai tant besoin. Je remercie également les familles Lemuhot et Poulaine de m'avoir accueilli aussi bien.

Je remercie mes directeurs de thèse, Otto Carlos Muniz Bandeira Duarte, Professeur à l'Université Fédérale de Rio de Janeiro, et Guy Pujolle, Professeur à l'Université Paris VI. Simplement merci !

J'exprime ma sincère gratitude à Ken Chen, Professeur à l'Université Paris XIII, et à Roland Schutz, Ingénieur à EolRing, d'avoir accepté de rapporter cette thèse. Je remercie très fortement Professeur Serge Fdida, qui m'a fait l'honneur de présider le jury. Je tiens aussi à remercier Professeurs Jean-Pierre Claudé et Marie-Claude Gaudel d'avoir accepté l'invitation de participer au jury de cette thèse.

Je tiens aussi à exprimer ma profonde reconnaissance à mes chers amis en France et au Brésil. Je ne prends pas le risque de les lister ici – ma faible mémoire me trahirait. Néanmoins, je pense très fort à vous tous.

Finalement, ma reconnaissance va aussi à l'institution brésilienne CAPES, pour son appui, et à l'accord CAPES/COFECUB, sans lesquels cette thèse n'existerait pas.

Que toutes les personnes qui m'ont aidé à un moment ou à un autre pendant la réalisation de cette thèse trouvent ici l'expression de ma gratitude.

Résumé de la thèse présentée à l'Université de Versailles Saint-Quentin-en-Yvelines comme partie des exigences pour l'obtention du grade de Docteur en Informatique.

Distribution de flux en couches dans les réseaux multicast hétérogènes

Marcelo Dias de Amorim

Juin 2001

Une des caractéristiques des réseaux multicast à large échelle est l'hétérogénéité des récepteurs en terme de capacité de réception. Un flux de données a donc son débit contrôlé par les récepteurs les moins ou les plus rapides. Dans le premier cas, on observe une sous utilisation des ressources pour les récepteurs les plus puissants, tandis que dans le second cas un certain nombre de récepteurs est obligé de quitter la session par manque de capacité. En plus de l'hétérogénéité des récepteurs, le système doit aussi contourner la variation de bande-passante disponible dans les liens. Cette thèse propose un système adaptatif de contrôle du débit de transmission de la source en utilisant la technique de codification hiérarchique. Le contrôle adaptatif est utilisé pour maintenir le débit de transmission conformement aux ressources disponibles dans les liens. La codification hiérarchique (en multi-couches) vise à régler l'hétérogénéité des récepteurs.

Nous proposons dans cette thèse un système de contrôle des débits des couches basé sur des paquets d'acquittement envoyés par les récepteurs. Ces paquets sont combinés dans les nœuds intermédiaires pour éviter l'implosion à la source. La première partie du travail propose l'Algorithme Direct qui optimise le calcul des couches afin d'augmenter la qualité globale de la session multicast dans les récepteurs. Le calcul de la qualité de la communication est basé sur la dégradation (débit demandé moins le débit alloué) dans les récepteurs. Nous vérifions que pour certaines applications l'utilisation d'une seule métrique ne correspond pas au cas idéal. Nous proposons donc ensuite une approche multi-métriques qui prend en compte les limitations de l'approche avec un seul paramètre. La troisième et dernière partie de cette thèse est une extension de la première partie mais considère que le réseau n'est pas capable d'exécuter les fonctions de fusion de paquets. Nous proposons un système de contrôle aux bouts (source et récepteurs) qui utilise un mécanisme de temporisateurs et d'échange de messages entre les récepteurs pour supprimer les paquets d'acquittement avant qu'ils ne soient envoyés à la source. L'ensemble du système se montre efficace en ce qui concerne la réponse face aux variations de l'état du réseau et très équitable en terme de degré de satisfaction des récepteurs.

Abstract of the thesis presented to the University of Versailles Saint-Quentin-en-Yvelines as partial fulfillment of the requirements for the degree of Doctor in Computer Science

Distribution of multi-layered flows in heterogeneous multicast networks

Marcelo Dias de Amorim

June 2001

One of the characteristics of large-scale multicast networks is the heterogeneity in terms of receivers' receiving capacities. A data flow is controlled by either the slowest receivers or by the fastest ones. In the first case, network resources are underutilized, whereas in the second case the slowest receivers are obliged to leave the multicast session because they miss enough capacity. Beyond the problem of heterogeneous receivers, the system must deal with variable avaiable bandwidth in the links. This thesis proposes an adaptive system where the data rate at the source is controlled by using hierarchical coding techniques. The control system keeps the source's throughput in conformity with network resources. The hierarchical (multi-layering) coding aims at reducing heterogeneity among receivers.

We propose a control system that computes the rates of the layers based on feedback packets sent by the receivers. These packets are combined in internediate nodes in order to avoid feedback implosion at the source. The first part of the work proposed the Direct Algorithm that optimizes the rates of the layers so as to increase the global quality of the multicast session. This computation is based on the degradation (difference between the required rate and the received rate) at the receivers. We note that for some applications the inefficiency of using a single metric. We then propose a multi-criteria approach that takes into account the limitations of the single-metric method. The third part of this thesis is an extension of the first one but consider that the network is unable to execute the merging procedure. We propose an end-system control (at the source and receivers) that uses a timer mechanism and message exchanging in order to supress feedback before they are sent to the source. The whole system shows to be efficient and adaptive to network state changes and fair in terms of the degree of satisfaction at the receivers.

Resumo da tese apresentada à Université de Versailles Saint-Quentin-en-Yvelines como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências da Computação (Dr.)

Distribuição de fluxos em camadas em redes multicast heterogêneas

Marcelo Dias de Amorim

Junho de 2001

Uma das características das redes multicast em larga escala é a heterogeneidade dos receptores em termos de capacidade de recepção. Um fluxo de dados tem sua vazão controlada ou pelos receptores mais lentos ou pelos mais rápidos. No primeiro caso, observa-se uma sub-utilização dos recursos pelos receptores mais velozes, ao passo que no segundo caso uma parte dos receptores deve abandonar a sessão por falta de capacidade. Além da heterogeneidade dos receptores, o sistema deve contornar a variação de banda-passante disponível nos enlaces. Esta tese propões um sistema adaptativo de controle da vazão da fonte através da técnica de codificação hierárquica. O controle adaptativo é utilizado para manter a vazão de transmissão conforme os recursos disponíveis nos enlaces. A codificação hierárquica (em múltiplas camadas) visa contornar a heterogeneidade dos receptores.

Esta tese propõe um sistema de controle da vazão das camadas baseado em pacotes de reconhecimento enviados pelos receptores. Este pacotes são combinados nos nós intermediários de maneira a se evitar implosão de pacotes na fonte. A primeira parte do trabalho consiste no Algoritmo Direto que otimiza o cálculo das camadas de forma a aumentar a qualidade global da sessão multicast nos receptores. O cálculo de qualidade da comunicação é baseado na degradação (vazão pedida menos vazão recebida) nos receptores. Observa-se que por certas aplicações a utilização de uma única métrica não corresponde ao caso ideal. É proposto então um sistema multi-métricas que considera as limitações do método com apenas um parâmetro. A terceira parte da tese é uma extensão da primeira parte, mas considera que a rede é incapaz de executar as funções de fusão de pacotes. É proposto um sistema de controle nos elementos fins (fonte e receptores) que utiliza um mecanismo de temporisadores e de troca de mensagens entre os receptores de forma a suprimir os pacotes de reconhecimento antes mesmo que eles sejam enviado à fonte. O sistema como um todo se mostra eficiente em relação àas variações do estado da rede e equilibrado em termos do grau de satisfação dos receptores.

Mots-clés

Applications adaptatives

Vidéo multi-couches

Dégré de qualité

Réseaux multicast

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Un des fondements de la société moderne est sans doute la maîtrise de l´information, et plus précisément de son stockage et de sa transmission. Avec l´arrivée des systèmes de communication et récemment des réseaux d´ordinateurs, la société moderne s´est retrouvée face à un nouveau progrès de l´humanité. Il est clair que l´Internet joue un des rôles les plus importants. En 2000 le mot “web” a été élu le mot des années 90 selon l’*American Dialect Society*, ayant été battu uniquement par le mot “elle”, qui a été élu le mot du centenaire.

La recherche et le développement des réseaux[1] dans tous leurs aspects sont très intenses depuis plusieurs années déjà, mais beaucoup de questions restent encore sans réponse. Parmi ces nombreux domaines, les communications adaptatives du type multicast sont particulièrement intéressantes.

D´une façon très brève, nous pouvons définir une communication du type multicast comme la diffusion contrôlée de l’information à des récepteurs appartenant à un même groupe d’intérêt. Le trafic multicast permet d’envoyer un paquet vers plusieurs destinations sans le dupliquer. Deux notions sont à retenir :

- le message envoyé ne l’est qu’à un groupe fermé de destinataires qui auront fait la demande de participer au groupe ;
- un seul paquet est envoyé à un routeur multicast. Cela permet une réelle économie

[1]Nous allons utiliser désormais le terme “réseaux” pour désigner les “réseaux d´ordinateurs.”

de bande passante ; les routeurs se chargent de copier et d'envoyer les paquets vers les différentes sorties des nœuds.

Un des plus grands problèmes intrinsèques aux communications du type multicast est l´hétérogénéité des récepteurs. Les données qui arrivent aux différentes destinations peuvent traverser des chemins entièrement distincts en terme de bande-passante disponible dans les liens. La cause de cette hétérogénéité peut être la vitesse variable des liens et/ou le niveau de congestion dans les nœuds du à la quantité de trafic transportée. De la même manière, une limitation de mémoire ou de vitesse d´exécution peut avoir lieu dans les systèmes terminaux.

Malgré ces limitations, nous pouvons constater qu´un nombre croissant d´applications du type multicast apparaît chaque jour. De plus, ces applications exigent des garanties de qualité de service (QoS – *Quality of Service*) qui n´étaient pas prévues lors de la conception des protocoles de l´Internet. C´est ainsi que d´autres architectures ont été proposées avec le but de fournir aux trafics un niveau minimum de QoS. Contrairement à ce qui a été prévu, ces architectures n'ont pas remplacé la base déjà installée, et les services *best-effort* continuent d'exister. La tactique naturelle est d´introduire dans les systèmes existants des mécanismes capables d´ajuster le comportement inadapté d´autrefois aux nouveaux types de trafics [1, 2].

Une des techniques proposées pour offrir aux trafics un minimum de garantie est la réservation de ressources, qui a été démocratisée avec l´arrivée du protocole RSVP [3, 4]. Cependant, l´utilisation de réservation ne peut être la seule solution. En effet, il est nécessaire de connaître les caractéristiques du trafic au moment de l´établissement de la réservation. Malgré l'existence de nombreux travaux dans ce domaine [5, 6, 7, 8, 9, 10, 11, 12], il est évident qu´il n´est pas toujours facile de caractériser un trafic à l´avance. Si la définition des paramètres n´est pas réussie, les ressources peuvent être sous-utilisées ou indisponibles. Dans le premier cas, si un trafic est spécifié comme ayant un débit moins important que son débit réel, une réservation au dessous de la valeur idéale sera faite ; cela se traduit par une sous-utilisation des ressources. De la même manière, le trafic peut être décrit comme ayant un débit plus important que son débit réel – dans ce cas le réseau souffre d´un gâchis de ressources.

Plus récemment, la technique qui se montre la plus adaptée aux réseaux existants consiste à introduire dans les applications la caractéristique d'*adaptation* [1, 13, 14, 15, 16, 17, 18, 19, 20, 21].

Considérons le modèle de service de l'Internet qui est du type *best-effort*. Ainsi, il n'existe ni borne supérieure pour le délai ni borne inférieure pour la bande-passante disponible. La qualité du service offert par le réseau change dans le temps ; un contrôle de congestion effectif pourrait mener à des variations aléatoires et importantes de la bande-passante disponible. Les applications modernes doivent donc être capables de réagir d'une manière robuste à ces variations et opérer d'une façon dynamique dans des conditions flottantes.

1.1 Applications adaptatives

Une grande part de la discussion autour de la réservation de ressources considère qu'une partie des applications possède des besoins fixes. Ces applications ne fonctionnent que si le réseau leur offre la quantité juste de ressources demandée. Par contre, de nombreuses applications possèdent un degré quelconque de tolérance aux variations de ressources. Elles peuvent donc d'une manière ou d'une autre s'adapter.

Il y a deux types principaux d'adaptation du point de vue du réseau : adaptation du délai et adaptation du débit.

1.1.1 Adaptation du délai

Beaucoup d'applications possèdent la faculté de s'adapter facilement au délai subi dans le réseau. Un des exemples classiques est l'outil pour la transmission de l'audio `vat` [22]. Une estimation actualisée des déviations moyenne et standard du délai est maintenue par l'outil. Une fois ces paramètres connus, il est facile de déterminer la taille idéale du "play-out buffer". Les distorsions dues aux délais variables (gigue) sont ainsi éliminées.

1.1.2 Adaptation du débit

Les applications s'ajustent à une certaine quantité de ressources disponibles si elles sont capables d'adapter leurs débits à l'état actuel du réseau.

Très fréquemment, le signal devant être transmis est redondant. Par exemple, une image perdue peut être remplacée par l'image précédente. Un segment audio peut également être obtenu par l'interpolation des segments précédent et suivant. Bien évidement, ces techniques sont dépendantes du média et changent selon l'algorithme de compression utilisé.

Il y a en effet trois façons différentes pour une application de s'adapter à un changement de débit :

– *compression variable* : les sources sont constamment informées sur l'état de congestion du réseau et sur la qualité de réception aux destinations. Le taux de compression est donc ajusté pour accommoder les variations des paramètres ;
– *codage hiérarchique* : les données sont partagées entre différents flux suivant une hiérarchie pré-définie. Si le réseau n'est pas congestionné, tous les paquets arrivent. Sinon, les paquets de basse priorité sont rejetés ;
– *approche hybride* : fonctionne comme une fusion des deux techniques présentées ci-dessus, où la variation de débit est appliquée sur les différents niveaux hiérarchiques. Comme nous allons préciser dans la section suivante, cette approche sera utilisée tout au long de cette thèse.

1.2 Options pour la vidéo adaptative

Avec l'évolution des applications adaptatives, la façon dont les médias sont transportés (notamment la vidéo) a beaucoup changé au fil des temps. À présent, on peut aisément les classer en trois groupes principaux [23] :

– flux simples ;
– flux multiples ;
– flux multi-couches.

Chacune de ces approches se montre plus indiquée pour certains scénarios spécifiques ; l'affirmation qu'une est meilleure qu'une autre n'est donc pas exacte. Nous allons présenter maintenant ces approches et discuter des cas pour lesquels elles se montreraient les plus indiquées.

1.2.1 Approche par flux simples

On dit qu'une transmission suit l'approche par flux simples si un seul flux vidéo est transmis de la source vers les récepteurs et des messages d'acquittement sont envoyés dans le sens contraire[2] [14, 24, 25]. À partir de ces messages d'acquittement, la source ajuste le débit de transmission pour mieux servir les récepteurs. Cette approche est celle qui donne le plus intéressant rapport 'qualité de la vidéo' / 'quantité de données'. La transmission par flux simples est aussi la plus indiquée pour les communications unicast et pour les sessions multicast homogènes, où les chemins entre la source et les récepteurs sont équivalents en termes de ressources disponibles.

L'hétérogénéité dans les réseaux est le principal inconvénient pour l'approche par flux simples. Si les récepteurs possèdent des capacités très variables (à cause d'une limitation des ressources du réseau ou du système terminal), la source sera confrontée au problème de décider quel débit sera le plus indiqué pour la session multicast entière. Par conséquent, les récepteurs "puissants" seront sous-utilisés si la source choisit un débit contrôlé par les récepteurs plus "lents". D'un autre côté, les routes vers les récepteurs lents peuvent se congestionner si la source envoie avec un débit correspondant aux récepteur puissants.

1.2.2 Approche par flux multiples

Cette approche est en fait une extension de l'approche par flux simples présentée ci-dessus. L'objectif est simplement de limiter les effets de l'hétérogénéité du réseau. La source transmet des multiples flux obtenus à partir de la même vidéo originelle. Chacun de ces flux est caractérisé par différents débits et qualités visuelles, et est obtenu en utilisant des paramètres variables de compression au moment du codage [26, 27, 28].

[2] Cela s'applique, bien sûr, aux types d'applications qui nous intéressent.

Chaque flux est envoyé avec une adresse multicast différente. Ainsi, un récepteur s'abonne à la session multicast qui convient le mieux à ses besoins (ou capacité). Comme pour l'approche par flux simples, des messages d'acquittement peuvent être utilisés pour que la source ajuste dynamiquement le débit de transmission.

Pour un bon fonctionnement de cette approche, le système doit permettre aux récepteurs de changer assez efficacement de groupe multicast. Ce dynamisme est indispensable étant donné que la capacité des récepteurs peut être très variable dans le temps. L'abonnement/désabonnement des groupes multicast doit être donc exécuté assez rapidement.

Une autre limitation de cette approche est liée à la mauvaise utilisation des ressources du réseau. Malgré la simplicité de ce système en ce qui concerne le traitement de l'hétérogénéité des récepteurs, le réseau doit transporter des données redondantes, puisque les différents flux sont en fait la même séquence vidéo.

1.2.3 Approche par flux multi-couches

Ce schéma est basé sur l'habileté d'un certain nombre de codeurs vidéo à diviser leur signal en *couches*. Il y une couche principale, nommé *couche de base*, et une ou plusieurs *couches d'amélioration*. La couche de base peut être indépendemment décodée et donne le niveau minimum de qualité de la vidéo. Les couches d'amélioration ne peuvent être décodées que si la couche de base est présente. Les informations portées par les couches d'amélioration sont ajoutées à celles de la couche de base dans le but de perfectionner la qualité visuelle de la vidéo. Ainsi, la couche de base plus deux couches d'amélioration donnent une vidéo possédant une qualité supérieure à celle obtenue avec la couche de base et seulement une couche d'amélioration.

Un système de communication multicast qui utilise l'approche multi-couches pourrait donc envoyer chaque couche à une adresse multicast différente. Les récepteurs s'abonneraient bien évidement à la session correspondante à la couche de base et, tant que leurs capacités leur permettraient, aux sessions relatives aux autres couches. Encore une fois, on doit ici fournir des mécanismes protocolaires permettant aux récepteurs de changer de groupe multicast selon les variation de capacités subies lors des modifications de l'état du réseau [16, 17, 18, 20, 29, 30, 31, 32, 33, 34].

Cette approche se montre la plus intéressante pour les réseaux multicast hétérogènes, puisqu'elle offre une gamme d'options de débit aux récepteurs sans l'inconvénient de la redondance de données intrinsèques à l'approche par flux multiples.

Dans le cadre de cette thèse, le but est d'optimiser la qualité des applications lorsqu'il s'agit de la distribution multicast dans les réseaux hétérogènes. L'approche par multi-couches est celle qui s'intègre le mieux à nos besoins et a donc été choisie comme un des composants de base de notre système.

1.3 Contexte

Le type de réseaux que nous allons étudier est constitué d'une source et d'un certain nombre de récepteurs. La source génère une ou plusieurs couches de données[3] qui sont transmises comme des flux distincts, où chaque flux correspond à une adresse multicast différente. Les récepteurs, étant indépendants et libres de s'abonner aux groupes multicast qui les intéressent, choisissent le nombre de couches qui s'adapte le mieux à leurs capacités.

Nous distinguons ici une session multicast d'un "collectif multicast". Une vidéo multi-couches dont chaque couche possède une adresse multicast différente est considérée comme un collectif multicast (chaque couche est associée à une session multicast indépendante). Ainsi, un collectif multicast peut être lié à une ou plusieurs sessions multicast.

La problématique dans laquelle s'inscrit cette thèse est l'amélioration de la qualité des applications dans les réseaux hétérogènes dynamiques. Dynamisme signifie, ici, que la capacité de réception d'un récepteur quelconque est variable dans le temps. Notre système assemble des mécanismes qui permettent à la source de recalculer les débits des couches selon l'état actuel du réseau.

Nous nous sommes intéressés aux mécanismes basés sur des acquittements explicites des récepteurs [16, 18, 30, 34]. Chaque récepteur envoie de temps en temps à la source

[3]Nous allons utiliser la terminologie "données" pour tout ce qui peut être transmis à travers un flux quelconque. Souvent, et comme nous allons le faire, ces données peuvent être traduites par "vidéo".

un paquet de contrôle contenant entre autres le débit maximum qu'il peut recevoir. En assemblant tous les paquets de contrôle provenant des récepteurs du collectif multicast, la source est apte à calculer les débits des couches de façon à optimiser la qualité de l'application.

Il n'y a dans notre système aucune restriction en ce qui concerne le nombre de récepteurs dans la session multicast. La quantité de paquets de contrôle qui arrive à la source est proportionnelle au nombre de récepteurs. Par conséquent, pour un groupe de récepteurs large, la source peut être victime de ce que l'on appelle *implosion de paquets d'acquittement*.

Nous proposons également dans cette thèse des mécanismes de réduction du nombre de messages d'acquittement. En fait, une des phases d'optimisation de la qualité de l'application est réalisée essentiellement en parallèle avec les mécanismes de réduction du nombre des paquets.

Sauf pour le calcul de la complexité de l'Algorithme Direct dans le chapitre 3, qui utilise le langage C, toutes nos simulations ont été implémentées sur l'outil de simulation NS [35].

1.4 Organisation du document

Cette thèse est organisée en 5 autres parties.

Au chapitre 2, nous présentons un état de l'art du domaine. Nous commençons par les principales caractéristiques des codeurs multi-couches. Le but n'est pas d'étudier leur fonctionnement interne, mais de fournir un minimum de compréhension sur un des composants principaux de cette thèse. Nous présentons ensuite les plus importants travaux sur la distribution adaptative de la vidéo que l'on trouve dans la littérature et qui sont plus liés au cadre de notre étude.

Le chapitre 3 propose l'Algorithme Direct de combinaison de paquets d'acquittement dans les nœuds intermédiaires de façon à éliminer le problème d'implosion à la source. L'algorithme agrège les paquets en amont (vers la source) d'une manière distribuée, car chaque nœud dans l'arbre multicast est responsable du traitement des paquets venus de son sous-arbre associé. Le principal avantage de cet algorithme est l'optimisation de la qualité de l'application du point de vue des récepteurs. Nous détaillerons aussi une extension de l'Algorithme Direct qui utilise la notion de "couches virtuelles". Les résultats obtenus montrent que la méthode proposée procure un très bon contrôle de la qualité de l'application dans les récepteurs pour tous les types de topologie de réseau.

L'optimisation de la qualité de l'application atteinte par l'Algorithme Direct est réalisée sur l'ensemble de données qui arrivent aux récepteurs. Ainsi, on dit qu'une configuration de couches est "de meilleure qualité" qu'une autre si la première mène à une quantité de données plus importante reçue par l'ensemble des récepteurs. Ce concept nous pousse à nous poser la question suivante : "La quantité de données dans les récepteurs est-elle la seule façon de mesurer la qualité du collectif multicast ?" La réponse à cette question est l'objet principal du chapitre 4. Nous proposons dans ce chapitre une approche qui prend en compte des multiples métriques pour le calcul de la qualité de la communication. Nous argumentons que pour certaines applications, l'algorithme de calcul des couches devrait considérer plus d'un seul paramètre. Cette approche prouve être avantageuse dans de nombreux cas, comme nous allons le montrer au travers d'analyses et de simulations.

Il est aisé de voir qu'il est nécessaire d'introduire dans les routeurs des nouvelles fonctionnalités pour la mise en œuvre des mécanismes proposés dans les chapitres 3 et 4. Le chapitre 5 est dédié aux communications sans altération de la structure des nœuds intermédiaires. Nous proposons dans cette partie un système d'échange d'informations entre les récepteurs de façon à réduire le nombre de messages d'acquittement avant même que ces informations soient envoyées par les récepteurs. Ceci se déroule sans la participation active des nœuds intermédiaires.

Enfin, le chapitre 6 clot cette thèse en présentant les conclusions et les perspectives de travaux futurs.

Chapitre 2

Vidéo adaptative en multi-couches

Le travail développé par McCanne sur la distribution multicast de vidéo en multi-couches a été le début d'une nouvelle phase d'études sur les applications adaptatives [16, 17, 18, 29, 34, 36, 37]. Bien évidemment, il y avait en cours d'autres travaux dans le même domaine, mais l'utilisation de groupes IP Multicast distincts pour la transmission indépendante des couches vidéo a spécifiquement marqué le début d'une branche de recherche.

Aujourd'hui, ce domaine est bien plus large [13, 16, 18, 37, 32, 38, 39, 40]. Ce chapitre a pour but de montrer la base des systèmes de distribution de vidéo multi-couches en multicast et aussi de présenter un certain nombre de propositions trouvées dans la littérature. D'une façon brève, nous présenterons les concepts d'un système de codage en multi-couches, ainsi que les avantages de son utilisation pour améliorer la qualité globale de la communication. Ensuite, nous décrirons le système RLM (*Receiver-driven Layered Multicast*) proposé par McCanne [17, 36], puis la proposition faite par Vickers *et al.* [18, 30] qui est plus liée au travail développé dans cette thèse. Finalement, nous allons décrire d'autres systèmes qui appartiennent au même domaine de recherche.

2.1 Vidéo en multi-couches

Une source qui suit le principe du codage en multi-couches possède les caractéristiques suivantes [41, 42, 43, 44, 45, 46, 47]. Les données sont divisées en groupes hiérarchiques dont la première couche, la *couche de base*, est la plus importante de toutes et doit être présente pour que les autres puissent être décodées. Les couches supplémentaires sont appelées *couches d'amélioration* et dépendent de la couche de base et des autres couches d'amélioration possédant un niveau hiérarchique plus élevé. Il existe donc un rapport de causalité entre les couches. La figure 2.1 illustre ce principe. Les flèches d'une couche x à une couche y indiquent que x est dépendante de y.

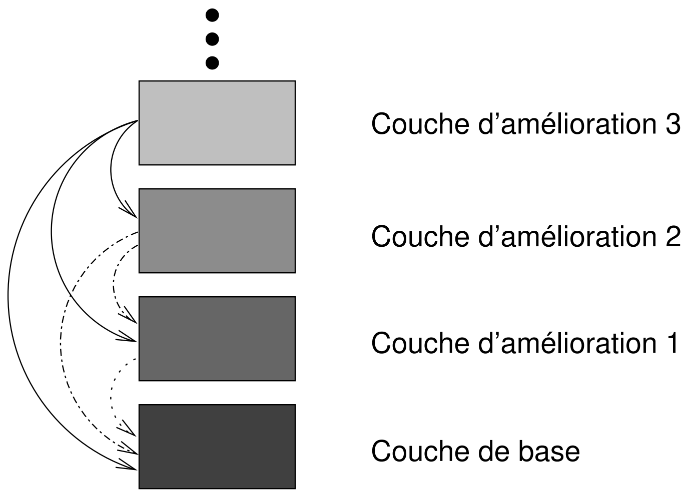


FIG. 2.1 – Causalité entre les couches.

La qualité de la vidéo est directement liée au nombre de couches disponibles. Ainsi, si un client possède la couche de base plus une couche d'amélioration, la qualité obtenue est moins bonne que celle d'un autre client possédant la couche de base plus deux couches d'amélioration. À chaque fois qu'une couche est ajoutée, la qualité s'améliore.

Le codage en multi-couches augmente la flexibilité d'un système de communication numérique. Les couches peuvent être décodées de façon à fournir la vidéo en différentes résolutions (spatiales ou temporelles) ou qualités visuelles [42]. Malheureusement, le codage en multi-couches réduit normalement l'efficacité de compression de la vidéo originelle. Néanmoins, il devrait toujours être plus efficace qu'un système utilisant de multiples codages de la même vidéo (approche par flux multiples, cf. chapitre 1), car l'information transmise contient toutes les couches codées. Le décodeur n'a qu'à décoder les couches nécessaires pour fournir la qualité requise. Cela signifie que différentes séquences de qualités distinctes sont issues d'un seul flux codé.

Le codage en multi-couches peut être aussi utilisé pour améliorer la performance d'un système en présence d'erreurs de transmission. Les couches ne sont normalement pas identiques en ce qui concerne la tolérance aux erreurs. Très fréquemment, la couche de base donne la plus importante contribution à la qualité de la vidéo décodée, ce qui réserve aux couches d'amélioration moins d'importance. Il est aisé de voir que les couches d'amélioration peuvent donc supporter un taux d'erreur un peu plus élevé que celui de la couche de base. La couche de base pourrait ainsi être transmise à travers un canal de priorité plus élevée, tandis que les couches d'amélioration le seraient dans un canal plus sensible aux erreurs. Cette technique est normalement utilisée pour fournir ce que l'on appelle dégradation souple (*graceful degradation*). Les couches de basse priorité sont les premières à être affectées, ce qui donne une légère perte de qualité visuelle. La couche de base n'est touchée que si le taux d'erreurs est très élevé.

Un exemple de l'utilisation de la "dégradation souple" est la diffusion terrestre ou satellitaire pour la télévision numérique. Si les conditions de propagation sont mauvaises, ou si un récepteur est localisé à l'extrémité d'une cellule satellitaire, le signal reçu peut contenir un certain nombre d'erreurs. La correction d'erreurs en aval (*forward error correction* – FEC) réduit le nombre d'erreurs dans beaucoup de cas. Toutefois, tous les schémas de FEC sont limités sur le nombre d'erreurs qui peuvent être automatiquement corrigées. Si le rapport signal-bruit (*signal-to-noise rate* – SNR) dans le récepteur est assez faible, l'efficacité du FEC s'arrêtera brusquement, ce qui correspondra à l'apparition d'un nombre important d'erreurs dans les données décodées. Dans un système à une couche, ce "débranchement" du FEC engendre un changement en escalier[1] de la qualité de la vidéo, ce qui peut éventuellement aboutir à une dégradation inacceptable. La figure 2.2 compare l'effet du bruit sur la télévision analogique et sur une vidéo MPEG codée en une couche. Le signal analogique se dégrade "souplement", tandis que la vidéo MPEG souffre d'un changement brusque de qualité lorsque le système de FEC ne peut guère corriger les erreurs.

[1]De l'anglais *step change*, pour noter une variation rappelant des marches.

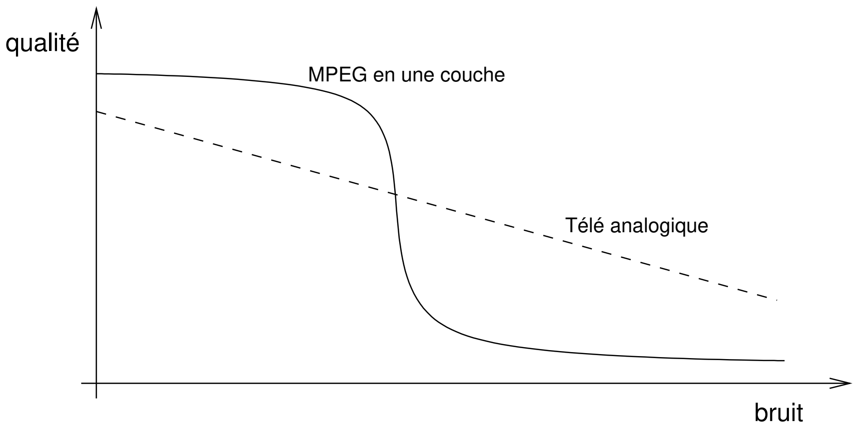


FIG. 2.2 – Dégradation de la qualité en présence de bruit.

2.1.1 DCT pour multi-couches

Les algorithmes de codage d'images et de vidéo basés sur la transformée discrète du cosinus (*discrete cosine transform* – DCT) ont eu un grand succès et sont maintenant utilisés par un large nombre de standards internationaux. On peut par exemple mentionner le standard vidéo MPEG (*Motion Picture Expert Group*) [48] et le standard d'images JPEG (*Joint Picture Expert Group*) [49] comme deux des plus importants utilisateurs de la DCT.

Les algorithmes basés sur la DCT suivent le codeur générique dessiné dans la figure 2.3. Dans ce modèle, une transformée **T** est appliquée sur l'image originelle de façon à extraire la corrélation entre les *pixels*. Si cette phase est exécutée avec succès, on applique un quantificateur **Q** indépendamment sur chaque coefficient issu de la première phase. Le résultat est obtenu sans perte de qualité. La représentation par transformée est spécialement intéressante car, en pratique, de petites variations de perception sont plus facilement manipulées dans le domaine des transformées que dans le domaine des pixels. Par exemple, les coefficients DCT en haute fréquence sont typiquement codés d'une façon plus grossière que celle des coefficients à basse fréquence, ce qui ne peut pas être fait dans le domaine des pixels.

Les coefficients obtenus après le quantificateur sont par la suite agrégés par le codeur d'entropie **E**. Les phases **T** et **E** sont l'une et l'autre réversibles et le seul responsable de l'introduction de distorsion est le quantificateur.

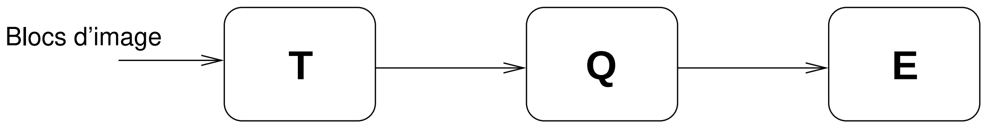


FIG. 2.3 – Codeur générique.

Les standards MPEG et JPEG utilisent la DCT selon le diagramme illustré dans la figure 2.4. Dans l'algorithme de base d'un codeur JPEG, par exemple, les blocs d'images de dimension 8x8 passent premièrement par la transformée DCT. Ensuite, les coefficients sont quantifiés au travers d'un quantificateur scalaire uniforme. Finalement, les codes de Huffman sont utilisés pour produire les données finales.

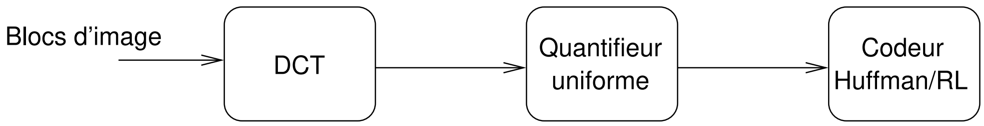


FIG. 2.4 – Diagramme des standards MPEG et JPEG.

2.1.2 Le codage en multi-couches

Le codage en multi-couches d'une vidéo n'est pas l'objet principale de cette thèse. Nous nous intéressons à son utilisation comme outil d'amélioration de la qualité de réception des données dans les réseaux hétérogènes supportant des applications multicast. Néanmoins, nous allons présenter par la suite une brève description du fonctionnement de base de ce type de système. Un exemple classique du fonctionnement pratique d'un codeur multi-couches est l'annexe O du standard H.263 de l'ITU-T [50].

Le codage en multi-couches suit le principe de scalabilité qui peut être classé en trois groupes distincts :

- scalabilité spatiale ;
- scalabilité temporelle ;
- scalabilité signal-bruit (SNR).

Scalabilité spatiale

La scalabilité par résolution spatiale implique le codage de la séquence originelle de la vidéo en différentes résolutions spatiales. Chaque trame est convertie en une image de résolution inférieure, c'est-à-dire, de dimensions plus réduites. Typiquement, un facteur égal à deux est utilisé, ce qui signifie que l'image originelle est réduite à la moitié. Ensuite, cette nouvelle image est travaillée encore une fois, donnant une troisième image d'un quart des dimensions de la première. Ce processus peut être répété autant de fois que l'on désire. La figure 2.5 illustre une "pyramide" issue d'un décodeur dont le signal d'entrée a été codé en différentes résolutions spatiales.

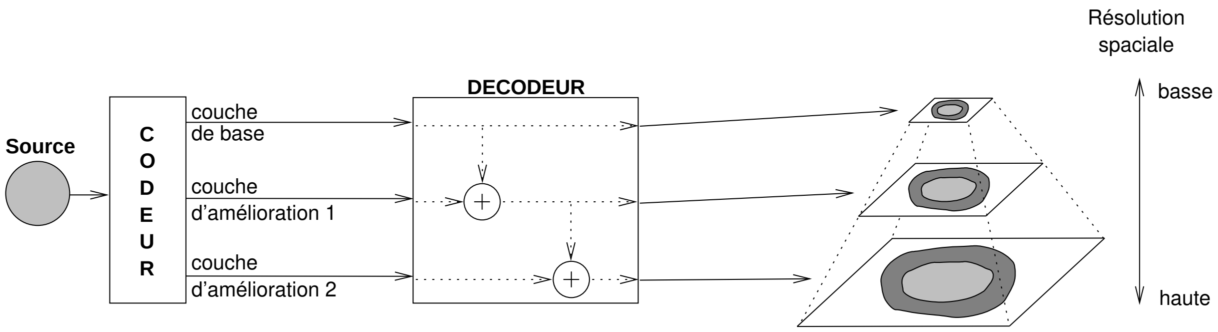


FIG. 2.5 – Scalabilité spatiale.

Scalabilité temporelle

La scalabilité temporelle permet à une séquence vidéo d'être codée en différentes résolutions temporelles ou taux de trames. Cette technique est intégrée par exemple dans l'extension de scalabilité temporelle du format MPEG-2.

Dans cette extension, la vidéo originelle est sous-échantillonnée de façon à produire une séquence avec un plus faible taux de trames. Cette nouvelle séquence est normalement codée et donne lieu à la couche de base. Les trames restantes sont travaillées et se transforment en couches d'amélioration. La figure 2.6 montre les taux de trame obtenus à partir de trois couches codées entre elles avec un facteur égal à deux.

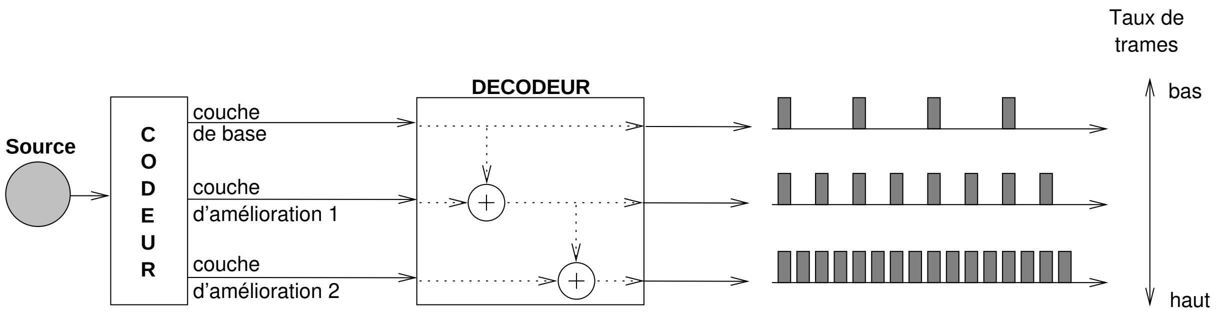


FIG. 2.6 – Scalabilité temporelle.

Scalabilité signal-bruit

Une séquence codée avec différents rapports signal-bruit donne comme résultat des vidéos avec différentes qualités visuelles. Ce type de codage implique que n'importe quel taux peut être obtenu à partir de la même vidéo originelle. Le taux est déterminé par les besoins des utilisateurs et/ou par la bande-passante disponible dans le réseau ou même par la capacité des systèmes terminaux. Les fonctions de scalabilité correspondantes rendent possible une dégradation souple de la qualité de la vidéo face aux congestions du réseau ou au manque de ressources des systèmes. La figure 2.7 illustre un exemple de scalabilité signal-bruit obtenue sur une image d'une séquence vidéo.

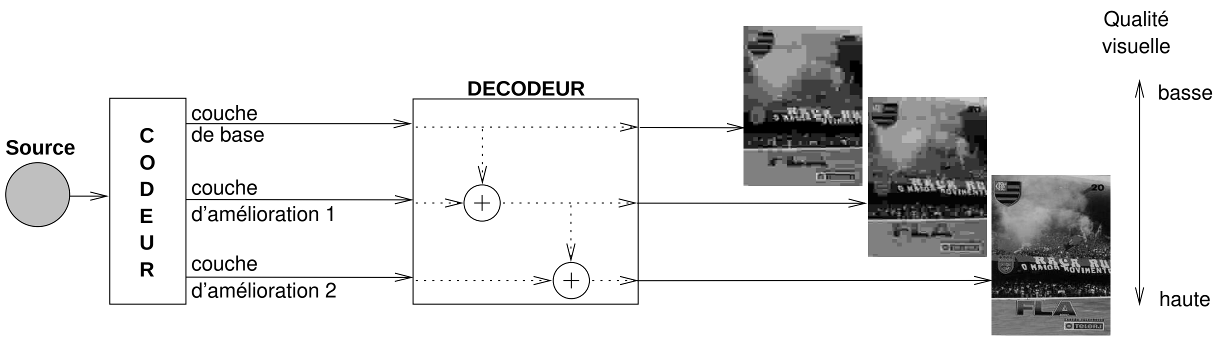


FIG. 2.7 – Scalabilité signal-bruit.

2.2 Multicast en multi-couches

Nous allons voir maintenant les avantages de l'utilisation du codage en couches pour la distribution en multicast. Soit un réseau dont les chemins entre la source et les récepteurs sont hétérogènes en terme de bande passante disponible. L'hétérogénéité des récepteurs peut être en réalité très forte. Il suffit de considérer un terminal mobile (un téléphone portable, par exemple) et une station de travail connectée à un réseau ATM. Il est aisé de voir que si ces deux terminaux s'abonnent à la même session multicast, une de deux situations suivantes aura lieu : si la source transmet à la vitesse du récepteur le plus lent (terminal mobile dans notre exemple), la capacité de la station de travail sera sous-utilisée ; si le débit de la source est contrôlé par le terminal le plus rapide (station de travail), le récepteur moins puissant sera incapable de rester longtemps connecté sans saturation des ressources.

Lorsque la technique de codage en multi-couches est utilisée dans des cas tels que celui décrit ci-dessus, le problème de l'hétérogénéité des récepteurs peut être réduite. La figure 2.8 montre un réseau hétérogène dont la source produit trois couches vidéo. Les récepteurs r_1 et r_3 sont plus performants et peuvent donc recevoir toutes les couches. Le récepteur r_2, par contre, est très limité et ne peut manipuler que la couche de base. Dans une position intermédiaire se trouve le récepteur r_4, qui reçoit la couche de base et la première couche d'amélioration.

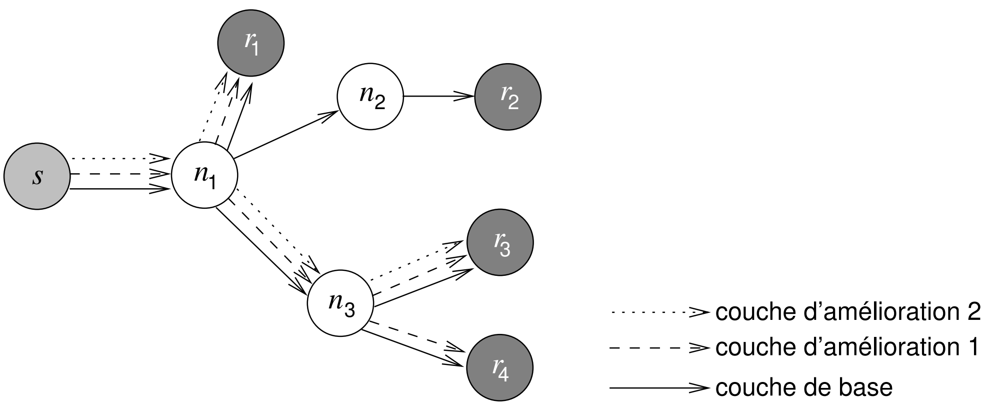


FIG. 2.8 – Hétérogénéité dans les réseau.

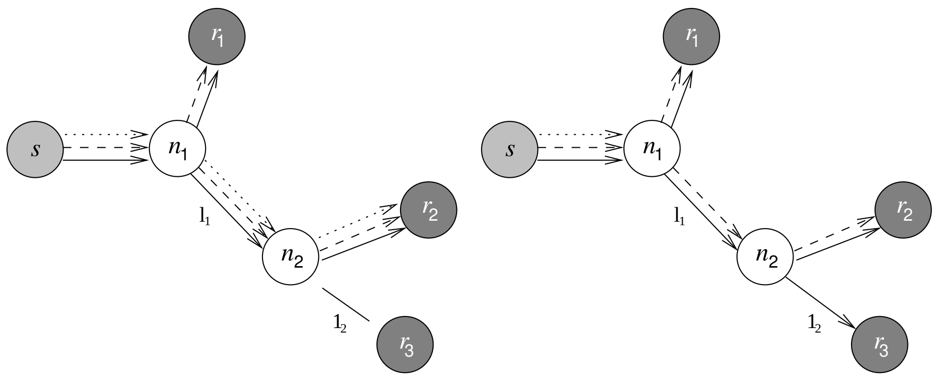
Il est évident que de cette façon un plus grand nombre de récepteurs peut être servi avec le débit qui a été demandé. Cependant, il est également facile d'observer que, pour que tous les récepteurs puissent être entièrement servis, la source doit produire autant de couches que le nombre de récepteurs. Il est clair qu'en pratique la source n'est pas illimitée. Le réseau doit donc être capable de calculer les débits des couches de façon à optimiser la qualité du service livré à l'ensemble des récepteurs. Ceci est le but de cette thèse.

2.3 Receiver-driven Layered Multicast

McCanne et al. [17, 36] ont proposé pour la première fois un protocole de contrôle de congestion pour la transmission de vidéo en multi-couches sur Internet. Ce protocole, nommé *Receiver-driven Layered Multicast* (RLM), est la base de nombreuses autres architectures.

RLM opère sur la traditionnelle pile de protocoles de l'architecture Internet et se repose sur l'efficacité de l'IP Multicast [51]. Le protocole lui-même n'exige aucune garantie de la part du réseau pour les trafics en temps réel et assume uniquement la livraison de paquets selon une politique multicast de meilleur service (*best-effort*). Un des aspects importants de l'IP Multicast est l'abstraction de groupes d'hôtes (*host groups*) ; il n'est pas indispensable que les récepteurs connaissent les émetteurs dans un groupe multicast ni que les sources prennent conscience des récepteurs existants. Les émetteurs envoient simplement leurs paquets à une "adresse de groupe" et les récepteurs informent le réseau (par l'intermédiaire du protocole de gestion de groupes IGMP [52] – *Internet Group Management Protocol*) qu'ils sont intéressés par recevoir les paquets adressés à ce groupe.

La figure 2.9 illustre comment le protocole de gestion d'abonnements pourrait exécuter l'acheminement sélectif des couches. Admettons que la source S transmet trois couches vidéo et que, dans une premier instant (figure 2.9(a)), r_1 reçoit deux couches, r_2 les trois couches et r_3 aucune couche. Supposons que cette configuration provoque la congestion dans le lien l_1. Le récepteur r_2 réagit en excluant la troisième couche. Par conséquent, le réseau élimine le flux correspondant dans le lien l_1 (cf. figure 2.9(b)). En même temps, le récepteur r_3 décide de s'abonner à la première couche, ce qui est fait automatiquement par le nœud l_2.



(a) état à l'instant $t = 0$. (b) état après congestion.

FIG. 2.9 – Protocole de gestion d'abonnements.

Il est aisé de voir que la décision du choix des couches est uniquement de la responsabilité des récepteurs – dont le terme *receiver-driven* (contrôlé par les récepteurs). La source n'a pas un rôle actif dans le système adaptatif. Elle transmet simplement les différentes couches à des groupes multicast différents. Le noyau du protocole est placé dans les récepteurs et, fondamentalement, fonctionne de la façon suivante :

- si période de congestion, éliminer une couche ;
- sinon, ajouter une couche.

Dans ce schéma, un récepteur recherche constamment le meilleur *niveau d'abonnement* d'une façon similaire à celle d'une source TCP qui calcule son débit de transmission avec le mécanisme de *slow-start* [53]. Un récepteur ajoute des couches jusqu'à ce qu'il y ait congestion. À ce moment, il retourne au point d'opération précédent.

Un récepteur peut facilement détecter si le réseau est congestionné en observant le comportement de la séquence des paquets arrivants. Toutefois, il ne peut obtenir avec la même facilité les caractéristiques de la bande-passante disponible. L'utilisation exclusive du monitoring passif ne permet pas la différenciation entre un débit qui est juste au-dessous de la capacité totale et un autre bien inférieur à cette capacité. De façon à surmonter cet handicap, le protocole RLM utilise des "expériences d'abonnement" pour vérifier la présence de bande-passante marginale. Un récepteur teste occasionnellement la disponibilité des ressources en ajoutant une couche supplémentaire. Si cette expérience provoque la congestion du réseau, le récepteur revient à son état précédent et augmente exponentiellement le temps qu'il doit attendre pour refaire le test. Dans le temps, les récepteurs "apprennent" que certaines couches sont plus "problématiques" que d'autres.

La figure 2.10 illustre la stratégie utilisée par RLM pour un récepteur qui peut s'abonner jusqu'à quatre couches. Tout au début, le récepteur s'abonne à la première couche et configure un temporisateur (A). La période du temporisateur est courte car il n'y a pas encore eu de congestion à ce niveau. À la fin de cette période, le récepteur s'abonne à la deuxième couche et configure un autre temporisateur (B). Le processus se répète et la troisième couche est aussitôt ajouté. Lorsque le récepteur s'abonne à la quatrième couche, une congestion dans le réseau a lieu (C). Par conséquent, une file (ou des files) quelconque se remplit et provoque la perte de paquets. Le récepteur détecte cette perte et revient à l'état précédent, c'est-à-dire avec trois couches. Le temporisateur qui mène le récepteur à essayer l'abonnement à la quatrième couche est donc étalé (D). Encore une fois, à la fin de cette période le récepteur passe par le même problème de congestion et le temporisateur est encore une fois prolongé (E). Ensuite, une congestion transitoire a lieu et force le récepteur à éliminer également la troisième couche. À la deuxième couche, la période du temporisateur est encore courte (F) et le récepteur revient rapidement à son état stable.

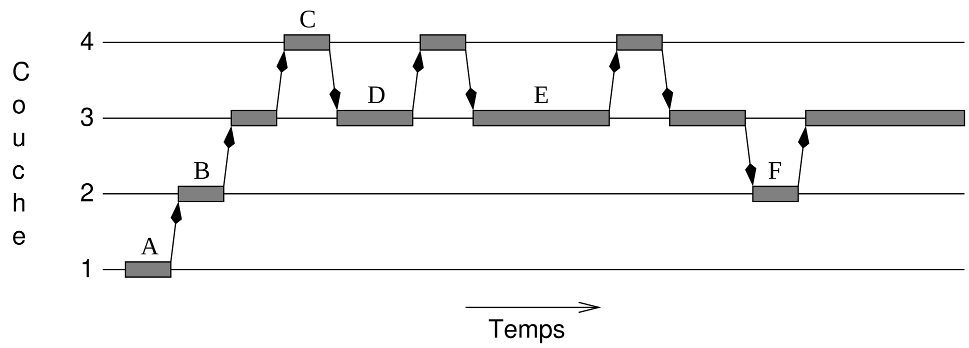


FIG. 2.10 – Schéma d'adaptation.

Si le nombre de récepteurs est large et si chacun d'eux exécute cet algorithme indépendamment, le protocole se déstabiliserait à cause du nombre croissant d'expériences d'abonnement et provoquerait des congestions fréquentes. Pour éviter cet inconvénient, le protocole RLM applique la technique d'apprentissage partagé, où les récepteurs sont informés à propos des expériences échouées des autres récepteurs. Les détails sur cet algorithme se trouvent dans [17, 36].

2.4 Algorithme d'adaptation à la source

Nous allons maintenant présenter une nouvelle classe d'algorithmes d'adaptation qui a été mieux étudiée et développée dans les travaux de Vickers et al. [18, 30]. Dans ce paradigme, la source ajuste les paramètres de transmission, y compris le nombre de couches et leurs débits, en réponse à un flux d'acquittements provenant du réseau et/ou des récepteurs.

L'architecture nécessaire à l'implantation des algorithmes d'adaptation à la source qui nous intéressent possède quatre composants de base : la source multi-couches adaptative, des récepteurs également multi-couches, des routeurs multicast et des nœuds possédant l'aptitude de manipulation de données. Ces composants sont décrits ci-dessous ;

- *source multi-couches adaptative* : la source doit être capable de produire un certain nombre de couches à partir du média originel. La génération des couches est la seule caractéristique exigée de la source, indépendemment du codage utilisé. Le codage peut être simplement le choix d'un sous-ensemble des trames comme la couche de base et les trames restantes comme les couches d'amélioration. Le codage peut être aussi développé sous la forme d'un ajustement des paramètres de codification pour générer la couche de base plus l'addition de raffinements pour obtenir les couches d'amélioration ;
- *récepteurs multi-couches* : les récepteurs doivent recevoir les couches vidéo séparément et être capables de les rassembler de façon à reconstruire le média originel. Les récepteurs coopèrent avec l'algorithme en rapportant les informations concernant l'état de congestion du réseau ;
- *routeurs multicast* : les routeurs doivent, au minimum, posséder les fonctionnalités suivantes : (a) routage et acheminement multicast ; (b) élimination de paquets par priorité ; (c) isolation des flux ; (d) contrôle de congestion. Des exemples de protocoles

multicast sont le DVMRP [54], le MOSPF [55], le CBT [56], le PIM [57] et le HBH [58] ;
- *nœuds de manipulation* : le rôle premier des nœuds de manipulation est d'éviter l'implosion de paquets d'acquittement à la source. Lorsque plus qu'un paquet d'acquittement est reçu par un nœud intermédiaire, ils sont automatiquement combinés et le paquet résultant est acheminé vers la source. Ainsi, la manipulation des informations de contrôle (plus précisément d'acquittement) est faite d'une façon distribuée dans le réseau. La source ne reçoit qu'un nombre de paquets égal à la quantité de nœuds qui lui sont attachés. L'idée de désigner les nœuds intermédiaires pour réduire la probabilité d'implosion d'acquittements à la source a déjà été l'objet d'autres travaux, notamment ceux sur le multicast fiable [59, 60, 61].

Dans cette thèse, nous nous intéressons particulièrement à ce type de système, c'est-à-dire, les système d'adaptation à la source. Nous réservons plus de détails sur le fonctionnement des architectures avec adaptation à la source pour le chapitre suivant.

2.5 Systèmes adaptatifs

Nous présentons dans cette section d'autres systèmes adaptatifs pour les applications multimédia que l'on peut trouver dans la littérature.

2.5.1 Mécanisme scalable pour la vidéo sur Internet

Une des premières approches pour la transmission adaptative de vidéo était le système IVS (*INRIA Video-conference System*), proposé par Bolot, Turletti et Wakeman [14, 24, 25]. Le système IVS est dans son essence un *software* de vidéo conférence pour l'Internet qui utilise l'approche adaptative mono-couche.

Le mécanisme de contrôle de débit est implémenté dans le codeur H.261 [14]. Pour l'ajustement du débit de sortie du codeur, trois paramètres sont considérés : le taux de rafraîchissement, le quantificateur et le seuil de détection de mouvement. Les besoins spécifiques de l'application vidéo déterminent lequel de ces paramètres doit être modifié au moment de l'ajustement du débit de sortie du codeur. Les informations d'acquittement sont obtenues à partir du taux de perte dans les récepteurs, qui à son tour est calculé par l'intermédiaire des numéros de séquence du protocole RTP (*Real Time Protocol*) [62]. Le protocole de contrôle associé à RTP, nommé RTCP (*Real time Transport Control Protocol*) [62], est chargé d'envoyer des *rapport de réception* contenant les informations d'acquittement.

Pour éviter le problème d'implosion de paquets d'acquittements, Bolot, Turletti et Wakeman proposent en [25] un mécanisme de requêtes scalable. Chaque récepteur répond à la source en envoyant un paquet d'acquittement contenant certaines informations sur l'état de congestion. Une réponse est faite selon une probabilité p et une fréquence f dépendante du nombre total de récepteurs dans la session multicast.

Comme nous avons mentionné précédemment, un des problèmes potentiels de l'IVS, et de n'importe quel autre système à une seule couche, est son incapacité à fournir un traitement équitable à l'ensemble des récepteurs dans un environnement hétérogène. On peut traduire cette limitation par le schéma dessiné dans la figure 2.11. D'un côté, si la distribution de la vidéo est faite par l'intermédiaire de canaux d'acquittements du type point-à-point individuels l'économie de bande-passante est faible. Par contre, la finesse de contrôle est très élevée car la source peut négocier les paramètres de communication avec chacun des récepteurs. D'un autre côté, si l'on utilise un seul flux multicast on obtient une bonne économie de bande-passante, mais une très faible finesse de contrôle ; le calcul des paramètres peut ne pas être intéressant pour beaucoup de récepteurs.

2.5.2 Destination Set Grouping

L'algorithme DSG (*Destination Set Grouping*), proposé par Cheung, Ammar et Li [26, 28, 27], a été un des premiers travaux à examiner le problème d'hétérogénéité dans les réseaux multicast. Cet algorithme combine autant l'approche contrôlée par la source que l'approche contrôlée par les récepteurs. Les principaux objectifs du protocole DSG sont : l'amélioration de l'équité dans un groupe multicast et la scalabilité. L'équité est obtenue par la transmission de différents flux vidéo (avec des débits différents) issus de la même vidéo originelle (cf. figure 2.12). Cela permet aux récepteurs de choisir le débit qui se montre

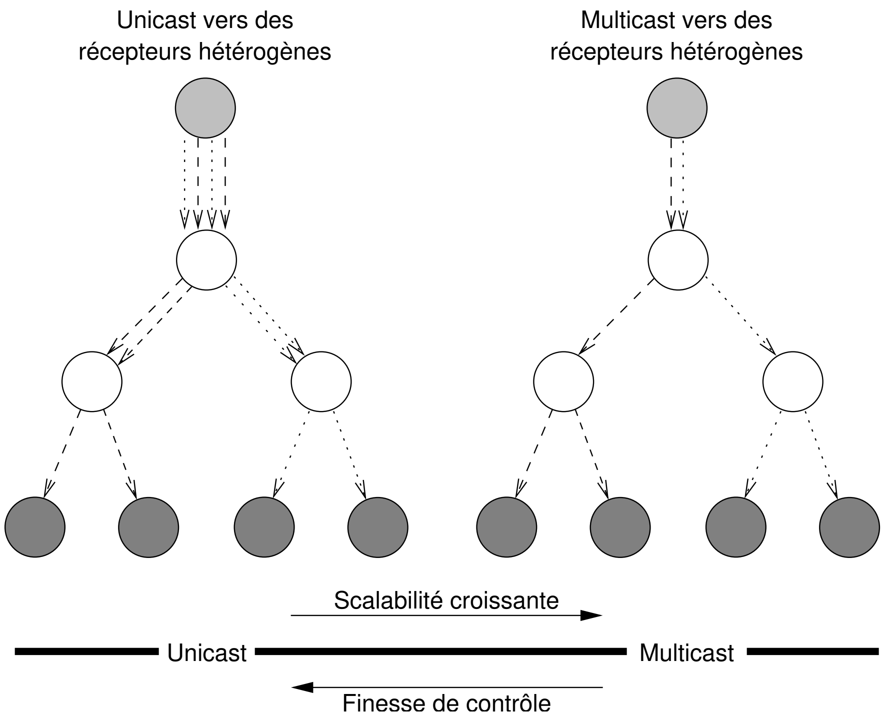


FIG. 2.11 – Compromis entre les distributions multicast et unicast.

le mieux adapté à leurs besoins. La scalabilité est atteinte car la décision d'abonnement aux groupes multicast est prise uniquement par les récepteurs.

Les récepteurs sont constamment informés sur l'état du réseau. Ainsi, ils peuvent prendre la décision de changer ou non de groupe multicast. Le protocole DSG possède deux composants principaux :

- un protocole intra-flux utilisé par les récepteurs appartenant au même groupe pour ajuster le débit de transmission selon certaines limites préétablies ;
- un protocole inter-flux utilisé par les récepteurs pour les changements des groupes multicast.

Le protocole DSG obtient des bons résultats en termes de scalabilité et d'équité. Cependant, une très importante limitation est la mauvaise utilisation des ressources due à la répétition de la même vidéo en différents flux indépendants.

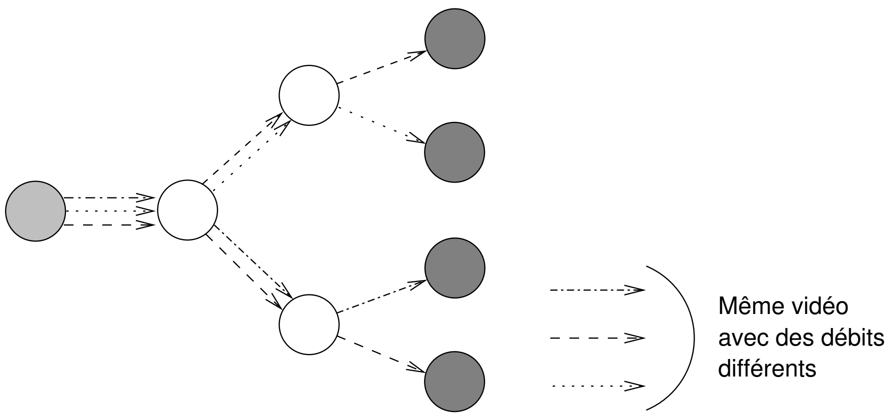


Fig. 2.12 – Le schéma de distribution du protocole DSG. Toutes les vidéos sont en fait la même vidéo codée en différents débits.

2.5.3 Autres schémas

Plusieurs autres travaux sur la distribution de la vidéo dans les réseaux hétérogènes sont apparus plus récemment. Un certain nombre entre eux sont en effet des extensions ou variations du protocole RLM. Le protocole LVMR (*Layered Video Multicast with Retransmission*) [29, 63] apporte deux contributions principales : (*i*) amélioration de la qualité de réception dans chaque couche vidéo au travers la retransmission des paquets perdus et (*ii*) adaptation à l'état de congestion du réseau en utilisant un contrôle hiérarchique du débit de transmission. Le travail développé par Vicisano *et al.* (TCP-like congestion control for layered data) [37] présente un algorithme de contrôle de congestion pour la transmission de flux en multi-couches sur le MBone. L'algorithme se comporte d'une façon similaire aux algorithmes de congestion proposés pour TCP. Un des objectifs principaux est de partager d'une façon équitable la bande-passante du réseau avec d'autres instances du protocole avec des flux TCP.

Une autre solution potentielle pour la transmission multicast de la vidéo dans les réseaux hétérogènes est le transcodage [64, 65, 66, 67]. Dans cette approche, une seule couche vidéo est codée à la source en utilisant le plus élevé taux de transmission. Les nœuds intermédiaires, au moment de l'acheminement des paquets, transcode (c'est-à-dire, décode et re-code) la vidéo à un taux moins élevé tant que les liens sont congestionnés. Malgré le fait de résoudre le problème de variation de la bande-passante disponible, cette approche est assez complexe et exige une puissance de calcul très élevée.

2.6 Conclusion

Nous avons présenté dans ce chapitre quelques principes sur la distribution multicast des applications adaptatives en multi-couches, ainsi qu'un survol sur les principaux travaux trouvés dans la littérature.

L'utilisation de la technique du codage en multi-couches est une réalité dans les propositions les plus modernes pour la transmission des applications dans les réseaux hétérogènes. On peut constater également que le problème de l'hétérogénéité est une des plus grandes préoccupations lors de la conception de ces systèmes.

Malgré l'existence de plusieurs propositions d'algorithmes, beaucoup de domaines de recherche restent ouverts. Dans cette thèse, nous nous intéressons à un certain nombre de ces domaines, à savoir :

- étude des systèmes de contrôle adaptatif du débit de transmission des couches dans les réseaux multicast hétérogènes ;
- proposition d'algorithmes de combinaison de paquets de contrôle de manière à empêcher l'implosion de messages d'acquittement à la source, tout en optimisant la qualité de réception des destinataires ;
- conception de systèmes de contrôle de transmission basés sur des paramètres multiples – la qualité est étendue à un niveau où l'on considère aussi bien le point de vue des récepteurs que le point de vue du réseau ;
- adaptation des systèmes proposés à des architectures sans capacité de calcul dans les nœuds intermédiaires – cela n'exige pas l'altération de la configuration des éléments du réseau.

Chapitre 3

L'algorithme de calcul direct

Le chapitre précédent a montré la technique que nous allons utiliser tout au long de cette thèse : le calcul des débits des couches transmises par la source à partir des paquets de contrôle envoyés par les récepteurs. Dans ce chapitre, nous nous intéresserons précisément au problème de la combinaison des paquets de contrôle dans les nœuds intermédiaires afin d'éviter l'implosion d'acquittements à la source.

Tout d'abord, nous allons détailler le système classique qui s'appuie sur le concept de "goodput rate" pour calculer le paquet de sortie d'une procédure de combinaison de paquets [18, 30]. Nous avons identifié quelques caractéristiques de cette approche qui limitent, en terme de dégradation du flux multimédia, la qualité globale de la session multicast. Ces observations nous ont poussés à bien étudier les aspects liés aux algorithmes de combinaison de paquets, ce qui a abouti en une proposition qui considère tous les paquets d'acquittement comme un ensemble [16, 33, 34]. Nous montrerons que ceci permet d'améliorer la qualité globale de la session multicast, tout en réduisant la dégradation de la vidéo dans les récepteurs.

3.1 Le contrôle des débits par des paquets d'acquittement

Le contrôle des débits des couches vidéo, au travers d'échanges d'informations entre la source et les récepteurs, permet aux trafics de la session multicast de toujours s'adapter aux conditions du réseau.

Nous décrirons en détail le schéma de contrôle par acquittements utilisé dans notre système adaptatif [16, 18, 30, 33, 34]. Dans ce système, la source envoie périodiquement des paquets d'investigation vers tous les récepteurs appartenant à la session multicast. Ce paquets d'investigation parcourent le réseau et marquent la bande-passante disponible dans chacun des liens traversés.

Soit $L_{s,r}$ l'ensemble des liens traversés par le paquet de contrôle entre la source s et le récepteur r. La bande-passante disponible dans le lien l est représentée par $c_s(l)$ et, lorsque l'un des paquets atteint une des destinations, le récepteur correspondant calcule le débit maximum $b_{max}^{r,m}$ qu'il peut recevoir :

$$b_{max}^{r,m} = \min_{l \in \mathbf{L}_{s,r}} c_s(l). \tag{3.1}$$

La capacité de réception peut être calculée de différentes façons. Le mécanisme *packet pair*, proposé par Keshav dans [68], permet à la source de calculer la bande-passante disponible dans le chemin entre la source et le récepteur [68, 69]. Legout et Biersack proposent dans [70] le protocole PLM (*packet Pair receiver-driven cumulative Layered Multicast*) qui utilise le concept introduit dans [68] pour découvrir la bande-passante disponible dans le réseau. Dans [18], Vickers *et al.* utilisent une variation de l'algorithme ERICA [71] qui a été originellement proposé pour le service ABR de l'ATM. Le calcul de la bande-passante disponible n'est pas l'objet principal de cette thèse et nous allons considérer que les nœuds intermédiaires possèdent déjà cette fonctionnalité.

Les récepteurs, après avoir calculé leurs capacités, créent un paquet d'acquittement contenant le débit maximum et un second champ indiquant le nombre de récepteurs souhaitant avoir ce débit. Dès lors que les paquets quittent les récepteurs, ce champ est bien sûr égal à un. Son but est, comme nous allons le voir en détail dans ce chapitre, d'accumuler le nombre de récepteurs dans le sous-arbre d'un nœud qui vient d'effectuer une opération de combinaison de paquets. Bien évidemment, si aucun paquet n'arrive par les liens d'entrée, nœud n'exécute aucune opération, ce qui correspond au cas trivial.

La structure d'un paquet d'acquittement est constitué de k composantes, $1 \leq k \leq L$, où L est le nombre de couches vidéo transmises par la source. La figure 3.1 montre le format de base d'un paquet d'acquittement. Chaque composante e_i, $i = 1, \ldots, k$, est divisée en deux champs : le débit f_{e_i} et le nombre de récepteurs $f^*_{e_i}$ souhaitant recevoir f_{e_i}. Ainsi, si l'on représente le paquet d'acquittement p de taille k sous la forme $p = \{(f_1, f^*_1), (f_2, f^*_2), \ldots, (f_k, f^*_k)\}$, le paquet $p_x = \{(1,3), (4,2)\}$ possède donc, dans le sous-arbre qui lui est associé, 5 récepteurs dont 3 souhaitant recevoir le débit 1, et 2 souhaitant recevoir le débit 4[1].

f_{i-1}	f^*_{i-1}	f_i	f^*_i	f_{i+1}	f^*_{i+1}	f_{i+2}	f^*_{i+2}

FIG. 3.1 – Structure d'un paquet d'acquittement.

3.1.1 La dégradation globale de la communication

Lorsque les paquets d'acquittement arrivent à la source, celle-ci se sert des informations des paquets pour calculer les débits qu'elle doit attribuer aux différentes couches vidéo. L'algorithme qui exécute cette tâche choisit la configuration de débits qui se montre la plus équitable possible au regard des clients. Nous considérons le paquet le plus équitable comme celui qui satisfait les définitions suivantes :

Définition 1 *La dégradation globale d'une session multicast en multi-couches est égale à la somme des différences entre le débit demandé pour une vidéo et le débit réel reçu pour l'ensemble des récepteurs de la session.*

Définition 2 *Après la procédure de combinaison de paquets, le plus équitable paquet de contrôle est celui qui minimise la dégradation globale de la vidéo dans les récepteurs.*

[1] Pour des raisons de simplicité, dans les exemples les unités des variables seront supprimées.

Bien entendu, plus importante est la dégradation de la session multicast, moins intéressante est la qualité de l'algorithme (voir section 3.2).

3.1.2 La procédure de combinaison

Les paquets d'acquittement qui arrivent à un nœud n sont stockés dans un vecteur temporaire $\vec{t}$. L'exécution de la procédure de combinaison par le nœud n est déclenchée par un des évènements suivants :

- tous les tampons des liens en amont (vers la source) ont reçu un paquet d'acquittement. Cela signifie que tous les récepteurs connectés au sous-arbre associé au nœud n ont réagit à l'arrivée du paquet d'investigation ;
- un temporisateur τ configuré à l'arrivée du premier paquet d'acquittement est épuisé. Cela empêche que la perte d'un des paquets dans le sous-arbre provoque la génération d'une boucle au niveau du nœud n et que des délais trop importants soient produits.

La première étape de la procédure de combinaison de paquets est l'organisation du vecteur temporaire pour que l'algorithme puisse lui-même réaliser les opérations nécessaires. Les composantes du vecteur temporaire sont disposées de telle façon que $f_{e_{i-1}} \leq f_{e_i} \leq f_{e_{i+1}}, \forall i, i \geq 1$. Si, par coïncidence, deux ou plus de composantes sont identiques (c'est-à-dire, au moins deux récepteurs dans le sous-arbre demandent la même bande-passante), elles sont automatiquement combinées de la façon suivante. Soit $f_{e_i} = f_{e_{i+1}}$, alors e_i et e_{i+1} sont combinées suivant les équations :

$$\begin{cases} f_{e_{i,\oplus}} = f_{e_{i,\ominus}}, \\ f^*_{e_{i,\oplus}} = f^*_{e_{i,\ominus}} + f^*_{e_{i+1,\ominus}}, \end{cases} \tag{3.2}$$

où les symboles $\ominus$ et $\oplus$ représentent les valeurs des variables respectivement avant et après l'exécution de la procédure de combinaison de paquets.

En réalité, le paquet de sortie est le résultat de la fusion de tous les paquets arrivés, moins le nombre de composantes éliminées. Les informations contenues dans les composantes rejetées sont ajoutées à celles de la composante suivante dans la hiérarchie. Ainsi, si le vecteur temporaire possède un nombre de composantes c_T plus grand que le nombre de couches vidéo transmises par la source (L), le nœud exécutera l'algorithme de réduction pour qu'à la fin il ne reste que L composantes. La variable L est connue des nœuds intermédiaires par l'intermédiaire des paquets d'investigation transmis par la source.

On définie *paquets candidats* tous les paquets pouvant être choisis à partir d'un vecteur temporaire provenant de l'agrégation de k paquets d'entrée. Celui qui va être choisi dépend de la politique appliquée par le mécanisme de combinaison de paquets.

Le premier point à définir est le comportement de l'algorithme classique de combinaison de paquets. Nous allons maintenant présenter une étude détaillée de cet algorithme et identifier les faiblesses qui peuvent être résolues par l'intermédiaire d'un algorithme capable de traiter les informations de contrôle des récepteurs comme un ensemble. Puis nous allons détailler notre proposition, à savoir l'Algorithme Direct, qui apporte des solutions à ces problèmes.

3.2 Le schéma classique

Le schéma classique de combinaison de paquets de contrôle utilise le concept de "goodput rate" pour calculer le paquet final qui suivra le chemin vers la source [18, 30]. À partir du vecteur temporaire $\vec{t}$, le *goodput rate* (on va l'appeler désormais "débit conforme") est le débit total obtenu après l'élimination de la composante j :

$$G = \sum_{i=1}^{c_T} f_i f_i^* - f_j^*(f_j - f_{j-1}). \tag{3.3}$$

Le deuxième terme de l'équation est la perte due à l'agrégation de la composante j à la composante $j-1$ (cf. équation 3.2). Ces actions sont répétées jusqu'à ce que $c_T - L$ composantes soient éliminées.

Nous allons maintenant détailler toutes les phases de la procédure de réduction des paquets de contrôle pour le réseau illustré dans la figure 3.2. Les chiffres à coté des liens représentent les respectives bandes-passantes disponibles à un certain instant t_0.

Considérons tout d'abord le récepteur r_1 (en haut de la figure). Sachant que le chemin

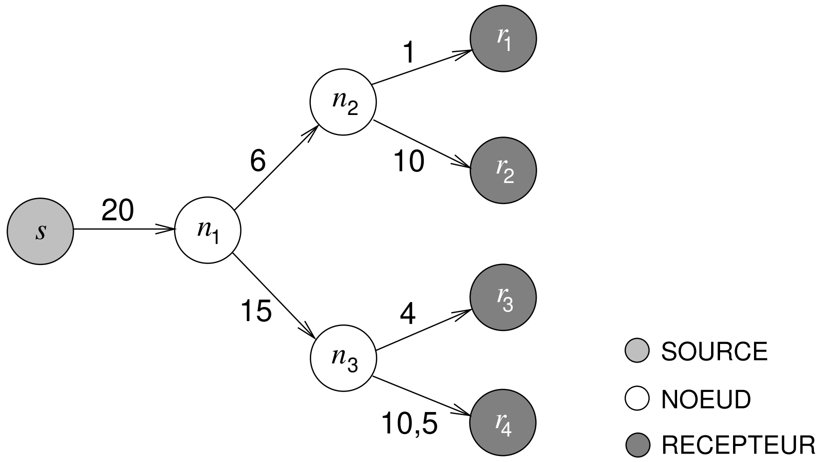


FIG. 3.2 – État initial du réseau.

entre la source et r_1 est $\mathbf{L}_{s,r_1} = \{l_{s,n_1}, l_{n_1,n_2}, l_{n_2,r_1}\}$, le débit maximum que r_1 peut recevoir est donc

$$b_{max}^{s,r_1} = \min\{c_s(l_{s,n_1}), c_s(l_{n_1,n_2}), c_s(l_{n_2,r_1})\} = \min\{20, 6, 1\} = 1.$$

Le récepteur r_2 exécute la même opération, ce qui donne $b_{max}^{s,r_2} = 6$, et ainsi de suite pour tous les autres récepteurs.

Le détail le plus important concerne le récepteur le plus lent qui est responsable de la première composante du vecteur temporaire. Il est possible que pendant une procédure de combinaison quelconque, cette composante donne le plus grand débit conforme. Une politique doit donc être mise en place pour déterminer si cette composante est candidate ou non à l'élimination. Nous avons choisi de servir tous les clients de la session multicast même si cela pénalise les récepteurs les plus rapides au bénéfice de ceux qui sont limités. L'autre choix serait, si nécessaire, d'éliminer la première composante pour que les débits ne soient pas gouvernés par des liens congestionnés ou par les clients les plus lents.

Soit G_{e_j} le débit conforme obtenu après l'élimination de la composante e_j du vecteur temporaire. Ce dernier, dans le nœud n_1 de la figure 3.2, aura quatre composantes dont deux seront éliminées. La figure 3.3 illustre les phases décrites ci-dessous :

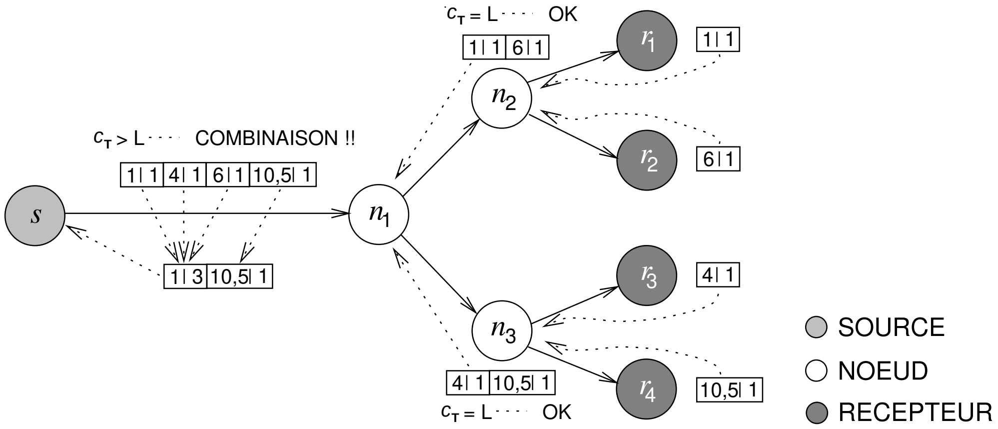


FIG. 3.3 – Procédure de combinaison de paquets.

Phase 1 $G_{e_2} = 18,5$

$G_{e_3} = 19,5$

$G_{e_4} = 17$

G_{e_3} est le débit conforme le plus important. Dans ce cas, e_3 est éliminée et $f^*_{2,\oplus} = f^*_{2,\ominus} + f^*_{3,\ominus}$. Il nous reste encore l'élimination d'une deuxième composante. Juste avant d'exécuter l'étape suivante, le vecteur temporaire possède cet aspect :

$e_{1,\ominus} \rightarrow \quad f_{1,\ominus} = 1 \qquad f^*_{1,\ominus} = 1$

$e_{2,\ominus} \rightarrow \quad f_{2,\ominus} = 4 \qquad f^*_{2,\ominus} = 2$

$e_{3,\ominus} \rightarrow \quad f_{3,\ominus} = 10,5 \quad f^*_{3,\ominus} = 1$

On a donc pour la deuxième étape le scénario suivant :

Phase 2 $G_{e_2} = 13,5$

$G_{e_3} = 13$

On peut observer que G_{e_2} est plus important que G_{e_3}. Enfin, e_2 est éliminée et le nœud envoie vers la source le paquet résultant.

Il est intuitif de penser que l'approche par le débit conforme donne le meilleur résultat en terme de dégradation de la vidéo dans les récepteurs. Néanmoins, nous avons observé que l'exécution de l'algorithme en plusieurs étapes (comme dans le cas de cette approche) peut s'avérer inexacte en ce qui concerne les composantes éliminées. En effet, le plus grand débit conforme en une étape précise peut ne pas refléter les caractéristiques du vecteur temporaire pendant tout le processus.

Revenons à l'exemple de la figure 3.3. Le paquet d'acquittement p_{out} résultant de la procédure de combinaison nous informe que les récepteurs dans le sous-arbre associé au nœud n_1 auront deux possibilités de débits. La première est donnée par $f_{e_1} = 1$ et la deuxième par $f_{e_2} = 10.5$. Si l'on observe pourtant le vecteur temporaire originel, il est facile de voir que le paquet de sortie le plus équitable serait $p_{out}^{+} = \{(1,2),(6,2)\}$. La dégradation globale de la communication associée au paquet obtenu par la méthode classique est $\delta = 8$ et celle associée au paquet p_{out}^{+} serait $\delta_{p_{out}^{+}} = 7.5$. En complément, il y a la consommation de bande-passante dans chacun des deux cas. Pour une dégradation globale plus importante, les résultats obtenus par l'approche classique satisfont moins de clients tout en consommant plus de ressources.

Nous formaliserons par la suite pourquoi l'exécution de l'algorithme en plusieurs étapes peut ne pas donner les meilleurs résultats en terme de la dégradation globale de la session multicast.

Soit $\mathbf{S}$ un sous-ensemble quelconque des composantes du vecteur temporaire. La taille de $\mathbf{S}$ est $|\mathbf{S}| = c_T - L$. On définie $\delta_{\mathbf{S}}$ comme la dégradation globale de la session multicast lors de l'élimination de la composante $e_i \in \mathbf{S}$. Par exemple, dans l'exemple ci-dessus les possibles valeurs de $\delta(\cdot)$ sont :

$\delta_{2,3} = 8,5,$

$\delta_{2,4} = 7,5,$

$\delta_{3,4} = 8.$

Remarquons que la plus petite dégradation globale de la session multicast correspond à la meilleure qualité globale de la vidéo dans les récepteurs. Soit $\mathbf{U}$ l'ensemble de toutes les combinaisons possibles du paquet de sortie et $\mathbf{O} \subset \mathbf{U}$, $|\mathbf{O}| = L$, le sous-ensemble de $\vec{t}$ qui mène à la meilleure qualité de la session multicast, où

$$\vec{t} = \{(f_{o_1}, f_{o_1}^{*}), \ldots, (f_{o_2}, f_{o_2}^{*}), \ldots, (f_{o_L}, f_{o_L}^{*}), \ldots, (f_{c_T}, f_{c_T}^{*})\}.$$

Considérons aussi un autre sous-ensemble $\hat{\mathbf{O}} \subset \mathbf{U}$, $|\hat{\mathbf{O}}| = L$, possédant au moins un élément e_x non appartenant à $\mathbf{O}$, c'est-à-dire $e_x \in \mathbf{O}, e_x \notin \hat{\mathbf{O}}$.

Le plus équitable paquet respecte l'inégalité :

$$\sum_{\forall k, k \notin \mathbf{O}} f_k^*(f_k - f_{\mathbf{O}_x}) < \sum_{\forall k, k \notin \hat{\mathbf{O}}} f_k^*(f_k - f_{\hat{\mathbf{O}}_x}), \quad \forall \hat{\mathbf{O}}, \tag{3.4}$$

où $\mathbf{O}_x$ est choisi de telle façon que $\mathbf{O}_x \leq f_k < \mathbf{O}_{x+1}$. Néanmoins, l'approche classique est incapable de produire ce paquet si pendant *au moins* une des étapes il y a une composante $e_v \in \mathbf{O}$ dont :

$$f_{e_v}^*(f_{e_v} - f_{e_{v-1}}) < f_{e_k}^*(f_{e_k} - f_{e_{k-1}}), \quad \forall k, k \neq v. \tag{3.5}$$

Nous allons maintenant réaliser cette analyse dans le cas où le vecteur temporaire possède 4 composantes dont deux doivent être éliminées (c'est-à-dire, la source transmet 2 couches vidéo). Dans ce cas, $\vec{t}$ est :

$$\vec{t} = \{(f_{e_1}, f_{e_1}^*), (f_{e_2}, f_{e_2}^*), (f_{e_3}, f_{e_3}^*), (f_{e_4}, f_{e_4}^*)\}. \tag{3.6}$$

Supposons que les composantes qui donnent le paquet le plus équitable soient e_1 and e_3. Pour que cela soit vrai, les composantes doivent respecter les deux inégalités suivantes :

$$\begin{cases} f_{e_2}^*(f_{e_2} - f_{e_1}) < f_{e_3}^*(f_{e_3} - f_{e_2}) + f_{e_4}^*(f_{e_3} - f_{e_2}), \\ \\ f_{e_4}^*(f_{e_4} - f_{e_3}) < f_{e_3}^*(f_{e_3} - f_{e_2}) + f_{e_3}^*(f_{e_2} - f_{e_1}). \end{cases} \tag{3.7}$$

En combinant le équations (3.4) et (3.7), et après quelques manipulations, on obtient :

$$\begin{cases} f_{e_3}^*(f_{e_3} - f_{e_2}) < f_{e_2}^*(f_{e_2} - f_{e_1}) < f_{e_3}^*(f_{e_3} - f_{e_2}) + f_{e_4}^*(f_{e_3} - f_{e_2}), \quad \text{et} \\ \\ f_{e_3}^*(f_{e_3} - f_{e_2}) < f_{e_4}^*(f_{e_4} - f_{e_3}) < f_{e_3}^*(f_{e_3} - f_{e_2}) + f_{e_3}^*(f_{e_2} - f_{e_1}). \end{cases} \tag{3.8}$$

Ainsi, la méthode classique ne produit pas le paquet d'acquittement le plus équitable si l'équation 3.8 est respectée.

3.3 L'Algorithme Direct

Nous proposons dans cette thèse l'Algorithme Direct [16, 33, 34] qui vise à éliminer les limitations intrinsèques à l'approche classique présentées dans la section précédente. Pour un nœud quelconque dans le réseau, l'Algorithme Direct donne *toujours* comme résultat le paquet le plus équitable, grâce à son mécanisme interne d'exécution en une seule boucle.

Revenons à notre session multicast de base où la source transmet L couches et dans un nœud intermédiaire le vecteur temporaire est $\vec{t}$, $|\vec{t}| = c_t$. L'Algorithme Direct évolue de la façon suivante. On fait $l_1 = f_{e_1}$ et calcule $(l_2, \ldots, l_L)$ de sorte à minimiser la *somme des pertes* Δ_S :

$$\Delta_S = \sum_{\forall k, k \notin \mathbf{O}} f_k^* (f_k - f_{\mathbf{O}_x}). \tag{3.9}$$

Les valeurs $(l_2, \ldots, l_L)$ sont choisies de telle façon qu'elles coïncident avec un sous-ensemble des composantes du vecteur temporaire. Ainsi, dans un premier temps, on fait

$$\begin{aligned} l_2 &= f_{e_2}, \\ l_3 &= f_{e_3}, \\ \vdots &\ \vdots \ \vdots \\ l_L &= f_{e_L}. \end{aligned}$$

et on calcule la dégradation globale. On modifie la combinaison, par exemple en faisant $l_L = f_{e_{L+1}}$, et ainsi de suite jusqu'à ce que toutes les possibilités soient testées.

Avec l'Algorithme Direct, toutes les composantes en surplus sont éliminées en une seule étape, car tous les paquets candidats sont examinés. La somme des pertes Δ_S étant calculée, on assure que la qualité globale est localement optimisée dans chacun des nœuds intermédiaires.

Nous présentons dans le cadre ci-dessous le pseudo-code de l'Algorithme Direct.

Algorithme Direct (**in :** vecteur temporaire ;
out : paquet le plus équitable)

$\delta = 0$;

$\delta_{tmp} = 0$;

finish = 1 ;

Calcule $\mathbf{O}^{tmp} \leftarrow$ "premier paquet candidat ;

tant que ($\exists$ "prochain paquet candidat")

{

  pour ($k = 1$;$k \leq c_T$;$k + +$)

    $\delta_{tmp} = \delta_{tmp} + f_k^*(f_k - f_{\mathbf{O}_x})$;

  si ($\delta_{tmp} < \delta$) **alors**

  {

    $\delta = \delta_{tmp}$;

    $\mathbf{O} = \mathbf{O}^{tmp}$;

  }

  Calcule prochain $\mathbf{O}^{tmp} \leftarrow$ "paquet candidat" ;

}

3.3.1 L'approche hybride

Il est clair que l'on se pose des questions en ce qui concerne la complexité de la méthode proposée en termes de scalabilité. Effectivement, si la quantité de paquets rentrant dans un nœud est trop grande, le nombre de paquets candidats sera également élevé. Nous discutons maintenant cette limitation et soutenons qu'en réalité elle n'arrive pas souvent. Dans le cas échéant, nous montrons comment cette restriction peut être résolue en utilisant l'approche hybride.

Soit n le nœud intermédiaire qui exécute à un certain instant t une procédure de combinaison de paquets. Lui sont connectés k liens d'entrée p_i^n, $i = 1 \ldots k$, et un lien de sortie p_{out}^n vers la source. Nous allons analyser la complexité de l'algorithme dans le cas extrême où la charge de traitement dans le nœud est maximale. Cela permettra l'établissement de bornes supérieures sur la performance du système proposé.

Nous allons tout d'abord adopter quelques hypothèses, à savoir :

- tous le liens rentrant $p_i^n, i = 1 \ldots k$, appartiennent à l'arbre multicast et possèdent par conséquent dans un "sous-nœud" au moins un récepteur qui est en train d'envoyer des paquets d'acquittement ;
- au moment d'exécuter l'algorithme de combinaison de paquets, un paquet d'acquittement est arrivé par lien rentrant. Cela signifie que tous les paquets arrivent au nœud n dans un intervalle $\Delta(t)$ inférieur ou égal au temporisateur T démarré au moment de l'arrivée du premier paquet d'acquittement. Autrement dit, si le premier paquet d'une séquence de liens arrive à t_0 et le dernier à t_{last}, alors l'inégalité $t_0 + T > t_{last}$ doit être respectée ;
- tous les paquets en amont qui arrivent au nœud possèdent le nombre maximal de composantes, qui est donné par la quantité de couches transmises par la source ;
- après l'organisation des paquets dans le vecteur temporaire $\vec{t}$, toutes les composantes sont "non-compatibles", c'est-à-dire qu'elles n'ont pas les débits suffisamment proches pour être automatiquement combinées (voir section 3.1.2).

Nous insistons ici sur le fait que la première tâche de l'algorithme de combinaison de paquets est de fusionner (et d'organiser, bien entendu) les paquets d'acquittement. Dans le cas de notre analyse, le nombre de composantes dans le vecteur temporaire est :

$$\hat{n}_e = kL. \tag{3.10}$$

Les paquets candidats sont obtenus quand $\hat{n}_e - L$ composantes sont éliminées du vecteur temporaire. Le nombre de paquets candidats est calculé par la formule combinatoire suivante :

$$\delta = \frac{(\hat{n}_e - 1)!}{(L-1)!(\hat{n}_e - L)!} = \frac{\prod_{i=1}^{kL+1}(kL-i)}{\prod_{i=1}^{L+1}(L-i)\prod_{i=L}^{kL+1}(kL-i)} = \frac{\prod_{i=1}^{L-1}(kL-i)}{\prod_{i=1}^{L+1}(L-i)} \tag{3.11}$$

La figure 3.4 montre le comportement de δ face à différentes valeurs de L et de k. On peut observer que l'influence de L et k sur δ devient très importante à partir d'un certain seuil k' mais aussi à partir d'un certain nombre de couches L'. Dans le réseau multicast pour l'Internet que l'on connaît actuellement, à savoir le réseau MBone, le nombre de liens attachés à un seul nœud s'est toujours maintenu à des niveaux assez faibles. C'est pourquoi les valeurs k et L sont, en pratique, relativement petites. Considérons par exemple une source vidéo avec un codeur MPEG-2 intégré, dont le nombre de couches vidéo peut monter jusqu'à trois. Dans les courbes de la figure 3.4, pour $L = 3$ on peut remarquer que δ reste dans une tranche raisonnable même pour les valeurs de k les plus élevées. Par exemple, si $k = 15$ (c'est-à-dire, le nœud doit avoir quinze branches appartenant à l'arbre multicast) et si chaque paquet possède le nombre maximum de composantes, alors le nombre de paquets candidats est 1000. Même pour cette valeur, le calcul du paquet le plus équitable est rapide étant donné que l'algorithme de combinaison de paquets n'utilise que des opérations de somme et de comparaison.

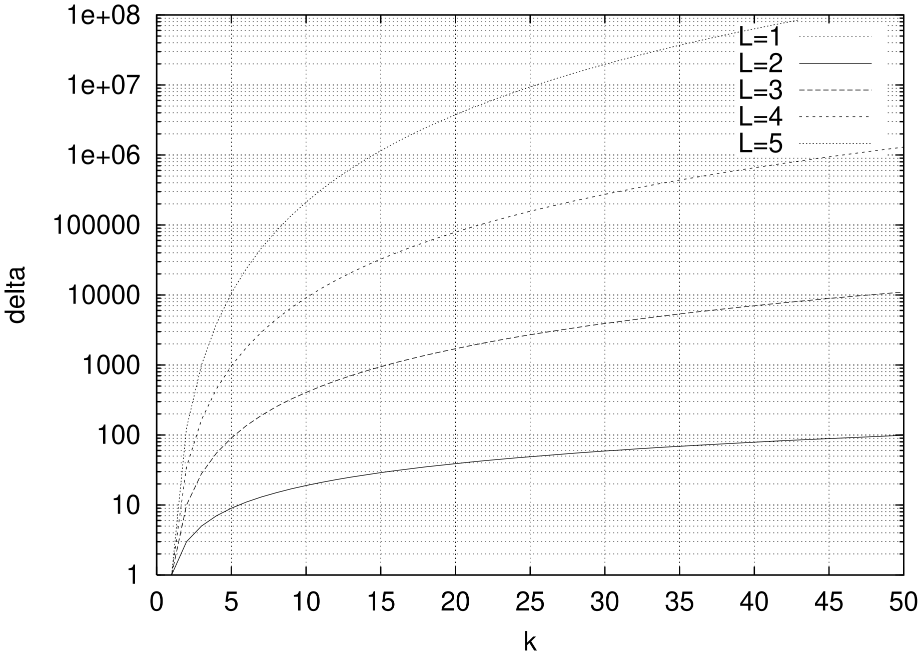


FIG. 3.4 – Comment δ réagit à k et L.

Pour des valeurs plus élevées, nous proposons l'approche hybride qui marie l'Algorithme Direct à l'approche classique. La technique hybride est exécutée en deux étapes principales. La première consiste à appliquer l'algorithme classique jusqu'à ce que le nombre de composantes du paquet temporaire soit réduit à un certain seuil β pré-défini. Lorsque ce seuil est atteint, la procédure de combinaison de paquets est commutée à l'Algorithme Direct. Cela maintient la complexité du système sous contrôle (directement proportionnelle au nombre d'étapes exécutées par l'approche classique plus une constante donnée par le seuil β) tout en permettant à la procédure de combinaison d'améliorer l'équité du paquet de sortie.

La figure 3.5 dépeint le nombre de paquets candidats obtenus par l'approche hybride pour une série de valeurs de L et de k. Nous avons choisi pour le seuil de commutation entre les deux algorithmes la valeur $\beta_{max} = 100$. En confrontant ces résultats à ceux de la figure 3.4, on peut observer que la charge introduite par l'Algorithme Direct est facilement contrôlée par l'approche hybride.

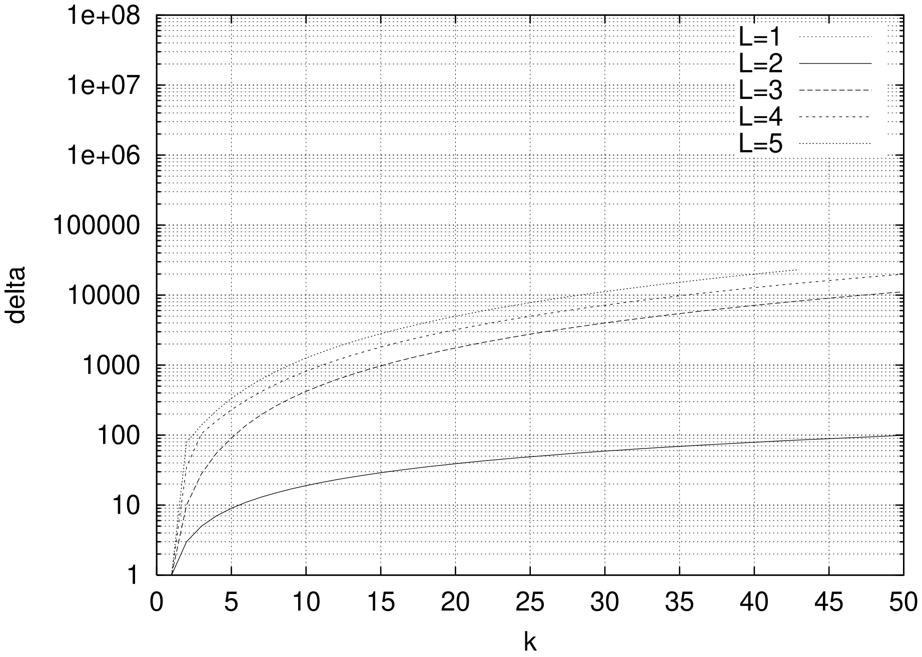


FIG. 3.5 – Comment δ réagit à k et L pour l'approche hybride.

3.4 Les couches virtuelles

Il existe une limitation dans les réseaux que nous allons étudier dans cette section. Il s'agit des informations existantes dans les sous-réseaux adjacents à un nœud qui exécute une procédure de combinaison de paquets.

Soit n_x un nœud quelconque dans le réseau et n_y un autre nœud non appartenant au sous-arbre associé à n_x et vice-versa. Il est facile de voir que tous les récepteurs dans le sous-arbre de n_y ne provoquent aucune interférence sur le paquet de sortie de n_x. Bien que la signalisation du réseau se maintienne faible, le mécanisme de combinaison de paquets est limité en ce qui concerne la qualité maximale qu'il peut obtenir.

Dans tel scénario non-coopératif, le schéma qui utilise les couches virtuelles entraîne les nœuds intermédiaires à conserver une quantité extra d'informations de contrôle. Cela se fait par une réduction du nombre de composantes éliminées par l'algorithme de combinaison.

En effet, dans chaque nœud intermédiaire exécutant la procédure d'élimination de composantes, la valeur de L (le nombre de couches transmises par la source) est localement modifiée en une autre valeur $\lambda > L$. Au lieu d'éliminer $c_T - L$ composantes, le nœud en supprime $c_T - \lambda$. Ainsi, un nombre supplémentaire de composantes suit le chemin vers la source et peut donc rencontrer d'autres composantes "compatibles" auxquelles elles pourront être combinées. Les trois cas particuliers sont :

- $\lambda = 0$: aucun paquet n'arrive à la source ;
- $\lambda = L$: on revient au cas normal sans couches virtuelles ;
- $\lambda = \infty$: tous les acquittements arrivent à la source.

Nous allons voir par la suite comment le schéma avec couches virtuelles peut améliorer la qualité globale de la session multicast.

Considérons par exemple le scénario de la figure 3.6 où la source est censée transmettre 2 couches vidéo. Dans les conditions normales où aucune couche virtuelle n'est utilisée, le nœud n_2 élimine $c_T - L = 1$ composante, qui correspond, dans notre exemple, à $e_2 = (5, 1)$. Évidemment, cette information est égarée et ne pourra plus être récupérée dans les nœuds suivants dans le chemin restant vers la source.

Dans le sous-arbre qui lie le nœud n_3 aux récepteurs il y a aussi un client qui demande le même débit $f = 5$. Toutefois, si cette composante n'avait pas été éliminée dans le nœud n_2, l'algorithme dans le nœud n_1 aurait supprimé la bonne composante, c'est-à-dire celle relative à $f = 10$, ce qui nous donnerait le meilleur niveau de qualité.

En effet, le but de l'utilisation des couches virtuelles est de réduire la probabilité qu'une composante soit éliminée et qu'une autre composante compatible dans un sous-arbre "proche" puisse lui être incorporée.[2]

[2]On dit qu'un sous-arbre est "proche" d'un autre si leurs paquets de sortie respectifs se rencontrent au

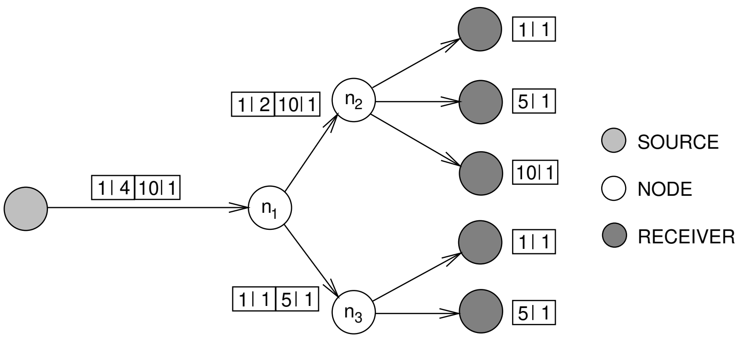


FIG. 3.6 – Schéma sans couches virtuelles.

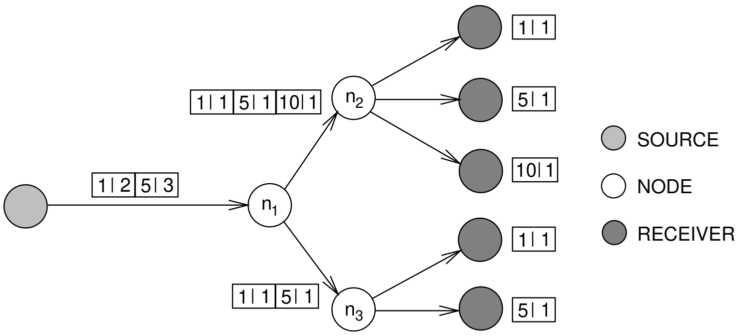


FIG. 3.7 – Schéma avec une couche virtuelle supplémentaire.

3.5 Analyse

Nous avons étudié dans les sections précédentes comment améliorer le mécanisme de combinaison de paquets dans les nœuds intermédiaires afin d'augmenter la qualité globale de la session multicast. Cet objectif a abouti à la proposition de l'Algorithme Direct qui exécute toute la procédure de combinaison de paquets en une seule étape.

Cette section est réservée à la performance de l'Algorithme Direct face aux résultats obtenus par l'approche classique. Nous nous intéressons à la dégradation globale de la vidéo (autrement dit, l'inverse de la qualité globale, cf. section 3.1.1) ainsi qu'à la consommation de bande-passante.

Nous réalisons un certain nombre de simulations en utilisant deux types distincts de

maximum après un nombre h_{max} de *hops*.

topologie. Le premier type est consacré aux réseaux avec des récepteurs concentrés (que nous allons désormais appeler 'en mode dense'). Un réseau est considéré dense si les nœuds sont connectés à des grandes quantités de liens. Le second type de réseau est en mode dilué, dont chaque nœud n'est connecté qu'à un nombre limité de liens. Un troisième type serait un réseau du type hybride, où certains nœuds seraient en mode dense et d'autres en mode dilué. Nous nous restreignons cependant aux deux premiers types pour bien identifier les différences de comportement entre l'approche classique et l'Algorithme Direct, dans des situations distinctes.

Dans les topologies en mode dense, le nœuds sont censés éliminer un grand nombre de composantes à chaque procédure de combinaison de paquets. L'objectif est d'observer combien l'Algorithme Direct peut être proche de la borne empirique. La borne empirique est le meilleur résultat qui peut être obtenu. Elle correspond au cas où une seule procédure de combinaison de paquets est exécutée (à la source).

Pour les topologies du type dilué, on évalue le comportement de l'Algorithme Direct en cascade et on vérifie que, pour éviter les limitations des sous-réseaux non-coopératifs, l'approche qui utilise les couches virtuelles mène le système à des résultats plus intéressants.

Nous rappelons encore une fois que la variable utilisée pour évaluer l'efficacité des techniques est la dégradation globale de la vidéo dans les récepteurs. Pour chacun des récepteurs, on calcule la différence entre le débit requis et le débit reçu (cf. section 3.1.1). On vérifiera également le rapport entre le nombre de couches transmises par la source et le degré de satisfaction des clients.

La topologie utilisée pour simuler un réseau en mode dense est illustrée dans la figure 3.8. Il s'agit d'une source et de dix récepteurs connectés à un nœud. Tous les liens ont la même capacité $C = 1$ Mbps et délais identiques $\tau = 10$ ms. L'utilisation de délais identiques et d'un réseau symétrique nous permet de confiner les résultats à des valeurs plus dépendantes de la bande-passante disponible et moins à d'autres paramètres du réseau. Pour faire varier la bande-passante disponible, nous insérons dans chaque lien un trafic de fond du type exponentiel.

La figure 3.9 représente la dégradation globale de la session multicast pour la topologie

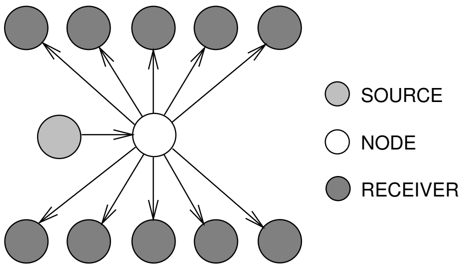


FIG. 3.8 – Topologie en mode dense.

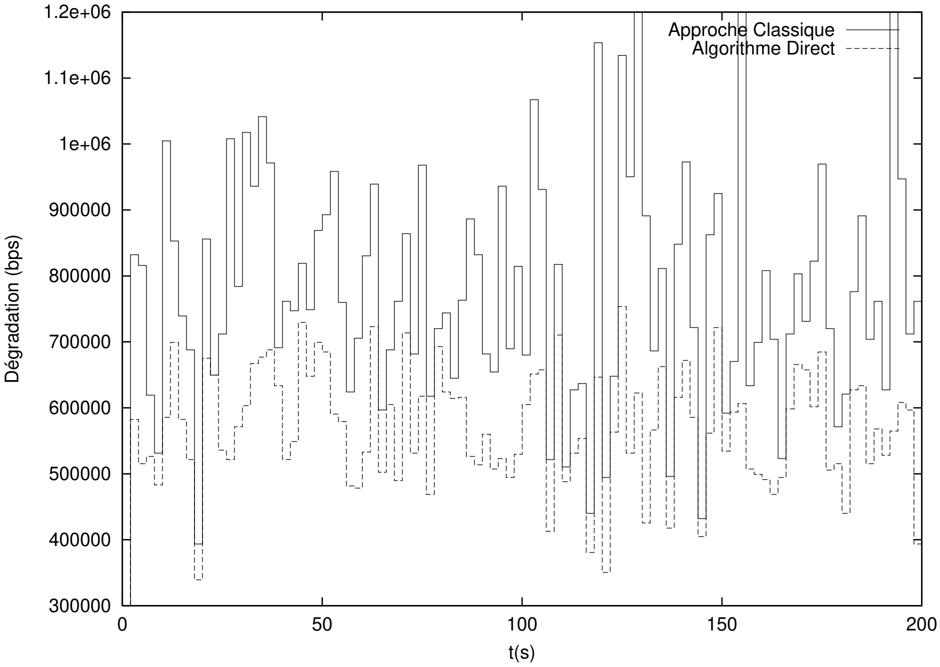


FIG. 3.9 – Dégradation globale pour la topologie 1.

de la figure 3.8. Il est aisé de voir que l'Algorithme Direct mène *toujours* aux meilleurs résultats en termes de dégradation de la qualité de la vidéo par rapport à l'approche classique. Ce rapport est mieux représenté par la courbe de la figure 3.10. Il est intéressant d'observer que le rapport moyen est $\sim 25\%$, ce qui est assez considérable pour un réseau relativement petit comportant 10 récepteurs.

Une particularité est liée à ces résultats. L'Algorithme Direct donne non seulement les meilleurs résultats mais aussi la borne empirique. Bien évidemment, c'était exactement ce que l'on attendait car, l'arbre de diffusion ayant une profondeur égale à 2 (une seule procédure de combinaison de paquets est exécutée), toutes les possibilités de combinaison de paquets ont été analysées dans la même boucle de calculs. De toute façon, l'Algorithme

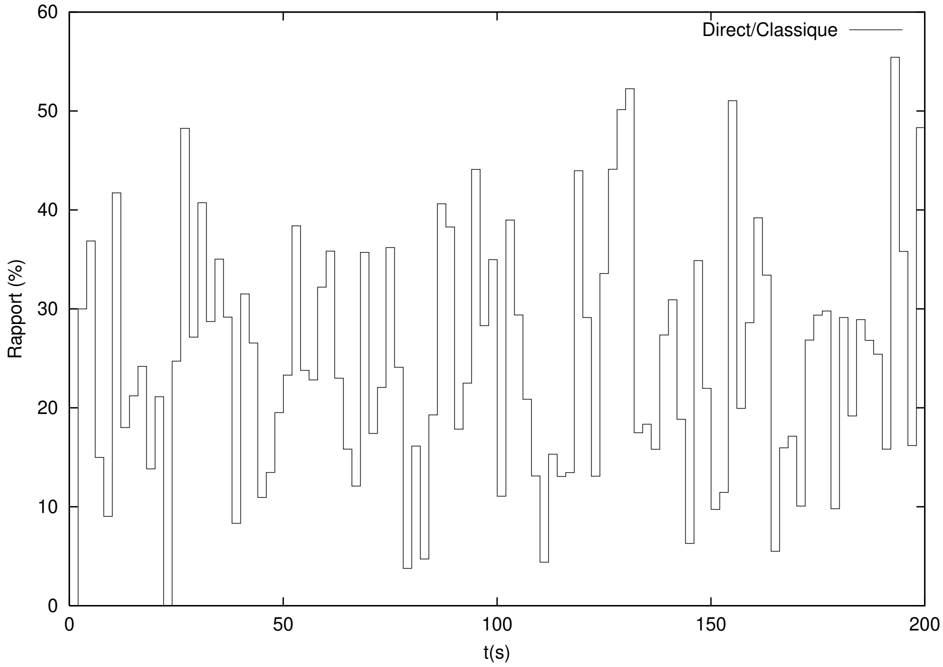


FIG. 3.10 – Rapport entre l'Algorithme Direct et le schéma classique pour la topologie 1.

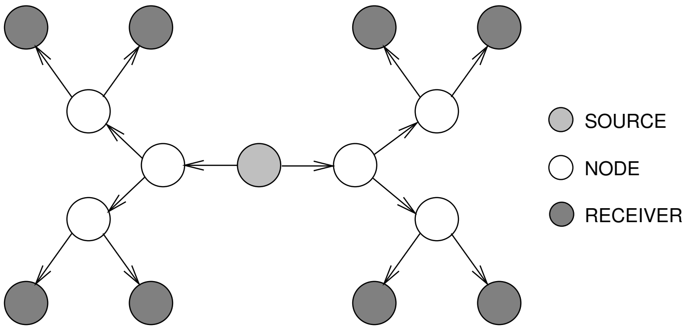


FIG. 3.11 – Topologie en mode dilué avec 8 récepteurs.

Direct est intrinsèquement plus performant que n'importe quel autre mécanisme dans les réseaux non-coopératifs.

Le même type d'analyse a été effectué sur le réseau en mode dilué de la figure 3.11. Cette topologie possède 8 récepteurs distribués dans une structure du type arbre binaire. Les paramètres de simulation sont les mêmes de la topologie précédente, c'est-à-dire avec des liens symétriques où $C = 1$ Mbps et délais $\tau = 10$ ms.

Les dégradations globales de la session multicast pour l'Algorithme Direct et pour l'approche classique sont illustrées dans la figure 3.12. Il est clair que dans ce cas-ci les

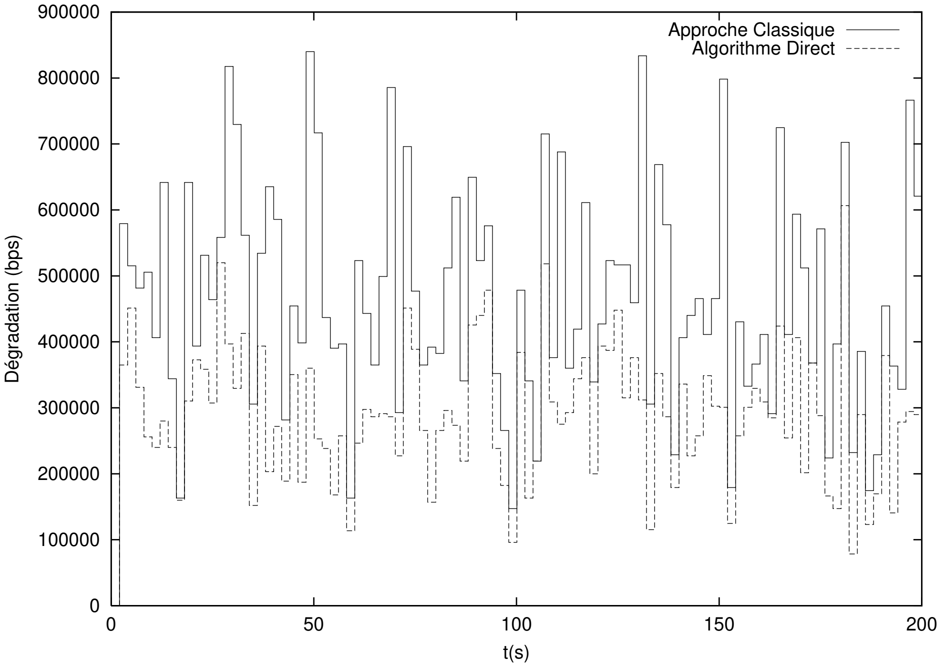


FIG. 3.12 – Dégradation globale pour la topologie 2.

résultats des deux techniques sont moins éloignés que dans le cas de la topologie 1. Il est également évident d'observer que l'approche classique est toujours délimitée par la courbe issue de l'Algorithme Direct. Les raisons pour lesquelles les deux techniques obtiennent des résultats plus proches sont le nombre de composantes éliminées dans chaque nœud et, bien sûr, la profondeur de l'arbre multicast.

Dans le cas de la topologie 1, le nœud qui exécute l'algorithme doit supprimer à chaque tour $c_T - L = 7$ composantes. La probabilité que l'approche classique se trompe de composante à éliminer est plus importante étant donné le nombre élevé d'étapes consécutives qu'elle exécute. De l'autre côté, pour la topologie 2 l'algorithme ne doit éliminer que $c_T - L = 3$ composantes au maximum dans chaque nœud. Cela s'explique par le fait que l'algorithme est distribué sur toute l'arborescence.

De la même manière que pour la topologie 1, la figure 3.13 montre le rapport entre l'Algorithme Direct et l'approche classique pour la topologie 2. Les résultats obtenus par l'Algorithme Direct sont en moyenne 35% meilleurs que ceux de l'approche classique en au moins 50% du temps de simulation.

Maintenant, nous allons approfondir l'analyse des topologies du type dense, plus pré-

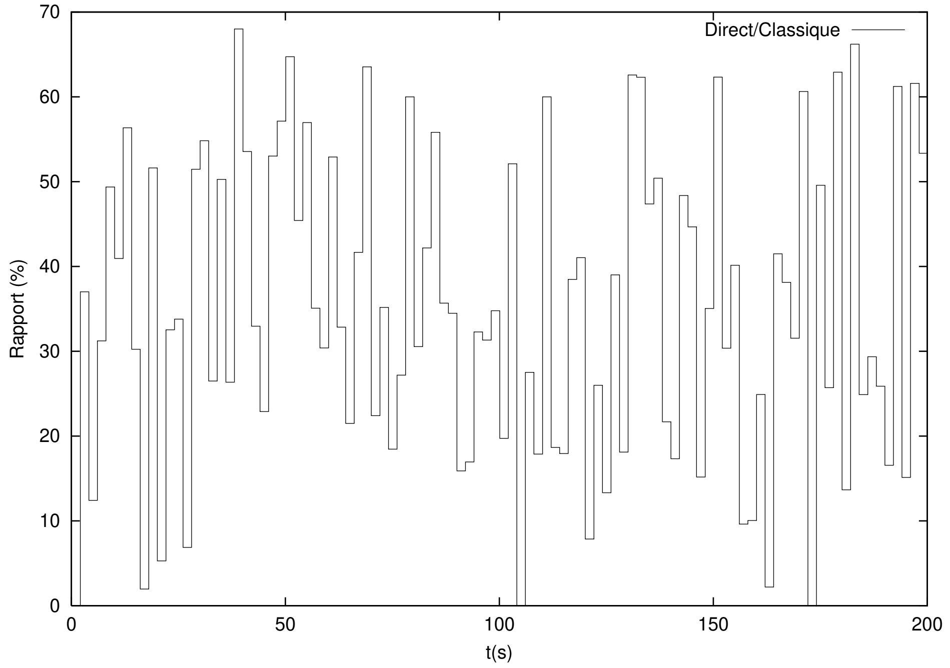


FIG. 3.13 – Rapport entre l'Algorithme Direct et le schéma classique pour la topologie 2.

cisément sur l'influence de la profondeur de l'arbre multicast sur les techniques de combinaison de paquets. Le réseau des prochaines simulations est dessiné dans la figure 3.14. Il s'agit de l'arborescence de la topologie 2 avec un degré de profondeur de plus. Ce qui change aussi, bien évidemment, est le nombre de récepteurs servis par la même vidéo. Nous avons gardé les paramètres de simulation des topologies précédentes. Le nombre de composantes supprimées dans chaque nœud après l'exécution d'une procédure de combinaison de paquets est égal à 3, comme dans le cas avec 8 récepteurs. Nous pouvons par inférence conclure que 3 est aussi le nombre maximum d'étapes successives exécutées par l'approche classique.

Nous faisons un petit rappel que, pour la topologie 1, l'approche classique exécutait le calcul des couches 7 fois de suite. Pour la topologie 2, cette valeur était de 3. Maintenant, la valeur est aussi 3, mais dans un arbre binaire plus profond. Il est intuitif de penser que l'utilisation de l'Algorithme Direct sera encore plus avantageuse pour la topologie 3.

Les figures 3.15 et 3.16 montrent que ce gain a effectivement lieu. La simple augmentation d'un niveau dans la profondeur de l'arbre multicast a généré des valeurs de dégradation globale de la session multicast pour l'Algorithme Direct qui sont en moyenne

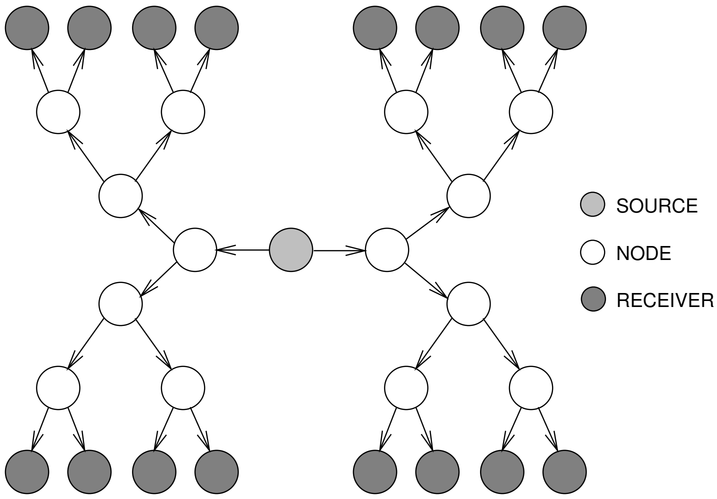


FIG. 3.14 – Topologie en mode dilué avec 16 récepteurs.

55% meilleures que celles de l'approche classique, contre 35% dans le cas précédent.

Nous allons maintenant considérer l'utilisation des couches virtuelles par l'Algorithme Direct. Nous nous concentrerons uniquement sur la topologie 3 pour deux raisons principales. D'abord, il est inutile d'utiliser l'approche par couches virtuelles dans le cas de la topologie 1 puisque les résultats obtenus sont déjà ceux correspondants à la borne empirique. Rien ne peut alors être amélioré.

Étant donné que la proposition de l'utilisation des couches virtuelles a été faite pour perfectionner le comportement de l'Algorithme Direct face à des réseaux plus "profonds", la topologie 2 sera en fait représentée par la topologie 3.

La figure 3.17 montre les courbes obtenues lorsque, respectivement, 4, 5 et 6 couches virtuelles sont utilisées. Comme pour les autres simulations, nous montrons aussi, dans la figure 3.18, le rapport entre l'Algorithme Direct et l'approche classique pour 6 couches virtuelles. Il est aisé de remarquer qu'avec un simple changement de variable (le nombre de couches transmises par la source), il est possible de réduire la dégradation globale de la vidéo dans les récepteurs.

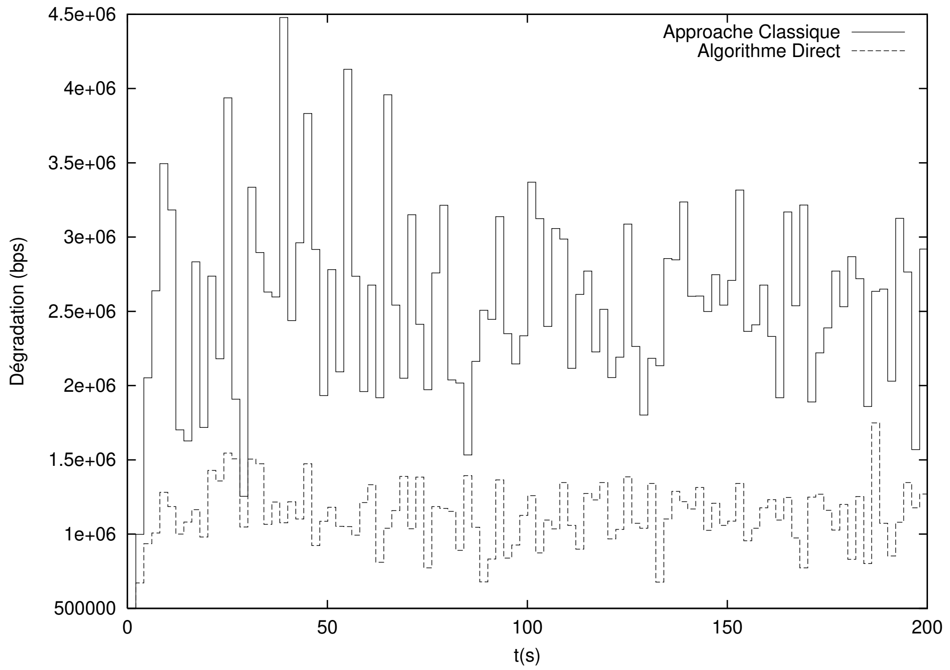


FIG. 3.15 – Dégradation globale pour la topologie 3.

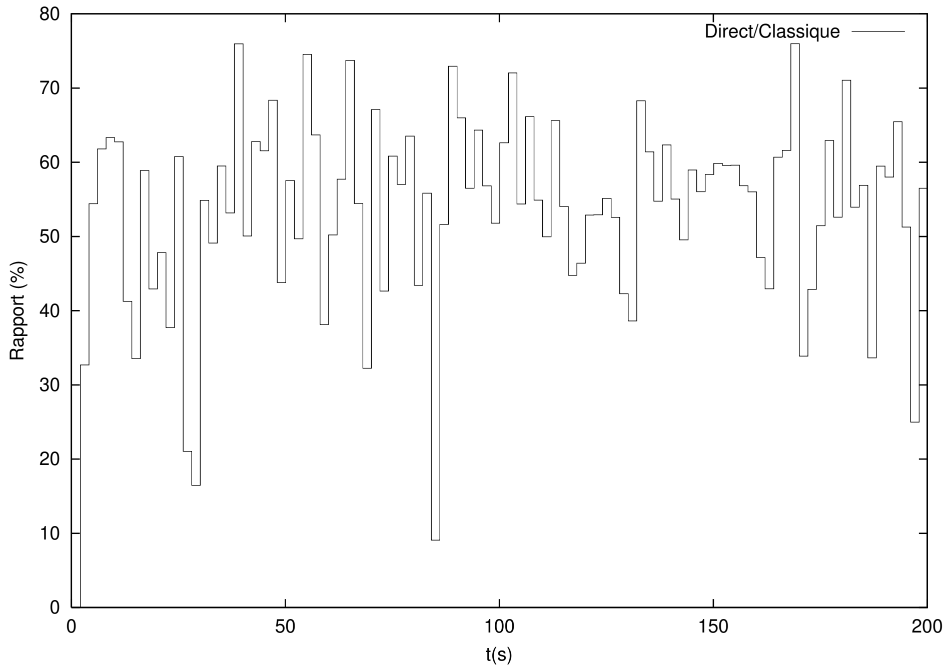


FIG. 3.16 – Rapport entre l'Algorithme Direct et le schéma classique pour la topologie 3.

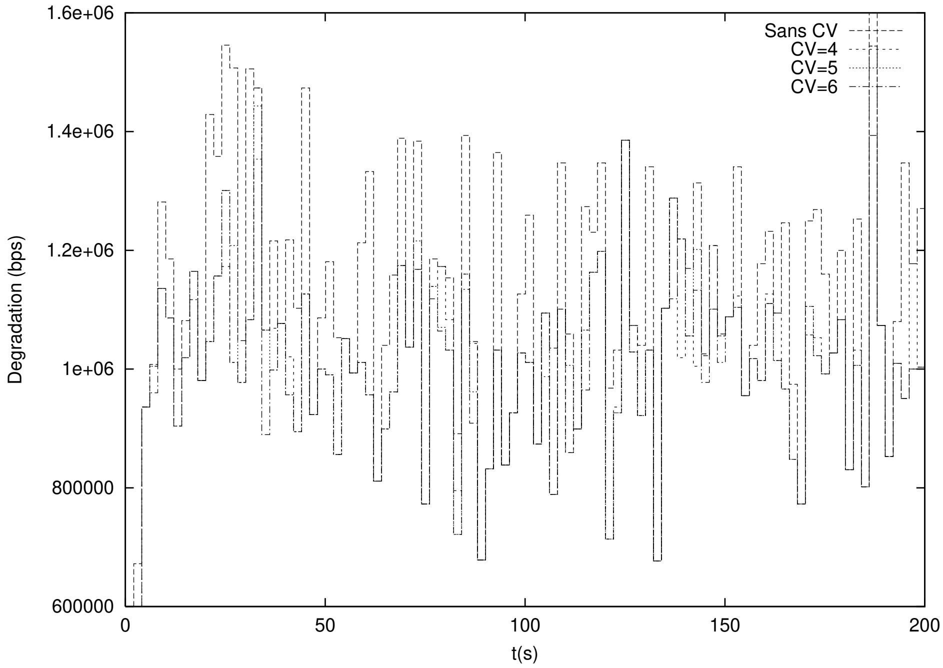


FIG. 3.17 – Dégradation globale pour la topologie 3 lorsque l'on utilise des couches virtuelles.

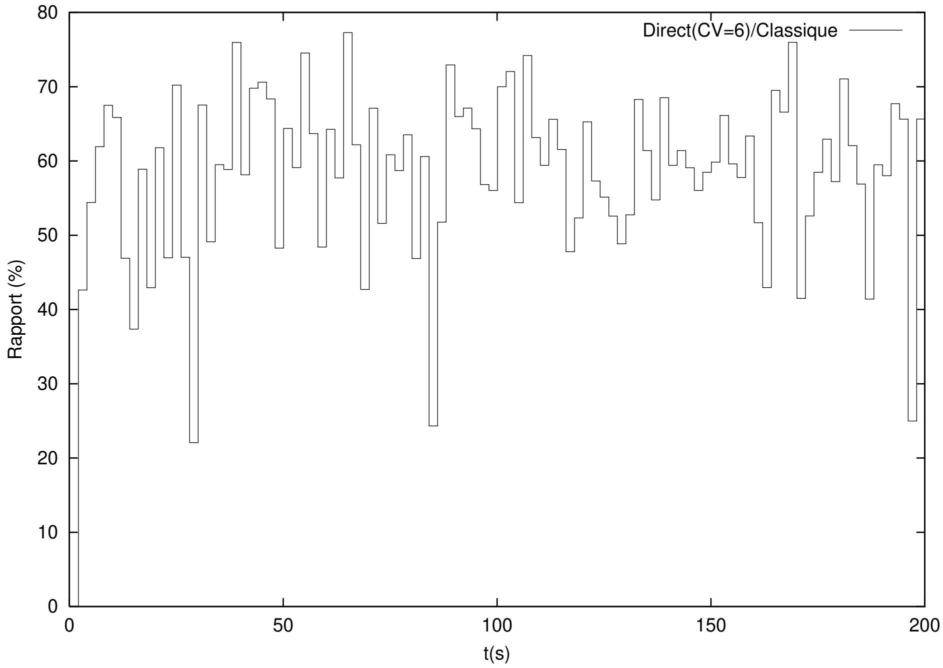


FIG. 3.18 – Rapport entre l'Algorithme Direct et le schéma classique pour la topologie 3 avec 6 couches virtuelles.

3.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème d'implosion de paquets d'acquittement dans la source lorsque le nombre de récepteurs appartenant à une session multicast est trop important. Nous avons montré que l'utilisation de paquets de contrôle contenants des informations sur la capacité de réception des clients permet à la source de calculer les débits d'une vidéo codée en mode multi-couches.

Les approches jusqu'à présent proposées dans la littérature utilisent le concept de "débit conforme". Elles visent à améliorer la qualité globale de la communication en calculant dans chaque nœud le débit maximum qui pourrait être transmis après l'exécution d'une procédure de combinaison de paquets. Les études que nous avons réalisées nous ont pourtant montré que la façon dont l'approche classique calcule les paquets de sortie peut ne pas être performante lorsque un nombre plus élevé de paquets d'entrée doit être analysé.

La raison pour laquelle les résultats ne sont pas optimaux est liée à la succession de calculs exécutés pour chaque ensemble de paquets entrant. L'Algorithme Direct que nous avons proposé dans ce chapitre résout ce type de problème en réalisant la combinaison des paquets en une seule étape. Cela signifie que les résultats obtenus sont les meilleurs possibles dans les réseaux du type non-coopératifs (cf. section 3.4).

La clef de l'Algorithme Direct est l'analyse totale (exhaustive) des possibilités. Lorsque dans un nœud, l'algorithme examine tous les paquets candidats, il est possible d'obtenir la configuration exacte qui mène à la meilleure qualité globale de la session multicast. Le paquet de sortie qui est envoyé en aval correspond donc aux valeurs optimales obtenues dans le nœud.

De façon à contourner la limitation naturelle des réseaux non-coopératifs, nous avons aussi proposé l'Algorithme Direct avec le concept de couches virtuelles. Avec celles-ci, les paquets de sortie transportent une quantité supplémentaire d'information sur les récepteurs connectés au sous-arbre duquel ils sont arrivés.

Nous avons analysé l'Algorithme Direct ainsi que l'approche classique à travers plusieurs simulations. Nous nous sommes intéressés à deux types principaux de réseaux : en mode dense, dont les nœuds intermédiaires sont connectés à un grand nombre de liens entrants ; et en mode dilué, dont les récepteurs sont bien distribués dans les liens.

Les résultats ont montré que l'Algorithme Direct donne *toujours* les meilleures valeurs en terme de qualité de la session multicast. Pour les topologies du type dilué, l'approche classique est bornée par la qualité de l'Algorithme Direct et la distance entre les deux techniques est intimement liée à la profondeur de l'arbre multicast.

Dans le cas des topologies du type dense, la différence de performance entre les deux approches est encore plus remarquable. Puisque cette différence est surtout liée au nombre d'étapes successives exécutées par l'approche classique, les réseaux à caractère concentré forment le scénario idéal pour l'Algorithme Direct. De plus, la qualité obtenue est inversement proportionelle à la profondeur de l'arbre multicast et, lorsque cette profondeur s'approche de l'unité, les résultats obtenus par l'Algorithme Direct se dirigent vers la borne optimale, c'est-à-dire la borne empirique.

Chapitre 4

La qualité avec des paramètres multiples

Calculer les débits des couches en utilisant la dégradation de l'application dans les récepteurs a été le sujet abordé dans le chapitre précédent. Durant le développement de cette partie, nous avons remarqué que pour certains scénarios il serait plus intéressant d'introduire d'autres arguments au moment du calcul des couches. Seule l'utilisation de la dégradation de qualité est déjà efficace, mais nous allons montrer que la qualité globale de la communication peut être encore améliorée si d'autres métriques sont utilisées [31, 32, 20].

Dans ce chapitre nous allons étudier cette approche multi-métriques en utilisant comme paramètres la densité de récepteurs satisfaits, l'allocation de la bande-passante dans les réseaux et, bien évidemment, celle que nous utilisions déjà, la dégradation de la qualité.

4.1 La métrique simple

Nous allons montrer dans cette partie que l'utilisation d'une métrique unique pour décrire le niveau de qualité d'une session multicast peut être assez restreinte autant du point de vue de l'utilisateur que celui du réseau. Dans le chapitre précédent, nous avons étudié des mécanismes pour améliorer la qualité globale de la session multicast à travers l'optimisation de l'algorithme de combinaison de paquets dans les nœuds intermédiaires.

Les techniques appliquées étaient caractérisées par la distribution des débits de manière à minimiser la dégradation globale dans les récepteurs. Maintenant, nous nous intéresserons à la conception des systèmes qui intègrent aussi bien la qualité de la session multicast du point de vue des utilisateurs que du point de vue du réseau.

Soit toujours $\vec{t}$ le vecteur temporaire dans un nœud quelconque dans le réseau. On sait que $|\vec{t}| = c_T$ correspond à la taille du vecteur et que $\mathbf{O} \subset \vec{t}$ est le plus équitable sous-ensemble de $\vec{t}$.

Notons $\Psi[\mathbf{0}]$,

$$\Psi[\mathbf{O}] = \sum_{i=1}^{|\mathbf{O}|} f_i f_i^*, \tag{4.1}$$

la puissance du vecteur $\mathbf{O}$. On dit qu'un vecteur $\mathbf{O}$ est plus puissant qu'un autre vecteur $\mathbf{O}'$ si $\Psi[\mathbf{O}] > \Psi[\mathbf{O}']$.

En effet, $\Psi[\mathbf{O}]$ nous fournit une idée de l'allocation moyenne de la bande-passante dans l'ensemble des liens du réseau. Notre but est de montrer que des configurations quasi identiques des couches peuvent donner des résultat extrêmement différents en terme de consommation des ressources des liens.

Revenons à notre analyse. Soit $\Delta^{\mathbf{O}}_{\vec{t},\tau_\ominus}$ la dégradation de la session multicast lorsque le vecteur de sortie à l'instant $\tau_\ominus$ est $\mathbf{O}$. Le dégradation obtenue pour le même sous-ensemble $\mathbf{O}$ à l'instant $\tau_\oplus$ est $\Delta^{\mathbf{O}}_{\vec{t},\tau_\oplus}$. Il est possible qu'à un instant précis $\hat{\tau}$, $\tau_\ominus < \hat{\tau} < \tau_\oplus$, le contenu d'une composante e_k du vecteur $\vec{t}$ ait changé d'état :

$$e_{k,\tau_\ominus} \in \vec{t} \ : \begin{cases} f_{e_k} \\ f^*_{e_k} \end{cases} \quad \rightarrow \quad e_{k,\tau_\oplus} \in \vec{t} \ : \begin{cases} f_{e'_k} \\ f^{*'}_{e_k} \end{cases} \tag{4.2}$$

Sans perte de généralité, mais uniquement pour simplifier l'analyse, nous allons considérer que $f^{*'}_{e_k} = f^*_{e_k}$. Par contre, $f_{e'_k}$ diffère de f_{e_k} d'une valeur ϵ pouvant être même très petite :

$$f_{e'_k} = f_{e_k} + \epsilon. \tag{4.3}$$

Pour le même sous-ensemble de composantes $\mathbf{O}$, la nouvelle valeur de la dégradation est simplement :

$$\Delta_{\vec{t},\oplus}^{\mathbf{O}} = \Delta_{\vec{t},\ominus}^{\mathbf{O}} + f_{e_k^*}\epsilon. \tag{4.4}$$

Il se peut que $\exists \mathbf{O}'$ tel que

$$\Delta_{\vec{t},\oplus}^{\mathbf{O}} > \Delta_{\vec{t},\oplus}^{\mathbf{O}'}, \tag{4.5}$$

même si

$$\Delta_{\vec{t},\ominus}^{\mathbf{O}} < \Delta_{\vec{t},\ominus}^{\mathbf{O}'}. \tag{4.6}$$

Il n'existe aucune contrainte qui empêche $\Psi[\mathbf{O}'] \gg \Psi[\mathbf{O}]$. On voit que, en pratique, il est possible qu'une faible variation en une des composantes du vecteur temporaire entraîne un grand changement de la puissance du vecteur de sortie. La question que l'on peut se poser est relative aux avantages que l'on peut obtenir en améliorant d'une valeur ϵ la dégradation de la session multicast tout en gardant une grande utilisation des ressources. Il est intuitif de penser qu'une petite perte de qualité peut être compensée par un niveau équilibré d'utilisation de la bande-passante.

Nous allons maintenant présenter, à travers d'un exemple qui montre en petite échelle ce qui peut se passer dans un réseau de dimensions réelles, le comportement que nous venons d'examiner.

Soit la topologie avec une source et quatre récepteurs illustrée dans la figure 4.1. La source est configurée pour transmettre deux couches vidéo, ce qui se traduit par l'exécution de l'algorithme de combinaison de paquets uniquement dans le nœud n_1. Nous allons vérifier que n'utiliser que la dégradation de la qualité est parfois la cause d'une mauvaise performance globale du système.

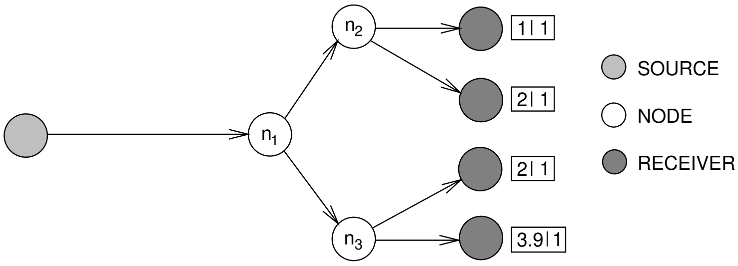


FIG. 4.1 – Topologie exemple avec une source et quatre récepteurs.

Le récepteur le moins performant (r_1) est capable de recevoir la vidéo au débit maximum égal à 1, ce qui donne la valeur de la couche de base l_1 codée à la source. Le calcul principal concerne donc l'obtention du débit de la couche d'amélioration l_2.

Notons b^{r_j} le débit demandé par le récepteur r_j et c_k,

$$c_k = \sum_{i=1}^{k} l_i, \tag{4.7}$$

le débit donné par l'ensemble des couches des k premiers niveaux hiérarchiques. La variable c_k n'a en réalité aucune signification spéciale. Nous l'utilisons uniquement pour alléger l'analyse.

Nous avons dans les circonstances présentées ci-dessus deux choix pour c_2. On fait soit $c_2 = b^{r_2} = b^{r_3}$, soit $c_2 = b^{r_4}$. Le tableau 4.1 montre les résultats obtenus pour les deux cas.

Bien évidemment, l'algorithme choisit la première configuration puisqu'elle donne la plus faible dégradation de la vidéo. Cette configuration est aussi plus économique que la deuxième en terme de la quantité de données injectée dans le réseau (c'est-à-dire, un débit moins élevé pour une meilleure qualité de la vidéo).

Supposons qu'à un certain instant $\hat{\tau}$, la valeur de b^{r_4} change de 3.9 à 4.1. Le tableau 4.2 illustre les possibilités.

Le nouvelles valeurs des couches qui optimisent la dégradation de la vidéo sont $c_1 = 1$ et $c_2 = 4.1$ (la dégradation globale correspondante est $\Delta_s = 2$). Il est aisé de voir qu'une simple variation de 5% de la capacité de réception de r_4 a provoqué une augmentation de

TAB. 4.1 – Dégradation pour $c_2 = b^{r_2} = b^{r_3}$ et $c_2 = b^{r_4}$.

Récepteur	Choix 1 : $c_2 = b^{r_2} = b^{r_3} = 2$			Choix 2 : $c_2 = b^{r_4} = 3.9$		
	requis	reçu	dégradation	requis	reçu	dégradation
r_1	1	1	0	1	1	0
r_2	2	2	0	2	1	1
r_3	2	2	0	2	1	1
r_4	3.9	2	1.9	3.9	3.9	0
Dégradation globale			1.9			2

TAB. 4.2 – Dégradation pour $c_2 = b^{r_2} = b^{r_3}$ et $c_2 = b^{r_4}$.

Récepteur	Choix 1 : $c_2 = b^{r_2} = b^{r_3} = 2$			Choix 2 : $c_2 = b^{r_4} = 4.1$		
	requis	reçu	dégradation	requis	reçu	dégradation
r_1	1	1	0	1	1	0
r_2	2	2	0	2	1	1
r_3	2	2	0	2	1	1
r_4	4.1	2	2.1	4.1	4.1	0
Dégradation globale			2.1			2

216% du débit de transmission de la source (de 1.9 à 4.1). De plus, pour le premier choix il y a trois récepteurs entièrement satisfaits et seulement un à 50% de sa capacité totale. Pour le second choix, deux récepteurs sont 100% servis et les deux autres à 50%.

Intuitivement, nous sommes poussés à affirmer qu'il serait plus intéressant de garder la première configuration (choix 1) même si elle implique une dégradation globale un peu plus basse que celle de la seconde configuration (choix 2).

À partir de l'analyse et de l'exemple précédents, on en conclut que l'utilisation de paramètres autres que la dégradation globale de la session multicast, notamment l'utilisation de la bande-passante et le nombre de clients satisfaits, est une voie intéressante pour la distribution multicast en couches. Passons maintenant à l'étude des métriques, puis à la proposition du système à multiples métriques.

4.2 L'étude des métriques

À l'issue de l'étape précédente, nous sommes dans la situation où il faut déterminer le comportement du système face à l'inclusion des nouveaux paramètres. Il est clair qu'il s'agit d'un problème de nature subjective et la décision du choix de la configuration dépend essentiellement de l'application en question.

Voyons quelques exemples. Une application du type VoD (*video-on-demand*) qui utilise la technologie *pay-per-view* exige que la totalité des clients soit servie avec la meilleure qualité possible tout le temps. On peut traduire ces exigences par l'optimisation de la dégradation globale de la session multicast. Dans le cas d'une application de vidéo-surveillance, le plus important est de garantir une qualité au-delà d'un seuil minimal. Il est cependant acceptable que cette qualité soit un peu dégradée au bénéfice d'une économie des ressources (communication moins chère). La publicité en large échelle est un cas classique où l'intérêt serait de toucher le plus grand nombre possible de clients.

Nous suivons l'étude faite dans la section précédente et proposons un système basé sur trois métriques pour déterminer la qualité de la session multicast. Les trois métriques sont :

– γ : la densité d'utilisateurs satisfaits ;
– ω : la bande-passante pondérée allouée ;
– δ : la dégradation de qualité de la session multicast.

Il est implicite que le paquet d'acquittement doit être modifié pour ajouter toutes ces informations de contrôle. La figure 4.2 représente le nouveau format des paquets contenant les métriques ajoutées. La suite montre comment ces métriques se comportent et la façon dont elles sont calculées lors d'une procédure d'agrégation de paquets dans les nœuds intermédiaires.

f_{i-1}	f^*_{i-1}	γ_{i-1}	ω_{i-1}	δ_{i-1}	f_i	f_i^*	γ_i	ω_i	δ_i	f_{i+1}	f^*_{i+1}	γ_{i+1}	ω_{i+1}	δ_{i+1}

FIG. 4.2 – Structure du nouveau paquet d'acquittement.

La première métrique est la *densité d'utilisateurs satisfaits*, représentée par γ. Pour un récepteur r_i demandant b^{r_i} unités de bande-passante, on dit que r_i est "partiellement satisfait" s'il reçoit $b_r^{r_i}$, où $b_r^{r_i} < b^{r_i}$.

On définit le degré de satisfaction du récepteur r_i :

$$\gamma_{r_i} = \frac{b_r^{r_i}}{b^{r_i}}. \tag{4.8}$$

Dans une session multicast avec des récepteurs hétérogènes, nous calculons donc la degré globale de satisfaction des utilisateurs en combinant les γ_{r_i} de chacun des récepteurs.

Soit un nœud quelconque dans le réseau possédant k liens entrants et un lien sortant comme illustré par la figure 4.3. Afin d'analyser le système dans le cas extrême, tous les liens entrants possèdent un paquet d'acquittement au moment de l'exécution de la procédure de combinaison de paquets. Pour chacun des paquets $p_i, i = 1 \ldots k$, on choisit $l_j \in \mathbf{S}_L$ de telle façon que

$$\sum_{a=1}^{j} l_a \leq f_{p_i} < \sum_{a=1}^{j+1} l_a. \tag{4.9}$$

Ce calcul est exactement celui de la section 4.1, où on appelle l_j la couche adoptive. On fait ensuite :

$$c_{p_i} = \sum_{a=1}^{j} l_a, \tag{4.10}$$

qui correspond au débit accumulé associé à p_i.

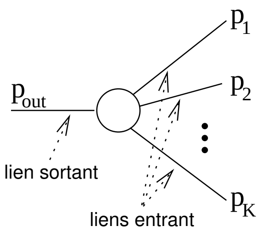


FIG. 4.3 – Nœud d'analyse.

Le paquet de sortie p_{out} possède comme résultat la somme pondérée de tous les γ_i, $i = 1 \ldots k$, plus la perte associée à l'opération d'agrégation $f_{p_i} \to c_{p_i}$, dont :

$$\gamma_{out} = \frac{\sum_{i=1}^{K} m_{i,c} f^*_{p_i} \frac{c_{p_i}}{f_{p_i}} \gamma_{p_i}}{\sum_{i=1}^{K} f^*_{p_i}}. \tag{4.11}$$

Dans l'équation ci-dessus, $m_{i,j}$ est un élément de la matrice unitaire $M_{k,L}$ respectant les règles suivantes :

$$m_{i,j} = \begin{cases} 1 & \text{si } \sum_{a=1}^{j} l_a \leq f_{p_i} < \sum_{a=1}^{j+1} l_a, \\ 0 & \text{sinon.} \end{cases} \tag{4.12}$$

La deuxième métrique est la *bande-passante pondérée allouée* au travers de l'arbre multicast, notée par ω. Le degré d'utilisation de la bande-passante est calculé directement à la source en combinant les débits des couches et le nombre de requêtes d'abonnements à ces couches. En effet, à chaque saut qu'un paquet d'acquittement effectue dans le réseau, le champ réservé pour ω dans chaque composante est augmenté d'une unité. Lorsque ce paquet arrive à la source, on connaît exactement le nombre de liens qui seront traversés par chacune de couches. Il est donc possible d'estimer la quantité de ressources utilisée par la session multicast en question.

Dans un nœud quelconque du réseau, la valeur globale de ω est $\omega^{out} = 1 + \sum_{i=1}^{K} \omega_{p_i}$. En fait, ω assume deux différents types de valeurs. Dans les nœuds du réseau, ω correspond à un chiffre indiquant le nombre de liens traversés par le paquet d'acquittement. À la source, ce chiffre est multiplié par le débit correspondant pour donner la bande-passante pondérée. Faire de ω à la fois "poids" et bande-passante est uniquement pour faciliter le calcul exécuté dans le nœuds.

Ainsi, il faut à la source calculer l'allocation pondérée de la bande-passante qui est la suivante :

$$\omega_{out} = \frac{\sum_{i=1}^{L} (f_{r_i}^* f_{r_i})}{\sum_{i=1}^{L} f_{r_i}^*}. \tag{4.13}$$

Le comportement de ω suit le même principe de γ : il est inversement proportionnel à la qualité de la session multicast ; il faut minimiser ω pour augmenter la qualité.

La troisième et dernière métrique est bien évidemment celle déjà utilisée précédemment : la dégradation de la session multicast δ. Nous répétons ici le calcul de δ suivant la notation utilisée pour les deux autre métriques. On définit tout d'abord la *dégradation locale* dans le récepteur r_i la différence entre le débit requis b^{r_i} et le débit reçu $b_r^{r_i}$:

$$\delta_{r_i} = b^{r_i} - b_r^{r_i}. \tag{4.14}$$

Du point de vue de la source, la dégradation est interprétée comme étant l'ensemble des dégradations locales, ce qui donne :

$$\delta = \sum_{i=1}^{N} \delta_{r_i}, \tag{4.15}$$

où N est le nombre total de récepteurs appartenant à la session multicast.

De la même manière que pour les analyses précédentes, un certain nombre de paquets sont combinés en un paquet de sortie contenant un nombre de composantes égal au nombre de couches transmises par la source[1]. Dans le nœud d'étude de la figure 4.3, chaque

[1] Nous considérons ce scénario uniquement pour alléger la notation, bien que l'on pourrait appliquer la technique des couches virtuelles. De toute façon, cela n'implique aucune altération des résultats.

composante $e_k \in \vec{t}$ porte la dégradation dans le sous-arbre associé. La procédure de combinaison de paquets calcule la dégradation du paquet de sortie :

$$\delta_{out} = \frac{\sum_{i=1}^{K} m_{i,c} f^{*}_{e_i} \frac{c_{e_i}}{f_{e_i}} \delta_{e_i}}{\sum_{i=1}^{K} f^{*}_{e_i}}. \tag{4.16}$$

4.2.1 Discussion

Les procédures d'optimisation des métriques doivent procéder différemment. D'une part, la densité des clients satisfaits γ doit être calculée de façon à maximiser le résultat. D'autre part, l'objectif est de minimiser la dégradation de la qualité et la bande-passante allouée. Il est cependant aisé de voir que ces objectifs ne sont pas forcément convergents. Nous sommes donc obligés de trouver un compromis entre les métriques et d'établir des conditions d'équilibre dans lesquelles le système comme un ensemble est considéré optimal[2].

Dans le but d'illustrer le comportement des trois métriques, nous allons montrer un exemple qui prouve la divergence des objectifs. La topologie du réseau est illustrée dans la figure 4.4. Il s'agit d'un scénario simple mais qui met bien en évidence les caractéristiques que nous voulons étudier.

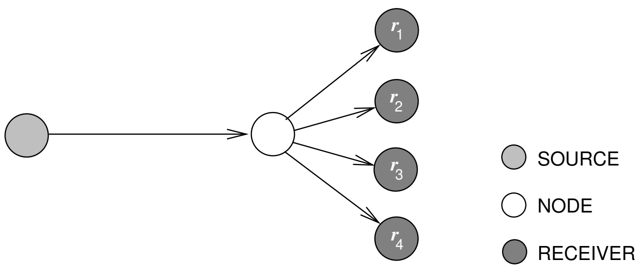


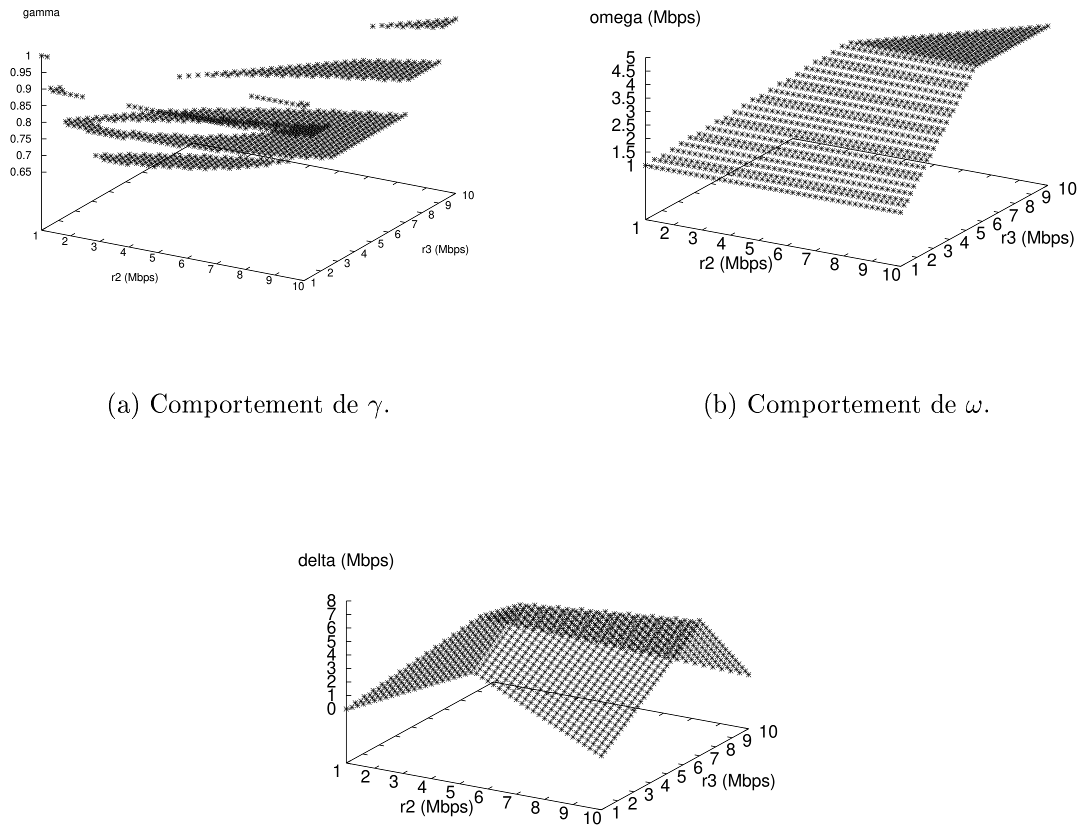
FIG. 4.4 – Topologie exemple pour l'étude de la divergence entre les métriques.

L'étude se poursuivra de la façon suivante. Conformément à la notation que nous avons déjà utilisée dans les chapitres précédents, appelons b^{r_i} la capacité de réception du récepteur r_i. Nous maintenons constantes les capacités de réception des récepteurs r_1 et

[2] Il n'est pas difficile de voir qu'il s'agit ici d'un problème de nature subjective. Ainsi, l'adjectif "optimal" dépendra de définitions préétablies liées à une application spécifique.

r_4 et varions celles de r_2 et de r_3. L'intervalle auquel sont soumises les b^{r_2} et b^{r_3} varie de b^{r_1} à b^{r_4}.

La figure 4.5 exhibe le comportement de chacune des métriques face au scénario présenté ci-dessus. La première courbe (figure 4.5(a)) montre l'évolution de la densité d'utilisateurs satisfaits. Pour des valeurs de b^{r_2} et b^{r_3} proches des extrémités de l'intervalle, γ s'oriente vers la valeur maximale. Pour des capacités de réception au milieu de l'intervalle, on obtient une moins bonne densité.



(a) Comportement de γ.

(b) Comportement de ω.

(c) Comportement de δ.

FIG. 4.5 – Comportement des métriques dans la topologie exemple.

La dégradation δ exhibe un comportement antagoniste à celui de la densité d'utilisateurs satisfaits. La réponse de δ aux variations de b^{r_2} et b^{r_3} tel qu'il a été décrit est illustré dans la figure 4.5(c). La courbe montre que l'on obtient des dégradations plus importantes pour les valeurs de b^{r_2} et b^{r_3} au milieu de l'intervalle.

La même analyse a été faite pour la quantité de bande-passante pondérée allouée dans le réseau. Les résultats sont illustrés dans la figure 4.5(b). Uniquement dans le but d'alléger la représentation de la courbe, nous avons varié b^{r_3} de b^{r_1} à b^{r_4} et b^{r_2} de b^{r_3} à b^{r_4}. Cette décision ne change en rien l'analyse car pour les valeurs de b^{r_2} entre b^{r_1} et b^{r_3} on obtiendrait une courbe symétrique à celle de la figure 4.5(b). Le plus important est d'observer que ω se comporte encore différemment, autant de γ que de δ. ω est minimisée lorsque b^{r_2} et b^{r_3} sont également minimaux. Nous nous attendions bien sûr à avoir ce résultat étant donné que la bande-passante allouée dans le réseau est directement proportionnelle aux débits reçus par les récepteurs individuellement.

Revenons à la situation idéale où (a) la densité d'utilisateurs satisfaits est maximale ; (b) la dégradation globale de la session multicast est minimale ; et (c) le degré d'utilisation des ressources du réseau est également minimal. Nous pouvons constater à partir des courbes de la figure 4.5, et au moins pour ce scénario, que les trois objectifs ne peuvent être atteints en même temps.

Nos propositions consistent à fournir une base de contrôle sur les métriques que nous étudions dans ce chapitre. Le système permet, selon des critères que nous allons établir, de désigner des niveaux de priorité aux différentes métriques. Il est possible ainsi de déterminer jusqu'à quel point une métrique sera dégradée au profit d'une autre.

La première des propositions travaille sur l'attribution de poids aux métriques. Il s'agit en fait d'un mécanisme qui introduit dans l'algorithme de combinaison de paquets une nouvelle mesure de qualité. Cette qualité est calculée à partir d'une somme pondérée des métriques et donne une estimation de l'influence de chacune d'elles sur le calcul effectué dans les nœuds.

La seconde proposition utilise une approche différentielle lors de l'élimination des composantes du vecteur temporaire $\vec{t}$ dans les nœuds du réseau. Le calcul de la qualité est fait à partir de la variation de (γ, δ, ω) lorsqu'une composante est éliminée, et non plus de leurs valeurs absolues.

4.3 La qualité globale de la communication

Nous avons analysé dans la session précédente le comportement des trois métriques que nous avons proposées pour augmenter le degré de contrôle sur le comportement du système d'acquittements.

Nous avons tout d'abord classé les applications en trois groupes principaux qui exigent l'optimisation de :

- la densité d'utilisateurs satisfaits γ ;
- la bande-passante pondérée allouée ω ;
- la dégradation de qualité de la session multicast δ.

Basés sur les exigences décrites ci-dessus, nous définissons la *qualité globale de la session multicast* le paramètre $\mathcal{Q}$ issu de la création d'une métrique unique à partir de γ, ω et δ. On peut facilement souligner que $\mathcal{Q}$ ne possède aucune particularité. La seule contrainte sur le comportement de la qualité globale de la session multicast est établie par la définition suivante :

Définition 1 *La qualité globale de la session multicast, $\mathcal{Q}$, augmente de valeur si, selon les critères établis, les règles suivantes sont en moyenne respectées :*
- *la densité d'utilisateurs satisfaits augmente ;*
- *la bande-passante pondérée allouée diminue ;*
- *la dégradation de qualité de la session multicast diminue.*

La qualité globale de la session multicast est donc une fonction de γ, ω et δ et, par conséquent, nous utiliserons dorénavant la notation $\mathcal{Q}(\gamma, \omega, \delta)$. Malgré des corrélations entre ces variables, nous avons décidé de les garder indépendantes. Cette simplification permet que la gestion de multiples variables dans le cas que l'on étudie soit plus facilement menée.

De façon à satisfaire nos besoins, mais toujours suivant les règles établies, nous avons défini la fonction $\mathcal{Q}(\cdot)$ comme suit :

$$Q(\gamma, \omega, \delta) = f_\gamma(\gamma) + f_\omega(\omega) + f_\delta(\delta). \tag{4.17}$$

Nous souhaitons également, pour des raisons de simplicité, que $\mathcal{Q}(\cdot)$ soit normalisé entre 0 et 1. À moins d'un facteur d'échelle, les fonctions $\{f_\gamma(\gamma), f_\omega(\omega), f_\delta(\delta)\}$ sont aussi normalisées entre 0 et 1.

Étant donné que γ, ω et δ sont directement liés à la topologie du réseau, nous ne pouvons pas définir d'une manière précise une fonction close qui correspondrait exactement à nos besoins. Pour implémenter telle caractéristique, nous calculons les valeurs minimales et maximales que chacune des métriques peut assumer, permettant ainsi de les normaliser.

Il est aisé de voir que dans les trois cas la valeur minimale est 0. La valeur 0 est équivalente à une transmission à 0 bps dans les cas de γ et ω et, pour δ, à la situation où tous les récepteurs seraient servis avec exactement les bande-passantes demandées.

Dans l'autre extrême, les valeurs maximales sont :

$$\gamma_{max} = 1 \tag{4.18}$$

$\Rightarrow$ Cette valeur correspond au cas où tous les récepteurs sont servis à 100%.

$$\omega_{max} = \lim_{\max[f^*_{p_i}] \to \sum_{i=1}^{L} f^*_{p_i}} \frac{\sum_{i=1}^{L} f^*_{p_i} f_{p_i}}{\sum_{i=1}^{L} f_{p_i}} = \max[f_{p_i}] \tag{4.19}$$

$\Rightarrow$ L'équation ci-dessus illustre la situation où tous les récepteurs sont servis avec toutes les couches.

$$\delta_{max} = \sum_{i=1}^{K} (f^*_{p_i} f_{p_i} + \delta_{p_i}) \tag{4.20}$$

$\Rightarrow$ La valeur maximale de δ correspond au cas où aucun récepteur n'est servi.

Nous pouvons maintenant définir le fonction $\mathcal{Q}$ exacte :

$$Q(\gamma, \omega, \delta) = \frac{a\frac{\gamma}{\gamma_{max}} + b\left(1 - \frac{\omega}{\omega_{max}}\right) + c\left(1 - \frac{\delta}{\delta_{max}}\right)}{3(a+b+c)}, \tag{4.21}$$

où les variables a, b et c sont les poids attribués à chacune des métriques. Ainsi, selon le type d'application en cours, il est possible de configurer le système de façon à donner aux différentes métriques les priorités qui leur sont mieux indiquées. À travers l'ajustement de ces variables, on peut donc placer l'application dans le plus adéquat "pool" de services.

4.4 Simulation

Dans cette partie nous allons analyser le comportement de l'algorithme avec la métrique unique proposée dans la section précédente. Nous utilisons pour cela une session multicast avec une source et dix récepteurs comme dépeint la figure 4.6. Les liens sont symétriques de même capacités et les délais homogènes de façon à faire des résultats dépendants uniquement de la bande-passante disponible. Les ressources variables dans les liens sont obtenues avec l'introduction de trafics de fond générés par des sources étrangères. Ces sources sont du type exponentiel dont les paramètres sont différents pour chacun des liens. La bande-passante disponible dans les liens est égale bien sûr à la différence entre la capacité du lien et le trafic de fond total.

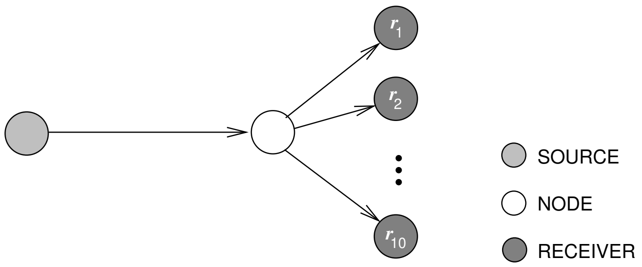


FIG. 4.6 – Topologie de test.

Comme nous en avons discuté auparavant, notre but est de permettre aux différentes applications de choisir quelle métrique doit avoir la plus importante priorité. Cela est entièrement dépendant de l'application et des clients. Nous avons choisi pour notre première simulation une application désirant l'optimisation de l'utilisation des ressources.

Nous avons sélectionné deux liens particuliers pour être représentés dans nos résultats. Ils sont les liens l_2 et l_7 qui représentent, respectivement, les liens le moins et le plus congestionnés du réseau (autrement dit, les liens le moins et le plus utilisés).

Nous illustrons dans la figure 4.7 le bande-passante demandée et la bande-passante reçue par le récepteur attaché au lien l_2. À partir de $t \approx 15s$, à peine la moitié de la bande-passante disponible dans le lien est utilisée. Cela signifie que les paquets d'acquittement envoyés par le récepteur r_2 sont combinés à d'autres paquets provenant d'un récepteur demandant à peu près la moitié de la capacité de r_2.

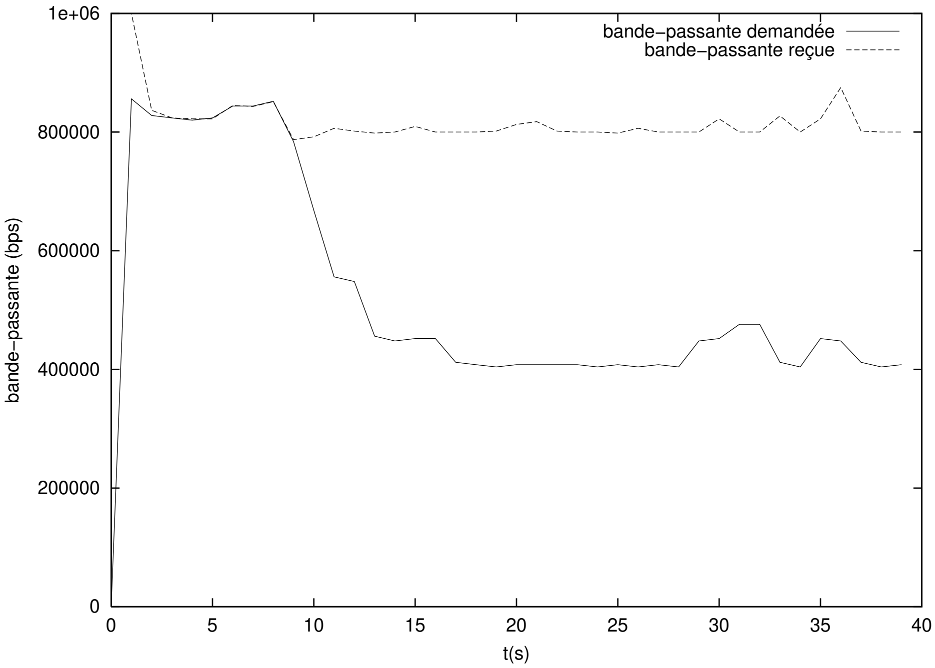


FIG. 4.7 – Utilisation du lien l_2.

La figure 4.8 montre la bande-passante disponible ainsi que son utilisation dans le lien l_7. Il est facile de conclure à partir des résultats que le récepteur (ou les récepteurs) demandant ce débit fournit un paquet d'acquittement qui n'est pratiquement jamais combiné à un autre paquet provenant d'un autre récepteur. Cela s'explique par le fait que la bande-passante requise est la même que celle reçue la plupart du temps.

Nous montrons maintenant le résultat qui réellement nous intéresse dans le contexte de cette analyse. Rappelons que la métrique prioritaire dans nos simulations est la bande-passante pondérée allouée à travers l'arbre multicast. La figure 4.9 illustre le gain obtenu par l'approche multi-métriques par rapport à l'approche classique, qui optimise uniquement la dégradation dans les récepteurs.

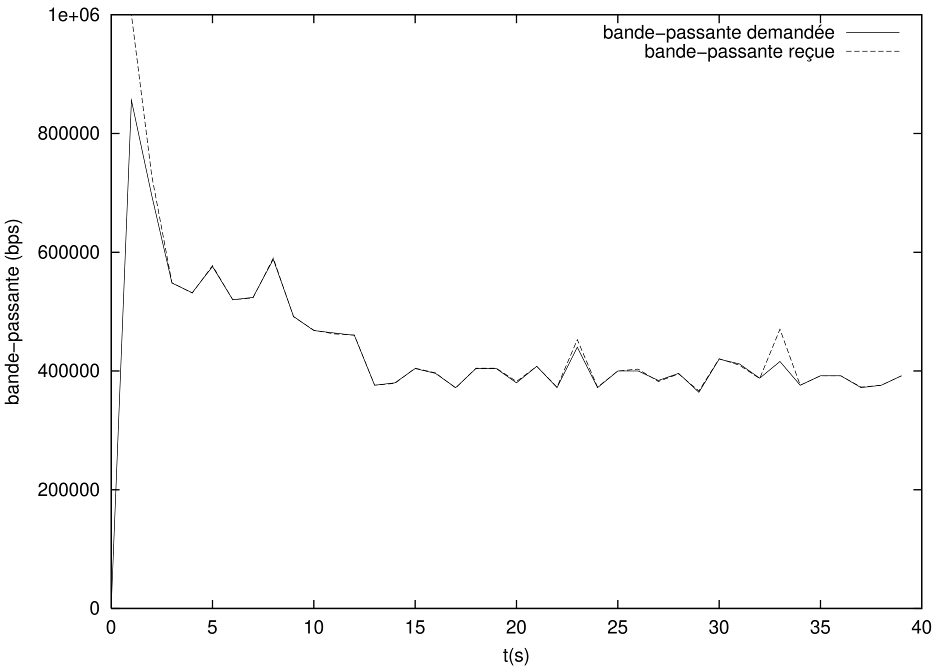


FIG. 4.8 – Utilisation du lien l_7.

4.5 L'approche différentielle

Le problème qui a été traité dans la section précédente réserve un certain nombre de caractéristiques particulières. Une des plus importantes est la nature subjective du problème. On peut définir assez facilement quelle est par exemple la métrique qui doit être prioritaire. Néanmoins, dire combien cette métrique doit valoir est une tâche qui ne peut pas être aisément exécutée. La difficulté s'appuie principalement sur les variations de l'état du réseau.

La technique que nous avons appliquée jusqu'à présent dans ce chapitre, puisqu'elle utilise les valeurs absolues des métriques, n'arrive pas à identifier ces variations du réseau. Le calcul de la qualité de la communication Q (cf. équation 4.21) nous donne un moyen d'identifier les composantes devant être éliminées lors de l'exécution de l'algorithme de réduction. Toutefois, ce calcul ne permet pas une séparation totale entre les variables. Une qualité $Q = Q_x$ peut représenter différentes configurations des métriques : γ grand, ω moyen et δ petit ; ou tous les trois moyens ; ou n'importe quelle autre configuration donnant comme résultat la même valeur Q_x. Ainsi, au cours d'une même communication,

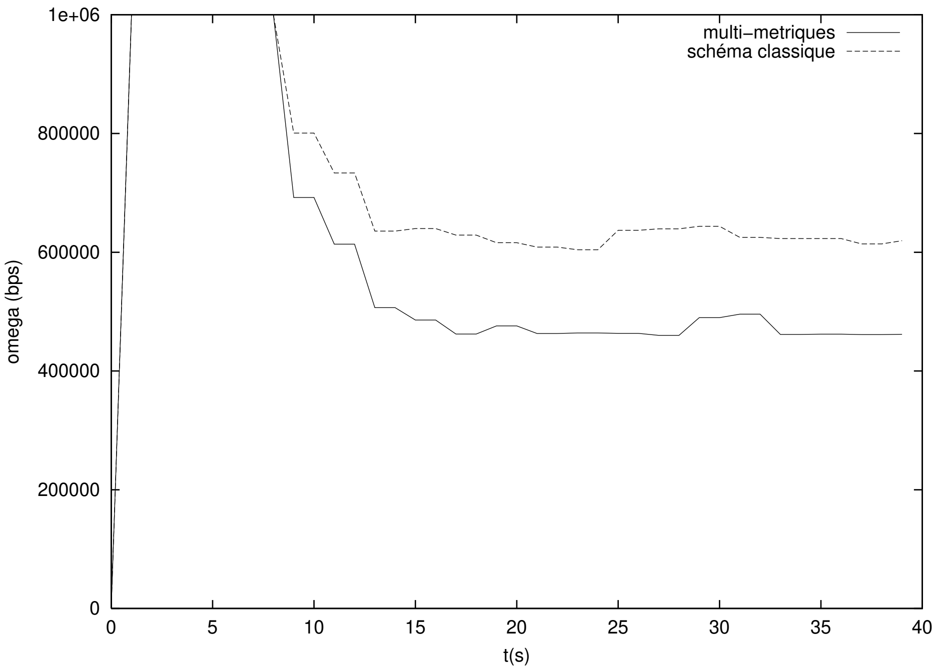


FIG. 4.9 – Gain de l'approche multi-métriques par rapport à l'approche classique.

rien n'empêche que les métriques varient tout en gardant une valeur stable de Q.

Nous proposons dans cette partie une modification dans le calcul de la qualité de la session multicast de façon à réduire, voire éliminer, le problème commenté ci-dessus. Une solution pour ce problème serait l'utilisation en boucle fermée, où les variables de contrôle a, b et c seraient ajustées au cours de la communication. En faisant a, b et c dépendantes aussi de γ, ω et δ, respectivement, il serait possible d'avoir un contrôle plus fin sur la qualité Q. Malgré l'amélioration du contrôle sur Q, nous croyons cependant que l'utilisation d'une boucle fermée rend le système beaucoup plus complexe et donc divergent de nos objectifs.

La solution que nous proposons consiste à utiliser une approche différentielle où on ne considère pas les valeurs absolues des métriques, mais leurs valeurs relatives. En effet, au moment de l'exécution de la procédure de réduction des paquets temporaires, une composante est choisie comme la composante de base. Toutes les autres composantes sont examinées et, si la variation des métriques par rapport à la composante de base est positive, cette nouvelle composante est choisie comme celle à éliminer. Nous allons dans la suite décrire en détail le fonctionnement du système.

Soit toujours $\vec{t} = \{e_1, e_2, \ldots, e_{c_T}\}$, $|\vec{t}| = c_T$, le vecteur temporaire devant être réduit. Considérons également $\mathbf{Z}_Q = \{z_{e_k} | 1 \leq k \leq c_T\}$, autrement dit

$$\mathbf{Z}_Q = \begin{bmatrix} z_{e_1} \\ z_{e_2} \\ \vdots \\ z_{e_{c_T}} \end{bmatrix} = \begin{bmatrix} \gamma_{e_1} & \omega_{e_1} & \delta_{e_1} \\ \gamma_{e_2} & \omega_{e_2} & \delta_{e_2} \\ \vdots & \vdots & \vdots \\ \gamma_{e_{c_T}} & \omega_{e_{c_T}} & \delta_{e_{c_T}} \end{bmatrix}, \tag{4.22}$$

la matrice des vecteurs de qualité. Chaque vecteur z_{e_k} correspond bien entendu à un triple du type $\{\gamma_{e_k}, \omega_{e_k}, \delta_{e_k}\}$. La première tâche qui doit être exécutée est de choisir un vecteur z_{e_k} qui servira comme "entrée de base" de notre système. Le vecteur de base fonctionne comme un point de départ pour la sélection de la composante devant être éliminée du vecteur temporaire. Toutes ces composantes sont comparées au vecteur de base et celle qui donne la plus grande variation individuelle positive des métriques sera choisie comme la composante devant être éliminée.

Notons e_β la composante du vecteur temporaire choisie comme entrée de base. Nous pouvons utiliser n'importe quels critères pour sélectionner l'entrée de base. Dans le cadre de cette thèse, nous avons choisi la composante qui donne la dégradation globale la plus petite. La raison pour laquelle nous avons pris cette décision est que dans la plupart des cas la dégradation globale est choisie comme la métrique prioritaire. Ainsi, il y a une légère mais plus importante probabilité que cette composante soit choisie comme la composante finale devant être éliminée.

Nous utilisons toujours les variables a, b, et c comme paramètres de pondération des métriques. Néanmoins, elles jouent ici un rôle légèrement différent. Elles ne fonctionnent plus comme les poids des valeurs absolues des métriques, mais comme des modérateurs des *variations* des métriques par rapport à elles-mêmes. L'utilité des variables a, b et c pourra être mieux visualisée à partir des équations que nous allons développer dans la suite.

Après avoir calculé l'entrée de base, nous passons directement au calcul des qualités différentielles. Pour chaque k, $1 \leq k \leq c_T$, $k \neq \beta$, on fait[3]

[3]La notation $\frac{dx}{dy}$ que nous utilisons sert uniquement à indiquer que l'équation utilise la différence

$$
\begin{aligned}
\frac{dz_{e_k}}{dz_{e_\beta}} &= a\frac{d\gamma_{e_k}}{d\gamma_{e_\beta}} - b\frac{d\omega_{e_k}}{d\omega_{e_\beta}} - c\frac{d\delta_{e_k}}{d\delta_{e_\beta}} \\
&= a\left(\frac{\gamma_{e_k}-\gamma_{e_\beta}}{\gamma_{e_\beta}}\right) - b\left(\frac{\omega_{e_k}-\omega_{e_\beta}}{\omega_{e_\beta}}\right) - c\left(\frac{\delta_{e_k}-\delta_{e_\beta}}{\delta_{e_\beta}}\right) \\
&= a\frac{\gamma_{e_k}}{\gamma_{e_\beta}} + b\frac{\omega_{e_k}}{\omega_{e_\beta}} + c\frac{\delta_{e_k}}{\delta_{e_\beta}} - (a-b-c).
\end{aligned} \tag{4.23}
$$

Même si on doit définir les valeurs de a, b et c, maintenant elles sont moins liées aux variations de l'état du réseau – la qualité est mesurée à partir du rapport entre les mêmes métriques de composantes différentes.

Une fois les qualités calculées, nous choisissons pour être éliminée la composante correspondante à la plus importante qualité. Le processus recommence jusqu'à ce que le nombre de composantes dans le vecteur temporaire soit réduit au nombre de couches transmises par la source.

Ainsi que pour l'analyse effectuée dans la section 4.4, la comparaison de l'approche différentielle avec d'autres mécanismes exige la définition de certains objectifs. Cette comparaison était directe dans le chapitre 3 car autant l'approche classique que l'Algorithme Direct que nous avons proposé avait le même but, à savoir la minimisation de la dégradation dans les récepteurs.

Les résultats que nous allons montrer dans la suite vise à fournir l'avantage de l'utilisation de l'approche différentielle selon les termes que nous avons utilisés pour décrire l'exemple de la section 4.1. C'est-à-dire, nous comparons les variations de δ, γ et ω par rapport à la composante de base dans nos calculs.

La topologie que nous utilisons est la même que celle de la section 4.4. De façon a simplifier la lecture, nous la répétons ci-dessous (figure 4.10).

Nous nous concentrons maintenant pour nos analyses sur les variations des métriques. Pour cela, nous comparons les résultats obtenus par l'algorithme différentiel avec les résultats correspondants aux composantes de base (qui est le résultat obtenu par l'approche classique). Les courbes que nous allons présenter dans cette partie illustrent le rapport entre le gain obtenus pour γ et ω et la perte liée à δ.

entre les métriques. Elle n'a aucun rapport direct avec la représentation mathématique des équations différenciées.

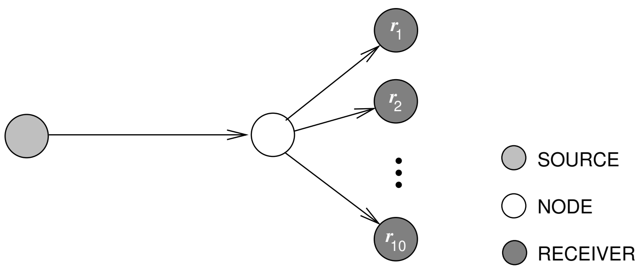


FIG. 4.10 – Topologie de test.

Dans un premier temps, nous considérons que le système est très peu tolérant aux variations de δ (par conséquent, la valeur du paramètre c est élevée). Nous utilisons $a = 0.5$, $b = 0.5$ et $c = 1$. La figure 4.11 illustre le rapport entre les variations de γ et δ pour cette configuration de système. Ensuite, dans la figure 4.12, nous montrons le même rapport pour ω. On peut aisément noter que dans les deux cas, même en utilisant une configuration peu tolérante, il y a des périodes où la variation de γ (et/ou ω) est si importante que l'algorithme décide de changer de composante à éliminer.

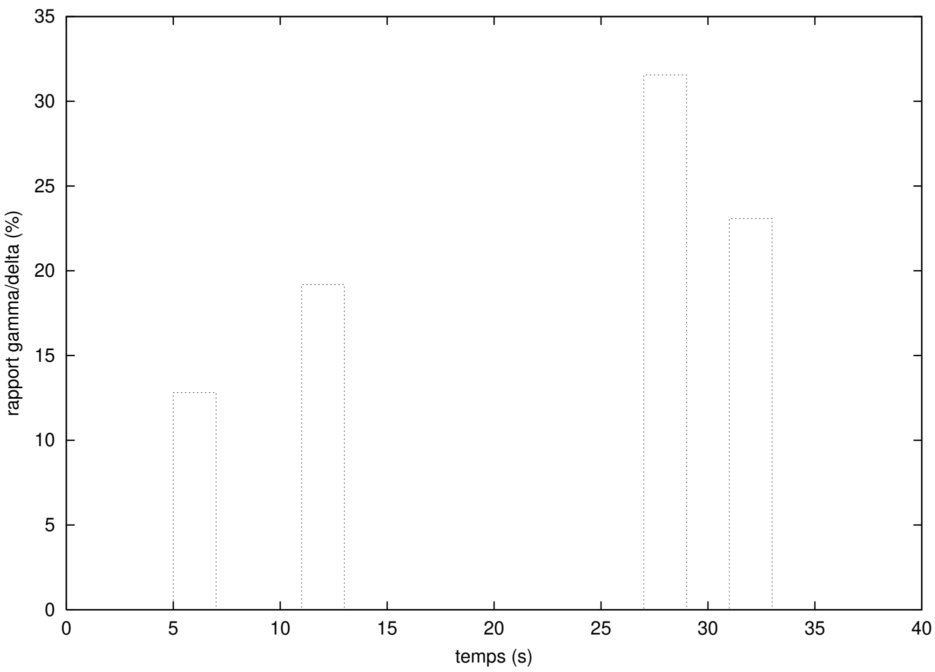


FIG. 4.11 – Rapport entre les variations de γ et δ pour le scénario peu tolérant.

Nous allons maintenant vérifier les résultats pour une configuration plus tolérante aux variations de δ. Nous utilisons pour cela $a = 0.8$, $b = 0.8$ et $c = 0.5$. La figure 4.13 illustre

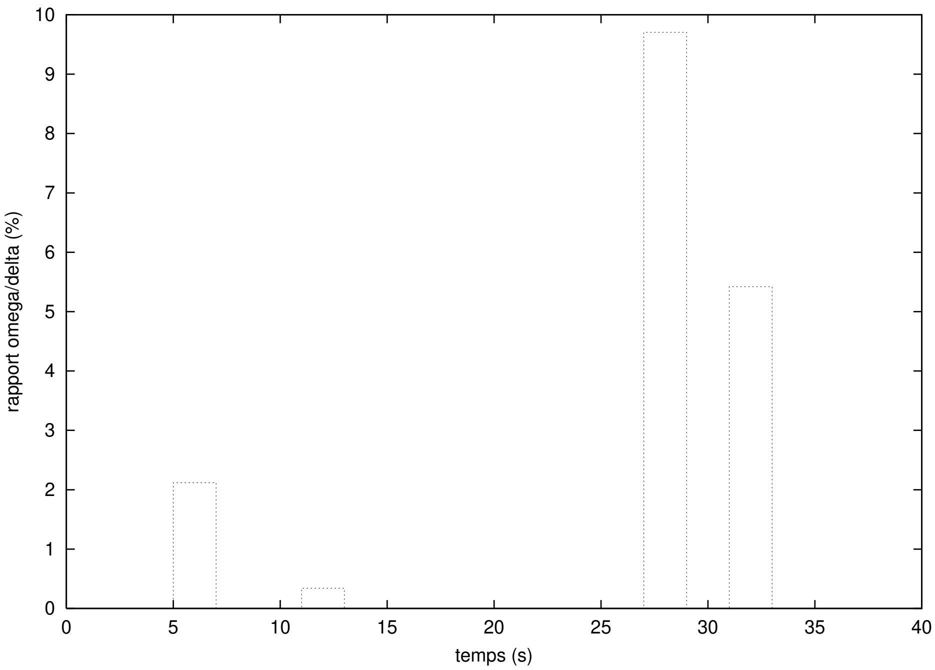


FIG. 4.12 – Rapport entre les variations de ω et δ pour le scénario peu tolérant.

le rapport entre les variations de γ et de δ. Le même est fait pour ω et δ (figure 4.14). Comme nous pouvions espérer, les rapports se montrent encore plus importants.

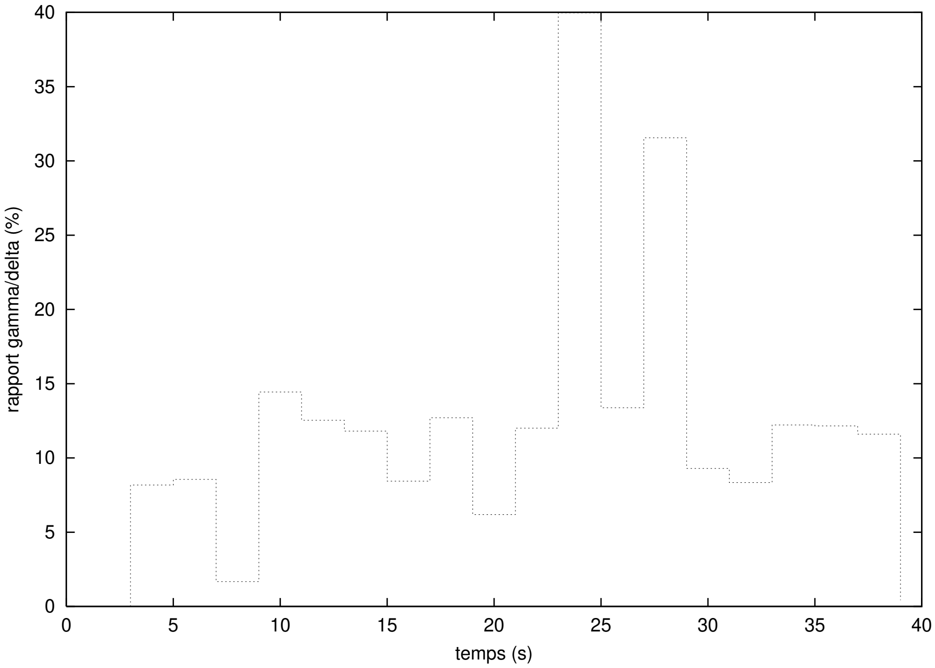


FIG. 4.13 – Rapport entre les variations de γ et δ pour le scénario plus tolérant.

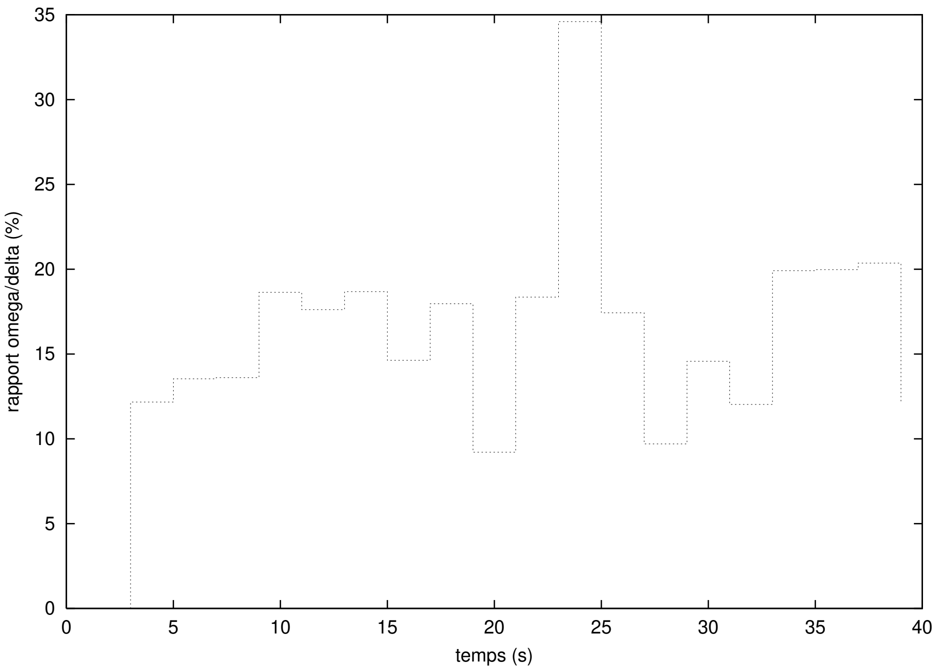


FIG. 4.14 – Rapport entre les variations de ω et δ pour le scénario plus tolérant.

Une version plus simples du système pourrait limiter l'action des paramètres a, b et c. En effet, ils pourraient fonctionner uniquement comme des facteurs limitant la variation des variables. Par exemple, un application quelconque pourrait supporter des ajustements de δ, mais pas au-delà d'un certain seuil (qui serait donné, dans le cas de δ, par c).

4.6 Discussion

Le premier point que nous soulignons concerne l'utilisation de la métrique simple pour le calcul des débits des couches. Intuitivement, la dégradation de la qualité joue un rôle important dans les systèmes que nous étudions. Néanmoins, la façon dont le calcul est fait peut suggérer un certain nombre de discussions. Du point de vue d'un récepteur seul, la simple différence entre le débit demandé et celui calculé par la source représente une quantité valable puisque le récepteur connaît l'échelle de sa capacité. Lorsque l'on rassemble les valeurs venues de différents clients afin de réaliser un calcul, la métrique simple diverge d'une base commune. Il faut donc bien séparer le concept "dégradation de qualité" du concept "dégradation perceptuelle". Le terme que nous utilisons dans cette thèse (dégradation de qualité) est lié uniquement au fait que, dans le cadre de notre étude, la 'qualité' est mesurée comme la différence entre le débit demandé et celui calculé par la source. Considérons, à titre d'exemple, le cas d'une vidéo. Une dégradation de qualité δ_i pour un récepteur r_i demandant b_i peut être moins gênante pour l'utilisateur qu'une dégradation δ_j pour un récepteur r_j demandant b_j, même si $\delta_i = \delta_j$.[4]

Autant dans le chapitre antérieur que dans ce chapitre (et, comme nous allons le voir, dans le chapitre suivant), nous utilisons le concept de dégradation de qualité. Cette quantité avait déjà été utilisée par d'autres mécanismes proposés dans la littérature. Le mécanisme basé sur le débit conforme, par exemple, utilise un calcul explicite sur la dégradation de qualité. Comme première approche, et de façon à montrer l'amélioration de nos propositions, la métrique simple utilisant la dégradation de qualité a été utilisée.

À notre avis, la dégradation de qualité est cependant limitée dans son fonctionnement comme base de calcul des débits des couches. En premier lieu, elle ne montre que le point de vue des récepteurs. L'exemple que nous avons montré dans la section 4.1 décrit un cas où une forte variation de la bande passante allouée dans le réseau est observée pour une dégradation de qualité pratiquement stable. Ce type de scénario nous a suggéré l'utilisation de la bande passante allouée comme un des paramètres de calcul des couches. En second lieu, la variation perceptuelle n'est pas directement liée à la dégradation de qualité. Comme nous l' avons déjà évoqué dans cette section, la variation perçue par l'utilisateur est fonction d'autres paramètres, et δ peut être beaucoup plus gênant pour un utilisateur x que pour un utilisateur y. Ainsi, nous avons proposé l'utilisation de la densité d'utilisateurs satisfaits, qui utilise le concept de dégradation relative à la place de la dégradation absolue.

Nous avons vu dans ce chapitre que l'utilisation d'un système multi-métriques peut être intéressant pour une bonne gamme d'applications. Il est clair que la configuration que nous avons construite peut être remplacée par une autre configuration utilisant d'autres métriques. Notre approche vise surtout à souligner les limitations de l'approche par la métrique simple et montrer qu'un système à multi-métriques est faisable et peut aboutir à une amélioration de la qualité de la session multicast à différents points de vue.

[4]Suivant les valeurs b_i et b_j, la gêne provoquée peut être même plus importante pour le récepteur r_j que pour le récepteurs r_i dans le cas où $\delta_i > \delta_j$.

Chapitre 5

Le contrôle aux bouts

Nous avons étudié dans les chapitres précédents des mécanismes capables de contrôler les débits des couches dans une session multicast avec une source et plusieurs récepteurs. Le but est de calculer ces débits de façon à réduire la dégradation globale de la session multicast. Un des composants principaux du système proposé est l'algorithme de combinaison de paquets d'acquittement qui évite l'implosion de paquets de contrôle à la source.

Il est évident que le système exige la présence d'un certain nombre d'éléments actifs dans les nœuds du réseau. Ces éléments sont responsables de l'exécution des algorithmes à l'arrivée des paquets de contrôle. Il n'est pourtant pas sûr que le réseau soit capable d'héberger ces algorithmes ; si l'on considère par exemple le réseau MBone d'aujourd'hui, les algorithmes que l'on a proposés ne peuvent pas être aussitôt implantés. D'autres solutions doivent donc être fournies.

Ce chapitre est réservé à une proposition qui tient compte de l'handicap des réseaux passifs. Il s'agit d'un mécanisme devant être introduit uniquement dans les éléments terminaux, à savoir la source et les récepteurs. Au travers de l'échange de messages entre ces entités, le nombre de paquets d'acquittement envoyés à la source est contrôlé. De plus, l'analyse de ces messages permet au système de conserver un niveau minimum d'équité de la vidéo face à l'hétérogénéité entre récepteurs. Le système est basé sur des temporisateurs qui contrôlent l'envoi des messages ainsi que l'échange d'informations entre les entités [72, 73, 74, 75].

À la fin de la période du temporisateur, un récepteur envoie son paquet d'acquittement aussi bien à la source qu'aux autres récepteurs appartenant au groupe multicast. Un récepteur qui reçoit un paquet d'acquittement d'un autre récepteur quelconque avant que son temporisateur soit expiré vérifie si son acquittement peut être supprimé. Soit b^{r_i} le débit demandé par le récepteur r_i et b^{r_j} le débit demandé par r_j. Si avant que son temporisateur ne soit épuisé le récepteur r_i reçoit l'acquittement de r_j et si $\frac{b^{r_j}}{b^{r_i}}$ est plus petit qu'un certain seuil donné, le paquet d'acquittement de r_i sera supprimé. Ainsi, tout le contrôle se passe aux bouts et aucune interférence ne doit être faite par les nœuds intermédiaires.

Nous proposons deux approches pour le système. Dans la première, les récepteurs se comportent d'une façon entièrement passive et, après vérification que leurs paquets d'acquittement peuvent être supprimés, ils ne réalisent plus aucune opération. Dans la deuxième approche, nous intégrons un système d'échange de messages entre les récepteurs qui apporte à la source l'information de combien de récepteurs chaque paquet reçu représente. Ces deux méthodes sont décrites en détail dans les sections suivantes, après un portrait du système de temporisateurs.

5.1 Le système de temporisateurs

L'objectif des temporisateurs dans le système que nous proposons est de distribuer dans le temps l'action des récepteurs pour éviter que tous répondent au même instant à la source. Le plus important est aussi de permettre à certains paquets d'acquittement d'atteindre d'autres récepteurs avant que ceux-ci répondent à la source. De cette manière, ces récepteur peuvent supprimer leurs acquittements dès la première phase du protocole.

Avant de détailler le fonctionnement du protocole, nous allons décrire le système de temporisateurs utilisé. À titre d'information, nous allons décrire deux lois probabilistes pour la génération des temporisateurs, des distributions uniforme et exponentielle [72]. Néanmoins, nous avons utilisé dans nos systèmes la loi uniforme pour des raisons que nous allons exprimer dans la section suivante.

Soit z_{r_i} le temporisateur distribué dans l'intervalle $[0, T]$ par le récepteur r_i. La distribution du type uniforme est gouvernée par la densité :

$$f_{Z_{r_i}}(z_{r_i}) = \begin{cases} \frac{1}{T}, & 0 \leq z_{r_i} \leq T, \\ 0, & \text{sinon.} \end{cases} \tag{5.1}$$

Si l'on utilise par contre la distribution exponentielle, la densité est représentée par l'équation :

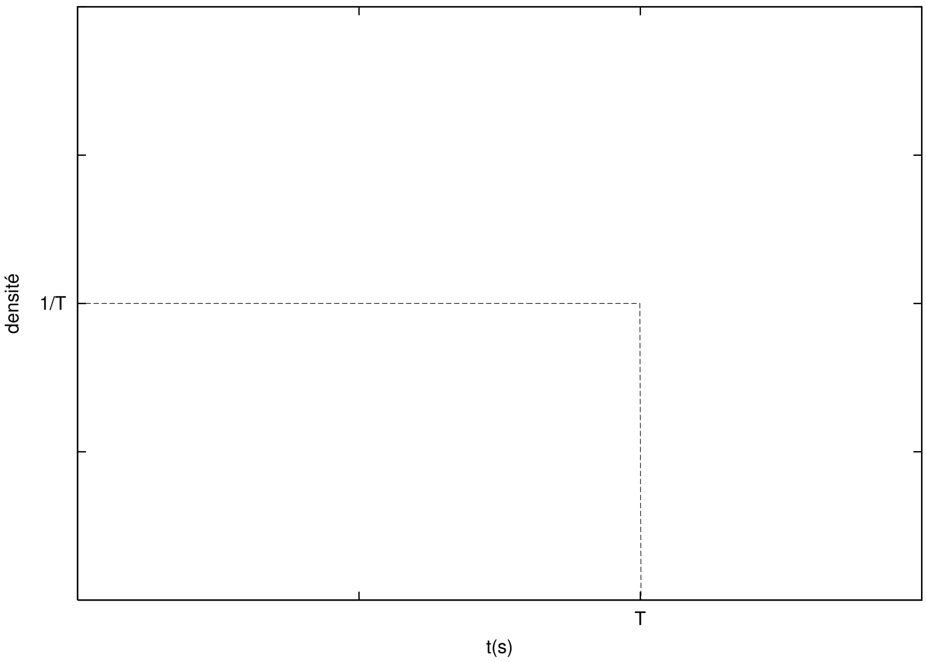
$$f_{Z_{r_i}}(z_{r_i}) = \begin{cases} \frac{1}{exp^{\lambda}-1}\frac{\lambda}{T}exp^{(\frac{\lambda}{T})z_{r_i}}, & 0 \leq z_{r_i} \leq T, \\ 0, & \text{sinon.} \end{cases} \tag{5.2}$$

Le paramètre λ de la distribution exponentielle fonctionne comme un “concentrateur” dans le sens où le poids de la distribution se dirige vers T avec la croissance de λ. La figure 5.1 montre les distributions uniforme et exponentielle avec différentes valeurs de λ.

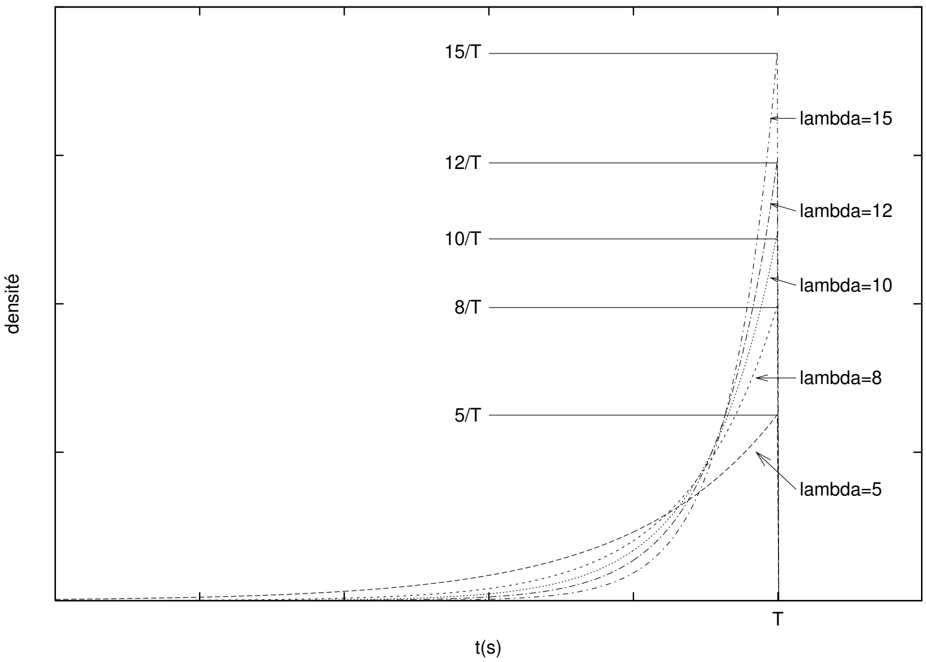
Le système présenté dans [72] montre que, malgré une complexité plus importante, l'utilisation de la distribution exponentielle est plus avantageuse que la distribution uniforme ; en effet, cette première donne aux paquets d'acquittement déjà en transit plus de possibilité d'arriver aux autres récepteurs avant que ceux-ci aient transmis leurs acquittements. Or, dans le cadre de cette thèse, l'avantage de l'utilisation de la loi exponentielle par rapport à la loi uniforme n'est pas significatif. La principale raison en est que la suppression des paquets d'acquittement n'est pas dépendante exclusivement de l'arrivée ou non des acquittements des autres récepteurs, mais aussi du débit transporté par ces acquittements.

5.2 Description générale du système

Pour des raisons de simplicité, nous nous sommes servis du système à une seule métrique. L'adaptation d'autres métriques est simple et n'exigerait que quelques modifications. A la réception d'un paquet de prospection envoyé par la source, le récepteur configure un temporisateur suivant une loi donnée (dans notre cas, la loi uniforme). À



(a) Distribution uniforme.



(b) Distribution exponentielle.

FIG. 5.1 – Distributions uniforme et exponentielle pour les temporisateurs.

la fin de cette période, le récepteur envoie à la source et à tous les autres récepteurs un paquet contenant sa capacité de réception.

Dans la première approche que nous proposons ici, si pendant la période du temporisateur un paquet d'acquittement provenant d'un autre destinataire arrive, le récepteur supprime sa réponse si le demandes des deux récepteurs sont proches.

Dans la deuxième approche, dès l'arrivée d'un paquet d'acquittement d'un autre récepteur, une politique "d'hébergement" (cf. 5.4) est mise en place pour déterminer si les deux récepteur seront rassemblés. En résumé, lorsqu'un récepteur est en attente et un paquet d'acquittement envoyé par un autre récepteur arrive, il doit procéder comme suit :

- vérifier si le paquet arrivé peut héberger le sien ;
  - si non, rien n'est fait et le récepteur continue en attente ;
  - si oui, envoyer à l'autre récepteur une demande d'hébergement.

5.3 L'approche sans coopération

Soit t_{r_i} l'instant d'arrivée du paquet de prospection au récepteur r_i. Nous considérons pour des questions de simplicité que tous les paquets de prospection arrivent en même temps dans tous les récepteurs. Nous appelons ce temps t_0 et :

$$t_0 = t_{r_1} = t_{r_2} = \ldots = t_{r_{N-1}} = t_{r_N},$$

où N est le nombre de récepteurs appartenant à la session multicast.

Soit τ_{r_i} la valeur du temporisateur correspondant également au récepteur r_i et t_{r_i,r_j} le temps de transfert d'un message entre les récepteurs r_i et r_j. Nous considérons aussi que $t_{r_i,r_j} = t_{r_j,r_i}$, c'est-à-dire, que les délais entre deux récepteurs sont symétriques, quel que soit le sens.

Dans la situation normale, un récepteur r_i configure un temporisateur dès qu'il reçoit le paquet de prospection. Lorsque ce temporisateur τ_{r_i} expire, le récepteur diffuse dans le groupe (vers la source et vers les autres récepteurs) son paquet d'acquittement. Nous allons représenter le paquet d'acquittement d'un récepteur r_i naturellement comme p_{r_i}.

Il est aisé de voir que le récepteur r_j n'exécute les procédures que si :

$$\tau_{r_i} + t_{r_i,r_j} < \tau_{r_j}. \tag{5.3}$$

Bien évidement, l'équation ci-dessus considère que le paquet de prospection arrive en même temps dans tous les récepteurs. Le cas échéant, il faudrait introduire les termes t_{r_i} et t_{r_j} dans les deux côtés de l'équation, respectivement.

Avant d'analyser et simuler le comportement du système, nous allons notifier une petite modification introduite dans le paquet de prospection envoyé par la source. Le paquet de prospection contient désormais un champ qui stocke l'indice d'accumulation toléré. Cet indice, représenté par α, sert à contrôler le seuil qui permet aux récepteurs de s'agréger (cf. définition 1). Par conséquent, la quantité de paquets d'acquittement est également contrôlée. Comme nous allons le voir dans ce chapitre, cette valeur peut être variable et maintenue sous surveillance exclusive de la source, selon la quantité de paquets d'acquittement reçue.

Définition 1 *Un récepteur r_i, demandant le débit b^{r_i} et en attente de la fin de la période de son temporisateur, supprime son paquet d'acquittement si, à l'arrivée d'un paquet d'acquittement provenant du récepteur r_j, l'inégalité $\frac{b^{r_j}}{b^{r_i}} \geq \alpha$ est respectée.*

5.3.1 Simulations

Nous allons présenter maintenant plusieurs simulations du système décrit ci-dessus. Nous varions les valeurs des paramètres tels que : la moyenne des périodes des temporisateurs (τ), le seuil de suppression (α) et le délai de transmission entre les récepteurs (c).

Le scénario que nous allons utiliser est composée d'une source et N récepteurs. Dans un premier temps, nous allons faire $N = 10$ et ensuite $N = 50$. La figure 5.2 illustre le scénario pour la première partie des tests.

Toutes les configurations considèrent le même délai $c = 10\ ms$. La figure 5.3 montre la dégradation globale de la session multicast pour le scénario de la figure 5.2. La courbe

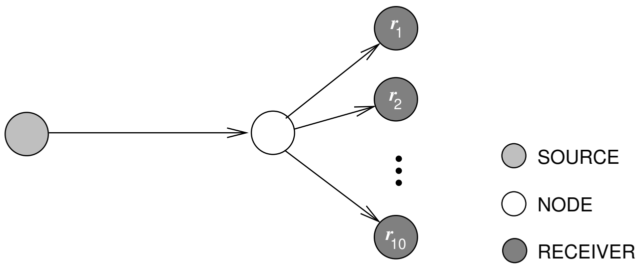


FIG. 5.2 – Scénario avec 10 nœuds.

représente les résultats obtenus avec $\alpha = 0.5$ et $T = 2c, 8c$ et $15c$. Ensuite, dans la figure 5.4, nous illustrons le nombre de paquets d'acquittement qui atteignent la source.

Dans les figures 5.5 et 5.7 nous représentons aussi la dégradation de la qualité. Nous utilisons cependant les seuils de suppression $\alpha = 0.7$ et $\alpha = 0.9$, respectivement.

De toutes ces courbes, nous pouvons extraire les conclusions suivantes. Pour une plus grande tolérance de suppression (α plus petit), le nombre d'acquittements supprimés est plus important. Cependant, puisque plus de récepteurs avec des débits plus éloignés suppriment leurs réponses, la source est nourrie avec moins d'information. Le résultat est naturellement une croissance de la dégradation globale de la session multicast.

Du côté de la valeur du temporisateur, plus grand est le rapport $\frac{T}{c}$, moins importante est la dégradation globale du système. Il est également important de noter que, comme on peut observer dans les figures présentées ci-dessus, la différence entre les courbes pour $T = 2c, T = 8c$ et $T = 15c$ est liée à la valeur du seuil de suppression.

Nous allons maintenant vérifier le comportement du système avec un nombre plus élevé de récepteurs. Nous utilisons le même type de scénario que dans la figure 5.2, mais avec $N = 50$ récepteurs. Les résultats sont illustrés dans les figures 5.9 et 5.10. Tout d'abord, le premier graphique montre l'écart que l'on peut obtenir si des valeurs très petites de α et T sont choisies. Il montre également qu'il existe une tendance du système à se stabiliser après certaines valeurs de (α,T).

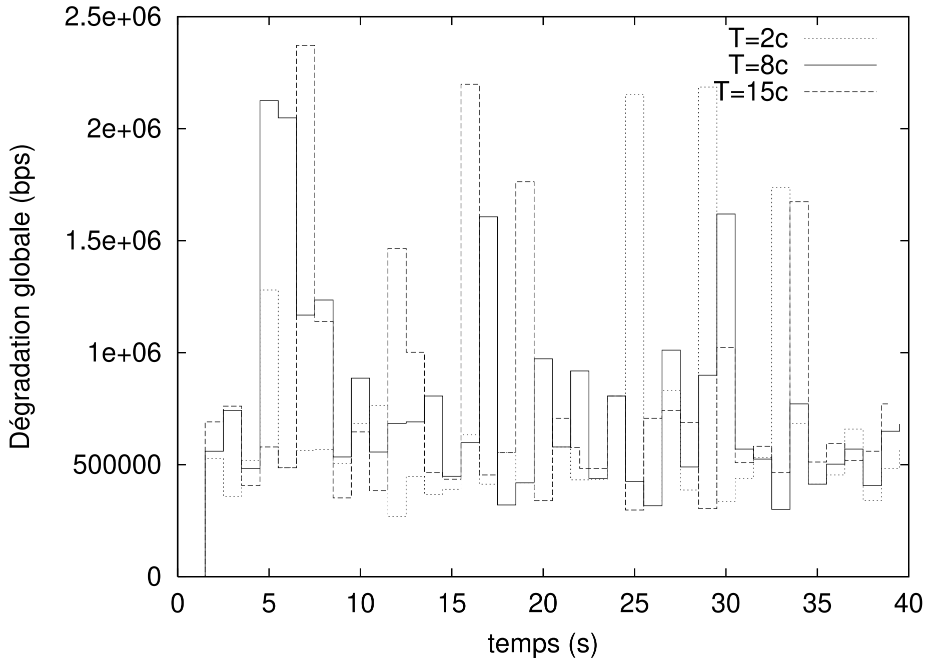


FIG. 5.3 – Dégradation globale de la session multicast avec $\alpha = 0.5$ et $T = 2c, T = 8c, T = 15c$.

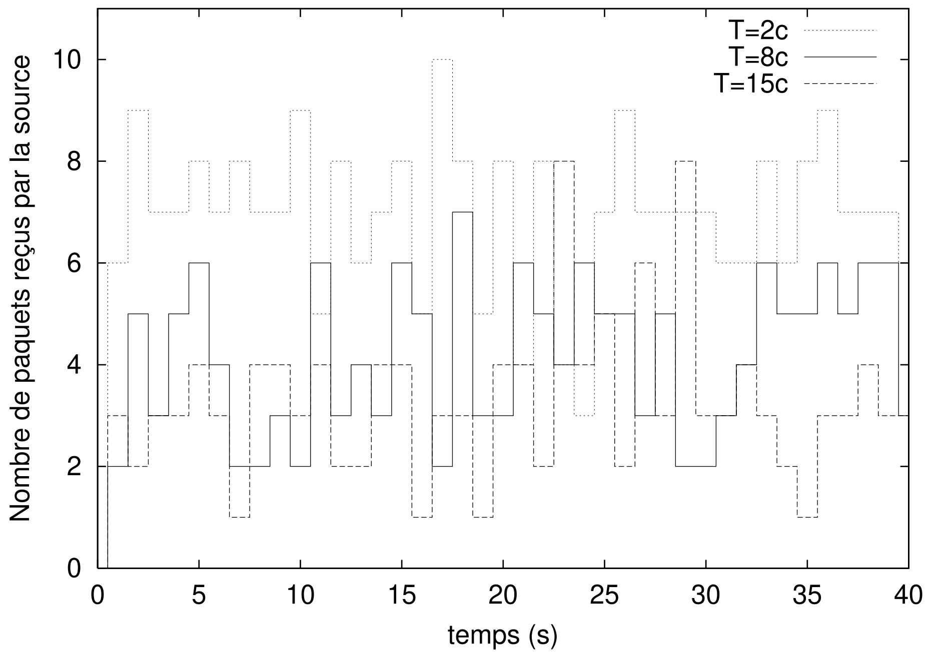


FIG. 5.4 – Nombre total de paquets d'acquittement qui arrivent à la source, avec $\alpha = 0.5$ et $T = 2c, T = 8c, T = 15c$.

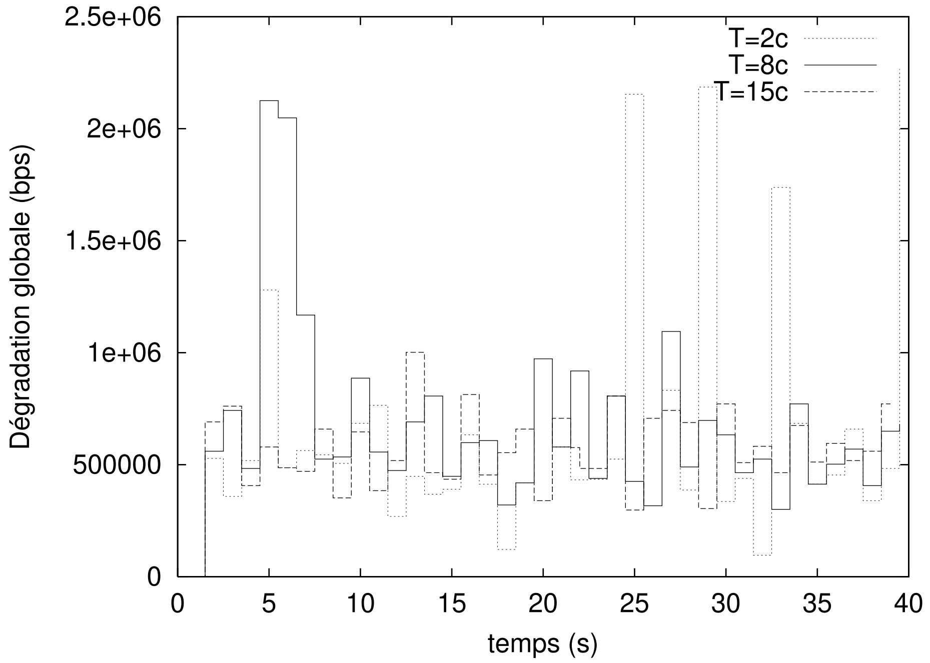


FIG. 5.5 – Dégradation globale de la session multicast avec $\alpha = 0.7$ et $T = 2c, T = 8c, T = 15c$.

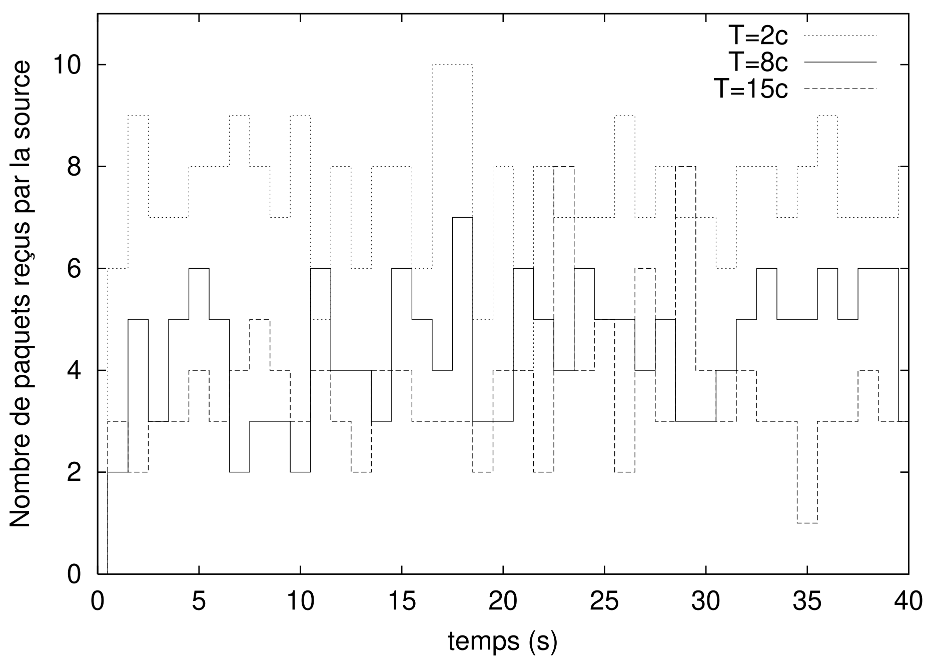


FIG. 5.6 – Nombre total de paquets d'acquittement qui arrivent à la source, avec $\alpha = 0.7$ et $T = 2c, T = 8c, T = 15c$.

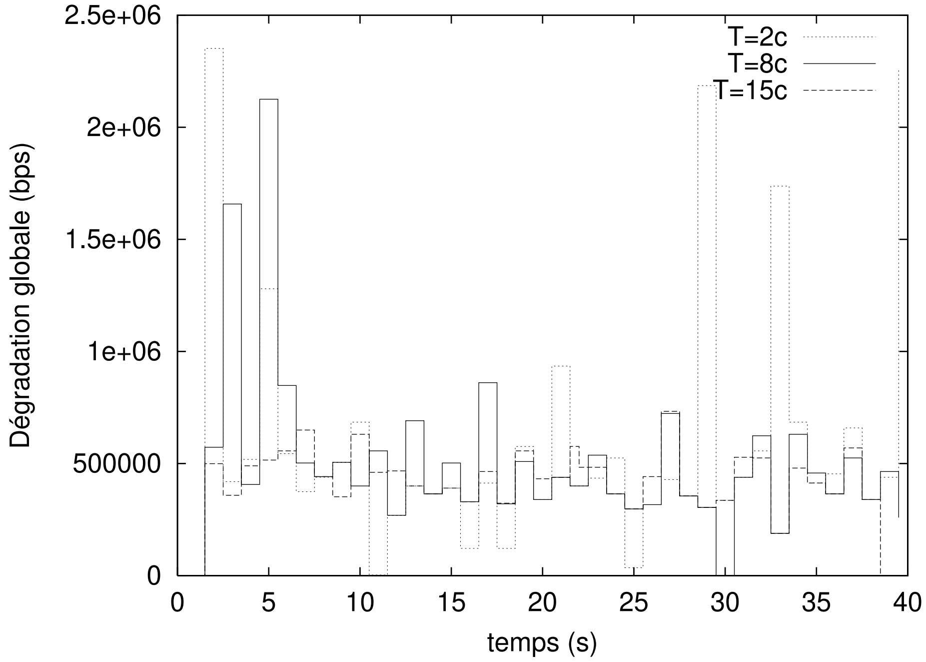


FIG. 5.7 – Dégradation globale de la session multicast avec $\alpha = 0.9$ et $T = 2c, T = 8c, T = 15c$.

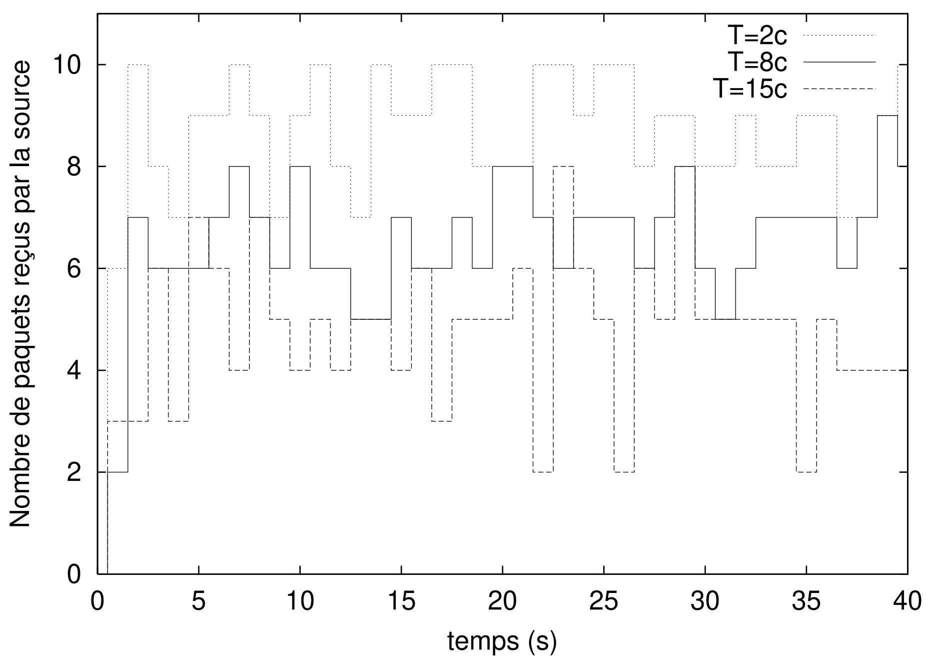


FIG. 5.8 – Nombre total de paquets d'acquittement qui arrivent à la source, avec $\alpha = 0.9$ et $T = 2c, T = 8c, T = 15c$.

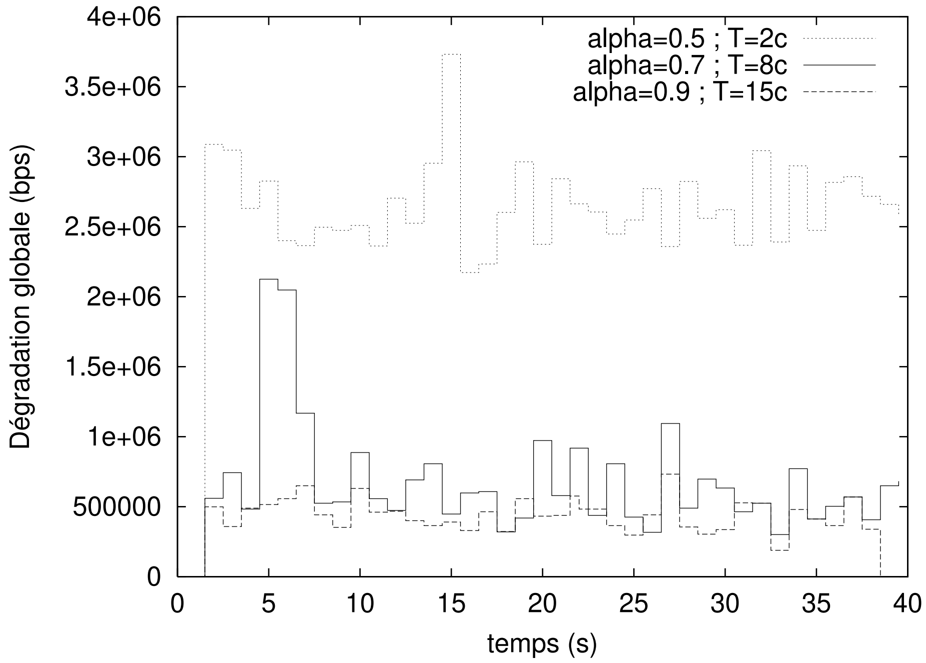


FIG. 5.9 – Dégradation globale de la session multicast avec ($\alpha = 0.5; T = 2c$), ($\alpha = 0.7; T = 8c$) et ($\alpha = 0.9; T = 15c$).

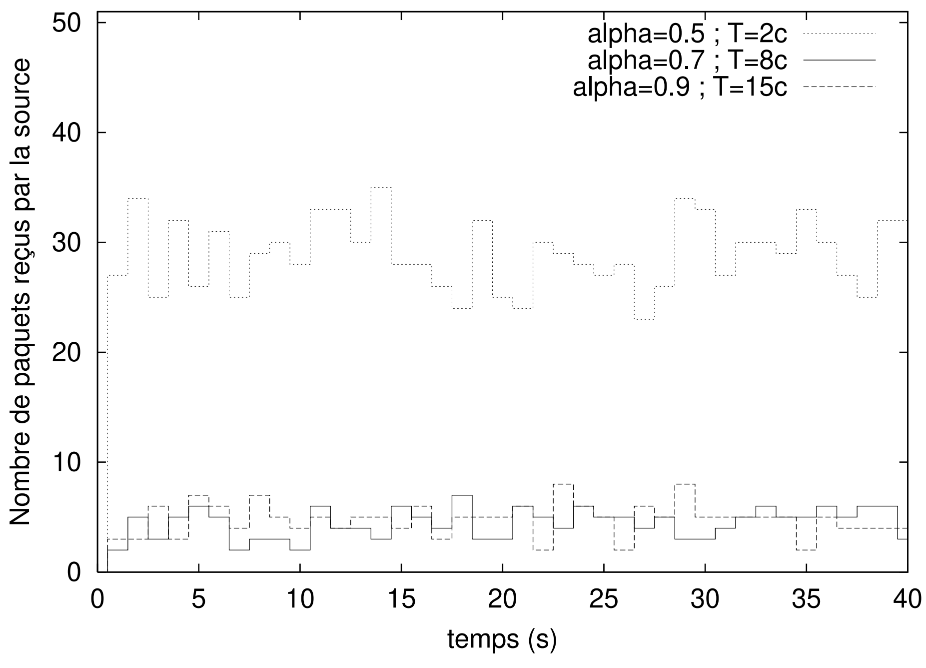


FIG. 5.10 – Nombre total de paquets d'acquittement qui arrivent à la source, avec ($\alpha = 0.5; T = 2c$), ($\alpha = 0.7; T = 8c$) et ($\alpha = 0.9; T = 15c$).

5.3.2 Discussion

Dans cette section, nous avons proposé une approche simple de suppression de paquets d'acquittement dans un réseau sans traitement de données dans les nœuds intermédiaires. Malgré le comportement passif des récepteurs, le système montre déjà qu'il apporte de bons résultats en termes de nombre de messages reçus par la source et de la dégradation globale de la session multicast.

Il est cependant important de noter qu'une variable significative n'est plus tenue en compte : le nombre de récepteurs souhaitant avoir le débit marqué dans le paquet d'acquittement. Cette intégration est l'objet de la section suivante. Nous proposerons un protocole de coopération entre les récepteurs qui vise à augmenter l'exactitude du calcul des débits des couches. Ceci est possible car les récepteurs qui suppriment leurs messages prennent une position active dans le système ; ils communiquent avec le récepteur dont l'acquittement les a faits ne pas répondre à la source.

5.4 Le protocole de coopération

Le protocole proposé dans cette partie est une extension du système de suppression d'acquittements présenté dans la section précédente. Cette extension apporte aux récepteurs, contrairement au système antécédent, une position active dans le processus de création des paquets d'acquittement. Le protocole proposé a pour but de permettre l'échange de messages entre récepteurs appartenant à la même session multicast. À partir des messages échangés, les récepteurs qui répondent à la source sont donc habilités à lui informer la quantité de récepteurs désirant le débit demandé.

L'arrivée d'un paquet p_{r_i} (cf. section 5.3) à un récepteur r_j déclenche une série de procédures de vérification et d'échange de messages entre les récepteurs. Le protocole de coopération que nous proposons ici est divisé en deux phases distinctes. Durant la première, les récepteurs sont dans un état d'observation, où ils attendent les paquets d'acquittement des autres récepteurs afin d'identifier qui pourrait éventuellement être son hébergeur. À la fin de cette période, de longueur égale à la durée du temporisateur, les récepteurs savent s'ils demanderont l'hébergement ou non.

La seconde phase est réservée à l'échange de messages entre les récepteurs. À ce moment, si un récepteur a reçu pendant le cycle du temporisateur un message envoyé par un autre récepteur qui "l'intéresse", un contact s'établira entre les deux. Essentiellement, le récepteur intéressé demande à l'autre d'inclure dans son paquet d'acquittement son désir de recevoir le même débit.

Par contre, si le récepteur n'a reçu aucun message, il crée simplement un paquet d'acquittement comme si rien ne s'était passé et l'envoie à la source et aux autres récepteurs du groupe. Ceci suscitera, à son tour, les mêmes procédures dans les autres récepteurs.

Ainsi comme dans le système de la section précédente, on dit qu'un récepteur r_i s'intéresse au récepteur r_j si le débit requis par le premier (b^{r_i}) est compris entre αb^{r_j} et b^{r_j}, avec $0 < \alpha \leq 1$. Nous introduisons ici le concept *d'hébergement*, qui associe r_i à r_j. On dit que r_j devient *l'hébergeur* de r_i s'il accepte d'insérer dans son paquet d'acquittement un indicatif de la présence de r_i.

Supposons qu'au début de la seconde phase le récepteur r_i s'intéresse à r_j. r_i envoie donc à r_j un message du type `LodgeMe` contenant son identificateur ainsi que le débit b^{r_i}. Le récepteur r_i entre ensuite dans un état d'attente jusqu'à ce que le récepteur r_j lui réponde.

Le récepteur hébergeur en potentiel r_j (qui ne demande pas à être hébergé) envoie à r_i (et éventuellement aux autres récepteurs qui lui avaient fait la demande) un message du type `Stable`. Ce message assure les récepteurs que leurs demandes ont été prises en compte. Dès l'arrivée du message `Stable`, r_i vérifie s'il y a des récepteurs qui lui avaient demandé de les héberger. À ce moment-là, la seule information que r_i possède est la dégradation qu'il subira lors de l'hébergement. Le récepteur r_i décline donc les sollicitations des autres récepteurs et leur envoie un message du type α-`Stable` contenant principalement l'identificateur de l'hébergeur effectif (r_j) et la dégradation composée. La dégradation composée est donnée par le rapport entre le débit demandé par le récepteur hébergé et le débit réclamé par l'hébergeur effectif. Ainsi, si le récepteur r_i demande l'hébergement à r_j qui, par contre, sera hébergé par r_k, la dégradation composée $\delta^c_{r_i}$ pour le récepteur r_i sera :

$$\delta_{r_i}^c = \frac{b^{r_j}}{b^{r_i}} \frac{b^{r_k}}{b^{r_j}} = \frac{b^{r_k}}{b^{r_i}}.$$

Si le récepteur r_i accepte l'hébergement de r_k, on dit qu'il sera sous-hébergé. Après avoir accepté le sous-hébergement, le récepteur r_i envoie aux récepteurs qui lui avaient demandé l'hébergement un autre message α-`Stable` avec la dégradation composée $\delta_{r_x}^c$, où r_x est un récepteur ayant demandé l'hébergement à r_i. Ce cycle se répète jusqu'à ce que les récepteurs n'acceptent plus le sous-hébergement puisque la dégradation dépasse le seuil donné par α.

Si, par contre, r_i reçoit de r_j un message α-`Stable` et

$$\delta_{r_i}^c \frac{b^{r_i}}{b^{r_x}} > \alpha,$$

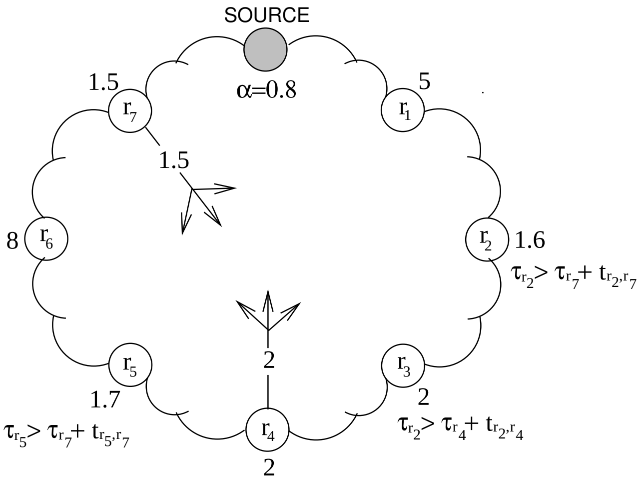
la dégradation composée pour le récepteur r_x dépasse le seuil donné par α. À ce moment-là, r_i accepte le sous-hébergement de r_k et envoie à r_x un message du type `Reject`.

Le récepteur r_x, ayant reçu un message `Reject`, sait qu'il ne pourra guère être hébergé. Ainsi, il devient un récepteur stable et envoie un message `Stable` à tous les récepteurs qui lui avaient demandé l'hébergement. Le cycle est donc clos jusqu'à la prochaine phase d'adaptation, déclenchée lors de l'arrivée d'un nouveau paquet de prospection.

Nous montrons dans la figure 5.11 un exemple du fonctionnement des deux étapes du protocole.

La concurrence de messages envoyés par différents hébergeurs potentiels est un point important à discuter. Supposons que le récepteur r_i reçoit pendant sa période d'attente deux messages d'acquittement envoyés par les récepteurs r_j et r_k. Naturellement, r_i choisirait comme hébergeur celui qui donne la plus petite dégradation : r_j, si $\frac{b^{r_j}}{b^{r_i}} < \frac{b^{r_k}}{b^{r_i}}$; ou r_k, si $\frac{b^{r_j}}{b^{r_i}} > \frac{b^{r_k}}{b^{r_i}}$.

Il n'est pas difficile de voir qu'il existe une forte probabilité que le récepteur r_j demande l'hébergement à r_k, si $b^{r_j} > b^{r_k}$, ou r_k à r_j, si $b^{r_j} < b^{r_k}$). Dans un des ces deux cas, le récepteur r_i "by-pass" son hébergeur directe (celui qui donne la plus petite dégradation) et demande directement à l'autre récepteur son hébergement. Cette procédure est illustrée



(a) Configuration des capacités. (b) Demande d'hébergement.

FIG. 5.11 – Illustration du mécanisme de demande d'hébergement.

dans la figure 5.12, où $\frac{b^{r_k}}{b^{r_i}} < \frac{b^{r_k}}{b^{r_j}} > \alpha$.

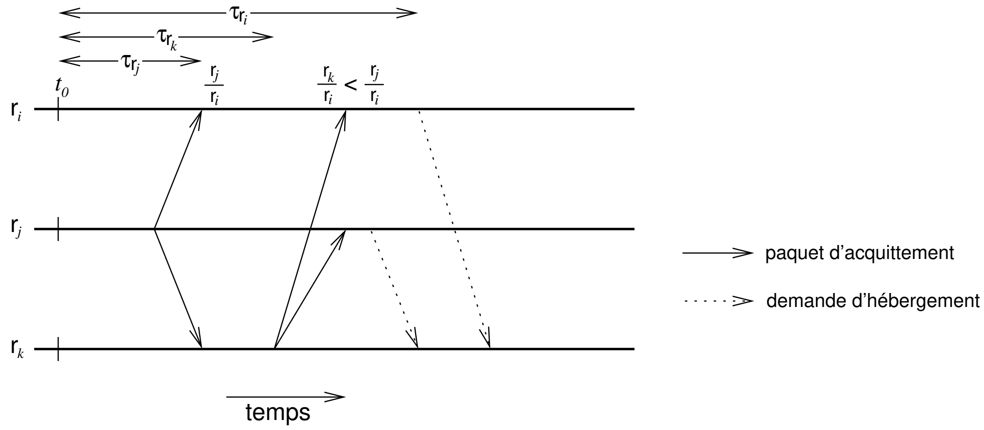


FIG. 5.12 – Le récepteur r_i "by-pass" le récepteur r_j.

Un dernier commentaire concerne la signalisation générée par le protocole. D'un côté, lorsqu'aucun des récepteurs n'a demandé l'hébergement, une signalisation égale à zéro est provoquée. Le problème est que la signalisation nulle correspond à tous les récepteurs envoyant un acquittement à la source. Cela ne règle pas un des nos problèmes qui est l'implosion de paquets d'acquittement à la source. Le système doit donc ajuster dynamiquement les paramètres α et/ou T pour surmonter ce problème. D'un autre côte, chaque récepteur demanderait à un autre récepteur l'hébergement, ce qui formerait une chaîne de requêtes successives. Dans ce cas, le nombre de paquets de contrôle généré serait équivalent au nombre de récepteurs. La différence est que la manipulation de ces paquets serait distribuée et pour à peine une courte période de temps, c'est-à-dire, jusqu'à ce que tous les récepteurs aient trouvé leurs hébergeurs définitifs.

Avant d'entamer l'analyse du protocole de coopération, nous montrons deux schémas simplifiés de son fonctionnement. La figure 5.13(a) illustre un schéma d'états pour un récepteur demandant l'hébergement à un second. Le récepteur est originellement dans un état inactif et n'appartient pas au groupe de récepteurs d'un hébergeur déjà existant. Ce cas est représenté dans le schéma de la figure 5.13(b), où le comportement adopté par un récepteur déjà hébergé et désirant changer d'hébergeur est illustré.

Dans la section suivante nous allons simuler le fonctionnement du protocole de coopération proposé dans ce chapitre. Nous allons utiliser, comme pour le système sans coopération de la section 5.3, le réseau de tests de la figure 5.2.

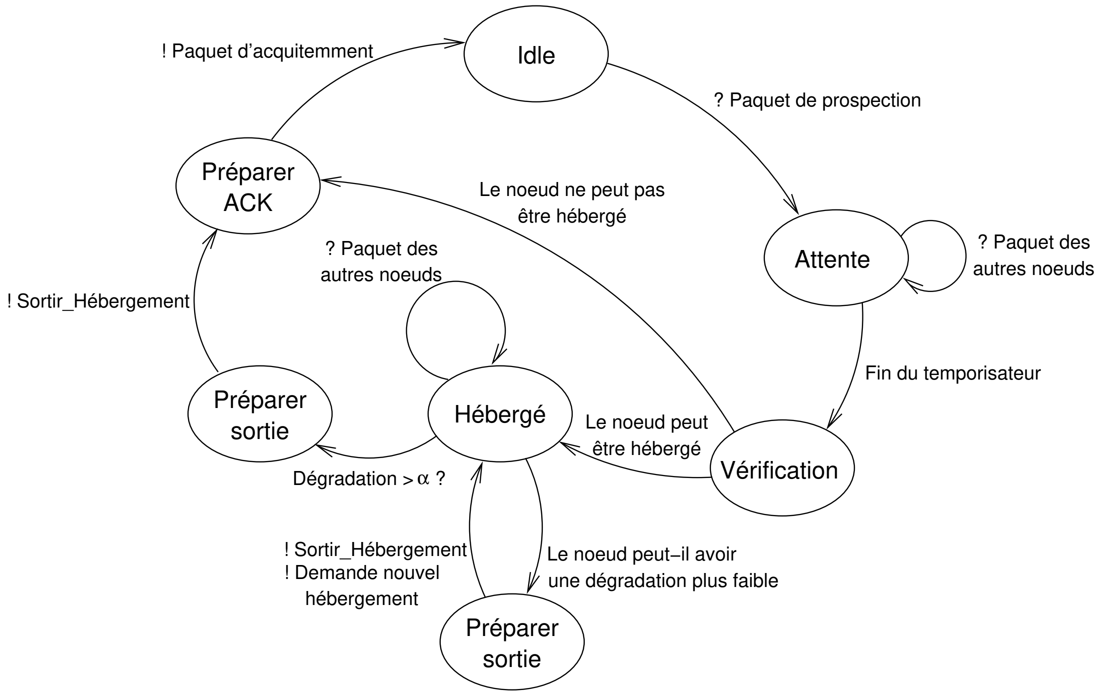
5.4.1 Simulations

Cette section est réservée à l'analyse par simulation du protocole de coopération proposé ci-dessus. Nous nous intéressons plus particulièrement à l'évolution des états des récepteurs hébergés et hébergeurs. Ainsi, nous utilisons des valeurs précises pour les paramètres α et T.

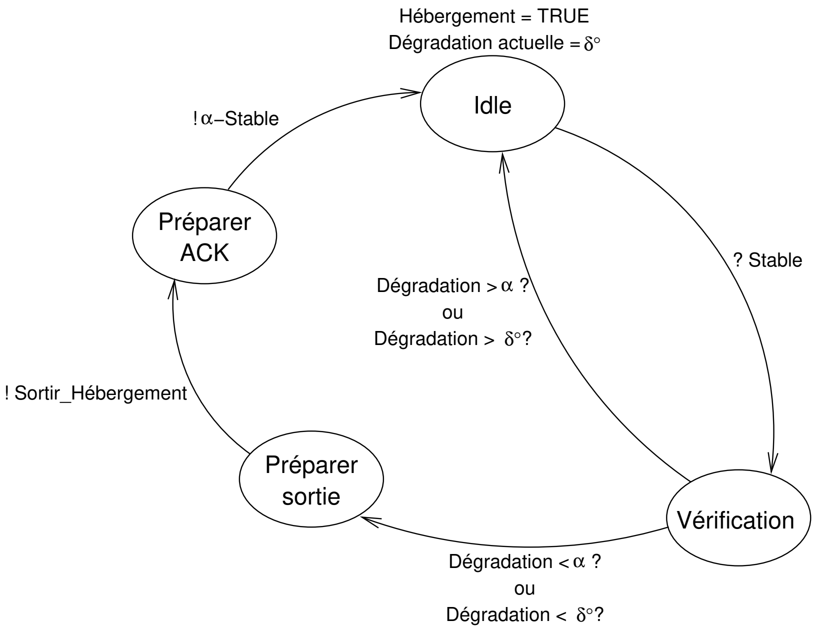
Le premier scénario analysé est composé de 10 récepteurs. Ainsi que pour la section précédente, nous simulons la bande-passante variable des liens avec un trafic de fond (du type exponentiel dans notre cas).

Les courbes des figures 5.14, 5.15, 5.16 et 5.17 montrent les états des récepteurs dans le temps. Nous avons choisi un sous-ensemble des récepteurs pour être représentés ici. Un nœud hébergeur égal à zéro correspond à un récepteur stable, c'est-à-dire, qui n'est pas hébergé. Nous utilisons pour nos simulations les paramètres $\alpha = 0.7$ et $T = 10c$, où c représente le délai de transmission entre deux récepteurs.

Il est facile de voir que certains des récepteurs sont plus stables que d'autres et trouvent assez aisément un hébergeur. Le récepteur r_7, par exemple, après quelques changements



(a) Demande d'hébergement à partir d'un état inactif.



(b) Demande d'hébergement pour un récepteur déjà hébergé.

Fig. 5.13 – Schémas de demande d'hébergement.

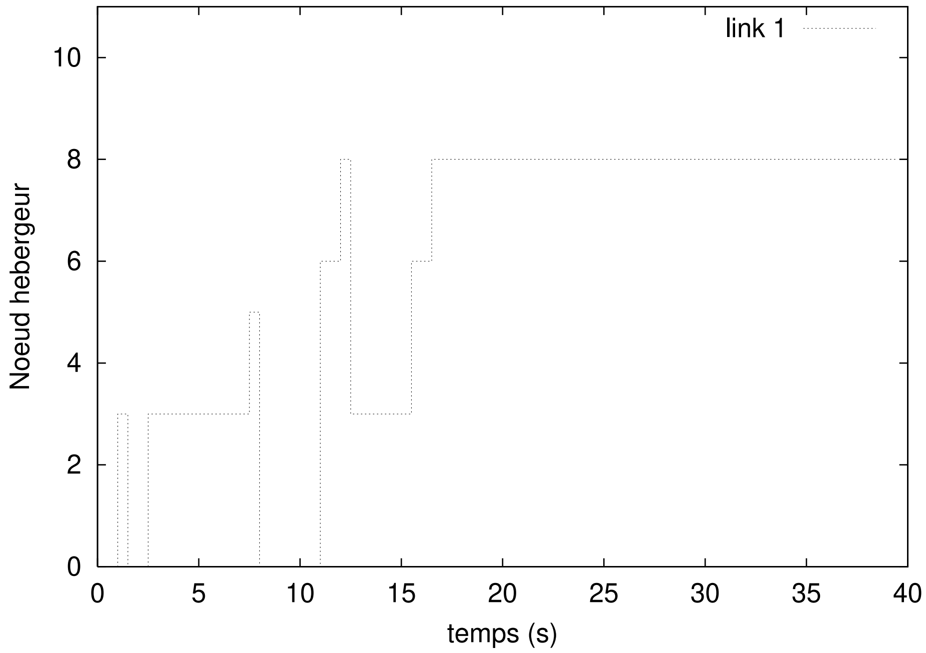


FIG. 5.14 – Hébergement du récepteur r_1.

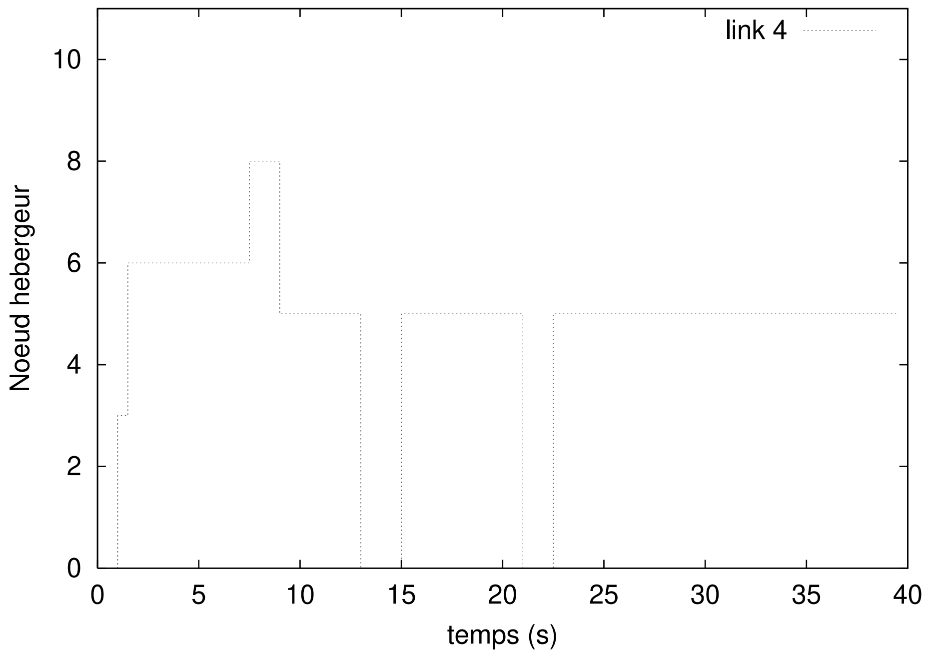


FIG. 5.15 – Hébergement du récepteur r_4.

est hébergé pour une longue période par r_2. Le récepteur r_8, par contre, est hébergé uniquement une fois et devient un récepteur stable et ne demande plus d'hébergement à d'autres récepteurs.

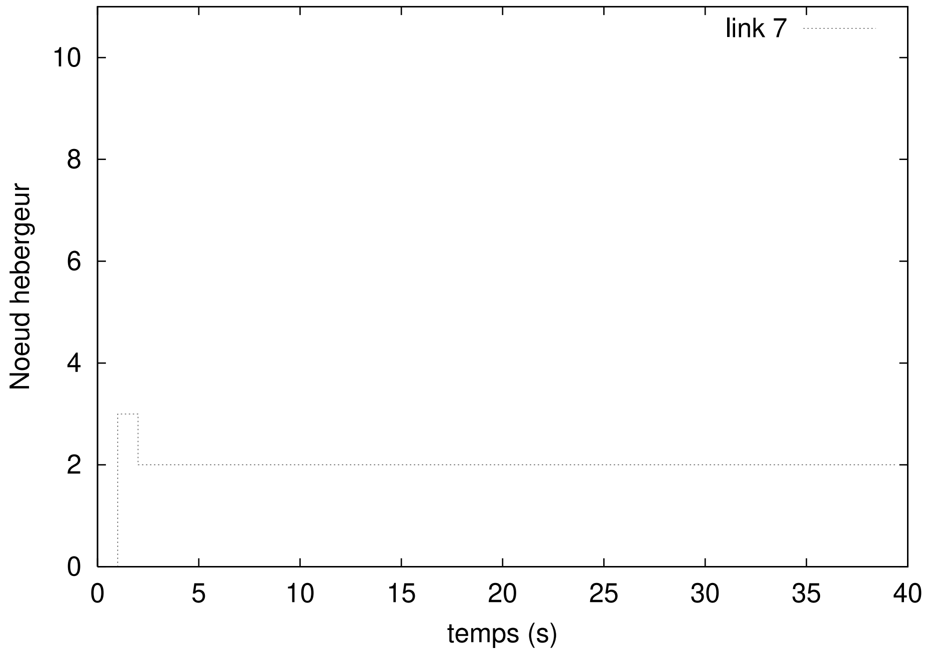


FIG. 5.16 – Hébergement du récepteur r_7.

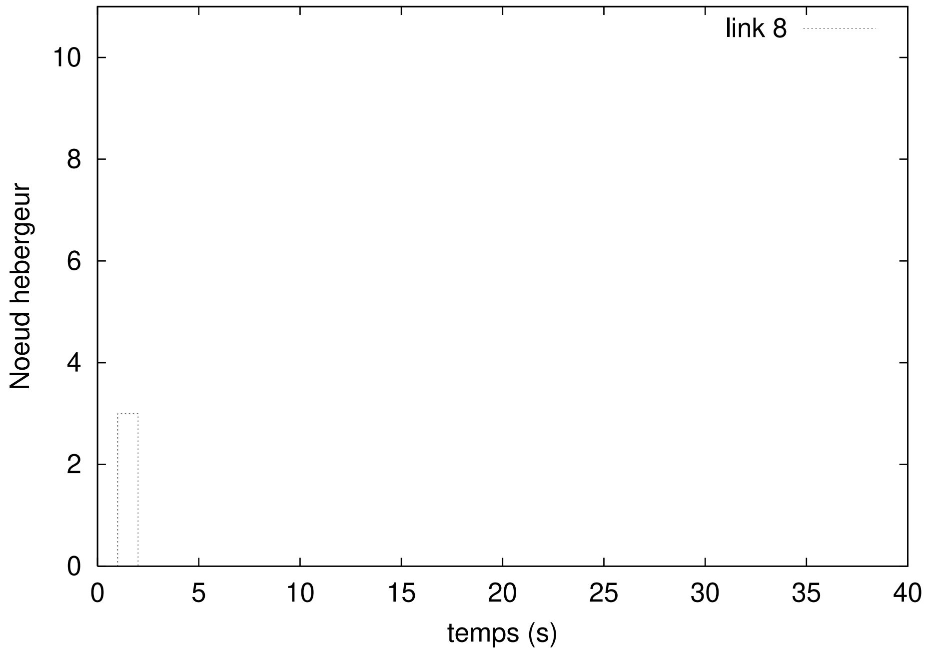


FIG. 5.17 – Hébergement du récepteur r_8.

Il est clair que ces résultats sont directement liés à l'état du réseau, autant pour l'intensité de variation de la bande-passante disponible dans les liens que pour le rapport entre les capacités des récepteurs. Ces simulations servent cependant à montrer que le protocole assume une posture robuste au moins dans les cas où le comportement du réseau n'est pas complètement aléatoire. Dans un tel cas, l'utilisation d'un système sans coopération se montrerait plus adapté.

Maintenant nous allons montrer, du côté de la source, le nombre de paquets d'acquittement reçus des récepteurs stables. Ce résultat est illustré dans la figure 5.18. La scalabilité du protocole de coopération se comporte d'une façon similaire à celle du système sans coopération. Nous montrons également la dégradation obtenue pour la même configuration. Ceci est illustré dans la figure 5.19.

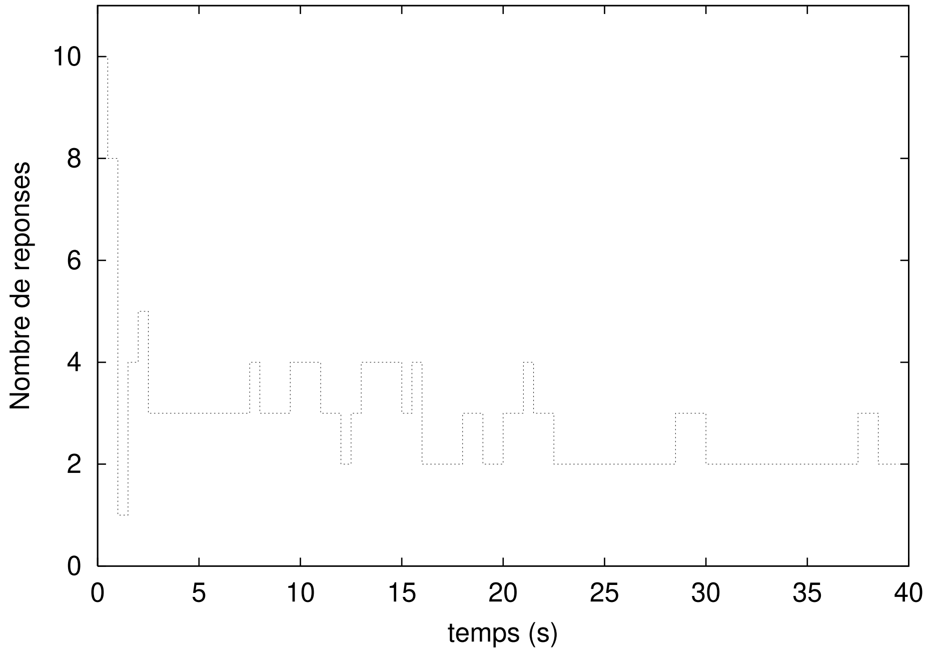


FIG. 5.18 – Nombre de paquets d'acquittement reçus par la source.

5.5 Discussion

L'utilisation d'un système de temporisateurs pour contrôler l'envoi d'acquittements par les récepteurs n'est pas un sujet tout à fait nouveau. L'originalité que nous apportons à ce système est l'analyse du contenu des paquets de contrôle. La suppression ou non de l'acquittement d'un récepteur dépend non seulement de l'arrivée d'un ou plusieurs paquets de contrôle d'autres récepteurs mais aussi de l'information que ces paquets portent.

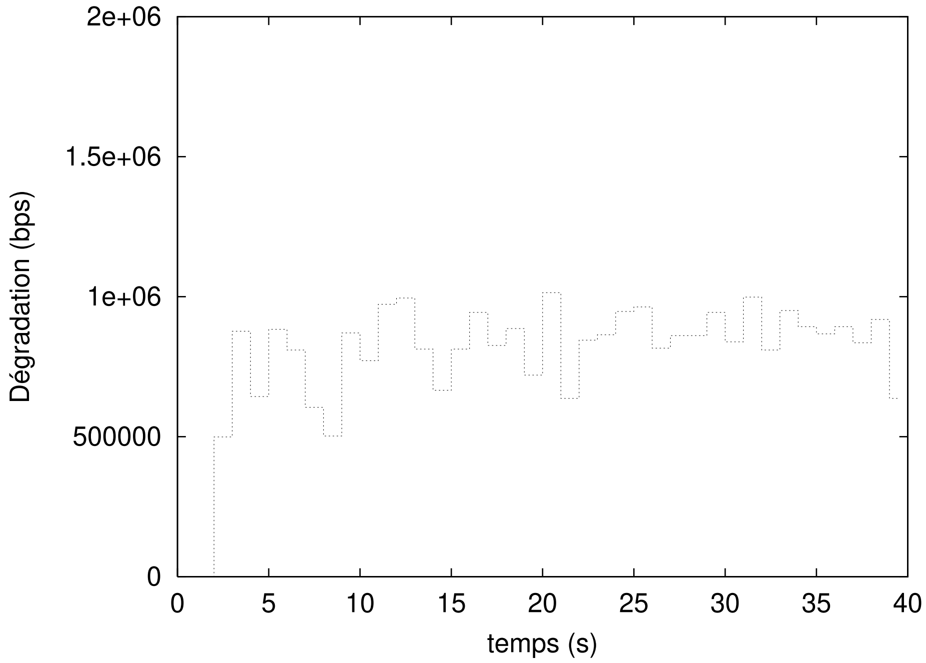


FIG. 5.19 – Dégradation subie par la session multicast.

Ce chapitre a évoqué principalement le problème de scalabilité lorsque l'on ne peut pas implanter des "fusionneurs" dans les nœuds intermédiaires du réseau. L'impossibilité (ou la difficulté) de l'introduction de nouveaux mécanismes dans le nœuds du réseau est un problème intrinsèque aux architectures actuelles. Le travail que nous avons développé dans ce chapitre est, *grosso modo*, l'adaptation des mécanismes proposés dans les chapitres précédents aux réseaux sans nœuds actifs[1].

Nous avons proposé deux approches dans le cadre de cette thèse. Le choix de l'utilisation d'une d'entre elles dépendrait essentiellement de l'état du réseau. L'utilisation du protocole de coopération serait bien évidemment réservée à un réseau dont la variation de la bande-passante disponible n'est pas complètement aléatoire et intense.

Les résultats que nous obtenons dépendent du choix des paramètres tels que le seuil de suppression et la période maximale du temporisateur. Suivant le délai observé sur l'échange d'informations entre les récepteurs et sur l'arrivée des paquets de contrôle, la source peut ajuster la valeur des paramètres de façon à contrôler la charge de réponses et

[1]Nous utilisons le terme "actif" pour désigner un élément capable d'exécuter une nouvelle procédure. Même s'il existe quelques similarités, on ne doit pas le confondre avec la notion habituelle de "réseaux actifs".

la qualité de la session multicast.

Chapitre 6

Conclusion

Nous avons traité dans cette thèse plusieurs aspects de la transmission adaptative de données codées en multi-couches dans les réseaux multicast. D'une part, la haute diversité des réseaux d'aujourd'hui fait de ces objectifs un grand défi. D'autre part, les avancement dans les domaines de représentation de données et de techniques de transmission indiquent chaque jour des nouvelles voies de solutions pour cette hétérogénéité.

Le noyau de notre étude est le système de contrôle explicite du débit de transmission de la source. Ce système est basé sur un mécanisme d'échange de paquets de contrôle où les récepteurs, par l'intermédiaire de messages d'acquittement, maintiennent la source constamment informée sur leurs capacités de réception. La source est ainsi capable de calculer son débit dynamiquement de façon à livrer aux récepteurs les flux les plus équitables possibles.

Nous nous soucions également de la scalabilité du système. Dans une session multicast dont le nombre de récepteurs est très élevé, la source est victime d'implosion de paquets d'acquittement. Cette implosion est due au fait que la quantité d'acquittements est directement proportionelle au nombre de récepteurs. Nous avons résolu ce problème avec l'utilisation d'entités qui agrègent les acquittements dans les nœuds intermédiaires de l'arbre multicast. En effet, ces entités utilisent des algorithmes d'optimisation qui obtiennent à la sortie un seul paquet possédant des informations résultantes de la fusion des contenus des paquets d'entrée.

Dans la première partie de cette thèse, nous avons proposé l'Algorithme Direct de combinaison de paquets d'acquittement. Cet algorithme utilise le principe que toutes les possibilités de calcul du paquet de sortie doivent être examinées. L'approche classique que l'on trouve dans la littérature fusionne les paquets en étapes successives. Nous avons analysé cette approche et les résultats montrent que souvent des dégradations de qualité au-delà des valeurs optimales sont obtenues.

Le comportement et l'efficacité des algorithmes sont aussi dépendants de la topologie du réseau. Dans les topologies en mode dense, où les récepteurs sont concentrés et liés à la source par l'intermédiaire de peu de nœuds intermédiaires, l'Algorithme Direct s'approche de la du résultat optimal (sans fusion dans les nœuds intermédiaires). En effet, pour une topologie où tous les récepteurs sont liés à un seul nœud, l'Algorithme Direct s'égalise à la borne empirique. Les avantages de notre proposition par rapport à l'approche classique sont accentués dans de nombreux cas. Premièrement, dans les réseaux en mode dense cités ci-dessus. Comme l'approche classique fusionne les paquets un à un à chaque fois, plus est grand le nombre de récepteurs attachés à un nœud, plus la différence de qualité entre les deux approches est importante. Deuxièmement, lorsqu'il s'agit d'une topologie en mode dilué (où les récepteurs sont distribués dans le réseau), l'écart entre l'approche classique et l'Algorithme Direct est lié principalement à la profondeur de l'arbre multicast et bien sûr au nombre de récepteurs. Nous avons effectué des tests dans des réseaux du type arbre binaire et les résultats ont montré que le rapport entre la dégradation donnée par l'approche classique et par l'Algorithme Direct avait quasiment doublé entre un arbre de profondeur 2 et un autre de profondeur 3.

Nous avons aussi étudié une autre limitation des systèmes adaptatifs basés sur des paquets envoyés par les récepteurs : l'isolation d'acquittements. Nous avons désigné ce problème comme une non-coopération entre sous arbres de la session multicast. En effet, un nœud exécutant une procédure de combinaison de paquets n'est influencé que par les paquets qui arrivent du sous-arbre qui lui est associé. Il est possible, et même fort probable, que deux paquets similaires (en terme des ressources demandées) soient éliminés dans des nœuds différents avant qu'ils se rencontrent. Dans le but de réduire (et non pas d'éliminer, bien entendu) la limitation de non-coopération entre les nœuds intermédiaires, nous avons proposé une extension à l'Algorithme Direct qui utilise le concept de couches virtuelles.

En faisant croire aux nœuds intermédiaires que la source transmet plus de couches que la quantité réelle, un nombre extra de composantes est conservé. Nous majorons ainsi la probabilité que deux composantes similaires provenant de sous-arbres différents puissent être associées.

Jusqu'à ce point de la thèse, nous nous étions intéressés au problème de la minimisation de la dégradation de qualité de l'application dans les récepteurs. Cela correspond uniquement au point de vue des récepteurs de la session multicast.

Dans la deuxième partie du travail, nous nous sommes intéressés à une conception plus large de qualité de la session multicast, qui considère autant le point de vue des récepteurs que les points de vue du réseau et de la session multicast. Au moment du calcul des couches, la source n'utilise plus seulement la dégradation dans les récepteurs. Nous proposons un système multi-métriques qui vise à fournir une évaluation de la session multicast en autres termes.

Le système multi-métriques que nous avons proposé est fondé sur trois grandeurs de base : la densité de satisfaction des récepteurs, le degré d'utilisation de la bande-passante du réseau et, bien évidemment, la dégradation dans les récepteurs. Ces métriques sont combinées pour donner le niveau de qualité qui sert comme base du calcul des débits des couches. Les tests que nous avons réalisés montrent que l'optimisation de ces métriques est du type divergente, c'est-à-dire, lorsqu'une est optimisée, une autre est dégradée. Il fallait donc trouver un compromis entre ces manipulations qui considère la corrélation entre les métriques.

Notre approche prend en compte le type d'application exécutée. Dans le calcul de la qualité, chaque métrique est liée à un paramètre qui détermine le poids qu'elle aura sur le résultat final. À chaque procédure d'élimination d'une composante dans un nœud du réseau, l'algorithme tente d'obtenir le paquet de sortie qui optimise la métrique prioritaire selon les proportions données par les poids. Nous avons analysé une session multicast qui demandait l'optimisation de l'utilisation des ressources dans le réseau. Malgré la réduction des ressources alloués, comme l'on désirait, nous avons constaté que le système peut être déstabilisé par des variations de l'état du réseau.

Cette limitation nous a poussés à proposer une deuxième approche pour le système multi-métriques : l'approche différentielle. Le but de cette proposition est de restreindre les effets de la variation de l'état du réseau sur les valeurs absolues de métriques (et par conséquent sur la qualité de la session multicast). En effet, la qualité de la session multicast est calculée à partir de la différence entre les métriques des composantes pouvant être éliminées. L'algorithme d'élimination de paquets travaille donc sur des valeurs relatives, et non plus sur les valeurs absolues qui sont directement affectées par les caractéristiques du réseau.

L'approche différentielle utilise aussi un système de poids qui, contrairement au cas précédent, fonctionne comme un modérateur des variations de métriques par rapport à elles-mêmes. En utilisant l'approche différentielle, une session multicast qui supporte une légère augmentation de δ (dégradation de qualité dans les récepteurs) peut profiter d'une grande économie de bande-passante (ou d'une amélioration de la densité de récepteurs satisfaits).

Les deux premières parties de cette thèse considèrent implicitement que les nœuds intermédiaires du réseau sont capables d'implémenter les algorithmes proposés. Néanmoins, il n'est toujours pas vrai que les nœuds peuvent être modifiés de façon à intégrer ces fonctionnalités.

La troisième partie de cette thèse a été dédiée aux réseaux sans nœuds de fusion. Nous avons proposé deux mécanismes devant être introduits uniquement dans les systèmes terminaux, à savoir la source et les récepteurs. Les mécanismes s'appuient sur un système de temporisateurs qui contrôle l'action des récepteurs. Un récepteur ne répond à la source que si son temporisateur est expiré. Bien évidemment, le but est toujours d'éviter le problème d'implosion de paquets d'acquittement à la source.

Dans le premier mécanisme, la réponse d'un récepteur est envoyée à la source et à tous les autres récepteurs appartenant à la session multicast. Un récepteur supprime sa réponse si, pendant sa période d'attente, il reçoit d'un autre récepteur un paquet d'acquittement contenant une demande proche de la sienne. L'utilisation d'un seuil de suppression permet au système de régler la quantité de réponses. De la même manière, ce seuil peut être dynamiquement ajusté par la source si celle-ci considère que le nombre de réponses est bas (ce qui peut provoquer une plus grande dégradation dans les récepteurs) ou trop élevé (implosion).

Le second mécanisme est une extension considérable du premier et introduit un protocole de coopération entre les récepteurs. Nous avons remarqué que si l'état du réseau n'est pas extrêmement variable, certains récepteurs peuvent être "hébergés" par d'autres au long de la communication. Ainsi, en addition à la capacité de réception, les paquets d'acquittement informent la source du nombre de récepteurs ayant cette capacité. Le protocole s'est montré stable et, malgré une réponse plus lente due aux temporisateurs, les résultats illustrent le fait qu'il est possible contrôler le nombre de réponses envoyées à la source.

Perspectives

Nous avons assumé tout au long de notre travail que les débits des couches transmises par la source sont dynamiquement calculés à partir des paquets d'acquittement envoyés par les récepteurs. Rien n'empêche que les capacités des récepteurs varient à de grandes amplitudes, ce qui pourrait provoquer des changements constants de la qualité des applications dans les récepteurs. Il serait intéressant d'étudier l'implantation d'un mécanisme à la source qui fonctionnerait comme un contrôleur des variations trop brusques des débits des couches. En utilisant des valeurs discrètes (plus ou moins fines) et/ou même une approche par hystérèse, il serait possible de réduire ces variations.

Un deuxième point concerne l'effet final sur les utilisateurs. Par exemple, comment une application vidéo doit exactement se comporter dans les sessions multicast et comment ce comportement pourrait être transposé dans notre système ? Ces questions sont aussi liées à la discussion du paragraphe précédent, car des variations constantes de la qualité de réception dans les récepteurs sont normalement très indésirables du point de vue des utilisateurs. De la même façon, on pourrait envisager l'intégration de mécanismes de tarification dans notre système afin de permettre la classification des récepteurs dans des groupes de priorités différentes. Un récepteur appartenant à un groupe plus prioritaire serait moins vulnérable aux changements de débit.

La cohabitation des flux en multi-couches analysés dans le cadre de cette thèse et d'autres flux existants, notamment les flux TCP s'il s'agit par exemple de l'Internet, est aussi un point important devant être développé. Plusieurs propositions ont déjà été faites dans ce domaine et pourraient servir comme point de départ pour l'intégration de ces caractéristiques dans notre système [37, 40, 76, 77, 78].

Finalement, nous avons également comme perspective l'étude de l'application de nos systèmes dans d'autres architectures. Nous avons déjà étudié dans [79, 80] l'application de la vidéo multi-couches avec contrôle explicite en utilisant les réseaux actifs. Dans ce travail, nous nous sommes intéressés au cas où les récepteurs appartiennent à plusieurs collectifs multicast. Ainsi, en présence de congestion, le réseau doit gérer les conflits entre les flux de différents collectifs. Un deuxième point serait l'étude de comment notre approche pourrait s'avérer utile pour les réseaux cellulaires, notamment ceux du type hiérarchique. Le but serait de vérifier si une bonne distribution des couches entre les niveaux micro et macro améliorerait la qualité de la vidéo dans les récepteurs.

Bibliographie

[1] C. Diot, C. Huitema et T. Turletti, “Multimedia applications should be adaptive,” in *HPCS’95*, (Mystic, CN, USA), août 1995.

[2] D. D. Clark et D. L. Tennenhouse, “Architectural considerations for a new generation of protocols,” in *ACM Sigcomm*, (Philadelphia, PA, USA), septembre 1990.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog et S. Jasmin, “Resource reSerVation Protocol (RSVP) – version 1 functional specification.” Request for Comments 2205, Internet Engineering Task Force, septembre 1997.

[4] P. P. White, “RSVP and integrated services in the Internet : A tutorial,” *IEEE Communications Magazine*, pp. 100–106, mai 1997.

[5] M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, “An improved MPEG behavioral analysis with autonomous parameter-extracting algorithm for strict video applications,” in *IEEE ICC’2000*, (New Orleans, USA), juin 2000.

[6] C.-S. Chang, “Deterministic traffic specification via projections under the min-plus algebra,” in *IEEE Infocom*, (New York, USA), mars 1999.

[7] M. D. de Amorim et O. C. M. B. Duarte, “A novel deterministic traffic model based on a two-level analysis of MPEG video sources,” in *IEEE Globecom*, (Sydney, Australia), novembre 1998.

[8] P. Tang et T.-Y. Tai, “Network traffic characterization using token bucket model,” in *IEEE Infocom*, (New York, USA), mars 1999.

[9] V. Elek, G. Karlsson et R. Rönngren, “Admission control based on end-to-end measurements,” in *IEEE Infocom*, (Tel-Aviv, Israel), mars 2000.

[10] D. E. Wrege, E. W. Knightly, H. Zhang et J. Liebeherr, “Deterministic delay bounds for VBR video in packet-switching networks : Fundamental limits and practical tra-

deoffs," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 352–362, juin 1996.

[11] M. Krunz et A. Makowski, "Modeling video traffic using M/G/infinity input processes : A compromise between markovian and LRD models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 733–748, juin 1998.

[12] M. D. de Amorim et O. C. M. B. Duarte, "A self-extracting accurate modeling for bounded-delay video services," rap. tech., Grupo de Teleinformática e Automação – Universidade Federal do Rio de Janeiro. Aussi soumis à *IEEE Transactions on Circuits and Systems for Video Technology*, juin 2000.

[13] R. Rejaie, M. Handley et D. Estrin, "Layered quality adaptation for internet video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, decembre 2000.

[14] J.-C. Bolot et T. Turletti, "A rate control mechanism for packet video in the internet," in *IEEE Infocom*, (Toronto, Canada), mars 1994.

[15] B. Li et K. Nahrstedt, "A control-based middleware framework for quality-of-service adaptations," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 9, pp. 1632–1650, septembre 1999.

[16] M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Single-loop packet merging for receiver-oriented multicast multi-layered video," in *ICCC'99*, (Tokyo, Japon), septembre 1999.

[17] S. McCanne, V. Jacobson et M. Vitterli, "Receiver-driven layered multicast," in *ACM SIGCOMM*, (Stanford, CA, USA), août 1996.

[18] B. J. Vickers, C. Albuquerque et T. Suda, "Source adaptive multi-layered multicast algorithms for real-time video distribution," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 720–733, decembre 2000.

[19] R. Rejaie, M. Handley et D. Estrin, "Architectural considerations for playback of quality adaptive video over the Internet," in *IEEE International Conference on Networks – ICON*, (Singapour), septembre 2000.

[20] M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Application-aware multicast," rap. tech., Grupo de Teleinformática e Automação – Universidade Federal do Rio de Janeiro. Aussi soumis à *IEEE Globecom 2001*, mars 2001.

[21] R. Rejaie, M. Handley et D. Estrin, "Quality adaptation for congestion controlled video playback over the Internet," in *ACM Sigcomm*, (Cambridge, MA, USA), septembre 1999.

[22] V. Jacobson, "VAT – X11 based audio conferencing tool." Manuel Unix, 1993.

[23] X. Li, M. Ammar et S. Paul, "Video multicast over the internet," *IEEE Network Magazine*, pp. 46–60, avril 1999.

[24] T. Turletti et C. Huitema, "Videoconferencing on the internet," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 340–351, juin 1996.

[25] J.-C. Bolot, T. Turletti et I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *ACM Sigcomm*, (Londres, Angleterre), septembre 1994.

[26] S. Y. Cheung, M. H. Ammar et X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *IEEE Infocom*, (San Francisco, CA, USA), mars 1996.

[27] T. Jiang, M. Ammar et E. W. Zegura, "On the use of destination set grouping to improve inter-receiver fairness for multicast ABR sessions," in *IEEE Infocom*, (Tel-Aviv, Israel), mars 2000.

[28] X. Li et M. H. Ammar, "Bandwidth control for replicated-stream multicast video distribution," in *HPDC*, (Syracuse, NY), août 1996.

[29] X. Li, S. Paul et M. H. Ammar, "Layered video multicast with retransmissions (LVRM) : Evaluation of hierarchical rate control," in *IEEE Infocom*, (San Francisco, CA, USA), mars 1998.

[30] B. J. Vickers, C. Albuquerque et T. Suda, "Adaptive multicast of multi-layered video : Rate-based and credit-based approaches," in *IEEE Infocom*, (San Francisco, CA, USA), mars 1998.

[31] M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Multi-critera arguments for improving the fairness of layered multicast applications," in *Networking 2000 – Lecture Notes in Computer Science 1815*, (Paris, France), mai 2000.

[32] M. D. de Amorim et O. C. M. B. Duarte, "Improving the quality specification of multi-layered applications." à paraître dans *International Journal of Networking and Information Systems*, 2001.

[33] M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Otimização da qualidade de vídeo em redes multicast não cooperativas." à paraître dans les proceedings du *congrès brésilien en réseaux – SBRC2001* (en portugais), 2001.

[34] M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Improving user's satisfaction in adaptive multicast video," rap. tech., Grupo de Teleinformática e Automação – Universidade Federal do Rio de Janeiro. Aussi soumis à *International Journal of Communications and Networks*, avril 2001.

[35] K. Fall et K. Varadhan, *The ns Manual.* UC Berkeley, LBL, USC/ISI et Xerox Parc, janvier 2001. Disponible dans http ://www.isi.edu/nsnam/ns/ns-documentation.html.

[36] S. R. McCanne, *Scalable Compression and Transmission of Internet Multicast Video.* PhD thesis, University of California at Berkeley, decembre 1996.

[37] L. Vicisano, J. Crowcroft et L. Rizzo, "TCP-like congestion control for layered multicast data transfer," in *IEEE Infocom*, (San Francisco, CA, USA), mars 1998.

[38] D. Saparilla et K. W. Ross, "Optimal streaming of layered video," in *IEEE Infocom*, (Tel-Aviv, Israel), mars 2000.

[39] S. Sarkar et L. Tassiulas, "Distributed algorithms for computation of fair rates in multirate multicast trees," in *IEEE Infocom*, (Tel-Aviv, Israel), mars 2000.

[40] D. Rubenstein, J. Kurose et D. Towsley, "The impact of multicast layering on network fairness," in *ACM Sigcomm*, (Cambridge, MA, USA), septembre 1999.

[41] S. McCanne, M. Vetterli et V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 982–1001, août 1997.

[42] M. J. Riley et I. E. G. Richardson, *Digital Video Communications.* Artech House, 1996.

[43] E. Amir, S. McCanne et M. Vetterli, "A layered DCT coder for Internet video," in *IEEE International Conference on Image Processing*, (Lausanne, Suisse), septembre 1996.

[44] D. Taubman et A. Zakhor, "Multi-rate 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572–588, septembre 1994.

[45] M. Vishwanath et P. Chou, "An efficient algorithm for hierarchical compression of video," in *IEEE International Conference on Image Processing*, (Austin, TX, USA), novembre 1994.

[46] N. Chaddha, "Software only scalable video delivery system for multimedia applications over heterogeneous networks," in *IEEE International Conference on Image Processing*, (Washington, DC, USA), octobre 1995.

[47] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, decembre 1993.

[48] D. L. Gall, "MPEG : A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 305–313, avril 1991.

[49] G. K. Wallace, "The JPEG still picture compression algorithm," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, avril 1991.

[50] ITU-T Recommendation, "H.263 : Video coding for low-bit rate communication." International Telecommunication Union, fevrier 1998.

[51] S. E. Deering, *Multicast Routing in a Datagram Internetwork*. PhD thesis, Standford University, decembre 1991.

[52] W. Fenner, "Internet Group Management Protocol." Request for Comments 2236, Internet Engineering Task Force, novembre 1997.

[53] V. Jacobson, "Congestion avoidance and control," in *ACM Sigcomm*, (Standford, CA, USA), août 1998.

[54] D. Waitzman, S. Deering et C. Partridge, "Distance Vector Multicast Routing Protocol." Request for Comments 1075, Internet Engineering Task Force, novembre 1988.

[55] J. Moy, "Multicast extensions to OSPF." Request for Comments 1584, Internet Engineering Task Force, mars 1994.

[56] T. Ballardie, P. Francis et J. Crowcroft, "Core Based Trees (CBT) : An architecture for scalable inter-domain multicast routing," in *ACM Sigcomm*, (San Francisco, CA, USA), septembre 1993.

[57] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu et L. Wey, "The PIM architecture for wide-area multicast routing," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 153–162, avril 1996.

[58] L. H. M. K. Costa, S. Fdida et O. C. M. B. Duarte, "Hop-by-hop multicast routing protocol," in *ACM Sigcomm*, (San Diego, CA, USA), août 2001.

[59] S. Paul, K. Sabnani, J. Lin et S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, avril 1997.

[60] C. Papadopoulos, G. Parulkar et G. Varghese, "An error control scheme for large-scale multicast appliations," in *IEEE Infocom*, (San Francisco, CA, USA), avril 1998.

[61] T. Speakman, D. Farinacci, S. Lin et A. Tweedly, "PGM reliable multicast specification." Internet Draft (travail en cours), Internet Engineering Task Force, août 1998.

[62] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : A transport protocol for real-time applications." Request for Comments 1889, Internet Engineering Task Force, janvier 1996.

[63] X. Li, S. Paul, P. Pancha et M. H. Ammar, "Layered video multicast with retransmissions (LVRM) : Evaluation of error recovery," in *Nossdav*, (San Louis, MI, USA), mai 1997.

[64] G. Iannaccone et L. Rizzo, "A layered video transcoder for videoconference applications," rap. tech., Mosaico Research Repport PI-DII/4/99, Université de Pisa, Italie, 1999.

[65] E. Amir, S. McCanne et H. Zhang, "An application-level video gateway," in *ACM Multimedia*, (Boston, MA, USA), novembre 1995.

[66] P. A. A. Assunção et M. Ghanbari, "Multicasting of MPEG-2 video with multiple bandwidth constraints," in *7th International Workshop on Packet Video*, (Brisbane, Australie), mars 1996.

[67] E. Amir, S. McCanne et R. Katz, "An active service framework and its application to real-time multimedia transcoding," in *ACM Sigcomm*, (Vancouver, Canada), septembre 1998.

[68] S. Keshav, *Congestion Control in Computer Networks*. PhD thesis, University of California at Berkeley, septembre 1991.

[69] V. Paxson, *Measurements and Analysis of Ent-to-End Internet Dynamics*. PhD thesis, University of California at Berkeley, avril 1997.

[70] A. Legout et E. Biersack, "PLM : Fast convergence for cumulative layered multicast transmission scheme," in *ACM Sigmetrics*, (Santa Clara, CA, USA), juin 2000.

[71] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy et R. Viswanathan, "ERICA switch algorithm : A complete description." ATM-Forum / 96-1172, août 1996.

[72] J. Nonnenmacher et E. W. Biersack, "Scalable feedback for large groups," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 375–386, juin 1999.

[73] S. Pejhan, M. Schuartz et D. Anastassiou, "Error control using retransmission schemes in multicast transport protocols for real-time media," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 413–427, juin 1996.

[74] S. Floyd, V. Jacobson, C. Liu, S. McCanne et L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–803, decembre 1997.

[75] P. Sharma, D. Estrin, S. Floyd et V. Jacobson, "Scalable timers for soft state protocols," in *IEEE Infocom*, (Kobe, Japon), avril 1997.

[76] S. Floyd et K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, août 1999.

[77] R. Rejaie, *An End-to-End Architecture for Quality Adaptive Streaming Applications in the Internet*. PhD thesis, University of Southern California, decembre 1999.

[78] D. Sisalem et H. Schulzrinne, "The loss-delay based adjustment algorithm : A TCP-friendly adaptation scheme," in *Nossdav*, (Cambridge, Angleterre), juillet 1998.

[79] N. Achir, M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Active bandwidth sharing for multi-layered video in multi-source environments." à paraître dans *International Journal of Networking and Information Systems*, 2001.

[80] N. Achir, M. D. de Amorim, O. C. M. B. Duarte et G. Pujolle, "Gestion active de la vidéo dans un environnement multisource," in *JDIR*, (Paris, France), novembre 2000.

Résumé de Thèse

Distribution de flux en couches dans les réseaux multicast hétérogènes

Marcelo Dias de Amorim

Juin 2001

Une des caractéristiques des réseaux multicast à large échelle est l'hétérogénéité des récepteurs en terme de capacité de réception. Un flux de données a donc son débit contrôlé par les récepteurs les moins ou les plus rapides. Dans le premier cas, on observe une sous utilisation des ressources pour les récepteurs les plus puissants, tandis que dans le second cas un certain nombre de récepteurs est obligé de quitter la session par manque de capacité. En plus de l'hétérogénéité des récepteurs, le système doit aussi contourner la variation de bande-passante disponible dans les liens. Cette thèse propose un système adaptatif de contrôle du débit de transmission de la source en utilisant la technique de codification hiérarchique. Le contrôle adaptatif est utilisé pour maintenir le débit de transmission conformement aux ressources disponibles dans les liens. La codification hiérarchique (en multi-couches) vise à régler l'hétérogénéité des récepteurs.

Nous proposons dans cette thèse un système de contrôle des débits des couches basé sur des paquets d'acquittement envoyés par les récepteurs. Ces paquets sont combinés dans les nœuds intermédiaires pour éviter l'implosion à la source. La première partie du travail propose l'Algorithme Direct qui optimise le calcul des couches afin d'augmenter la qualité globale de la session multicast dans les récepteurs. Le calcul de la qualité de la communication est basé sur la dégradation (débit demandé moins le débit alloué) dans les récepteurs. Nous vérifions que pour certaines applications l'utilisation d'une seule métrique ne correspond pas au cas idéal. Nous proposons donc ensuite une approche multi-métriques qui prend en compte les limitations de l'approche avec un seul paramètre. La troisième et dernière partie de cette thèse est une extension de la première partie mais considère que le réseau n'est pas capable d'exécuter les fonctions de fusion de paquets. Nous proposons un système de contrôle aux bouts (source et récepteurs) qui utilise un mécanisme de temporisateurs et d'échange de messages entre les récepteurs pour supprimer les paquets d'acquittement avant qu'ils ne soient envoyés à la source. L'ensemble du système se montre efficace en ce qui concerne la réponse face aux variations de l'état du réseau et très équitable en terme de degré de satisfaction des récepteurs.

Mots-clés : applications adaptatives, vidéo multi-couches, dégré de qualité, réseaux multicast.