La Communication Multimédia de Groupe de Bout-en-Bout

José F. de Rezende[1]
Serge Fdida[1]
Andreas Mauthe[2]

(1) *Laboratoire MASI*
Université Pierre et Marie Curie
4, place Jussieu 75252 Paris Cedex 05
Adresse électronique: rezende/fdida@masi.ibp.fr.

(2) *Computing Department*
Lancaster University
Lancaster LA1 4YR - UK
Adresse électronique: andreas@comp.lancs.ac.uk.

Résumé. *Plusieurs protocoles réseau fournissent le multicast des messages à un groupe de récepteurs; cependant, le problème du support de la communication de groupe de bout-en-bout n'a pas reçu toute l'attention méritée. Dans ce contexte, les problèmes spécifiques de bout-en-bout concernent principalement l'intégrité du groupe et la fiabilité versus l'insensibilité au facteur d'échelle. Le M-Connection Service, présenté dans cet article, offre des différentes schémas de fiabilité. Nous introduisons ce service et montrons comment ces schémas de fiabilité sont fournis. À partir d'un ensemble de résultats de simulation, nous discutons l'insensibilité au facteur d'échelle des différents schémas et analysons les paramètres de performance.*

Abstract. *Many network level protocols allow the necessary multicast of messages to a group of receivers. However, the problem of end-to-end group communication support has not received appropriate attention. End-to-end specific issues in this context are communication integrity and the problem of reliability versus scalability. The M-Connection Service discussed in this paper offers different reliability schemes. We introduce this service and discuss how different reliability semantics are accommodated by it. From a set of simulation results we discuss the scalability of the different reliability schemes provided and analyse the contributing factors.*

Mots-clés : *Communication de groupe, Multicast, Applications multimédia.*

Keywords: *Group communication, Multicast, Multimedia Applications.*

1. Introduction

Plusieurs applications distribuées, comme les applications multimédia coopératives et les applications de vidéoconférence, demandent aux services et protocoles sous-jacents une diversité de mécanismes de support pour la communication de groupe. Le mode de communication de groupe considéré comme indispensable est le multicast (1 vers N); c'est à dire, une topologie avec un seul émetteur et plusieurs récepteurs. Les réseaux locaux et métropolitains actuels fournissent le support pour le multicast à travers leurs caractéristiques inhérentes de broadcast, cependant, il existe un manque de support pour la communication de groupe de bout-en-bout. Cet article relate les résultats obtenus dans l'investigation de problèmes importants liés au support de la communication de groupe multimédia.

Un problème spécifique de la communication de groupe de bout-en-bout est la fiabilité versus l'insensibilité au facteur d'échelle (scalability). Une communication est considérée comme fiable si les données envoyées par l'émetteur sont correctement reçues par la totalité des récepteurs. La fiabilité est assurée par l'utilisation des algorithmes de reprise sur erreur. Dans une approche orientée-émetteur, celui-ci est responsable de la correction des erreurs, en utilisant les informations d'état transmises par les récepteurs, ainsi que de la retransmission des données, quand elle est nécessaire. Or, cette approche pose un problème dès lors que la taille du groupe devient importante; à savoir, l'effet d'implosion de la source. Quand l'état global des récepteurs est maintenu par l'émetteur, il existe un problème de sensibilité de l'algorithme à la taille du groupe, même avec l'utilisation d'algorithmes efficaces et de machines puissantes.

L'étude de la fiabilité versus l'insensibilité au facteur d'échelle a fait récemment l'objet de quelques travaux [1, 2]. Dans [1], un cadre pour la communication de groupe fiable, dénommé SRM (Scalable Reliable Multicast), a été défini, où l'insensibilité au facteur d'échelle est obtenue par l'utilisation d'un mécanisme de garantie de la fiabilité orienté-récepteur et de l'approche ALF (Application Level Framing). Ici, les récepteurs sont responsables de la détection et de la récupération d'erreurs, d'une façon totalement distribuée. Les algorithmes spécifiés supposent que tous les récepteurs sont capables de retransmettre, une fois qu'un modèle de distribution multicast, tel celui défini par IP, est disponible. Par conséquent, ces algorithmes sont dépendants de ce modèle, ce qui rend leur implémentation difficile et coûteuse dans un environnement ATM homogène. Dans [2], un nouveau concept, appelé Local Group Concept (LGC), est présenté. Le groupe de récepteurs est divisé en sous-groupes distincts, où les récepteurs qui appartiennent au même sous-groupe sont géographiquement proches les uns des autres. Dans chaque sous-groupe, une entité est responsable de la collecte des informations d'état des récepteurs, et est chargée, pour éviter l'effet d'implosion, d'en informer l'émetteur par un seul message. Cette entité est aussi responsable de la retransmission des données au sous-groupe. Les deux travaux décrits ci-dessus supposent qu'il est possible d'obtenir de l'application les données nécessaires à une éventuelle retransmission. Aussi, pour exploiter au mieux leurs fonctionnalités, il faut avoir connaissance des réseaux sous-jacents.

Un second problème spécifique à la communication de groupe est l'intégrité du groupe qui n'a pas été traité par la plupart des études existantes. L'intégrité du groupe se réfère à certaines conditions, dénommées conditions d'intégrité, sous lesquelles une communication est considérée comme réussie ou non. Ces conditions portent sur le nombre et/ou l'identité des récepteurs, qui doivent recevoir les données correctement. La fiabilité et l'intégrité du groupe sont extrêmement liées et elles doivent être traitées dans un même cadre. Le M-Connection Service fournit ce cadre, il constitue un service de transport qui offre un support multicast aux applications multimédia de groupe [3].

Le M-Connection Service offre quatre classes de service avec différents degrés de fiabilité et d'intégrité. Un nouveau concept de fiabilité, la fiabilité probabiliste [4], est introduit, où il est possible de fournir un haut degré de fiabilité à des groupes dynamiques et de taille importante. Les résultats de simulation montrent comment les différentes sémantiques de fiabilité affectent la sensibilité au facteur d'échelle, et aussi, ils révèlent que les concepts de fiabilité totale et probabiliste sont alternatifs, tout en étant complémentaires.

Cet article est structuré de la manière suivante. Le M-Connection Service est introduit dans la section 2. La section 3 discute les différentes sémantiques de fiabilité proposées. Dans la section 4, une introduction de l'intégrité du groupe, ainsi qu'une discussion des effets des différentes sémantiques sur l'intégrité, sont présentées. La section 5 montre les résultats de simulation à partir desquels une comparaison de différentes sémantiques est réalisée. Et finalement, dans la section 6, les conclusions et les futures évolutions de ce travail sont présentées.

2. M-Connection Service

Pour satisfaire à toutes les exigences et réduire la complexité de la communication de groupe, une approche qui prend en compte l'ensemble du problème est nécessaire. En fonction de ce besoin, une architecture de service a été spécifiée, laquelle est composée de plusieurs modules de service indépendants afin d'accéder à une plus grande efficacité et une plus grande flexibilité. Le M-Connection Service est un module essentiel de notre architecture; son objectif est d'offrir aux applications multimédia un support de communication de bout-en-bout entre un émetteur et plusieurs récepteurs. Il constitue un service de transport qui peut reposer sur plusieurs protocoles de transport, dès lors que ces protocoles fournissent un certain niveau de support au multicast (1 vers N) et également une notion de QoS. Le service offert par le M-Connection Service est indépendant de tout type de protocole et de réseau, ses fonctionnalités ne dépendant ni des caractéristiques de transmission ni de la structure du réseau. Malgré cette indépendance, quelques protocoles ont été considérés comme candidats potentiels, tels que XTP 4.0 [5] et le protocole de transport multimédia développé à l'université de Lancaster [6]. Pour montrer la faisabilité du service, nous avons mis en correspondance les éléments de service du M-Connection Service avec les mécanismes proposés par XTP 4.0 [3].

Le M-Connection Service est composé d'un ensemble restreint d'éléments de service, qui permettent aux utilisateurs d'établir, joindre, quitter, modifier les propriétés et terminer une *m-connection*. Les m-connections sont des connexions de transport multicast établies dans le but de transmettre des données multimédia. Le but du M-Connection Service est double: i) proposer aux utilisateurs un ensemble d'éléments de service pour la communication multicast, et ii) cacher les détails et la spécificité des différents protocoles de transport.

2.1. *Les Classes de Service*

Le M-Connection Service a été conçu pour supporter le transfert des médias continus aussi bien que celui des médias discrets. De plus, différents types d'applications avec une variation dans le nombre de récepteurs sont considérés. Quatre classes de service ont été définies pour accommoder différents types de trafic: A - best-effort; B - intégrité du groupe et/ou QoS; C - intégrité du groupe et fiabilité totale; et D - intégrité du groupe et fiabilité probabiliste.

La classe A propose un service best-effort sans négociation de la QoS. Néanmoins, l'émetteur peut spécifier la QoS d'un flot de données de manière à informer les récepteurs du type de trafic qui sera véhiculé. Cette classe de service a été définie pour les services de diffusion, comme par exemple la radio et la télévision. La deuxième classe de service, la classe B, fournit la notion d'intégrité du groupe, ainsi que la possibilité de négocier la QoS. L'intégrité du groupe se réfère à la possibilité de spécifier des conditions d'intégrité associées à la m-connection. Il est toujours possible de demander la négociation de la QoS sans la spécification des conditions d'intégrité, et vice-versa. La QoS peut être négociée entre tous les participants, selon les schémas décrits ci-dessous. La classe de service B a été conçue pour transférer des médias continus et aussi des médias discrets qui ne requièrent pas de fiabilité. La classe C offre l'intégrité du groupe, ainsi que la fiabilité totale, dans le transfert des données. La fiabilité totale est obtenue par des mécanismes de retransmission des données perdues ou erronées, utilisant l'information d'état de tous les récepteurs. La QoS spécifiée dans cette classe donne une indication de la bande passante nécessaire et les contraintes temporelles associées à la m-connection. Aucune garantie de délai pour les données reçues correctement n'est donnée; en d'autres termes, la fiabilité des données, dans cette classe, a toujours priorité par rapport aux contraintes de temps. La classe D a été spécialement conçue pour fournir un haut niveau de fiabilité pour le transfert des données dans des groupes dynamiques et de taille importante. Un mécanisme particulier de retransmission basé sur la connaissance partielle de l'état du groupe est employé. La QoS est fondée sur la spécification d'un taux de transmission de l'émetteur et il n'y a pas de négociation.

2.2. *Les Schémas de Négociation de la QoS*

La négociation de la QoS a lieu entre l'émetteur, le fournisseur du service et les récepteurs. Trois schémas différents de négociation ont été définis: (1) QoS imposée: l'émetteur spécifie une certaine QoS qui ne peut pas être négociée. Chaque récepteur a seulement le droit de l'accepter ou de la refuser; (2) dénominateur commun: une QoS commune doit être trouvée pour tous les récepteurs; (3) sélectionnée par le récepteur: l'émetteur spécifie la QoS maximale qu'il peut offrir, et chaque récepteur sélectionne la QoS appropriée, qui peut être fournie par des mécanismes de filtrage [7].

3. La Fiabilité et le Multicast

La fiabilité dans la communication multicast ne dépend pas que du type de fiabilité fourni par le réseau et des fonctions de retransmission utilisées, mais aussi de la taille du groupe récepteur. Notons que la surcharge causée par la gestion du groupe croît exponentiellement en fonction du nombre de récepteurs. De ce fait, il n'est pas possible de fournir de fiabilité à des groupes de taille importante, ou alors, il faut trouver d'autres mécanismes pour pouvoir offrir un certain niveau de fiabilité. Le M-Connection Service est destiné à véhiculer différents types de médias et aussi à accommoder des groupes de tailles diverses. En raison de cette spécificité, quatre classes différentes de service ont été définies.

Le M-Connection Service offre trois types différents de fiabilité, à savoir, aucune fiabilité dans les classes A et B, une fiabilité totale dans la classe C et une nouvelle sémantique, appelée fiabilité probabiliste, dans la classe D. Les deux sous-paragraphes qui suivent se focalisent sur les deux sémantiques présentes dans le M-Connection Service.

3.1. *Communication Totalement Fiable*

Dans le M-Connection Service, une communication totalement fiable est obtenue à travers l'utilisation d'un schéma de gestion de groupe orienté-liste, proche de celui proposé par XTP 4.0 [5]. Dans ce schéma, une liste du groupe récepteur est créée à partir des réponses positives de récepteurs à une demande d'établissement d'une m-connection. Pendant la communication, l'émetteur collecte, à intervalles réguliers, des informations de contrôle du groupe récepteur pour mettre à jour cette liste. L'état de la m-connection est déterminé par l'émetteur en fonction de toutes les réponses des récepteurs. Si une réponse de l'un des récepteurs, pour quelque raison, n'arrive pas, un processus de synchronisation (synchronising handshake) est initié par l'émetteur. Pendant ce processus, la transmission est arrêtée et l'émetteur demande, un certain nombre de fois défini par l'utilisateur, des informations d'état de tous les récepteurs jusqu'à ce qu'il obtienne une réponse de la part de tous les récepteurs. À chaque nouvelle tentative, l'émetteur augmente la période pendant laquelle il collecte les réponses. À la fin de ce processus, si l'un des récepteurs n'a toujours pas répondu, il est considéré comme ayant quitté le groupe récepteur. Si après son départ, les conditions d'intégrité sont toujours satisfaites, la communication continue sans lui.

L'émetteur doit conserver une copie des paquets transmis jusqu'à ce que tous les récepteurs aient acquitté leur réception. Si une retransmission est nécessaire, les paquets seront envoyés à tout le groupe. Le M-Connection Service utilise pour l'instant le mécanisme go-back-n pour la retransmission.

Le multicast orienté-liste offre une communication fiable en suivant la trace de l'appartenance explicite de l'ensemble des récepteurs et aussi en assurant la retransmission de toutes les données perdues ou erronées. Par conséquent, une fiabilité totale est garantie pour le groupe récepteur en entier. Pourtant, même avec une implémentation optimale, la taille du groupe est limitée. De plus, si le groupe est très dynamique, une charge supplémentaire est ajoutée par l'arrivée et le départ des utilisateurs qui requièrent une fréquente mise à jour de la liste du groupe récepteur.

3.2. *La Fiabilité Probabiliste*

Pour répondre aux exigences de fiabilité des applications avec des groupes de taille importante et/ou dynamiques, un nouveau mécanisme de contrôle d'erreur a été développé. La sémantique de fiabilité fournie par ce mécanisme est différente de celle obtenue par l'approche orientée-liste. Quoique ce mécanisme, qui a été inspiré par l'algorithme de bucket du XTP 3.6, n'offre pas une fiabilité totale, il garantit un haut niveau de fiabilité avec une faible probabilité d'erreurs non-récupérables.

A contrario du schéma fiable orienté-liste qui est fondé sur une connaissance totale et globale de l'état des récepteurs, la décision de (re-)transmettre des données est basée sur la connaissance partielle de cet état. La philosophie centrale de cette sémantique consiste en ce que les récepteurs sont responsables de la détection et de la reprise sur erreurs.

L'algorithme de bucket amélioré

Cet algorithme utilise un mécanisme d'acquittement négatif qui laisse aux récepteurs la responsabilité de détecter la perte des paquets de données et de déclencher les mécanismes de reprise. Parallèlement à l'acquittement négatif, une stratégie d'acquittement basée sur l'émetteur est employée pour signaler efficacement à l'émetteur qu'il peut libérer ses tampons mémoire et aussi pour vérifier l'état global du groupe quand la communication se déroule dans des conditions normales. Dans cette stratégie, l'émetteur maintient une liste de structures de données, nommées buckets, et il envoie périodiquement un message de contrôle associé à un bucket pour solliciter des récepteurs des informations sur leur état. Pour chaque message de contrôle envoyé, l'émetteur crée un nouveau bucket. Dans chaque bucket, l'émetteur mémorise le plus grand numéro de séquence des paquets transmis jusqu'ici. Sur réception d'un message de contrôle, les récepteurs répondent avec un message d'écho qui contient le plus grand numéro de séquence des paquets correctement reçus. Ce message contient aussi les informations requises pour mettre à jour des variables utilisées par d'autres mécanismes, tels que le contrôle de flux, le contrôle de débit, l'estimation du round-trip-time (RTT), etc.

L'émetteur doit fusionner les messages d'écho dans une structure de données unique et représentative. Par conséquent, l'émetteur maintient une liste de variables globales qui suivent la trace de l'état des récepteurs. Ces variables comportent, par exemple, le plus petit numéro de séquence des données reçues correctement (*min_rseq*). Un récepteur qui détecte un trou dans les numéros de séquence envoie un acquittement négatif à l'émetteur. Si le numéro de séquence du paquet demandé est plus petit que le numéro de séquence actuellement maintenu dans *min_rseq*, cette variable est mise à jour avec cette nouvelle valeur. Elle est également mise à jour si un message d'écho acquitte un numéro de séquence plus petit que celui qui est mémorisé.

Le nombre de buckets est fini; ainsi, quand tous les buckets sont remplis, l'émetteur vide le bucket le plus ancien et le remplace par un nouveau bucket. À ce moment là, il vérifie s'il faut retransmettre des données. Une retransmission n'est initiée que si elle n'a pas déjà été antérieurement traitée ce qui a pour objectif d'éviter des retransmissions redondantes [8]. Tous les paquets qui suivent le numéro de séquence demandé sont retransmis à tout le groupe (retransmission en multicast go-back-n). De leur côté les paquets pour lesquels une demande de retransmission n'est plus attendue, sont effacés du tampon mémoire. Les données ne sont retransmises que lorsqu'un bucket est remplacé, ainsi un acquittement négatif n'est pas immédiatement traité. De cette façon, plusieurs demandes sont traitées par une seule retransmission au lieu d'une seule demande à chaque fois.

En fait, la liste de buckets correspond à une période de temps finie Tp, c'est-à-dire qu'un bucket reste toujours dans la liste pendant cette période de temps. Cette période doit donc être assez grande pour permettre aux récepteurs de répondre aux messages de contrôle avant que le bucket respectif soit remplacé. Une période de temps qui s'avère optimale est l'estimation du RTT maximum ($maxRTT$) multipliée par un facteur qui peut être modifié; par exemple, dans un environnement instable, un accroissement de sa valeur assure que les buckets n'expirent pas prématurément. Le nombre de buckets (N_{Buck}) dans la liste détermine la fréquence avec laquelle les messages de contrôle sont envoyés, les demandes de retransmission traitées et les tampons libérés. Notons que N_{Buck} est un paramètre de conception à dimensionner qui détermine la vitesse de libération des tampons mémoire et le niveau de fiabilité. Il doit être choisi selon le niveau d'activité de la connexion multicast.

L'insensibilité au facteur d'échelle et la fiabilité

Pour éviter que la source ne soit submergée par des messages d'écho (l'effet d'implosion de la source), un algorithme de *back-off* capable de réduire leur nombre est utilisé. Avec cet algorithme, un récepteur ne répond pas à chaque message de contrôle. Tout le groupe répond au premier message de contrôle. Au second message, un récepteur répond avec une probabilité de 0.5, au troisième avec une probabilité de 0.25, etc. Pour assurer qu'en moyenne au moins un récepteur répondra à chaque message de contrôle, la probabilité avec laquelle un récepteur doit répondre ne doit pas être plus petite que $\frac{1}{2^k}$ où $k = \ln_2 N$[1]. À chaque fois qu'un récepteur envoie un acquittement négatif, la probabilité avec laquelle il répondra au prochain message de contrôle est mise à 1. Ce mécanisme garantit également qu'un récepteur répond en moyenne une fois à tous les 2^k messages de contrôle, et qu'il ne devienne pas complètement silencieux.

Un choix raisonnable pour la période de temps, pendant laquelle un bucket est valable, doit faire en sorte d'éviter que les tampons soient libérés avant que tous les récepteurs aient la possibilité de faire une demande de retransmission. Pourtant, il peut exister une situation dans laquelle un acquittement négatif arrive après que les paquets ont été libérés du tampon mémoire. Cette situation peut se produire, par exemple, si le délai entre l'émetteur et l'un des récepteurs croît soudainement. Dans ce cas là, l'émetteur signale à ce récepteur que cette erreur ne peut pas être récupérée. Dans le message, il inclut aussi le plus petit numéro de séquence qui est encore disponible. À cause de ces erreurs non-récupérables, ce mécanisme n'offre pas un service totalement fiable.

4. L'Intégrité du Groupe

La communication multicast introduit un nouveau problème associé à l'intégrité du groupe récepteur et qui se rapporte aux conditions selon lesquelles l'établissement d'une connexion et le transfert des données sont considérés comme ayant abouti. Ces conditions portent sur le nombre et/ou l'identité des récepteurs qui doivent recevoir correctement les données transmises. Pour les médias continus, ces contraintes sont liées aux paramètres de QoS et non pas à un seul paquet de données. Chaque récepteur est tenu de valider la QoS reçue et, au cas où elle n'est pas conforme à sa spécification, il informe l'émetteur qui doit donc valider les conditions d'intégrité.

Dans le M-Connection Service, un ensemble de conditions a été défini pour permettre à l'utilisateur de spécifier l'intégrité de l'établissement et de la communication. Les conditions suivantes sont considérées: *membre-clé* (identité) et/ou k (nombre), *tous*, *quorum*[2].

4.1. *Les Médias Continus et l'Intégrité du Groupe*

Les contraintes temporelles associées aux médias continus rendent virtuellement impossible l'utilisation d'un mécanisme de retransmission destiné à la correction d'erreurs. En outre, un faible nombre d'erreurs est totalement acceptable pour la plupart des applications multimédia. Cependant, quelques

[1] N est égal au nombre des récepteurs

[2] Pour les conditions d'établissement, *quorum* et *tous* se rapportent à un groupe connu; en d'autres termes, le nombre et/ou l'identité des membres du groupe sont connus de l'utilisateur du service ou du M-Connection Service. Pour les conditions d'intégrité de la communication, *quorum* et *tous* se réfèrent au nombre de réponses positives pendant l'établissement de la m-connection.

applications de groupe utilisant des médias continus requièrent l'intégrité du groupe. Dans ces conditions, un (sous) groupe d'utilisateurs qui doit recevoir une certaine QoS est spécifié.

Dans le M-Connection Service, ces applications sont prises en compte par la classe de service B. Les conditions d'intégrité de la communication désignent le nombre et/ou l'identité des utilisateurs qui doivent être présents. Chaque récepteur détermine si la QoS reçue est conforme à la spécification. Dans le cas contraire, et si toutes les actions pour remédier à la situation n'aboutissent pas, le récepteur doit quitter la communication. Il incombe alors à l'émetteur de valider les conditions d'intégrité et de déclencher les actions nécessaires. Par ailleurs, pour éviter que les conditions d'intégrité ne soient pas satisfaites parce qu'un récepteur meurt silencieusement[3], l'émetteur demande périodiquement l'information d'état des récepteurs. La périodicité avec laquelle ces demandes sont envoyées, appelé *intervalle d'intégrité*, dépend des besoins de l'application.

4.2. *Fiabilité Totale et Intégrité du Groupe*

La plus forte notion d'intégrité de groupe, disponible dans le M-Connection Service, est fournie par la classe de service C, le service totalement fiable. Dans cette classe, il est garanti que les récepteurs reçoivent correctement les données ou que ces dernières sont encore disponibles à l'émetteur. Les conditions d'intégrité ne sont donc plus satisfaites lorsqu'un récepteur requis n'est plus présent, et non parce qu'il ne reçoit pas une QoS suffisante. Une invalidité momentanée de l'intégrité du groupe peut seulement avoir lieu entre deux messages de contrôle.

Pour la classe de service C, les conditions d'intégrité sont validées à chaque fois que l'émetteur découvre qu'un récepteur n'est plus présent; par exemple, quand un récepteur quitte ou est forcé à quitter une m-connection[4]. De plus, l'émetteur valide les conditions d'intégrité à chaque fois qu'il détecte qu'un récepteur meurt silencieusement. Si les conditions d'intégrité ne sont plus satisfaites, l'émetteur termine la m-connection. En contraste avec la classe de service B, l'intervalle d'intégrité n'est pas seulement déterminé par l'importance de l'intégrité du groupe; parmi les autres paramètres qui déterminent cet intervalle se trouvent: le nombre de récepteurs, la taille du tampon mémoire de transmission, le $maxRTT$ et la bande passante du réseau.

4.3. *Fiabilité Probabiliste et Intégrité du Groupe*

Dans la classe de service D, l'intégrité de groupe doit être considérée dans un contexte plus large. Elle ne se réfère qu'à un certain intervalle, pendant lequel la présence des récepteurs obligatoires doit être validée, comme dans la classe de service B. Elle n'est pas non plus un autre paramètre de fiabilité qui détermine le (sous) groupe qui doit recevoir une copie correcte des données. Ici,

[3]Les récepteurs peuvent par exemple être coupés à cause d'une rupture du réseau.

[4]Un utilisateur récepteur est, par exemple, forcé par le fournisseur du service à quitter une connexion quand il ne suit pas les autres récepteurs.

l'intégrité du groupe assure qu'un ensemble de récepteurs obligatoires est, en moyenne, présent, et reçoit une version correcte des données transmises. Cela se traduit, entre autres, par le fait qu'un taux d'erreur est défini, qui représente le nombre d'erreurs non-récupérables acceptables.

Le principal facteur qui détermine la taille de l'intervalle d'intégrité est l'algorithme de *back-off*. Avec cet algorithme, un récepteur répond, en moyenne, à chaque 2^k-ème message de contrôle, ainsi la taille minimum de l'intervalle d'intégrité est 2^k fois la période entre deux messages de contrôle consécutifs.

5. Analyse de Performance

Le M-Connection Service a été mis en correspondance avec XTP4.0 dans le but de démontrer qu'il peut être supporté par un protocole existant. De plus, il a été analysé par une séquence de simulations afin d'évaluer le service et l'influence des différents scénarios, sur son efficacité et ses performances. Cette étude focalise sur deux sémantiques différentes de fiabilité et sur leur sensibilité au facteur d'échelle. Ici, nous présentons et discutons les résultats obtenus par simulation des algorithmes conçus pour les classes C et D.

Pour cette étude, nous avons utilisé un outil de simulation reposant sur des modèles de files d'attente à événements discrets ($QNAP_2$). Nous avons ainsi développé un modèle de communication multicast avec un nombre variable de récepteurs (N). Dans ce modèle, l'émetteur aussi bien que les récepteurs disposent de tampons mémoire avec une capacité limitée à B paquets de taille constante S. Les temps de traitement de messages de contrôle (C), de données (D) et d'écho (E) sont les suivants: C, $E = x$, et $D = 3*x$. Le temps d'exécution de l'algorithme de contrôle d'erreur est x. Le réseau sous-jacent utilise la technique ATM et est caractérisé d'une part, par un taux de transfert de 155 Mbits/s et d'autre part, par un taux d'erreur par paquet L. La distance entre l'émetteur et l'$i-ème$ récepteur est représentée par un délai de propagation D_i, D_i étant uniformément distribué dans l'intervalle $[D_{min}, D_{max}]$. L'émetteur génère du trafic périodiquement avec un intervalle constant $SData$ séparant deux paquets de données consécutifs; les récepteurs les consomment selon un taux de service exponentiel de moyenne $TSData$.

L'indice de performance clé dans l'évaluation des algorithmes est le temps moyen de réponse des paquets de données (MRT). Il représente le temps moyen écoulé entre la génération d'un paquet par l'application et sa remise à l'utilisateur récepteur. Les algorithmes de contrôle d'erreur utilisés dans les classes C et D sont comparés selon les résultats obtenus pour ce paramètre.

5.1. *L'insensibilité au facteur d'échelle des classes C et D*

L'insensibilité au facteur d'échelle des classes C et D est comparée en utilisant le MRT en fonction de N. La figure 1 montre les performances propres aux deux algorithmes. L'axe x représente le nombre de récepteurs et l'axe y le MRT. Dans ces simulations, le délai de propagation dans le réseau est uniformément distribué sur [15,28] ms. Les temps de traitement des paquets sont obtenus en utilisant $x = 0{,}5$ ms. De plus, nous utilisons les valeurs suivantes pour les paramètres de simulation: $B = 32$ paquets, $S = 2000$ octets, $L = 10^{-5}$, $SData = 10$ ms, $TSData = 0{,}5$ ms. Dans l'algorithme de bucket amélioré, le nombre de buckets (N_{BUCK}) est pris égal à 5.

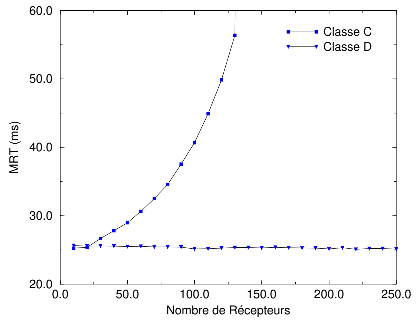

Figure 1: *Comparaison entre les classes C et D.*

La figure 1 montre que l'algorithme totalement fiable reste stable jusqu'à 20 récepteurs. À partir de ce point, une augmentation continue du MRT est observée. Avec plus de 130 récepteurs, une hausse rapide et abrupte du MRT est remarquée. Un comportement comme celui-là était prévisible; il peut s'expliquer essentiellement par deux facteurs: i) l'effet d'implosion de la source, ii) les restrictions sur les tampons mémoire de l'émetteur. Avec 130 récepteurs, un seuil est atteint, à partir duquel l'émetteur n'est plus capable de gérer efficacement les récepteurs. Une garantie de fiabilité totale génère toujours des coûts additionnels. Quelques optimisations devraient permettre d'accommoder un nombre plus important de récepteurs, mais il ne sera jamais possible d'obtenir une fiabilité totale sans aucun coût supplémentaire. Les résultats concernant l'algorithme de bucket amélioré attestent que, dans les mêmes conditions, le MRT reste pratiquement constant avec l'augmentation du nombre de récepteurs. Notons que dans ces simulations aucun cas d'erreur non-récupérable n'a pu être constaté. Pourtant, une fiabilité totale ne peut pas être garantie et la probabilité de telles erreurs demeure.

5.2. *Paramètres de Performance*

Les deux algorithmes sont influencés par différents facteurs externes liés à l'hétérogénéité du réseau, à celle du groupe de récepteurs, à la topologie, au taux d'erreur, etc. Chaque algorithme est affecté de manière différente; par conséquent, nous devons considérer séparément les optimisations adaptant les algorithmes à des conditions spécifiques.

Fiabilité Totale

Les principaux facteurs ayant de l'influence sur la performance de l'algorithme totalement fiable orienté-liste sont la surcharge engendrée par la gestion du groupe, le taux de perte (figure 2) et les restrictions des tampons mémoire de l'émetteur. La surcharge provoquée par la gestion du groupe, appelée effet d'implosion de la source, occasionne une augmentation continue du MRT du fait que, pour chaque nouveau récepteur, l'émetteur doit traiter davantage de paquets de contrôle et gérer un nombre croissant de récepteurs. Pendant ce temps là, il n'est pas capable d'envoyer des paquets de données. Dans le pire des cas, l'émetteur est continuellement en train de réaliser des tâches de gestion, ce qui lui permet difficilement d'envoyer une seule donnée de l'application. Cependant, les restrictions des tampons mémoire jouent également un rôle considérable dans la performance de cet algorithme.

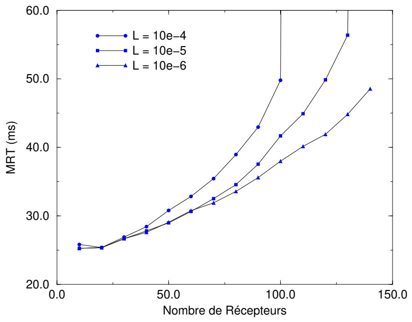

Figure 2: *L´algorithme totalement fiable avec différentes taux de perte.*

Avec un nombre croissant de récepteurs, le $maxRTT$ augmente parce que la surcharge de la gestion du groupe s'ajoute au calcul du $maxRTT$. L'émetteur ne peut transmettre que s'il a assez de tampons mémoire libres. Lorsque tous les tampons de transmission sont remplis, la transmission doit s'arrêter jusqu'à ce que tous les récepteurs aient acquitté les données déjà transmises. Avec l'augmentation du $maxRTT$, la fréquence avec laquelle les tampons mémoire sont libérés diminue; par conséquent, la transmission des données doit s'arrêter plus souvent, d'où l'effet négatif sur le MRT observé. Dans le cas de tampons mémoire de taille illimitée, l'augmentation du MRT est attribuée à l'effet d'implosion de la source et au nombre croissant de demandes de retransmission dû à une plus grande probabilité d'erreurs.

En raison des effets négatifs du $maxRTT$, sa valeur doit être soigneusement estimée. Si elle est surestimée, les tampons mémoire ne sont pas libérés le plus tôt possible. Par ailleurs, si elle est sous-estimée, l'émetteur entre plus souvent en phase de synchronisation. Pendant cette période, la transmission de données est arrêtée, augmentant ainsi le MRT. La valeur du $maxRTT$ est directement liée au délai d'acheminement des paquets dans le réseau et à la charge du récepteur. Une modification de cette valeur affecte immédiatement la performance de l'algorithme totalement fiable. Dans un environnement hétérogène, l'estimation du $maxRTT$ doit être accordée à la valeur du RTT observée avec le récepteur le plus lent.

Les résultats mettent en évidence que pour obtenir une meilleure performance, tout en fournissant la fiabilité totale, il faut non seulement optimiser l'utilisation des tampons mémoire, mais aussi remédier au problème d'implosion de la source. Le premier point demande une libération plus fréquente des tampons mémoire, pour amener une réduction de la sensibilité à l'estimation du $maxRTT$. Le second point, à son tour, exige une répartition des messages d'écho dans le temps.

Fiabilité Probabiliste

Dans l'algorithme de bucket amélioré, les problèmes liés aux tampons mémoire de l'algorithme totalement fiable sont évités par une libération plus fréquente des tampons. La fréquence avec laquelle ces derniers sont libérés est déterminée par le nombre de buckets qui devient alors un facteur primordial dans la performance de l'algorithme. La figure 3 montre son influence sur la performance pour différentes tailles de tampons mémoire, en utilisant la même charge de trafic. Avec l'augmentation du nombre de buckets, une quantité moindre de tampons mémoire dans l'émetteur suffit pour atteindre la même performance. Cependant, au-delà d'un certain seuil, aucune amélioration de performance ne peut être obtenue par ajout de buckets supplémentaires. Une étude antérieure indique que, si le nombre de buckets est trop grand, la performance commence à se dégrader suite au nombre croissant de messages de contrôle à envoyer, à une mise à jour plus fréquente de l'état des récepteurs et à une hausse du nombre de retransmissions [8]. Ainsi, la valeur la meilleure pour le nombre de buckets est la valeur minimum pour laquelle la performance est considérée comme optimale. En utilisant cette valeur minimum, le coût de gestion de buckets est minimisé.

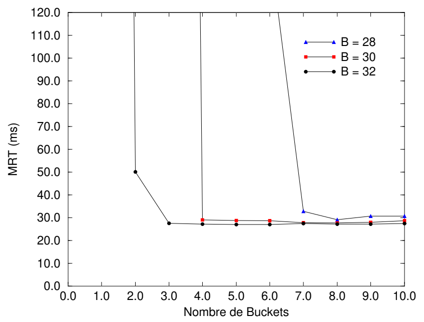

Figure 3: *L´influence de nombre de buckets sur le* MRT.

L'algorithme de bucket amélioré est beaucoup moins sensible aux variations du $maxRTT$, parce que ce paramètre ne détermine plus la fréquence de libération des tampons mémoire et, par conséquent, la cadence de transmission. Cependant, de larges variations du RTT d'un récepteur en particulier peuvent provoquer quelques erreurs non-récupérables. Dans ce cas, les tampons mémoire peuvent être libérés avant que le récepteur qui observe un plus large RTT puisse réagir. Pourtant, nos simulations montrent que le nombre d'erreurs non-récupérables est toujours borné et très faible. La stratégie utilisée pour réduire ces erreurs au minimum fait que l'émetteur considère tous les messages d'écho, même ceux qui arrivent en retard, pour estimer le $maxRTT$. L'émetteur ajuste le $maxRTT$ de façon à prendre en compte les récepteurs plus lents. Dans un environnement très dynamique et hétérogène, deux problèmes apparaissent. Tout d'abord, les modifications dynamiques influencent le $maxRTT$ en provoquant de grandes variations sur sa valeur, ce qui entraîne des effets négatifs dans le nombre d'erreurs non-récupérables.

6. Conclusions

La communication de groupe ainsi que le multicast suscitent une diversité de nouveaux problèmes qui affectent tous les composants de l'architecture de communication. La recherche actuelle se concentre essentiellement dans le réseau, le routage et la transmission de données; par contre, les problèmes de bout-en-bout n'ont pas jusqu'ici reçu l'attention qu'ils méritent. La question de la fiabilité en fonction de l'insensibilité au facteur d'échelle est l'un de ces problèmes. Une communication multicast fiable nécessite la gestion du groupe d'utilisateurs et la retransmission des données perdues ou erronées. Avec un nombre croissant de récepteurs, la surcharge causée par la gestion du groupe, de même que le nombre de retransmissions, augmentent. En conséquence, une façon efficace et insensible au facteur d'échelle capable de fournir des sémantiques différentes à cette grande diversité d'applications de groupe est nécessaire.

Le M-Connection Service est un service de transport qui fournit le support multicast aux applications multimédia. Il constitue un service indépendant qui peut être fourni au-dessus de tout type de réseau offrant le multicast et la notion de QoS. Le M-Connection Service propose quatre classes de service pour divers types de média et divers besoins de fiabilité.Les classes A et B sont conçues pour les services de distribution et de médias continus, et n'offrent aucune sorte de fiabilité de transfert. Cependant, pour la classe B, l'intégrité du groupe peut être définie, selon laquelle un (sous) groupe obligatoire doit recevoir les données transmises avec une certaine QoS. La classe C offre la fiabilité totale garantie par une gestion du groupe orientée-liste et un mécanisme de retransmission. Les résultats de simulation montrent que le nombre de récepteurs, qui peut être supporté par ce schéma, est limité. La classe D offre une fiabilité probabiliste: la fiabilité est fournie par des retransmissions en réponse à des acquittements négatifs. Toutefois, une faible probabilité d'erreurs non-récupérables existe. L'occurrence de ce type d'erreurs n'est pas fréquente dans des conditions normales, et, de plus, certaines applications peuvent s'en accommoder ou les récupérer elles-mêmes. Une question importante, qui est généralement omise dans ce contexte, porte sur l'intégrité du groupe. L'intégrité du groupe est une mesure selon laquelle la transmission de données est considérée comme avoir réussi, une fois qu'un sous-groupe spécifié de participants est présent et reçoit les données correctement. Dans le M-Connection Service, l'intégrité du groupe est fournie dans les classes de service B, C et D.

Plusieurs problèmes intéressants et difficiles demeurent. Pour l'algorithme de bucket amélioré, nous essayons un nombre de buckets qui devrait être ajusté dynamiquement selon la charge de trafic et les différents RTT. Pour remédier au problème de tampons mémoire de l'algorithme totalement fiable, nous étudions un schéma dans lequel les tampons mémoire sont divisés en sections, une section représentant la plus petite unité de gestion. Une libération constante et plus fréquente des tampons mémoire dans ces sections est obtenue par des mécanismes particuliers. L'implosion de la source est évitée par une répartition des acquittements dans le temps.

Références

[1] S. Floyd et al., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", in *Proceedings of ACM SIGCOMM'95*, Cambridge, USA, Aug. 1995.

[2] M. Hofmann, "A Generic Concept for Large-Scale Multicast", in *Proceedings of International Zurich Seminar on Digital Communication (IZS'96)*, Zurich, Switzerland, Feb. 1996.

[3] J. F. de Rezende, A. Mauthe, S. Fdida and D. Hutchison, "M-Connection Service: A Multicast Service for Distributed Multimedia Applications", in *Proc. of Multimedia Transport and Teleservices, International COST237 Workshop*, Copenhagen, Denmark, Nov. 1995.

[4] S. Fdida, "Multimedia Transport Protocol and Multicast Communication", in *High-Speed Networking for Multimedia Applications*. W. Effelsberg, O. Spaniol, A. Danthine, D. Ferrari (eds.), Kluwer Academic Publishers, Boston/Dordrecht/London, 1996.

[5] XTP Revision 4.0, "Xpress Transport Protocol Specification", XTP Forum, Santa Barbara, CA, Mar. 1995.

[6] F. García, "Continuous Media Transport and Orchestration Services", *PhD Thesis*, Lancaster University - Lancaster, UK, May 1993.

[7] N. Yeadon, F. García, D. Hutchison and D. Shepherd, "Filters: QoS Support Mechanisms for Multipeer Communications", To appear in *IEEE Journal on Selected Areas in Communications*, 1996.

[8] H. Santoso and S. Fdida, "Transport Layer Multicast: An Enhancement for XTP Bucket Error Control", in *Proc. of HPN'92, 4th IFIP Conference on High Performance Networking*, Liège, Belgium, pp. G2-1-18, Dec. 1992.