N° d'ordre 2176

THÈSE

Transport multipoint fiable à très grande échelle :

Intégration de critères de coût en environnement Internet hybride satellite / terrestre

Présentée pour obtenir

Le titre de docteur de l'Institut National Polytechnique de Toulouse

École doctorale : Informatique et Télécommunications
Spécialité : Réseaux et Télécommunications

Par :
Florestan de Belleville

Soutenue le 8 Décembre devant le jury composé de :

M.	*Michel Diaz*	Président du jury
M.	*Christian Fraboul*	Directeur de thèse
M.	*Richard Castanet*	Rapporteur
Mme	*Pascale vicat-Blanc Primet*	Rapporteur
M.	*Laurent Dairaine*	Membre (co-encadrant)
M.	*Isabelle Buret*	Membre

École doctorale Informatique et Télécommunications

Transport multipoint fiable à très grande échelle :

Intégration de critères de coût en environnement Internet hybride satellite / terrestre

Thèse pour le doctorat en Réseaux et Télécommunications de
l'Institut National Polytechnique de Toulouse
Octobre 2004

par :
Florestan de Belleville

Directeur de thèse :
Christian Fraboul, Professeur de l'I.N.P.T.

Co-encadré par :
Laurent Dairaine, Enseignant chercheur à l'E.N.S.I.C.A.

Remerciements

Je tiens à remercier :

Christian Fraboul pour son soutien et sa vision pragmatique des problèmes. En de nombreuses occasions, son recul s'est avéré essentiel pour dégager des problèmes imperceptibles et trouver de nouvelles solutions.

Laurent Dairaine pour son aide, ses conseils et sa disponibilité au cours de ces trois années. Sans oublier ses nombreuses idées ! (même si elles ont parfois dépassé les limites de la réalité :oD).

Un grand merci à Jérôme Lacan, Jean-Yves Tourneret, et Pierre de Saqui Sannes pour leur disponibilité et leur efficacité au cours de nos différentes collaborations. Le travail réalisé serait bien différent sans leurs précieuses contributions.

Merci également à Ludovic Apvril pour son aide avec le logiciel TTool (pour faire écho à ce qu'il m'a dit un jour, je pense que l'ouverture d'une hotline serait une bonne idée). Pas une fois il n'a fait défaut à mes requêtes (pourtant parfois à la limite du harcèlement).

Merci aux professeurs Richard Castanet et Pascale Vicat-Blanc Primet pour avoir accepté de rapporter ma thèse, ainsi qu'au professeur Michel Diaz pour avoir présidé mon jury de thèse. Tous trois ont accepté malgré un emploi du temps extrêmement chargé, ce dont je leur suis infiniment reconnaissant. Je suis de plus très honoré d'avoir pu les compter parmi les membres de mon jury.

L'ensemble des membres du laboratoire TéSA et du Département Mathématiques et Informatiques de l'E.N.S.I.C.A. pour leur accueil chaleureux, et en particulier Laurent Frank, David Bonacci, Olivier Florens, Milena Planells, Corinne Mailhes, Jean-Yves Tourneret, André-Luc Beylot, Sylvie Eichen, Nathalie Thomas, Emmanuel Chaput, Christian Fraboul et Francis Castanié, directeur du laboratoire TéSA. Avec un merci particulier à Sylvie et André-Luc, pour leurs contributions à la diffusion des informations au sein du laboratoire.

Merci également à Tanguy Perenou, René Espourteau, Yves Caumel, Bernard Jarlan, Fabrice Frances et Patrick Senac, directeur du DMI. Sans oublier Laurent Lancérica, Manu, Hervé, Jérôme (Fîmes) et Tarek qui ont sans aucun doute contribué à l'ambiance inimitable du DMI.

Merci à toute la fine équipe des doctorants en réseau, cuvée 2001 : Julien (Fasson),

Jérôme (Grieu) et Fabrice (Arnal). C'est en partie grâce à notre soutien mutuel que je suis venu à bout de ces trois années.

Merci à tous ceux qui m'ont soutenu, notamment au cours des derniers mois qui ont incontestablement été les ¡¡ meilleurs ¿¿ (la rédaction d'une thèse reste un moment . . .inoubliable).

Et pour finir merci à Hélène, ma compagne, qui n'a cessé de m'épauler et a tout fait pour que jamais je ne perde confiance ni espoir.

Avant-Propos

La problématique abordée au cours de cette thèse concerne les communications multipoints fiables à très grande échelle par satellite. Lors de telles communications, une source transmet des données à de nombreux récepteurs, et s'assure que ces derniers ont bien reçu les informations transmises. L'étude de cette problématique a été initialement proposée au sein du projet DIPCAST, labellisé par le Réseau National de Recherche en Télécommunications. Ce projet (le DVB support d'IP multiCAST) étudiait la possibilité d'utiliser un standard de diffusion satellite, DVB-S, comme support pour la transmission de données en mode multipoint. La transmission des données devait par ailleurs être effectuée au moyen d'un autre standard : IP (Internet Protocol), ou plus exactement avec son extension pour les communications multipoints, IP Multicast. Le travail réalisé dépasse cependant largement le cadre de ce projet. L'utilisation d'un satellite étant très intéressante pour la transmission de données à des groupes d'utilisateurs, nous nous sommes intéressés à l'utilisation de réseaux hybrides satellites / terrestres pour rendre un tel service. Plus précisément, nous avons analysé la possibilité de prendre en compte le coût de communications de manière à utiliser au mieux les services réseaux disponibles (le lien satellite et le réseau terrestre). C'est précisément cette problématique qui a fait l'objet des recherches présentées dans ce manuscrit.

Le travail effectué au cours de ces trois années de thèse a été réalisé dans le laboratoire TéSA (Télécommunications Spatiales et Aéronautiques), ainsi qu'au Département Mathématiques et Informatiques de l'École Nationale Supérieure d'Ingénieurs de Constructions Aéronautiques. Le fait d'être ainsi rattaché à deux laboratoires différents m'a, entre autres, permis de rencontrer davantage de personnes aux compétences très variées. Dans le cadre du travail de thèse réalisé, cela m'a permis d'aborder des sujets aussi variés que le codage d'informations, les problèmes liés à l'estimation de paramètres, ou encore la vérification de protocoles. Ces différentes collaborations se sont de plus déroulées de manières efficaces et agréables, ce qui n'a fait que renforcer mon attachement à ces laboratoires.

Enfin j'ai également eu la chance au cours de ces trois années, de découvrir l'enseignement. Si au départ c'est surtout la curiosité qui m'a poussé à commencer cette activité ; il me sera désormais difficile de m'en passer.

Table des matières

Introduction

De nombreuses applications requièrent un service de communication permettant à une source de joindre plusieurs récepteurs. Par exemple certains systèmes de communications multimédias nécessitent une communication multipoint car ils requièrent un échange d'informations au sein de communautés distribuées (vidéoconférence, vidéo à la demande, etc.). Les jeux interactifs distribués sont un autre exemple d'applications potentiellement utilisatrices de la communication de groupe. Enfin, dans le cadre des *Content Delivery Network* (CDN), l'alimentation de serveurs de cache répartis géographiquement représente également une application potentielle pour ce type de communication, car le même type d'information doit être acheminé vers des sites distribués géographiquement.

Dans le contexte d'Internet, des solutions techniques répondant à ce besoin ont déjà été proposées. Certaines solutions consistent à utiliser le réseau terrestre sans aucune modification, et à définir un service de communication multipoint au niveau applicatif (voir par exemple : [64], [84] ou encore [123]). Cette solution — communément appelée Multicast applicatif — est majoritairement utilisée à l'heure actuelle car elle présente l'avantage de se baser sur une technologie réseau standard. Cependant cette solution est très coûteuse en ressources réseau, notamment en terme d'utilisation de bande passante. Une autre solution, plus efficace, consiste à intégrer ce service au niveau réseau. Cette solution, qui se base sur *Internet Protocol* (IP [106]), est appelée IP Multicast [26]. Le chapitre 1 décrit la problématique du transport multipoint lorsque le support IP Multicast est utilisé. Dans ce cadre précis, de nombreuses solutions proposées pour répondre à cette problématique sont également présentées.

Toutefois, la mise en place du service IP Multicast dans les réseaux tarde, pour des raisons techniques et économiques [29]. Ainsi ce service est principalement proposé dans le cadre de réseaux de recherche ou d'enseignement comme le MBone [83] (pour *Multicast backBone*) ou encore à l'heure actuelle le Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche en France (RENATER [110]), et le projet GÉANT [46] qui couvre l'Europe. Une des raisons expliquant le retard de déploiement de IP Mulicast est un problème d'offre et de demande auto-entretenu. En effet comme ce service n'existe pas, les applications ont contourné le problème en adoptant des solutions *ad hoc* (comme le Multicast applicatif). Ainsi, il n'y pas de demande pressante pour le développement de ce service. L'intérêt économique lié au développement de ce service est donc a première vue masqué. Cela n'incite pas les gestionnaires des réseaux à investir dans le déploiement d'un tel service... les applications continuent donc à définir des solutions *ad hoc*. A ce problème s'ajoute le manque de

modèle économique pour la tarification de telles communications [127]. Ce problème se pose notamment lorsque le trafic généré traverse les réseaux de plusieurs opérateurs, car la définition d'accords commerciaux entre ceux-ci n'est pas simple.

Il est cependant possible de déployer efficacement un service de communication de groupe à large échelle. Une solution pour cela consiste à utiliser un satellite géostationnaire pour diffuser les informations. Plusieurs raisons permettent de penser que la technologie satellite constitue un candidat modèle pour ce type de communication. Tout d'abord ce type de système de communication permet de couvrir de vastes zones (de l'ordre d'un continent) avec un seul satellite. Ensuite les informations transmises par les satellites sont diffusées naturellement sur toute leur zone de couverture. En conséquence ces systèmes permettent d'atteindre un très grand nombre de récepteurs (tous ceux situés dans la zone de couverture) en un seul bond. Cette possibilité est donc étudiée dans plusieurs projets de recherche récents tels DIPCAST [31] et GEOCAST [48]. De plus, comme cette solution fait appel à un système central (le système satellite), elle permet de surmonter les problèmes de modèles économiques qui apparaissent dans les réseaux terrestres.

L'utilisation d'un lien satellite, bien qu'elle permette de déployer efficacement le service IP Multicast, pose toutefois un certain nombre de problèmes. Les caractéristiques des liens satellites diffèrent en effet complètement de celles des liens terrestres [82]. Ces caractéristiques induisent des dysfonctionnements et des manques d'efficacité des mécanismes protocolaires conçus pour les réseaux terrestres. Ces problèmes apparaissent notamment au niveau des architectures de transport. Par exemple, dans le contexte du transport unicast, le protocole TCP (*Transport Control Protocol* [107]) pose de nombreux problèmes de performances en environnement satellite. Plusieurs travaux recensent ces problèmes de fonctionnements dans le cadre des protocoles de transport unicast et proposent des améliorations spécifiques à l'environnement satellite. C'est le cas en particulier pour le standard TCP qui a fait l'objet des propositions [86],[82], [25] et [57]. Les avantages et les difficultés liés à l'utilisation d'un lien satellite pour un service de communication multipoint sont détaillés dans le chapitre 2. La confrontation des solutions présentées dans le chapitre 1 au contexte particulier du satellite permet ainsi de statuer sur l'adéquation des propositions existantes à ce contexte.

Une des possibilités existantes pour résoudre certains problèmes liés à l'utilisation d'un lien satellite consiste à utiliser un réseau hybride satellite / terrestre. Avec un tel réseau, les récepteurs sont connectés au système satellite et au réseau terrestre. Ce type d'architecture de communication permet par exemple de palier l'absence de voie retour, présente dans la plupart des systèmes satellites [66] [130]. Dans un système hybride de ce type, les communications peuvent transiter sur le réseau terrestre ou sur le réseau satellite. Il est donc naturel de s'interroger, sur la manière de déterminer le support le plus judicieux à utiliser, pour transmettre des données de manière fiable. Cela nous a conduits à envisager la prise en compte de critères de coût pour effectuer ce choix. Une fonction de coût est ainsi définie dans la section 2.3, le choix du moyen de communication étant réalisé de manière à minimiser cette fonction. Une première évaluation de cette approche, présentée dans la section 2.3.3, montre que celle-ci est particulièrement intéressante sous certaines conditions. Cependant, la mise en oeuvre d'un protocole de

transport intégrant la possibilité d'un choix pour le support de communication nécessite la définition de mécanismes spécifiques. Trois mécanismes essentiels à la réalisation de la proposition sont ainsi étudiés au cours du chapitre 3.

Ces mécanismes étant définis, il a été possible de décrire en détail dans le chapitre 4 une proposition de protocole de transport répondant à la proposition ; un des objectifs étant, à terme, de proposer un protocole de transport multipoint fiable en adéquation avec un environnement réseau hybride satellite / terrestre. Enfin, la description détaillée de la proposition a permis de produire une modélisation UML de celle-ci sous forme de diagrammes de classe et d'activité, l'objectif de cette modélisation étant de valider la proposition. La chaîne d'outils TTool-RTL a été utilisée dans ce but : celle-ci permet d'explorer l'espace d'état d'un système sous forme d'une analyse d'accessibilité, et d'en extraire des vues abstraites qui s'avèrent fort utiles pour caractériser le service rendu par une couche de protocole. Toutefois, des problèmes logiciels ont limité la portée des résultats obtenus. Seules les fonctionnalités de quelques mécanismes de niveau transport ont ainsi pu être validées. Ces derniers travaux ne sont donc que le commencement d'une phase de validation *a priori* de la proposition. Ils devraient être complétés par la suite de manière à aboutir à la conception d'une architecture de programmation valide.

Chapitre 1

Couche transport dans les communications multipoints

1.1 ARCHITECTURES PROPOSÉES POUR L'INTÉGRATION D'UN SERVICE DE COMMUNICATION MULTIPOINT

A la différence d'une communication point-à-point (qui consiste à émettre des données vers un seul récepteur), dans une communication multipoint, l'adresse destination fait référence à un groupe de récepteurs. Les données sont envoyées vers un ensemble d'hôtes. En observant l'architecture de communication traditionnellement utilisée dans les réseaux (figure 1.1), il apparaît que le support de la communication multipoint peut être mis en œuvre à différents niveaux de l'architecture.

La technique consistant à mettre en œuvre ce service au niveau de l'application (voir par exemple [102]) est souvent appelé Multicast applicatif. Plus précisément, cela correspond à intégrer dans l'application la gestion des groupes et la diffusion du contenu. L'application prend ainsi en charge toutes les opérations de contrôle du groupe (par exemple les opérations consistant à rallier et quitter un groupe), et délivre les messages à envoyer à chacun des membres. Une approche particulièrement intéressante dans cette catégorie consiste à établir un réseau logique au dessus d'une technologie réseau donnée. Ces réseaux sont souvent appelés réseaux de recouvrement (*overlay network*) [10] [64], car le réseau est ¡¡ superposé ¿¿ au réseau physique. Les réseaux de pair-à-pair [22] [126] [49] (ou *peer-to-peer*) sont un exemple de tels réseau, car les informations sont transmises d'utilisateur — ou pair —en utilisateur. Avec ce type de technique, les nœuds du réseau logique ont la charge de transférer les informations vers d'autres pairs qui en ont fait la demande.

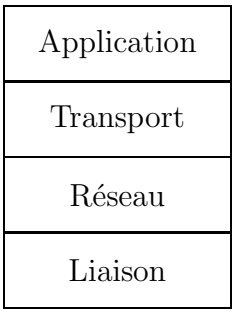


Figure 1.1 – Couches du modèle OSI traditionnellement mises en œuvre (la couche physique n'est pas représentée).

Les caractéristiques d'un système de communication de groupe pris en charge par la couche transport (e.g. [17]) sont relativement similaires à celles du système précédent : les groupes sont gérés par la couche de transport, et les messages pour chaque membre sont délivrés à la couche réseau. Cette solution présente cependant un avantage non négligeable : le service proposé par la couche transport est générique. Ainsi toute application désirant transmettre des informations à un groupe peut utiliser le service de transport correspondant (sans aucune autre implémentation).

Dans ces deux cas précédents, la communication peut se baser sur un service réseau point à point ou multipoint. La première solution, bien qu'actuellement employée sur Internet par le manque de disponibilité du service réseau multipoint, constitue rarement une utilisation optimisée de la bande passante du réseau. En effet, dans ce cas il faut envoyer autant de paquets de niveau réseau (soit des datagrammes avec IP) qu'il y a de récepteurs. Ainsi, lorsque les chemins permettant de joindre plusieurs récepteurs passent par un ou plusieurs liens identiques, la même information peut transiter plusieurs fois sur ces liens[1]. Il suffirait cependant de faire transiter un seul paquet par ces liens et de le répliquer lorsque les chemins divergent. C'est d'ailleurs ainsi que fonctionnent les communications multipoints lorsque le service est implémenté au niveau réseau : le service crée un arbre permettant de joindre la source et les récepteurs, de manière à ce qu'un seul paquet circule sur toutes les branches de cet arbre.

Le support des communications multipoints au niveau réseau présente donc en particulier l'avantage de diminuer la bande passante utilisée pour diffuser des informations au groupe. L'optimisation complète de la transmission est assurée lorsque la couche liaison de données assure également le support de communication de groupe. Évidemment, ce support est particulièrement efficace dans le contexte de supports à diffusion tels que les bus de données (e.g. Ethernet [61]), les liaisons radios ou satellites. Dans le contexte Internet, le support des communications multipoints au niveau réseau a été partiellement déployé (dans la mesure où ce n'est pas le cas dans tout le réseau) avec une extension au protocole IP classique appelée IP Multicast. IP Multicast offre un

[1]Pour pouvoir éviter cela, le réseau de recouvrement doit refléter la répartition des hôtes sur le réseau physique. Les paquets peuvent alors être répliqués de proche en proche sans impliquer une utilisation trop coûteuse du réseau. La problématique de ¡¡ convergence ¿¿ d'un réseau de recouvrement vers la répartition physique de ses membre reste cependant un problème non résolu. De plus cela n'assure pas que l'utilisation du réseau soit optimisée.

service de diffusion de groupe efficace et flexible. Il permet la gestion des membres des groupes et de l'émission d'informations en leur sein.

Néanmoins, le service offert par IP Multicast reste un service ¡¡ Au mieux ¿¿ (ou *Best Effort*) et s'appuie sur l'utilisation de protocoles de transport de bout en bout pour adapter la qualité de service du réseau aux besoins des applications. Ainsi, de très nombreuses propositions de protocoles ont été publiées. Citons parmi les plus connus MFTP [85], XTP [42], PGM [125] et RAMP [67]. La très grande majorité de ces propositions a cependant été développée dans le contexte des réseaux IP Multicast ¡¡ terrestre ¿¿. De plus ces solutions sont proposées en réponse à une situation spécifique, que ce soit en terme de qualité de service (par exemple pertes d'informations par le réseau rares et uniquement dues à des congestions du réseau), ou de topologie (grands groupes de récepteurs, petits groupes, etc.). La suite de ce document se situe dans ce contexte et considère exclusivement des protocoles de transport utilisant un service réseau IP Multicast.

Afin de proposer un protocole de transport multipoint en adéquation avec l'environnement satellite, il est nécessaire d'étudier en détail ces propositions. La transposition des architectures étudiées à l'environnement satellite permettra ensuite de détecter les dysfonctionnements éventuels, et d'y remédier. La partie suivante recense les problèmes techniques liés au déploiement d'un service de communication multipoint. Une fois définis les points durs liés à la conception de ce service, les différentes solutions adoptées par les propositions existantes sont analysées. Cette section se termine par l'état de la normalisation à l'IETF qui résume bien les tendances des travaux actuels dans le domaine des communications fiables multipoints.

1.2 LA PROBLÉMATIQUE DU TRANSPORT MULTIPOINT AVEC *IP MULTICAST*

La conception de protocoles liés aux communications multipoints demande de prendre en compte un certain nombre de problèmes spécifiques à la communication de groupe [30]. Parmi ces paramètres, les plus caractéristiques du niveau transport sont :

- la dynamique des groupes multicast ;
- la mise à l'échelle ;
- la sécurité dans la communication de groupe ;
- les services et la qualité de service du transport de données ;
- la prise en compte de liaisons spécifiques.

Dans la suite du document, nous appellerons, selon une définition normalisée, *session Multicast*[2] l'ensemble des opérations permettant de transmettre des informations vers un groupe. La session commence dès qu'elle est annoncée et se termine lorsque la source arrête la transmission des informations. Les paragraphes suivant exposent plus en détail les principes de base de chacun des problèmes énumérés plus haut.

[2]Le terme session Multicast désigne une source et un volume de données à transmettre. A cela s'ajoute l'adresse de classe D qui sera utilisée pour joindre le groupe et les récepteurs qui désirent recevoir les informations. Ce terme est à rapprocher de la définition du terme Session définie dans le standard SDP [53]. Cette distinction a été faite de manière à éviter toute confusion possible avec le terme Session défini par la terminologie OSI.

1.2.1 La dynamique des groupes *IP Multicast*

Dans le modèle IP Multicast, les groupes créés sont évolutifs au cours d'une session. Ainsi plusieurs récepteurs peuvent se joindre au groupe, ou le quitter pendant une session. Or, le fait que le groupe évolue au cours du temps implique que le protocole gérant le service multipoint (IP Multicast et le protocole de transport dans les architectures considérées dans ce document) puisse suivre l'évolution du groupe et évoluer en fonction de celle-ci. Par exemple certains protocoles — tel que RMP [87] — demandent des informations aux récepteurs (comme le débit maximum en réception afin de ne pas saturer de récepteur) pour pouvoir paramétrer la transmission. Si de nouveaux membres arrivent en cours de session, il faut donc ajuster les paramètres de la session en prenant en compte ces derniers.

1.2.2 Mise à l'échelle

La mise à l'échelle d'un protocole de transport multipoint recouvre globalement sa capacité à accroître la taille du groupe qu'il est en mesure de gérer sans influer sur ses performances. Et dès lors que l'application implique un grand nombre de sites, le problème de la mise en œuvre des mécanismes de niveau transport se pose. Ce problème est crucial dans la mesure où ces mécanismes permettent de supporter la qualité de service spécifiée par l'application.

Le premier aspect de la mise à l'échelle concerne la dissémination des informations à transmettre. Ceci peut être problématique, en particulier sur un réseau terrestre. En effet lorsque le nombre de récepteurs croît, le flux de données généré peut affecter une grande partie du réseau. Or les protocoles de transport multipoint ne doivent pas dégrader le fonctionnement du réseau sous-jacent. Ce problème implique la mise en œuvre d'un mécanisme de contrôle de congestion. Cela afin de garantir le bon fonctionnement du réseau sur tout le domaine qui sera utilisé par la transmission. Ce problème se pose en particulier avec les algorithmes de contrôle de congestion de bout en bout[3]. Notons que l'utilisation de lien satellite restreint ce problème. En effet la zone de couverture de ces liens étant très étendue, ils touchent un très grand nombre d'utilisateurs en utilisant un nombre très restreint d'équipements (le satellite et les routeurs du réseau d'accès). Il semble donc plus aisé de contrôler l'état du réseau dans un tel contexte.

Un autre aspect de la mise à l'échelle concerne la signalisation générée par le protocole. Le trafic de signalisation généré pour rendre à l'application le service désiré augmente en effet *a priori* avec la taille du groupe. Ce problème se pose en particulier pour le mécanisme de correction d'erreurs, car les récepteurs envoient généralement des messages pour signifier qu'ils ont (ou n'ont pas) reçu toutes les données. Ainsi si la taille du groupe augmente, deux problèmes se posent. Tout d'abord l'émetteur risque d'être saturé par les demandes de retransmission. Ensuite le réseau aux alentours de la source risque d'entrer en congestion. Ces deux problèmes sont référencés sous le terme de *Nack Implosion*.

[3]L'architecture de l'Internet est basée sur le report des fonctions de contrôle telles que le contrôle de congestion aux extrémités du réseau, concept illustré notamment par le protocole TCP.

Le problème de mise à l'échelle reste toujours un problème ouvert, et un certain nombre de solutions ont été proposées dans le but d'améliorer la possibilité de mise à l'échelle des protocoles [143], [73], [59], [58].

1.2.3 La sécurité des communications de groupe

Le problème de la sécurité sur Internet est un large problème toujours en cours d'étude. Le nombre de groupes de travail de l'IETF travaillant sur ce sujet [137] reflète la complexité du problème.

L'extension de ce problème à la communication multipoint est plus délicate encore, car le nombre d'interlocuteur dans la communication est plus important. De plus, les protocoles de transport étudiés dans ce document utilisent IP Multicast. Comme ce service ne permet pas de limiter le groupe à une liste de personnes connues, le contrôle des interlocuteurs est moins aisé que dans le contexte point à point. La mise en œuvre de gestion de groupe à un niveau supérieur (p. ex . applicatif ou transport) se révèle donc nécessaire. Ces services doivent également gérer les problèmes de cryptages et de partage de clefs par tous les membres du groupe. A l'heure actuelle, ce sont encore des problèmes ouverts, notamment pour l'échange des clefs. Ces problèmes sont en particulier étudiés au sein du groupe de travail MSEC de l'IETF [91]. Il existe également une autre approche consistant à intégrer des services plus complets au niveau de la couche réseau de manière à assurer la sécurité de ces communications [129].

1.2.4 Les services demandés aux protocoles de transport

Un certain nombre de choix effectués lors de la conception d'un protocole de transport dépendent d'une part des besoins des applications qui vont utiliser ce service, et d'autre part des contraintes et caractéristiques du réseau sous-jacent. Les besoins de l'application en termes de transport diffèrent par exemple si l'application diffuse de la vidéo ou si elle transmet des fichiers. Dans le cadre des transmissions point à point, on peut déjà recenser un protocole pour ces deux types de transmission : RTP/UDP et TCP. TCP propose un service totalement fiable et ordonné destiné à la transmission de fichiers, tandis que RTP (*Real Time Protocol* [122]) offre un service plus souple adapté à la diffusion de médias (audio, vidéos). D'autre part, les opérations de contrôle (contrôle de congestion, contrôle de débit, etc.) sont très dépendantes des caractéristiques du (ou des) réseau(x) traversé(s). L'adaptation de ces mécanismes dans le contexte multipoint constitue un large problème dû à la multiplicité des récepteurs et à la gestion déjà délicate du trafic de contrôle (cf. 1.2.2). Ce problème de disparité des besoins applicatifs et des conditions d'utilisation du protocole de transport est référencé sous le terme de *no one-size-fits-all*. Le développement d'un protocole de transport universel qui pourrait répondre aux besoins de toutes les applications reste en effet aujourd'hui du domaine de la recherche.

1.3 SERVICES DE COMMUNICATION ET PROTOCOLES DE TRANSPORT MULTIPOINTS

Située entre l'application et la couche réseau, la couche transport a un rôle essentiel dans l'architecture en couches. Elle a le rôle d'offrir le service de communication adéquat directement aux processus applicatifs exécutés sur les machines distribuées. Un de ses challenges est de s'adapter aux caractéristiques du réseau afin d'obtenir la qualité de service de transport spécifiée par l'application. Nous étudions dans cette section les concepts clés des architectures de transport proposant une communication multipoint.

Il existe de nombreux paramètres influençant la conception d'un protocole de transport multipoint [60], [47]. En premier lieu, les services demandés au protocole de transport sont dépendants des besoins des applications cibles. Ainsi les choix faits sur la conception du protocole de transport dépendent avant tout du (ou des) service(s) qu'il doit rendre. Par exemple, proposer ou non une fiabilité totale ou un contrôle de l'ordre de délivrance des paquets influe notablement sur l'architecture du protocole. Le protocole de transport est également tributaire de la qualité de service rendue par le réseau. Or le réseau Internet classique offre un service ¡¡ Au mieux ¿¿. Ainsi actuellement ni fiabilité, ni ordre, ni aucune gestion de qualité de service ne sont proposés au niveau de l'interface de service réseau. La gestion de critères de qualité de service a cependant déjà fait l'objet d'études au travers de projets comme Intserv [62] et Diffserv [27]. Toutefois de nombreux problèmes restent non résolus, et de tels services ne sont donc pas encore mis en œuvre. Avec un service réseau au mieux, le protocole de transport est une couche d'adaptation du service réseau aux besoins de l'application (voir figure 1.2).

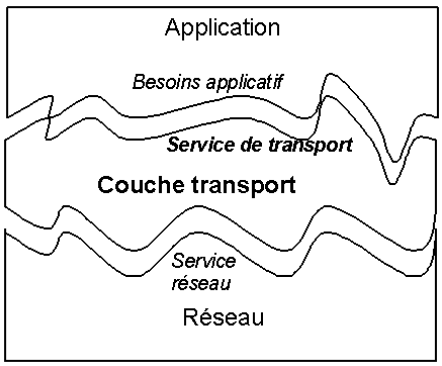


Figure 1.2 – La couche transport, adaptation du réseau à l'application

De plus, de par les principes de conception du réseau Internet, certaines fonctions indispensables à son fonctionnement —comme le contrôle de congestion— sont reléguées au niveau des stations hôtes, et généralement dans la couche transport (par exemple dans TCP). Il est ainsi nécessaire d'assurer ces fonctions dans les protocoles de transport pour assurer un fonctionnement convenable du réseau IP. En outre dans le cas d'un protocole de transport multipoint, le service IP Multicast n'offre aucune garantie de qualité de service. Le protocole de transport peut donc, selon les besoins de l'utilisateur, avoir à assurer la fiabilité de la communication. D'autres services, tels

que la gestion de l'ordre ou la gestion du temps peuvent être également du ressort du protocole de transport.

1.3.1 Multiplexage/démultiplexage

Le démultiplexage assure au niveau d'un récepteur l'aiguillage des unités de données véhiculées par le réseau vers le processus applicatif destinataire. Au niveau de l'émetteur, le travail de regroupement des données des processus applicatifs pour les soumettre au service réseau caractérise le multiplexage. Ces services de base sont assurés dans TCP et UDP par le biais du numéro de port.

1.3.2 Fiabilité

Ce service vise à contrôler la perte des unités d'information véhiculées lors de la communication. Le protocole de transport a traditionnellement pour mission d'assurer une fiabilité totale (cf. le service de TCP) ou nulle (cf. le service de UDP [105]). Un service de fiabilité totale indique que l'application dispose d'un canal de communication sans perte, ni erreur de transmission. Le protocole de transport doit donc implémenter dans ce cas un mécanisme permettant de reprendre les pertes ou erreurs éventuelles. Plus généralement, il est possible de définir une fiabilité partielle [119]. Dans ce cas, le niveau de fiabilité est différent selon les unités de données véhiculées par le protocole de transport, le temps écoulé, etc.

1.3.3 Ordre

Ce service concerne le contrôle de l'ordre de délivrance des paquets à l'application au niveau des récepteurs. L'ordre de délivrance des paquets est généralement lié à leur ordre de soumission au service de transport de l'émetteur. Il s'agit traditionnellement d'un service d'ordre total (même ordre de délivrance qu'à l'émission) ou nul (pas de relation entre la délivrance et l'émission). Cependant, de même que pour le service de fiabilité, il est possible de définir un service d'ordre partiel [119]. Ce service est alors basé sur une relation plus ou moins complexe entre la soumission et la délivrance des unités de données au service de transport. Le relâchement de contraintes d'ordre sur certaines des unités d'échange du protocole de transport (en fonction des besoins applicatifs) permet notamment d'améliorer la communication de bout en bout. Pour des communications de groupe, l'application peut également spécifier des contraintes inter-récepteur sur l'ordre de délivrance des données reçues par les couches transport des stations distribuées (e.g. synchronisation de la délivrance des paquets).

1.3.4 Gestion des groupes

Ce service consiste à gérer l'appartenance des utilisateurs à un groupe. On cite notamment les fonctions d'ajout et de retrait à un groupe de diffusion.

1.3.5 Gestion du temps

Ce service a pour but d'assurer un contrôle du délai de bout en bout et de sa variation (gigue de délai). Il est particulièrement utile lorsque les communications intègrent des médias à contraintes temporelles tels que la vidéo, le son, etc. A l'heure actuelle, avec l'architecture en couche classique, ce service est souvent relégué à des couches d'un niveau supérieur. On note cependant que le service du réseau IP classique ne peut assurer aucune garantie de qualité de service en général, ni de temps en particulier. Des extensions aux services IP (telle que diffserv [27], intserv [62], etc.) doivent être considérées pour pouvoir, au niveau d'un service de transport, garantir que ce type de contrainte sera respectée.

1.3.6 Contrôle de flux

Ce service vise à assurer que l'entité applicative émettrice ne sature pas le ou les récepteurs, en émettant trop d'unités de données, trop rapidement. Le contrôle de flux consiste à modifier le débit en émission du flux en fonction de la capacité de traitement des récepteurs, afin que les récepteurs ne soient pas surchargés par celui-ci.

1.3.7 Contrôle sur le réseau

Dans le cadre du réseau Internet, le protocole de transport doit également assurer des services contrôlant l'utilisation du réseau. C'est le cas du contrôle de congestion qui limite le trafic émis par une station pour prévenir ou diminuer la saturation du réseau (i.e. des routeurs IP du chemin suivi par le flux de paquets). Le fonctionnement global du réseau dépend de ce service. Ce risque est encore plus présent pour des communications multipoints, car les informations émanant d'une source se propagent vers plusieurs utilisateurs. Ces données ont donc généralement plus de poids sur l'utilisation du réseau (un datagramme émis peut affecter beaucoup plus de routeurs).

Les mécanismes de contrôle de congestion consistent donc à évaluer ¡¡ l'état ¿¿ du réseau, et à modifier les caractéristiques du trafic injecté en fonction de cet état. Les modifications sont effectuées en termes de débit et/ou en termes de nombres de messages présents dans le réseau. Le service de contrôle de congestion est généralement assuré par le protocole de transport (p.ex., TCP dans le cadre de la communication point à point). Cependant, il peut être implanté au niveau applicatif lorsque le protocole de transport n'intègre pas directement cette fonctionnalité. C'est le cas par exemple des applications utilisant le protocole UDP.

1.4 PRINCIPAUX MÉCANISMES ASSOCIÉS AUX SERVICES DE TRANSPORT

L'objectif de ce paragraphe est d'introduire les principaux mécanismes développés pour rendre les services décrits dans la section précédente. Un grand nombre de protocoles de transport multipoint ont servi de base à cette étude. La liste des protocoles étudiés n'est pas exhaustive mais constitue un ensemble représentatif des propositions

actuelles. Les protocoles présentés sont regroupés en différentes catégories selon l'optique dans laquelle ils ont été développés. Trois catégories se dégagent en effet des différents protocoles étudiés :

- Les protocoles développés pour rendre un service de communication multipoint générique (i.e. sans prendre en compte de contraintes applicatives particulières). Les protocoles MTP [4], RMP [87], XTP [42], STORM [140], LRMP [74], RMDP [81], [116] font partie de cette catégorie. Il faut également mentionner les protocoles RBP, LGMP et URGC cités dans [98] et [68].
- Les protocoles développés pour des applications interactives multipoints. Cette catégorie inclue les protocoles SRM [40], LBRM [60], RAMP [67], TRM [6], SRRTP [74], PGM [125], POC [119], et ECSRM [47].
- Les protocoles développés pour la dissémination de données, comme par exemple MDP [1], AFDP [19], TMTP [140], RMTP [121] MFTP [85], TRAM [142], RRMP [143], BMTP [88], et OTERS [73].

Le grand nombre de propositions dans ce domaine rend compte du problème de *no one-size-fits-all* exposé dans la section 1.2.4. Les paragraphes suivants présentent les mécanismes utilisés par ces protocoles dans le but de rendre les services voulus.

1.4.1 Gestion de la fiabilité

Le service de fiabilité est assuré par plusieurs mécanismes complémentaires. Les mécanismes associés à la gestion de la fiabilité sont en premier lieu orientés par le niveau de fiabilité (fiabilité totale, nulle ou partielle, voir 1.3.2) proposé par le protocole. Par exemple, dans le cas de la fiabilité partielle le protocole est généralement amené à limiter le nombre de requête de retransmission, à reprendre les pertes seulement pendant un intervalle de temps donné, ou à réserver les reprises d'erreurs pour des messages jugés plus importants que d'autres. Cette dernière proposition a été étudiée par exemple pour des flux vidéos utilisant des codages différentiels d'une image à une autre (p.ex. MPEG [90]) : l'utilisation d'une fiabilité partielle du flux en favorisant les images de référence paraît avantageuse dans ce cas [119].

1.4.1.1 Mécanisme de requête de retransmission

Le premier mécanisme associé à la gestion de la fiabilité est le mécanisme gérant les requêtes de retransmission. Ce mécanisme spécifie la manière dont les récepteurs doivent envoyer les requêtes de retransmission, ainsi que le destinataire de cette requête. Pour ce mécanisme, il existe deux alternatives majeures [133].

La première possibilité correspond au cas où les requêtes sont implicites. Cette possibilité est proposée par exemple dans les protocoles MFTP et XTP. Dans ce cas, les récepteurs envoient des acquittements positifs (ou ACK pour *ACKnowledgement*) au responsable des reprises d'erreur. Ces messages contiennent les numéros de séquence des paquets reçus. Ainsi, lorsque le responsable de la reprise d'erreur détecte un numéro de séquence manquant, il renvoie le message correspondant. On dit alors que le protocole

est *orienté émetteur* dans la mesure où c'est un responsable qui détecte les pertes et qui envoie ensuite les données aux récepteurs concernés.

A l'inverse, pour un protocole *orienté récepteur* chaque récepteur a la charge de détecter les erreurs, et de demander une retransmission de données lorsqu'il le juge nécessaire. Ainsi, un récepteur qui détecte une perte en remarquant un saut dans les numéros de séquence des messages reçus va par exemple envoyer un acquittement négatif (Negative ACKnoledgement : NACK) à qui de droit pour signaler la perte. Cette approche permet, pour des transmissions ayant des taux d'erreur de transmission faibles, de limiter la bande passante utilisée sur la voie retour. Ces requêtes peuvent être faites :

- En mode point à point : la requête est dirigée vers une entité précise (AFDP [19]).
- En mode multipoint : tout le groupe va recevoir la requête (LRMP [74]).
- Vers un serveur local (i.e. un responsable d'un sous-groupe), comme dans le protocole LGMP [98].

Le choix d'un mécanisme dépend grandement de la topologie du réseau sous-jacent, ainsi que du service ciblé par le protocole de transport. En outre, le choix d'un mécanisme adéquat, peut permettre d'améliorer les performances du protocole qui l'utilise [70].

1.4.1.2 Mécanisme de retransmission

Le second mécanisme concerne la retransmission des données manquantes. Ce mécanisme définit en particulier quel type de transmission (point à point ou multipoint) sera utilisé. Il peut en effet être judicieux d'adapter la retransmission au contexte [70]. Par exemple, dans le cadre d'une communication multipoint, si un grand nombre d'utilisateurs n'a pas reçu l'information, il est préférable de retransmettre cette information à tout le groupe.

En particulier la connaissance des propriétés du réseau sous-jacent permet parfois des optimisations importantes. Il est en effet utile de savoir si une perte détectée ne concerne qu'une personne, ou bien si compte tenu du support utilisé, tout un groupe de personnes n'a pas reçu l'information. Par exemple, une erreur de transmission entraînant la perte d'un datagramme IP Multicast au niveau d'un lien montant d'un lien satellite doit être considérée comme une perte pour l'ensemble des membres du groupe. Inversement, la perte d'un datagramme sur un lien permettant de joindre un récepteur situé en feuille d'arbre de diffusion, ne concerne *a priori* qu'un seul destinataire.

Selon la manière dont les requêtes de retransmission sont effectuées, plusieurs possibilités se présentent pour les retransmissions. En effet si les demandes de réémission sont faites de manière point à point vers la source, seule la source pourra opérer la retransmission. Cependant, si la demande de retransmission est faite de manière multipoint, le message pourra alors être retransmis par un des membres du groupe [70]. En particulier le membre le plus proche du récepteur (en termes de proximité réseau) émettant la demande peut retransmettre le message, améliorant ainsi le temps de réponse du système. Dans ce but, RMP, SRM, SRRTP et TRM ont adopté cette approche.

Une autre possibilité consiste à organiser le groupe en sous-groupes [59], [58]. Pour

chaque sous-groupe, un membre est nommé responsable. Ainsi les retransmissions seront effectuées par le responsable du groupe, s'il a bien reçu les messages, sinon il effectuera lui-même une requête de retransmission vers une autre entité. Cette solution a été notamment adoptée par les protocoles LGMP, OTER, STORM, TMTP et TRAM.

1.4.1.3 Mécanisme de fiabilisation

Le dernier élément constitutif d'un service de fiabilisation correspond au mécanisme utilisé pour augmenter la fiabilité de la transmission (idéalement jusqu'à obtenir un taux de perte nul). Ce mécanisme spécifie les informations qui seront retransmises en cas de perte.

Une première possibilité pour assurer la fiabilisation consiste à insérer dans le flux de transport des messages redondants. Cette technique est référencée sous le terme de *Forward Error Correction* (FEC). En pratique ce sont des messages générés au moyen d'un codage approprié de l'information à transmettre [95], [11]. Ceux-ci permettent de générer T' messages à partir de T (avec $T' \geq T$). Il suffit ensuite de recevoir T messages (ou $T(1+\epsilon)$ messages, avec $\epsilon << 1$, selon le code utilisé) parmi les T' pour pouvoir reconstruire la totalité des données. Ainsi chaque récepteur peut subir jusqu'à $T' - T$ pertes sans que cela soit problématique.

Dans le cas d'un mécanisme basé sur des retransmissions, une solution consiste à renvoyer simplement le message manquant. Une possibilité plus intéressante consiste à envoyer un message encodé [95]. L'utilisation de FEC dans ce contexte présente l'avantage de pouvoir coder la reprise de plusieurs pertes différentes en un seul message. Par exemple, considérons le cas où x récepteurs ont reçu $T-1$ paquets, chacun ayant perdu un paquet différent. Sans utilisation de codage, il faudrait retransmettre x paquets. L'utilisation d'un code permet de reprendre ces pertes avec un unique paquet encodé, car après réception d'un paquet, tous les récepteurs ont reçu T paquets. L'utilisation de FEC permet donc de diminuer le nombre de messages retransmis, et en conséquence le temps de transfert des informations. Cette solution, adoptée par les protocoles ECSRM, MDP, PGM, RMDP, RMTP II, est généralement appelée *hybrid ARQ type 2* (noté HARQ2 dans la suite de ce manuscrit).

L'*hybrid ARQ type 1* peut être vu comme un mélange des techniques exposées dans les deux paragraphes précédents : une couche de codage est insérée entre le protocole de transport et la couche réseau. Dans ce cas le codage est transparent pour les entités de niveau transport : la source et les récepteurs échangent normalement des messages. Par contre la sous-couche de codage code et décode tous les messages envoyés par le protocole de transport. La fiabilité de toutes les communications est donc améliorée.

1.4.2 Gestion de l'ordre

Les garanties d'ordre concernent l'ordre de la délivrance des messages à l'application réceptrice. On peut distinguer plusieurs niveaux d'ordre dans les services offerts par les protocoles de transport multipoint.

Tout d'abord certains protocoles comme MTP, RBP et URGC proposent un ordre

total sur l'ensemble des émissions au sein du groupe. En d'autres termes ces protocoles vont synchroniser les émissions des différentes sources pour que les messages émis par les sources soient délivrés dans le même ordre pour tous les récepteurs [98]. Cet ordre total est obtenu grâce à une entité centrale qui va distribuer le temps de parole entre les différentes sources conformément à l'ordre désiré.

D'autres protocoles permettent de réordonner les paquets pour les transmettre à la couche supérieure dans l'ordre où ils ont été émis. De manière classique — comme avec TCP — cela est rendu possible en associant un numéro de séquence aux trames émises. C'est le cas par exemple de AFDP, BMTP, ECSRM, LGMP, LRMP, MDP, OTERS, PGM, RAMP, RMDP, RMTP, RRMP, SRRTP, TRAM, TRM et XTP

1.4.3 Gestion des groupes

De manière générale, ce service est rarement implémenté. Une grande majorité de protocoles n'ont en effet pas besoin d'avoir une connaissance explicite du groupe. Ils s'appuient alors directement sur le service réseau multipoint sous-jacent : les informations sont envoyées vers un point de communication de groupe au niveau réseau (une adresse IP de classe D avec IP Multicast). La gestion du groupe est alors totalement transparente pour le protocole de transport.

Il existe cependant des protocoles proposant une gestion explicite des groupes. Cela est utile pour des raisons de sécurité par exemple (pour restreindre l'accès au groupe), ou plus généralement dans l'objectif de rendre un service de gestion de groupe différent de celui proposé par le service réseau. Dans ce cas le protocole prend en charge les opérations d'abonnement au groupe et de départ. Il est alors nécessaire que tout utilisateur désireux de devenir membre du groupe fasse une demande à un responsable. Ce dernier est ensuite libre d'accéder à cette requête ou non [98]. Notons que l'implémentation de ce genre de service paraît plus efficace au niveau de la couche réseau. Dans ce cas, la gestion du groupe est entièrement prise en compte par la couche réseau. Cette alternative est par exemple étudiée au sein du projet MCOP/TUT [129] [128].

Ces mécanismes de gestion des groupes sont répartis en deux catégories : ils sont soit centralisés, soit distribués. Pour des mécanismes centralisés, comme pour les protocoles AFDP, MFTP, MTP, RAMP, RBP, RMP, RRMP, RTP, URGC et XTP il existe un membre particulier du groupe qui connaît tout le groupe et qui prend en charge les opérations de connexion et de déconnexion au groupe. Pour des protocoles avec une gestion distribuée des groupes, les opérations de connexion au groupe sont prises en charge par des membres particuliers. Chacun gère un sous-groupe local et ne connaît de manière précise que ce sous-groupe.

1.4.4 Gestion du temps

Les mécanismes associés à la gestion de critères temporels sont également optionnels. Ils ne sont en effet utiles que pour des protocoles prenant en charge des applications ayant des contraintes de délai de bout en bout et de gigue (e.g. applications multimédias telles que les vidéoconférences). Ce type de mécanismes est très spécifique et demande

d'avoir dans le système des primitives très ciblées pour pouvoir obtenir la qualité de service recherchée. Ils sont en général, implantés dans l'application, et utilisent souvent un protocole de transport adapté (tel que RTP).

Il existe plusieurs niveaux de gestion du temps. Le simple fait d'estampiller les messages rend déjà un service aux mécanismes applicatifs multimédias (RTP [122]). Coupler la retransmission avec une validité temporelle d'un message permet de rendre plus efficace la gestion de la fiabilité. Dans ce cas les pertes ne seront reprises que pendant un certain intervalle de temps. L'approche développée dans [119] vise à augmenter le pouvoir de configuration du protocole de transport. Cela afin d'éviter de développer des mécanismes nouveaux pour chaque nouveau besoin identifié. Les contraintes temporelles sont ici décrites à l'aide de réseau de Petri temporisés (TSPN : Time Stream Petri Net). A partir de cette description le protocole est capable d'agencer et de configurer divers mécanismes internes pour rendre le service voulu.

1.4.5 Contrôle de congestion

C'est une partie très importante du protocole de transport multipoint dans le contexte d'Internet. Le contrôle de congestion vise à éviter la saturation du réseau en contrôlant la charge introduite dans le réseau. Le problème consiste d'une part à évaluer l'état du réseau, et d'autre part à modifier, si besoin est, le flux injecté dans le réseau en fonction de l'état de celui-ci.

1.4.5.1 Contrôle de congestion sur la voie aller

Le mécanisme le plus courant — utilisé par TCP — consiste à utiliser des rapports de réception émis par le récepteur (dans le cas de TCP ce sont des acquittements) : la détection de pertes suivant l'analyse de ces messages est alors associée à une congestion au sein du réseau. Cette hypothèse est en général vérifiée dans les réseaux Internet filaires. Elle peut par contre constituer un problème pour les réseaux où les erreurs de transmissions ne sont pas marginales (comme les liens sans fils). Pour ces liaisons, il existe des mécanismes qui tentent d'analyser les causes des pertes de messages (non réception ou erreur de transmission). Ceci permet de ne pas diminuer inutilement les performances de transfert lors du passage par ces liaisons [5]. L'adaptation de ce mécanisme à des communications multipoints a été étudiée dans le groupe de travail *Reliable Multicast Transport* (RMT). Cela a conduit à proposer les mécanismes appelés *TCP-Friendly Multicast Congestion Control* (TFMCC [139]) et PGMCC [115]. PGMCC est un mécanisme utilisant un membre représentatif de tout le groupe : le flux global est ajusté en fonction des réponses de cet hôte particulier (celui-ci peut changer au cours de la transmission). Pour TFMCC, le taux de transfert est adapté au récepteur subissant les pires conditions de réception.

La limitation des mécanismes précédents est évidente : lorsqu'un seul lien de l'arbre de diffusion est en congestion, la diminution du débit d'émission va résoudre le problème, mais en pénalisant tous les récepteurs du groupe. En réponse à ce problème, le RMT a proposé un contrôle de congestion ¡¡ en couches ¿¿ [78]. La transmission utilise dans ce cas plusieurs flux de données numérotés, chacun étant envoyé sur une adresse de classe D différente. Grâce à un codage adéquat des données transmises dans

chaque flux, les différents flux sont complémentaires : le flux 0 permet de recevoir les données à faible débit, la réception des flux 0 et 1 permet de recevoir plus rapidement les données (puisque les débits des deux flux s'ajoutent), de même pour la réception des flux 0, 1 et 2, et ainsi de suite. Avec une telle transmission les récepteurs qui détectent une congestion se désabonnent du flux ayant le numéro le plus élevé. Les débits des flux envoyés sont choisis pour reproduire l'action de TCP lorsqu'il détecte une congestion : le débit global de réception est divisé par deux.

1.4.5.2 Contrôle de congestion sur la voie retour

Comme cela a été dit plus haut, ce contrôle peut avoir une grande importance pour les performances du protocole. En effet, si la voie de retour entre en congestion parce que le récepteur envoie trop d'acquittements à l'émetteur, ce dernier va supposer que le réseau est entré en congestion. Il va donc diminuer le débit d'envoi des messages. Dans ce cas, l'émetteur aurait très bien pu continuer à envoyer des messages au même débit, c'est le récepteur qui aurait dû diminuer le taux d'envoi des acquittements. Il existe deux niveaux à ce contrôle d'utilisation de la bande passante de la voie retour.

Contrôle du nombre de messages envoyés par chaque récepteur

Une solution souvent utilisée pour diminuer le nombre de messages envoyés est de rassembler un ensemble de messages rendant compte de l'état du récepteur. Ces messages peuvent être envoyés périodiquement, ou en réponse à une demande de la source. Ces messages sont souvent appelés acquittements sélectifs (SACK), car ils correspondent à plusieurs acquittements concaténés. Avec cette technique, dans la mesure où la périodicité de l'envoi des messages est connue, le trafic généré sur la voie retour est partiellement contrôlé.

Certains protocoles se contentent d'utiliser des demandes de retransmission (ou NACK pour Negative ACKnoledgement). Ainsi, lorsque les pertes sur le canal de communication sont rares, peu de messages sont générés sur la voie retour. Cette hypothèse a été retenue entre autres par les protocoles AFDP, ECSRM, LGMP, LRMP, MDP, MTP, Muse, OTERS, PGM, RAMP, RMDP, RRMP, SRM, SRRTP, STORM, TRM et URGC.

Il est possible de coupler les deux approches précédentes. Cette approche est notée ACK/NACK : les récepteurs envoient périodiquement des messages donnant leur état de réception et envoie des NACK pour demander une retransmission. L'utilisation couplée de ACK et de NACK — mise en œuvre dans BMTP, LBRM, MFTP, RBP, RMP, TRAM et XTP — permet d'améliorer la fiabilité du protocole : si la source ne reçoit pas de NACK d'un récepteur, elle ne reçoit pas non plus les ACK de ce récepteur, et détecte donc un problème de communication.

Réduction du nombre total de messages envoyés par le groupe

Ce mécanisme s'attache à limiter le nombre de messages envoyés par la totalité du groupe vers le responsable des réémissions. Cela est nécessaire pour assurer la possibilité d'utilisation à grande échelle du protocole.

Dans le cas où des requêtes de retransmission (NACK) sont utilisées, une solution correspond à faire ¡¡ écouter ¿¿ les demandes de retransmission à tout le groupe (cela suppose que la retransmission, ainsi que la requête, soient envoyées à tout le groupe). Dans ce cas, lorsqu'un membre du groupe détecte une requête sur un paquet qu'il n'a pas reçu non plus, il n'envoie pas de nouvelle requête sur ce paquet. Pour que cette technique soit vraiment efficace, il est nécessaire d'espacer suffisamment les différentes requêtes. Le mécanisme de *NACK suppression* [93] a été conçu dans ce but. Il consiste à générer aléatoirement un temps d'attente lorsqu'une perte est détectée (selon une fonction de répartition prédéfinie). Ainsi si un récepteur décèle une requête sur la perte qu'il a détectée avant que son timer expire, il le réinitialise et annule l'envoi de sa requête. Sinon, il envoie la requête. Ce mécanisme est détaillé dans la section 3.3 car il est apparu comme le plus efficaces parmi ceux existants : il s'adapte très bien a toute taille de groupe, et réduit considérablement tout risque d'implosion des retours.

Une autre solution consiste à mettre en place une structure logique de manière à limiter le trafic généré sur la voie retour [59], [58]. Cela est généralement mis en œuvre en organisant les membres du groupe selon un arbre. Les nœuds de l'arbre sont alors responsables de leurs enfants directs : ils sont chargés de répondre à leurs requêtes, et le cas échéant de récupérer et d'agréger leurs requêtes pour les transmettre au niveau supérieur. L'utilisation d'une telle structure a été mise en place dans RMTP, STORM, LGMP, OTERS, et TMTP.

Une troisième solution est implémentée dans NTE [54] et BMTP. Dans ces protocoles, la source demande aux récepteurs de lui, le cas échéant, signaler les paquets qu'ils ont perdus. A chaque message de la source est associée une probabilité de réponse. Si après un temps prédéfini la source ne reçoit aucune réponse, elle renvoie un message avec une probabilité plus forte. Ce processus se répète jusqu'à ce que la source reçoive une réponse, ou jusqu'à ce que la probabilité passe à 1 (dans ce cas la source est assurée que tout le groupe a reçu l'information).

1.4.6 Contrôle de flux

Le contrôle de flux vise à éviter aux stations d'être saturées. La problématique est donc similaire au contrôle de congestion, si ce n'est que ce mécanisme n'est pas essentiel au fonctionnement du réseau. Cependant, le fait que des données correctement acheminées par le réseau soient perdues au niveau d'un récepteur, implique une retransmission ultérieure de ces données. La transmission utilise donc inutilement les ressources du réseau, ce qui a un impact néfaste sur le réseau. Il reste cependant un problème fondamental pour éviter que les stations ne perdent des données correctement transmises par le réseau. De même que pour le contrôle de congestion le problème de la multiplicité des capacités de réception des stations se pose : si un seul des destinataires du groupe est saturé, faut-il nécessairement diminuer le taux de transfert pour tous ? Il est donc possible d'altérer ou non les performances pour tout le groupe (comme le propose XTP). Notons que les deux approches développées pour le contrôle de congestion aboutissent également à un contrôle de flux. Avec TFMCC, si un récepteur est saturé, il va perdre des messages et la source va ajuster son débit d'émission en fonction de sa capacité (dans ce cas, tous les récepteurs sont pénalisés par ce récepteur). Avec

WEBRC, le contrôle de flux est implicite : tous les récepteurs saturés se désabonnent d'un certain nombre de flux jusqu'à ce qu'ils soient en mesure de recevoir correctement la communication.

Une autre approche consiste à se baser sur le taux minimum de réception de tous les membres du groupe, et à le comparer à un seuil minimum de taux d'envoi. Si le taux minimum des membres du groupe est inférieur au seuil, la source va éliminer du groupe les membres dont le taux de réception est trop faible. C'est le mécanisme utilisé par exemple par le protocole TRAM.

1.4.7 Conclusion

Un très grand nombre de propositions de protocoles de transport multipoint a donc été développé dans le cadre d'Internet multipoint. Ces différentes propositions s'appuient sur un ensemble de mécanismes généralement conçus dans le contexte précis de la proposition. La multiplicité de ces travaux amène à constater le manque de standardisation dans ce domaine à l'heure actuelle. C'est pour palier à ce manque que le groupe de travail *Reliable Multicast Transport* a été créé à l'IETF.

1.5 ÉTAT DE LA NORMALISATION À L'IETF : LE RMT-WG

L'objectif du groupe de travail *Reliable Multicast Transport* est de proposer un standard concernant les protocoles de transport multipoint sur Internet. Pour le moment ce groupe s'intéresse aux protocoles de transport fiables conçus pour diffuser des données d'une source vers un groupe de récepteurs (communication *one-to-many*).

Comme il existe de nombreuses applications demandant un tel service, et que leurs besoins sont relativement variés, le RMT étudie la standardisation de deux familles de protocoles :

- Un protocole basé sur des acquittements négatifs (NACK Based protocol) ;
- Un protocole utilisant un code correcteur d'erreur (Asynchronous Layered Coding protocol that uses Forward Error Correction).

Au départ trois approches étaient proposées. Aux deux protocoles cités précédemment s'ajoutait un troisième : TRACK (*Tree-Based ACK protocol*). L'objectif de ce protocole était d'organiser le groupe selon un arbre hiérarchique pour les raisons évoquées dans la section 1.4.5.2. Cependant par manque de résultats, cette orientation a été abandonnée. Pour standardiser ces protocoles, le RMT publie — au fur et à mesure de l'avancée des travaux — plusieurs RFC spécifiant :

- **Des *Building Blocks*** : ce sont des composants qui sont communs à de nombreux protocoles. En particulier les interfaces permettant l'accès aux méthodes des *Building Blocks* ainsi que leurs arguments sont définis.
- **La mise en œuvre des protocoles** : ceci spécifie le minimum de fonctions supplémentaires à mettre en œuvre pour réaliser un protocole à partir des *Building Blocks*.

– **L'*Application Programming Interface* (API)** : l'interface de programmation utilisable depuis l'application.

Les Building Blocks sont décrits dans la RFC 3048 [138], tandis que les différentes possibilités de conception pour des protocoles de transport multicast fiables répondant aux besoins des applications sont décrites dans la RFC 2887 [52]. Les paragraphes suivants exposent les deux familles de protocoles citées plus haut, en reprenant brièvement l'explication des mécanismes qu'ils utilisent.

1.5.1 *Nack-Oriented Reliable Multicast* : le protocole NORM

Ce protocole utilise l'approche par NACK (Negative Acknowledgement) décrite dans la section 1.4.5.2. Ainsi plutôt que de d'attendre régulièrement des messages signalant le dernier numéro de séquence reçu, la source suppose implicitement que les données qu'elle a émises sont reçues, tant qu'elle ne reçoit pas de message retour. Sur détection d'une rupture de séquence, c'est le récepteur qui indique qu'il n'a pas reçu certains paquets. Afin d'éviter le problème de *Nack Implosion* au cas où un grand nombre de récepteurs subirait des pertes, ce protocole impose l'implémentation du mécanisme de *Nack Suppression* décrit dans la section 1.4.5.2. Cette approche est associée aux mécanismes PGMCC et TFMCC pour le contrôle de congestion.

1.5.2 *Asynchronous Layered Coding* : protocole ALC

Cette proposition étudie une voie radicalement différente de l'utilisation de requêtes : ALC propose l'utilisation de *Forward Error Correction*. Ici, la fiabilité apportée est statistique. En protégeant au maximum la transmission de données, on réduit les risques de pertes de l'information. On peut également en plus retransmettre l'information en plusieurs passes pour permettre à des membres abonnés tardivement de récupérer toutes les données (prise en charge de *late-joining*).

La complexité est augmentée, du fait du codage/décodage additionnel requis, et un paramétrage adéquat (le taux de codage, la longueur des messages ...) doit être utilisé. Ce protocole est associé à l'implémentation du mécanisme de contrôle de congestion WEBRC. L'avantage par rapport à la proposition précédente est que ce mécanisme est déjà publié comme RFC, ce qui a permis de publier ALC comme protocole expérimental [77].

1.5.3 Conclusion sur la normalisation à l'IETF

Le groupe de travail de l'IETF donne une bonne idée des axes de recherche explorés actuellement. En particulier il exprime une volonté de standardiser plusieurs approches pour mettre en œuvre un service de communication multipoint fiable. De plus il confirme le fait qu'il semble difficile de concevoir un protocole de transport multicast répondant à tous les besoins, et dans toutes les situations. Pour cela il propose deux grandes familles de protocoles dont les caractéristiques sont bien distinctes en termes de gestion du groupe, de fiabilité, ou d'utilisation du réseau.

Cet effort de standardisation a porté ses fruits dans la mesure où plusieurs protocoles ont été développés ou modifiés de manière à être conformes aux consignes du

RMT. C'est le cas des protocoles MDP, LGMP, TCP-XM [131], FLUTE [101], de l'implémentation de NORM par le *Naval Research Laboratory*[97], et de la librairie MCL développée à l'INRIA [118]. Cela a même abouti à la proposition d'un module de communication multipoint parmi l'ensemble des modules java existant : c'est ce que propose le service JRMS (*Java Reliable Multicast Service* [65]) basé sur la proposition de protocole TRAM.

1.6 CONCLUSION : DE TRÈS NOMBREUSES PROPOSITIONS

Un grand nombre de protocoles de transport multipoint a donc été proposé au cours de ces dernières années. La multiplicité des approches adoptées a poussé l'IETF à définir un groupe de travail afin de standardiser un certain nombre de ces approches. Il faut remarquer cependant que cet effort de standardisation ne prend aucunement en compte la traversée de liaisons spécifiques. Or nous avons vu dans l'introduction que l'utilisation d'un satellite paraissait très intéressante pour déployer un service de communication multipoint. Ces liaisons ont toutefois des caractéristiques suffisamment différentes des liaisons terrestres pour qu'il soit nécessaire de les regarder en détail. Le chapitre suivant étudie donc les problèmes liés à l'utilisation de liaisons satellites, et ce afin de déterminer les mécanismes compatibles — ou non — avec celles-ci.

Chapitre 2

Problématique du transport multipoint en environnement satellite

2.1 AVANTAGES ET INCONVÉNIENTS DU SATELLITE

Les environnements de transmission à diffusion constituent un terrain très favorable au déploiement d'un service de communication multipoint. En particulier, les propriétés de diffusions naturelles des systèmes satellites, couplées ou non à des techniques de commutation embarquée et de transmission multi-spots permettent d'offrir un service efficace et en adéquation avec les objectifs de diffusion de groupe d'IP Multicast.

Cependant, l'intégration d'un lien satellite dans un réseau IP nécessite une adaptation préalable des protocoles utilisés que ce soit au niveau réseau [38], ou au niveau du transport [132]. En effet, la qualité de service d'un tel lien diffère grandement de celle des liens terrestres traditionnellement utilisés pour le support des réseaux IP. En conséquence les hypothèses adoptées pour la conception des protocoles standards (basées sur des liens terrestres) ne sont plus forcément valides. On cite par exemple le problème classique du délai de transit par un lien satellite, très important par rapport à celui d'un réseau terrestre ; ou encore les variations importantes du taux d'erreurs de transmission, dues à l'évolution des conditions climatiques. L'absence de lien retour est également un problème survenant traditionnellement dans un tel environnement. Il est nécessaire dans ce cas d'utiliser un mécanisme adéquat permettant d'intégrer des liens unidirectionnels dans des communications Internet comme LLTM [32] proposé par le groupe de travail UDLR [136]. Lorsqu'elles ne sont pas prises en compte, ces ca-

ractéristiques peuvent diminuer considérablement les performances des protocoles de transport. D'autre part, certaines caractéristiques de la transmission satellite ne sont pas toujours utilisées de manière optimale. Par exemple, dans le contexte d'un lien satellite connectant directement les utilisateurs, la possibilité d'accès à des informations d'état de la liaison est peu exploitée au niveau transport. Ce chapitre explique en quoi les liaisons satellites diffèrent des liens terrestres. Cela permet d'étudier dans la section 2.2.3 les différents mécanismes de transport afin de dégager les mécanismes potentiellement intéressants. Ce chapitre se termine par une étude statistique comparative des réseaux terrestres et satellites.

2.2 SPÉCIFICITÉ DES TRANSMISSIONS SATELLITES

2.2.1 Hypothèses sur le système satellite

De manière générale, les systèmes à satellites offrent la possibilité de fournir un service de communication sur une large étendue. on peut distinguer deux types de systèmes satellitaires : ceux utilisant un satellite géostationnaire (GEO), et ceux utilisant des satellites à défilement (LEO : *Low Earth Orbit*, et MEO : *Medium Earth Orbit*). Les satellites LEO et MEO permettent de réduire les coûts de lancement des satellites (le coût de lancement étant lié à l'altitude à laquelle le satellite doit être positionné). De plus, le fait que l'orbite soit basse ou moyenne réduit également le délai de transmission des informations. L'utilisation de satellite à défilement introduit cependant une complexité spécifique due à la nécessité d'acheminer les données entre les satellites. Avec ce type de système les satellites sont de plus mobiles par rapport aux stations terrestres. Il est donc nécessaire de mettre constamment à jour les chemins permettant de joindre des récepteurs. A l'inverse, avec un satellite géostationnaire permet d'atteindre directement tout récepteur situé dans sa zone de couverture. La zone de couverture de tel satellite est de plus très importante : la superficie couverte est de l'ordre d'un continent. Un tel système ne se heurte donc pas aux problèmes d'établissement de chemin (ou d'arbre de diffusion pour du multipoint). L'utilisation d'un système satellite utilisant un satellite géostationnaire semble donc appropriée pour un service de communication multipoint.

Dans le domaine des transmissions satellites, il existe plusieurs standards précisant le processus d'émission. Le standard DVB-S (*Digital Video Broacasting - Satellite* [34]) proposé par l'ETSI (*European Standard Telecommunication Series*) est cependant plus largement utilisé en Europe. L'utilisation de ce standard comme support d'un service de communication multipoint a donc fait l'objet de plusieurs études [31], [48]. Enfin, dans le système satellite il est possible d'utiliser plusieurs bandes de fréquence. Pour les télécommunications numériques, ce sont principalement les bandes Ku et Ka qui sont disponibles. Plus précisément, la bande Ku (de $10,7$ à $12,75$ GHz) est utilisée pour la transmission de télévision et de radio. Elle est actuellement la plus répandue en Europe, du fait de la petite taille des paraboles permettant de recevoir le signal (VSAT : *Very Small Aperture Terminal*). Cette bande de fréquence commence cependant à être saturée du fait du nombre d'opérateurs qui l'utilise. L'utilisation de la bande Ka ($20-30$ GHz) est donc étudiée pour étendre les fréquences utilisables pour la transmission de données.

L'architecture de communication sur laquelle se base l'étude est donc la suivante :

- le satellite est géostationnaire et mono-spot,
- il opère en bande Ka, et
- la transmission est conforme au standard DVB-S : les données sont encapsulées dans des trames MPEG2-TS (au moyen de la couche MPE —*Multi Protocol Encapsulation* [89]— ou de la couche ULE —*Ultra Lightweight Encapsulation* [36]— proposée par le groupe de travail IPDVB [63]).

Nous supposons de plus que le satellite propose un service de communication multipoint basé sur IP Multicast. Le système satellite reconnaît les adresses Multicast, et prend en charge les opérations d'abonnement et de départ des récepteurs. L'intégration d'un tel service à un système de communication satellite soulève plusieurs problèmes (voir par exemple [38]) qui ne sont pas considérés dans la suite du manuscrit. Les récepteurs quant à eux, sont connectés au réseau terrestre au moyen d'un réseau d'accès haut débit,et au système satellite soit directement avec une antenne VSAT (Very Small Aperture Terminal), soit au moyen d'un réseau d'accès haut débit.

Ces hypothèses sont appropriées par exemple, pour décrire un scénario de transmission de données vers des terminaux de réception de télévision par satellite. Dans ce cas chaque récepteur possède une antenne de réception avec un décodeur DVB-S. La connexion terrestre des récepteurs revient à faire l'hypothèse que le décodeur est relié au réseau terrestre au moyen d'une ligne téléphonique, d'une ligne ADSL, ou d'un réseau Ethernet par exemple (voir figure 2.1). Dans un tel système, la transmission de vidéo de haute qualité est, une application demandant de transmettre des volumes de données importants à de nombreux (voire très nombreux) utilisateurs. Les données reçues pourraient dans ce cas être stockées directement dans le décodeur. Ce scénario est réaliste à l'heure actuelle dans la mesure où certains décodeurs de télévision par satellite intègrent déjà la possibilité de se connecter au réseau téléphonique (généralement pour que l'utilisateur puisse demander la réception de films payants). De plus, l'intégration de moyen de stockage des données dans des systèmes de réception commence à se développer par l'intermédiaire des magnétoscopes numériques. Notons qu'une évolution récente de la norme DVB-S, DVB-RCS [33], permet d'utiliser une voie de retour satellite. Nous n'avons toutefois pas considéré cette possibilité car cela revient à utiliser la liaison satellite pour des communications point-à-point, ce qui n'est pas souhaitable (cette solution reste encore chère à l'heure actuelle).

Ces hypothèses sont également proches des hypothèses prises dans DIPCAST (projet labellisé par le Réseau National de Recherche en Télécommunication) : l'utilisation d'un satellite géostationnaire en bande Ku/Ku (lien aller / lien retour) ou Ku/Ka était retenue, et les récepteur recevaient le signal satellite par l'intermédiaire d'un réseau d'accès haut débit (directement connecté à une station de réception satellite).

2.2.2 Caractéristiques du système satellite

L'objectif de ce paragraphe est de présenter les caractéristiques du système satellite ayant une influence sur les performances et la conception des protocoles de transport. Il convient de noter que ce modèle doit être complété par une étude au niveau liaison permettant la description précise de la qualité de service du niveau réseau.

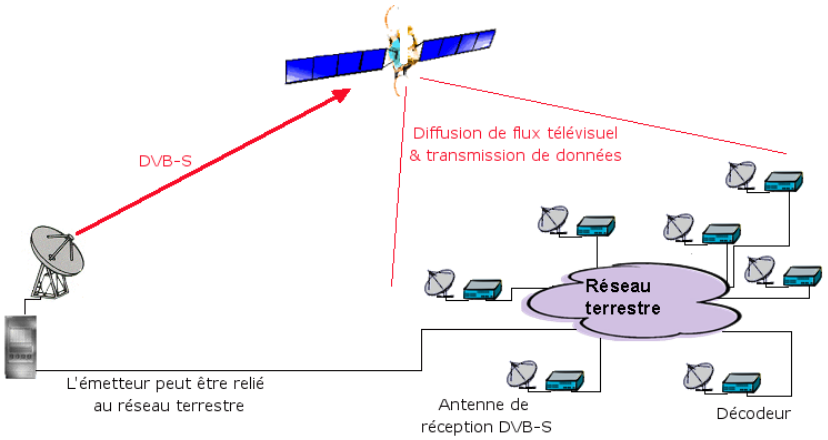


Figure 2.1 – Architecture de communication retenue dans le projet DIPCAST

Le codage utilisé pour transmettre les informations est conforme à la norme DVB-S. Classiquement, c'est un codage convolutif concaténé avec un code de Reed-Solomon [34][1]. Traditionnellement, le codage convolutif utilisé insère une redondance de 3/4 et, après celui-ci, un codage de Reed-Solomon $(188, 204)$ est utilisé. La capacité du système récepteur à décoder correctement le signal dépend ensuite du niveau du signal reçu (en dB). Selon les résultats présentés dans [41], les caractéristiques de décodage sont les suivantes :

- soit le niveau du signal est au-dessus du seuil de décodage, et les paquets sont tous correctement décodés (le taux d'erreur bit est inférieur à 10^{-9}) ;
- soit le niveau du signal est sous le seuil de décodage, et aucun paquet ne peut être décodé.

Il est ainsi fréquent de modéliser le système satellite par un modèle *on-off*. Les marges (en termes de puissance d'émission) prévues à la conception du système conditionnent le temps de disponibilité global du système. Généralement ces marges sont choisies de manière à obtenir une disponibilité de plus de 96%. En fonction de l'architecture utilisée, et des fonctions remplies par les différentes couches, cela se traduit au niveau transport par des pertes en séquence (due à des atténuations de signal) plus des pertes marginales (dues au taux d'erreur bit en sortie du décodeur). Comme les pertes isolées sont négligeables par rapport aux pertes en séquence, l'évolution temporelle des atténuations permettrait d'approximer l'évolution PLR au cours d'une transmission. Les systèmes satellites en bande Ka sont particulièrement sensibles aux perturbations atmosphériques. Des études ont été faites afin de quantifier les atténuations survenant pendant le fonctionnement de tels satellites [103]. Celles-ci permettent de connaître le pourcentage d'atténuation durant un certain temps, mais pas l'évolution ni la fréquence de ces atténuations.

A ce profil de pertes particulier s'ajoute le délai important de transit au sein du

[1]Il existe d'autres codages, notamment les turbo-codes proposées dans la norme DVB-RCS [33]. Le service obtenu en sortie du système satellite est cependant similaire : les capacités et/ou vitesse d'encodage et de décodage varient, mais le profil des pertes reste similaire.

système satellite. Tout d'abord le délai de propagation brut Terre-Satellite-Terre est d'environ 250 ms. A ce délai de bout en bout minimum s'ajoutent d'autres délais de traitement, plus ou moins importants selon le service. Par exemple, les délais d'obtention de ressources par des systèmes DAMA (*Demand Assignement Multiple Acces*) peuvent engendrer un délai d'attente relativement long.

Enfin le débit disponible pour le trafic IP sur un satellite varie entre quelques kilobits par seconde et plusieurs gigabits par seconde. A ces paramètres de qualité de service s'ajoutent des paramètres liés à l'architecture de la liaison satellite. Ces caractéristiques sont notamment :

- La propriété de diffusion naturelle des informations.
- La position fixe des passerelles et du satellite ;
- La position centrale du satellite qui traduit le fait que le satellite est le point de passage obligé entre la source et les récepteurs ;
- Le contrôle d'accès qui doit permettre de limiter les émissions vers le satellite ;

A ces paramètres il faut ajouter la position en fin de chaîne pour les passerelles et les utilisateurs ayant une antenne de réception satellite.

2.2.3 Impact sur les mécanismes de niveau transport

Ce paragraphe reprend les principaux mécanismes utilisés par les protocoles de transport et étudie dans quelles mesures ils sont adaptés ou non au satellite.

2.2.3.1 Fiabilité, requêtes de retransmission et retransmission

Cette partie étudie tout d'abord les différentes possibilités concernant le mécanisme de requête de retransmission. Le premier choix porte sur le fait que la retransmission est en mode unicast ou multicast. Pour un satellite mono spot, la retransmission sera physiquement diffusée sur toute la zone de couverture. Le coût de diffusion étant payé, il semble opportun d'exploiter le fait que la communication soit multipoint : chaque récepteur peut exploiter les données retransmises s'il en a besoin. En conséquence, l'utilisation de serveurs locaux n'a de sens que si les communications entre les serveurs et les récepteurs se font au moyen de liaisons terrestres. Cette dernière alternative est particulièrement intéressante lorsque les pertes subviennent dans le réseau situé entre la passerelle et les récepteurs finals : cela permet d'éviter l'utilisation du satellite, ce qui diminue le coût et améliore le temps de réponse du système.

Il est ensuite essentiel de s'interroger sur les informations qui seront retransmises : est-il intéressant d'utiliser un codage des informations au niveau transport. Dès lors que les retransmissions utilisent le lien satellite, comme les renvois sont transmis à tout le groupe, il est avantageux de maximiser leur utilité. Ainsi l'utilisation d'un codage à taux variable permettant d'envoyer le minimum de paquets pour que chacun des récepteurs puisse reconstruire l'information semble fort indiquée. Il est également possible d'introduire une certaine redondance sur ces paquets codés pour diminuer les taux de perte de paquet (au niveau transport), et éviter une retransmission ultérieure.

Les FEC apparaissent donc comme un moyen particulièrement adapté au satellite pour gérer la fiabilité (et sont de fait déjà largement utilisés). Ce mécanisme ajoute

une certaine redondance au niveau transport pour certaines unités d'information. Des données jugées prioritaires par l'application pourraient ainsi être envoyées de manière plus fiable que les autres afin d'éviter une retransmission. Le cas extrême correspondant à l'approche ALC est également possible : le flux de données est envoyé encodé avec une redondance préalablement choisie. Dans ce cas il n'y a plus besoin de retours. Certains problèmes sont cependant liés à cette approche :

- A cause de la nature statistique de cette technique, il peut toujours arriver des cas où le nombre minimum de messages n'est pas reçu. Alors, les messages reçus sont inutilisables, et la transmission doit être recommencée. Il est alors nécessaire qu'un mécanisme supplémentaire soit ajouté, par exemple pour que les récepteurs stipulent à la source que toute l'information, ou l'une de ses parties, ont été correctement reçues. Une utilisation minimale de la voie retour semble donc nécessaire pour garantir la fiabilité totale des transmissions.
- Une partie de la bande passante risque d'être gâchée, puisque dans le cas d'une transmission sans erreur, la redondance ajoutée n'apporte rien.

2.2.3.2 Gestion du groupe

Pour proposer une gestion explicite des groupes, le recours à une gestion distribuée de l'ensemble des membres est préconisé. Cela permet de limiter l'utilisation des liaisons satellites. En effet dans ce cas, en choisissant les responsables des sous-groupes après les liaisons satellites, les messages d'abonnement et de désabonnement aux groupes n'empruntent pas ces liaisons.

2.2.3.3 Gestion du temps

Le satellite GEO semble peu approprié aux applications hautement interactives, du fait du délai de bout-en-bout que son utilisation implique. De plus le contrôle d'accès introduit des délais supplémentaires variables. Au niveau de la communication cela se traduit par des variations de la gigue au cours de la communication. Il semble difficile pour le protocole de transport de répondre à une contrainte de temps s'il n'est pas possible de réserver des ressources.

Correctement paramétré, le mécanisme codage des informations permet d'éviter des demandes de retransmission. Il permet donc *a priori* de diminuer le délai d'obtention d'une information. Cependant ce mécanisme implique des temps de décodage des informations. De surcroît ce sont des codes en blocs qui sont utilisés au niveau transport [68], ce qui implique que c'est tout un ensemble de données applicatives qui sera décodé pour être transmis à l'application. Ce mécanisme semble donc utilisable pour des applications telles que de la diffusion vidéo (*streaming*) : les données peuvent être ajoutées à la mémoire tampon. Par contre le codage ne permet pas de proposer un service interactif avec le satellite. Notons de plus que cette technique utilise une grande partie de la bande passante, et qu'elle ne permet pas de régler les problèmes de gigue.

Enfin la capacité naturelle de diffusion des systèmes satellites est une caractéristique intéressante pour un type de service temporel : il est possible d'assurer une bonne synchronisation de l'arrivée des paquets au niveau des utilisateurs, éventuellement en pre-

nant en compte leur position géographique. Toutefois, dès que des pertes subviennent, la synchronisation des réceptions n'est plus de mise. Dans la mesure où les mécanismes existants pour ce service sont très lourds (en termes de nombre de RTT par exemple), il ne semble pas intéressant de les implémenter. Le protocole de transport peut par contre assurer que les informations correctement reçues sont transmises au même instant à l'application.

2.2.3.4 Contrôle de congestion

De manière classique, l'utilisation d'une fenêtre de transmission est liée à un mode de fiabilisation basé sur un envoi minimum de messages ACK de la part des membres, ce qui pose généralement des problèmes de mise à l'échelle. Ce mécanisme, utilisé dans TCP, utilise la détection de pertes suivant l'analyse des acquittements reçus : chaque perte est associée à une congestion au sein du réseau. Cette hypothèse est largement vérifiée dans le réseau Internet utilisant une technologie filaire. Elle constitue par contre un problème pour les réseaux satellites, où les erreurs de transmissions ne sont pas marginales. Dans ces cas, la diminution du taux de transfert est opérée sans raison et selon les algorithmes utilisés, le rétablissement du taux de transfert optimal peut être assez long à cause du RTT important. Notons que plusieurs projets de recherche ont cependant étudié la possibilité d'avertir explicitement les protagonistes d'une communication de congestions au sein de réseaux IP. Ces recherches ont en particulier aboutit à la proposition d'un mécanisme : ECN (*Explicit Congestion Notification* [108].

La deuxième approche pour le contrôle de congestion consiste à anticiper les états de remplissage des buffers en spécifiant un débit maximum d'émission. Dans le cas où un seul débit est utilisé, tous les récepteurs reçoivent les données à la même vitesse, et pour cela, il est nécessaire de s'adapter au membre le plus ¡¡ lent ¿¿ (PGMCC). Pour éviter ce problème, il est possible d'utiliser l'approche ALC avec le mécanisme WEBRC : plusieurs flux de données sont envoyés simultanément. Cependant ce mécanisme introduit une utilisation intensive de bande passante : dans la mesure où toutes les informations envoyées sur les couches $l > 0$ sont comprises dans la couche 0 (les flux de données sont complémentaires). Cela peut ne pas être acceptable pour des transmissions satellites, déjà réputées onéreuses. De plus, lorsque le lien satellite est en congestion, il faut que tous les récepteurs se désabonnent du nombre de couches adéquat avant que le flux émis par la source diminue. Cette opération peut prendre un temps non négligeable, selon le mécanisme de désabonnement implémenté au niveau réseau (c'est le cas par exemple lorsque IGMP est utilisé).

Un problème survenant lors de l'utilisation d'un lien satellite est que la source a tout intérêt à utiliser la capacité du satellite au maximum. Dans ce cas, le mécanisme d'allocation des ressources du satellite permet généralement d'assurer que le trafic injecté ne nuit pas au système satellite. Cela ne pose pas de problème lorsque les récepteurs reçoivent directement le trafic au moyen de VSAT. Il paraît donc plus intéressant dans ce cadre, d'établir une communication entre la source et la passerelle, de manière à ce que la source soit avertie des ressources qui lui sont attribuées. Elle pourra ainsi en tirer parti au maximum, sans pour autant saturer le réseau. Des problèmes surgissent cependant quand la source ou les récepteurs sont connectés au moyen de réseaux

d'accès : le trafic peut provoquer des congestions sur les liens terrestres traversés. Ainsi, le mécanisme de contrôle de congestion peut entraîner une sous-utilisation des ressources satellites. Ce problème n'est aujourd'hui pas encore résolu, et les projets étudiant les transmissions de données par satellite font souvent l'hypothèse que les utilisateurs sont connectés au moyen de réseau d'accès haut débit, ce qui diminue les probabilités d'occurrence de congestions.

2.2.4 Conclusion : restriction des services proposés avec le satellite

Le satellite permet de transmettre naturellement des informations à de très nombreux récepteurs. Ce système présente entre autres l'avantage de limiter le nombre de réseaux traversés, ce qui permet d'établir un modèle de tarification utilisable (ce problème a notamment été étudié dans le projet DIPCAST).

Certains des mécanismes développés pour les protocoles de transport multipoint fiables terrestres sont parfaitement adaptés à l'environnement satellite. C'est le cas en particulier du codage des informations au niveau transport. En effet, parmi toutes les propositions étudiées, tous les protocoles qui se destinent à des transmissions satellites utilisent ce mécanisme. Par exemple FCast, MFTP proposent une option permettant d'utiliser un codage pro-actif des données, associé à une transmission en plusieurs passes lorsque aucune voie retour n'est disponible. À l'inverse, lorsqu'une voie retour est disponible, les propositions plébiscitent le mécanisme HARQ2. C'est le cas par exemple de MDP, RMTP2, et SMDP. La plupart des offres commerciales spécifiques au satellite comme par exemple UDCast [135], Digital Fountain [28], et Castify [16] annoncent également qu'elles utilisent cette technique. Qu'il soit utilisé de manière pro-active ou pour coder les retransmissions, ce mécanisme permet d'améliorer les performances du protocole de transport. Le mécanisme de *Nack suppression* est également utilisable dans un tel environnement.

A l'opposé, certains services sont difficilement adaptables à l'environnement satellite. La gestion du temps pour des applications interactives en particulier paraît impossible : avec un RTT compris entre 300 ms (pour une architecture hybride terrestre / satellite) et 600 ms lorsque la voie retour utilise le satellite), le satellite ne peut satisfaire les exigences de communication de telles applications. De plus l'utilisation du mécanisme de *Nack Suppression* introduit des délais supplémentaires aléatoires. L'implémentation d'un mécanisme permettant de gérer le groupe semble également peu indiquée. En effet, dans la mesure où les messages sont diffusés sur toute la zone de couverture, il n'est pas possible d'empêcher un récepteur d'écouter les communications. L'implémentation d'un tel service (pour garantir la sécurité des communications par exemple) paraît plus efficace au niveau transport. Pour finir le contrôle de congestion pose un problème difficile à résoudre dans le cadre des communications multipoints en général, et pour le satellite en particulier.

Au vu de ces caractéristiques, le satellite paraît tout indiqué pour proposer un service de diffusion fiable à grande échelle, sans contraintes temporelles. Afin de dégager l'intérêt de déployer un tel service au moyen du satellite plutôt qu'avec le réseau terrestre, la section suivante expose une étude statistique portant sur le coût de communications multipoint terrestres et satellites.

2.3 ÉTUDE DU COÛT DE COMMUNICATIONS TERRESTRES ET SATELLITES

L'objectif de ce paragraphe est d'évaluer différentes possibilités pour transmettre des données vers un groupe. Dans ce but une fonction de coût est définie. Plusieurs hypothèses, portant notamment sur les pertes subies par les récepteurs, sont ensuite adoptées de manière à étudier cette fonction de coût. La fonction de coût ainsi que les hypothèses adoptées se veulent représentatives des techniques retenues pour les réseaux terrestres et satellites. En particulier le mécanisme HARQ2 est utilisé pour toutes les communications multipoints. Les résultats ainsi obtenus sont la base de la motivation d'un couplage des technologies terrestre et satellite.

2.3.1 Définition d'une fonction de coût

Ce paragraphe a pour objectif de proposer une fonction de coût prenant en compte :

- la charge induite sur le réseau (la bande passante utilisée), et
- un coût technologique (en terme financier par exemple) dû à l'utilisation des équipements de niveau réseau.

Pour cela nous avons adopté une approche ¡¡ par paquet ¿¿. C'est-à-dire que la charge induite sur le réseau est assimilée au nombre de paquets injectés dans le réseau. Nous avons ensuite considéré que la transmission de chaque paquet induisait un coût de transmission (différent selon la technologie utilisée), et que ce coût était constant. Cette fonction de coût est globalement représentative des communications terrestres et satellites. Notons cependant que cette fonction de coût pourrait être affinée, en intégrant par exemple le débit en émission comme paramètre du facteur de coût (dans la mesure où un abonnement à un débit plus haut est généralement plus onéreux), ou encore le volume de données envoyées.

Le calcul de la charge induite sur le réseau diffère selon le médium utilisé pour transmettre les données. Dans les paragraphes suivants, l'acronyme TU (*Terrestrial Unicast*) fera référence aux transmissions terrestres point-à-point, TM (*Terrestrial Multicast*) aux transmissions terrestres multipoints et SM (*Satellite Multicast* aux transmissions satellites multipoints.

2.3.1.1 Cas du réseau Terrestre

Lorsque le service de communication multipoint s'appuie sur un service réseau point-à-point, il est nécessaire d'émettre un flux de données vers chaque récepteur. Ainsi pour transmettre T paquets à un groupe de N hôtes, N flux sont transmis. En notant T'_i est le nombre de paquets transmis pour que le récepteur i ait reçu T paquets, la charge induite s'exprime par :

$$C_{TU}(N,T) = \sum_{i=1}^{N} T'_i \tag{2.1}$$

où :

T est le nombre de paquets à transmettre,

N est le nombre de récepteurs, et

T'_i est le nombre de paquets nécessairement transmis pour que le récepteur i ait reçu les données.

2.3.1.2 Cas des communications multipoints terrestres

Pour les communications multipoints terrestres, nous avons cherché à nous ramener à un problème équivalent aux communications point-à-point terrestres. Pour cela, nous avons utilisé le résultat de [18] qui donne le nombre de connexions équivalentes à un arbre de diffusion. Plus précisément pour une diffusion vers N récepteurs, selon les mesures effectuées sur le MBone présentées dans [18], le nombre de connexions équivalentes est égal à $N^{0,8}$. La charge induite sur le réseau peut donc être exprimée par :

$$C_{TM}(N,T) = N^{0,8}\, T'(N,T) \tag{2.2}$$

où $T'(N,T)$ est le nombre de paquets transmis de manière à ce que tout le groupe ait reçu les données (les retransmissions étant effectuées avec des paquets de parité transmis à tout le groupe). Il faut remarquer que dans [18] Chuang et Sirbu proposent de rajouter un surcoût dû à la mise en place et au maintient de l'arbre de diffusion. Dans ce cas ce coût est proportionnel au nombre de récepteurs, et s'exprime par N^{ϵ}. Le choix de ϵ dépend du fournisseur d'accès. La valeur de ϵ doit également être inférieure à $0,2$ de manière à promouvoir l'utilisation de communications multipoints pour la transmission de données vers un groupe (i.e. de manière à ce que $N^{0,8+\epsilon} < N$). Afin à ne pas pénaliser les communications multipoints terrestres dans notre comparaison, nous avons fixé $\epsilon = 0$. Ainsi le coût calculé sera le coût minimum d'une communication multipoint.

2.3.1.3 Cas des communications multipoints satellites

Pour les communications faisant intervenir le système satellite, nous avons supposé que les seules pertes à prendre en compte étaient les pertes survenant sur le lien satellite. Cela revient à négliger les pertes sur le réseau d'accès de la source au système satellite, ainsi que sur l'éventuel réseau d'accès des récepteurs. En d'autres termes, nous avons supposé qu'il n'y avait pas de congestions sur ces réseaux, i.e. que le débit en émission était toujours inférieur à la capacité du réseau. Comme la diffusion des données est naturelle, la charge induite s'exprime simplement comme :

$$C_{SM}(N,T) = T'(N,T) \tag{2.3}$$

(les notations sont les mêmes qu'au paragraphe précédent). De même que pour les communications terrestres multipoints, il faudrait ajouter un coût de maintien de la structure de diffusion. Cependant comme le satellite transmet les informations avec un seul bond, ces opérations de maintien sont locales à la passerelle satellite et aux récepteurs ou à leur réseau d'accès. La source doit en effet être capable d'envoyer un flux de données vers une adresse de classe D, ce qui implique que ce flux de données soit aiguillé vers la passerelle du système satellite. Cette opération est simplifiée lorsque la source appartient au réseau local de la passerelle : il n'est pas nécessaire d'établir

un tunnel IP Multicast entre la source et la passerelle (il faut cependant configurer le réseau local pour que les adresses de classe D soient destinées à la passerelle). C'est également le cas lorsque la source peut déposer les données à transmettre sur un hôte appartenant au système satellite, ou quand le flux de données est encapsulé dans une connexion point-à-point dirigée vers la passerelle. Pour ce qui est des récepteurs, seuls les récepteurs ne recevant pas directement le flux de données au moyen d'un VSAT sont concernés (car il n'y a pas de structure intermédiaire pour un récepteur utilisant un VSAT). Pour ceux-ci nous supposons que le nombre de routeurs traversés entre le point d'accès au système satellite et eux, est suffisamment faible pour pouvoir négliger ce coût.

2.3.1.4 Conclusion

Ce paragraphe a présenté le calcul de la charge induite sur le réseau pour les différents moyens disponibles pour transmettre des données vers un groupe de récepteurs. Afin de prendre en compte le coût induit par la technologie utilisée, ces grandeurs sont multipliées par un facteur α_t représentant le coût de transmission d'un paquet. Les fonctions de coût, notées $(F_t(N,T))_{t\in\{TU,TM,SM\}}$, qui seront comparées seront donc :

$$F_t(N,T) = \alpha_t \, C_t(N,T), \; t \in \{TU, TM, SM\} \tag{2.4}$$

Le paragraphe suivant expose les valeurs prises par ces fonctions de coût lorsque certaines hypothèses sont effectuées sur les pertes survenant sur le réseau.

2.3.2 Application aux communications terrestres et satellites

Dans cette partie, les différentes hypothèses prises sur le réseau sont d'abord présentées. Ensuite les différents scenarii qui seront étudiés sont décrits, et pour finir des expressions de la fonction de coût sont calculées (en se basant sur les hypothèses présentées ci-dessous) et commentées.

2.3.2.1 Hypothèses prises pour les pertes

En ce qui concerne le réseau terrestre, nous supposons que les pertes observées de bout-en-bout ne sont dues qu'à des congestions (la probabilité d'occurrence d'une erreur de transmission est négligée). Si le réseau n'offre pas de service multipoint, nous supposons qu'un service de communication est mis en place au niveau applicatif. L'utilisation de ce service se traduit au niveau réseau par l'ouverture d'une connexion point-à-point TCP pour chaque récepteur du groupe (vers la source ou vers un autre membre du groupe). Nous supposons de plus que les congestions représentent au niveau transport un taux de pertes de paquets (*Packet Loss Rate* : PLR) d'environ 5% sur l'ensemble de la transmission [100].

Dans le cas où un service de communication multipoint est disponible au niveau réseau (comme IP Multicast), nous supposons que ce service de diffusion est disponible sur tout le réseau de la source jusqu'au récepteur final. Nous avons également considéré que les pertes dues à des congestions sur l'arbre de diffusion se traduisent par un PLR

de 10% au niveau des récepteurs [141]. Dans les deux cas, les pertes sont supposées uniformément réparties et indépendantes au niveau des récepteurs. Dans la mesure où nous avons supposé qu'un codage était mis en œuvre dans les protocoles de transport multipoints, c'est le nombre de paquets perdus, plus que leur répartition qui importe. Ainsi, bien que réductrice, cette hypothèse classique a peu d'impact sur les résultats finals et permet de simplifier les calculs.

En ce qui concerne le lien satellite, nous supposons que le système satellite utilise un satellite géostationnaire qui émet en bande Ka. La source a conscience d'émettre au moyen d'un lien satellite, et peut donc choisir un protocole de transport adapté. Les récepteurs reçoivent directement le signal satellite et sont aussi connectés au réseau terrestre de manière à pouvoir communiquer entre eux, ou avec la source si besoin est.

Le système satellite offre un service de communication multipoint. Ainsi la source peut émettre un flux de données adressé à une adresse de classe D vers l'interface satellite. D'autre part le satellite reconnaît ces adresses et est capable de les diffuser sur sa zone de couverture. Les récepteurs quant à eux sont capables de filtrer les flux reçus (afin de ne recevoir que les flux auxquels ils sont abonnés). Pour ce scénario, afin de représenter les services Internet disponibles à l'heure actuelle, nous supposons qu'il n'y a pas de service de diffusion multipoint disponible sur le réseau terrestre. Une transmission terrestre est donc forcément point-à-point (et en particulier, la source ne peut pas transmettre les données à un ensemble de récepteurs par voie terrestre).

Les hypothèses précédentes supposent entre autres que les problèmes de traduction d'adresses entre les niveaux réseau et accès sont résolus (ces problèmes ont notamment fait l'objet d'une étude au sein du projet DIPCAST). Les difficultés liées à la mise en place de protocoles de routage multipoint nécessitent également une étude approfondie [38].

Pour finir, comme le satellite est supposé émettre en bande Ka, le signal émis est sujet à des atténuations dépendant des conditions climatiques [103]. Ces atténuations entraînent des pertes de connexion pendant des périodes plus ou moins longues. Au cours de ces périodes, les récepteurs ne reçoivent aucun paquet. En conséquence, le taux de perte de paquets dépend du débit alloué sur la liaison satellite, et du nombre de paquets transmis. Le lien entre le temps d'indisponibilité et le PLR peut être simplement obtenu *a posteriori* par la formule suivante :

$$PLR = \frac{D \times P_{Att}}{T'} \tag{2.5}$$

où :

T' est le volume total de données transmises par la couche transport pour que la destination ait reçu les données (T' diffère de la taille de l'objet à envoyer dès qu'une retransmission a été effectuée ou que des paquets de redondance ont été envoyés),

D est le débit en émission au niveau applicatif, et

P_{Att} est la durée cumulée des atténuations survenues pendant la transmission.

Il faut remarquer que (2.5) traduit un ensemble de pertes survenues en séquence par des pertes uniformément réparties. Cette conversion n'est pas gênante dans notre

contexte, car la couche transport utilise un codage des informations pour assurer la fiabilité de la transmission. Ainsi, c'est le nombre de paquets perdus qui importe, plus que la répartition de ces pertes.

Nous proposons à titre d'exemple dans le paragraphe suivant un scénario de communication. Les valeurs prises dans ce scénario ne sont pas des valeurs générales, ce ne sont que des valeurs possibles, mais réalistes, lors d'une transmission satellite.

2.3.2.2 Scenarii d'étude

Nous avons considéré les trois types de service réseau permettant de transmettre des informations vers un groupe de récepteurs (i.e. TU, TM, SM). Pour chacun des services nous avons défini un scénario protocolaire compatible avec un service de communication multipoint fiable :

Scénario Terrestre Unicast (scénario **TU**) : le fichier est transmis vers le groupe au moyen d'un mécanisme de transmission multipoint applicatif. La fiabilité de la transmission est assurée par l'utilisation de TCP au niveau transport. Les congestions sur le réseau sont traduites par un taux de pertes de paquets de 5% (TCP se charge de retransmettre les paquets perdus).

Scénario Terrestre Multicast (scénario **TM**) : le fichier est transmis vers le groupe de récepteurs en utilisant un arbre de diffusion IP Multicast. La fiabilité de la transmission est assurée par un protocole de transport multipoint. Ce protocole utilise un codage des données et un mécanisme de type Hybrid ARQ type 2. Les congestions sur le réseau sont traduites par un taux de pertes de paquets de 10%.

Scénario Satellite Multicast (scénario **SM**) : le fichier est transmis en utilisant un lien satellite. Les données sont diffusées sur toute la couverture du satellite. Les récepteurs reçoivent directement le flux de données. Le protocole de transport utilise un code correcteur d'erreurs et un mécanisme Hybrid ARQ type 2. Des paquets de redondance sont transmis tant que tout le groupe n'a pas intégralement reçu le fichier. Un lien terrestre est utilisé pour avertir la source de la bonne réception du fichier.

Nous avons considéré que les récepteurs peuvent être répartis en trois groupes. Un groupe d'utilisateurs ne subissant aucune perte, un groupe d'utilisateur subissant des pertes moyennes, et un groupe d'utilisateurs subissant de lourdes pertes.

Le groupe d'utilisateurs ne subissant aucune perte correspond aux récepteurs situés sous un ciel clair. L'existence de ce groupe implique qu'il n'y ait pas de pertes sur le lien montant (car ces pertes sont subies par tous les récepteurs). Nous considérons dans l'exemple développé par la suite que ce groupe représente $\beta_0 = 90\%$ des récepteurs.

Le groupe d'utilisateurs subissant des pertes moyennes correspond aux utilisateurs qui voient passer une perturbation provoquant peu de pertes (pluie fine par exemple) au cours de la transmission. Dans la suite nous avons considéré que le temps d'indisponibilité de la liaison satellite pour ces récepteurs correspondait à 20% du temps de la communication. En d'autres termes, cela revient à dire que le taux de perte PLR_1 de ces récepteurs était de 20%. Nous considérons dans ce document que ce groupe représente $\beta_1 = 9.9\%$ du nombre total de récepteurs.

Pour finir le dernier groupe, correspondant à $\beta_2 = 0.1\%$ des récepteurs, subit $PLR_2 = 60\%$ de pertes. Ce groupe correspond aux récepteurs pour lesquels la transmission est troublée par de lourdes perturbations climatiques (orage, forte pluie). Ici également le taux de perte représente la part d'indisponibilité du satellite pendant la communication.

Les résultats présentés ici sont conformes aux résultats présentés dans le document [103]. Nous sommes conscients que les hypothèses retenues sont simplificatrices. Il serait préférable par exemple d'utiliser un modèle prenant en compte la répartition et le déplacement de perturbations atmosphériques. Cependant le scénario présenté reste plausible (toujours selon [103]). Ce scénario de propagation ne doit donc être considéré que comme un exemple de communication satellite.

2.3.2.3 Expressions des fonctions de coût

Transmissions Terrestres Point-à-point

Pour le cas des transmissions point-à-point terrestres, avec les hypothèses faites ci-dessus, le coût moyen des communications peut être calculé à partir de l'équation 2.1. Pour qu'un paquet soit effectivement reçu au bout de m envois, il faut que les $m-1$ premiers soient perdus dans le réseau, et que le dernier soit reçu. Comme les pertes sont uniformément réparties et indépendantes — avec un taux de pertes de paquets PLR_{TU} — le nombre moyen d'envois à effectuer pour chaque paquets est donc de $\sum_{m=1}^{\infty} m\,(1-PLR_{TU})PLR_{TU}^{m-1}$. Ce qui permet d'exprimer le coût moyen :

$$\begin{aligned} C_{TU}(N,T) &= \sum_{i=1}^{N} T \sum_{m=1}^{\infty} m\,(1-PLR_{TU})PLR_{TU}^{m-1} \\ &= \frac{N\,T}{1-PLR_{TU}} \end{aligned} \tag{2.6}$$

Transmissions Terrestres Multipoints

Dans ce cas, comme un code est utilisé au niveau transport, la source transmet des paquets de parité jusqu'à ce que tout le groupe puisse reconstituer les données originales. Ainsi, sur l'ensemble du groupe, seul le nombre maximum de paquets perdus par les récepteurs va déterminer le nombre de paquets transmis. La charge moyenne dépend donc de la moyenne de ce maximum. Or la probabilité de perdre K paquets parmi T', et de recevoir les $T'-K$ restant s'exprime comme $\binom{T'}{K} PLR_{TM}^{K}\,(1-PLR_{TM})^{T'-K}$ (les pertes étant supposées indépendantes et uniformément réparties). La probabilité de recevoir au moins T paquets parmi T' s'exprime donc comme $\sum_{K=0}^{T'-T} \binom{T'}{K} PLR_{TM}^{K}\,(1-PLR_{TM})^{T'-K}$. Cette condition devant être vérifiée pour les N récepteurs, la probabilité pour que tout le groupe reçoive les données après transmission de T' paquets s'exprime ainsi (en considérant que les pertes entre

récepteurs sont indépendantes) :

$$P_1(T', PLR_{TM}, N) = \begin{cases} = & \left[\sum_{K=0}^{T'-T} \binom{T'}{K} PLR_{TM}^K (1-PLR_{TM})^{T'-K}\right]^N & \text{, si } T' \geq T \\ = & 0 & \text{, si } T' < T \end{cases} \tag{2.7}$$

On appelle

- A l'évènement ¡¡ tout le groupe a reçu au moins T paquets lorsque $T'-1$ paquets ont été envoyés ¿¿, et
- B l'évènement ¡¡ tout le groupe a reçu au moins T paquets lorsque T' paquets ont été envoyés ¿¿,

Avec ces notations la probabilité pour qu'il soit nécessaire et suffisant d'envoyer exactement T' paquets pour que les données soient reçues par tout le groupe, est la probabilité d'occurrence de $B \setminus A$. Or $B \setminus A = B \cap \bar{A}$, et $B = (B \cap \bar{A}) \cup (B \cap A)$. On en déduit que $P(B) = P((B \cap \bar{A}) \cup (B \cap A))$. En remarquant que $(B \cap \bar{A}) \cap (B \cap A) = \emptyset$, on obtient $P(B) = P(B \cap \bar{A}) + P(B \cap A)$. La définition des évènements A et B permet de plus de dire que $A \subset B$, et donc

$$P(B) = P(B \cap \bar{A}) + P(A). \tag{2.8}$$

En utilisant l'équation 2.7 pour obtenir les valeurs de $P(A)$ et $P(B)$, et les injecter dans l'équation 2.8, on peut calculer la probabilité pour qu'il soit nécessaire d'envoyer exactement T' paquets. Cette probabilité est la probabilité que tout le groupe n'ait pas reçu les données après émission de $T'-1$ paquets, *et* que tout le groupe ait reçu les données après émission de T' paquets. Cette probabilité a pour expression :

$$P(T', PLR_{TM}, N) \begin{cases} = & P_1(T', PLR_{TM}, N) - \\ & P_1(T'-1, PLR_{TM}, N) & \text{, si } T' > T \\ = & \left[(1-PLR_{TM})^T\right]^N & \text{, si } T' = T \\ = & 0 & \text{, si } T' < T \end{cases} \tag{2.9}$$

Cela nous permet de calculer la charge moyenne induite sur le réseau, selon les résultats présentés dans le paragraphe 2.3.1.2 :

$$C_{TM}(N,T) = N^{0,8} \times \sum_{T'=0}^{\infty} T' \; P(T', PLR_{TM}, N) \tag{2.10}$$

et on peut en déduire : $F_{TM}(N,T) = \alpha_{TM} \times C_{TM}(N,T)$.

Transmissions Satellites Multipoints

Dans le contexte du satellite (scénario SM), le calcul de la charge induite par la transmission s'apparente au calcul effectué pour une transmission multipoint terrestre (comme l'indique la similitude entre les équations 2.2 et2.3). Avec les hypothèses faites sur les pertes, la transmission peut être vue comme plusieurs transmissions multipoints :

- une vers un groupe ne subissant pas de pertes (90% du groupe total),

- une vers un groupe subissant des pertes moyennes (9.9% du groupe avec un PLR de 20%),
- et une vers un groupe subissant de lourdes pertes (0.1% du groupe avec un PLR de 60%).

Pour chacune de ces communications, on peut calculer la charge de la même manière que dans le paragraphe précédent. La charge induite par la transmission satellite est ensuite déduite des charges précédentes comme étant le maximum des charges calculées. En posant :

- $\beta_0 = 90\%$, $\beta_1 = 9,9\%$, $\beta_2 = 0,1\%$, et
- $PLR_{SM_0} = 0$, $PLR_{SM_1} = 20\%$ et $PLR_{SM_2} = 60\%$,

le coût de la communication s'exprime comme :

$$F_{SM}(N,T) = \alpha_{SM} \times \max\left\{ \sum_{T'=0}^{\infty} T'\ P(T', PLR_{SM_i}, \beta_i N),\ i \in [0,2] \cap \mathbb{N} \right\} \qquad (2.11)$$

2.3.2.4 Comparaison des scenarii

La figure 2.2 présente l'évolution du coût des communications terrestres et satellites en fonction de la taille du groupe. Les résultats présentés ont été obtenus avec les facteurs de coût suivants : $\alpha_{TU} = \alpha_{TM} = 1$, et $\alpha_{SM} = 200$[2].Les taux de pertes utilisés sont ceux qui ont été décrits dans la section 2.3.2.2 : $PLR_{TU} = 5\%$, $PLR_{TM} = 10\%$, et le satellite est caractérisé par trois zones de réception. Enfin, cette étude correspond à l'envoie de 100 paquets. Notons que 100 paquets ne représentent en général pas la totalité du fichier, mais une partie. Plus précisément lorsqu'un code en blocs est utilisé (cf. section 3.2), le fichier est d'abord découpé en paquets. Les paquets sont ensuite regroupés en blocs de T_B (ici $T_B = 100$). Pour chaque bloc de T_B paquets, il est alors possible de générer jusqu'à $T_{max} - T_B$ paquets de redondance, où T_{max} est un paramètre du code. Avec cette technique chaque bloc est encodé indépendamment des autres. Cela signifie que chaque paquet de redondance ne permet de réparer que les pertes survenues au sein du bloc à partir duquel il a été généré. Il est donc intéressant d'étudier la transmission d'un ensemble de paquets correspondant à un bloc. Le coût ainsi calculé est bien proportionnel à la taille du groupe. C'est l'analyse de la position de la courbe par rapport aux coûts des autres approches classiques qui est intéressante.

Notons que pour une communication :

- visant à transmettre un volume S de données,
- en utilisant un codage en bloc avec T_B paquets par blocs,

le nombre blocs à envoyer est de $\lceil \frac{S}{T_B \times TDU} \rceil$, où TDU est le volume de données compris dans les paquets de niveau transport. Or le coût moyen de transmission de chaque bloc est précisément donné par $F_x(N, T_B)$. On peut donc calculer le coût moyen de

[2]Le choix de ces paramètres dépend des coûts d'exploitation des fournisseurs d'accès. Il ne nous a cependant pas été possible d'obtenir de véritables valeurs auprès d'Alcatel Space pour ces paramètres. Les valeurs choisies représentent donc uniquement l'ordre de grandeur de la différence de coût entre une communication terrestre et une communication satellite.

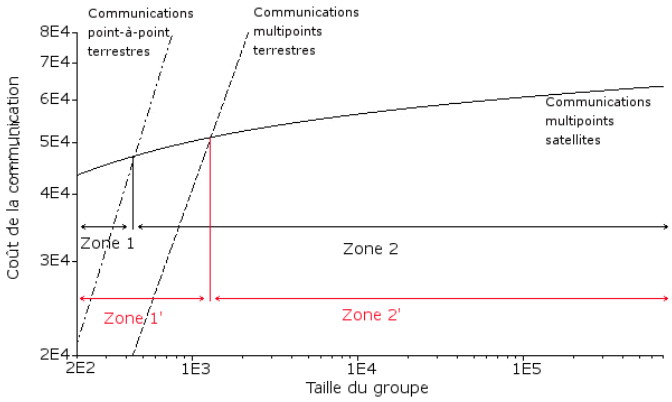


Figure 2.2 – Coût des transmissions en fonction de la taille du groupe, $T = 100$, $\alpha_{TU} = \alpha_{TM} = 1$, $\alpha_{SM} = 200$

transmission d'un objet de taille S :

$$G_x(N, S, TDU, T_B) = \lceil \frac{S}{T_B \times TDU} \rceil \times F_x(N, T_B).$$

Après avoir regardé plus en détail les résultats obtenus, nous avons remarqué que, dans tous les cas, ce sont toujours les quelques récepteurs ayant les pires conditions de réception qui imposent le coût de la transmission satellite. La figure 2.2 montre de plus que l'on peut distinguer plusieurs zones de transmission. Dans le cas où un service de communication multipoint n'est pas disponible au niveau réseau, il apparaît une zone (la *zone 1* : de 0 à 350 récepteurs environ dans la figure 2.2) où le coût d'une transmission satellite est plus important que celui de plusieurs transmissions terrestres. Il est donc plus intéressant d'utiliser le réseau terrestre. Le même constat peut être effectué dans le cas où un service de communication est disponible, si ce n'est que l'utilisation du réseau terrestre reste intéressante pour des tailles de groupe plus importantes. Cette zone est représentés par la *zone 1'* : de 0 à 1200 récepteurs dans la figure 2.2). Remarquons que les tailles de groupe inférieures à 200 récepteurs ne sont pas représentées sur la figure 2.2. cela se justifie dans la mesure où le service visé concerne des groupes de nombreux utilisateurs. De plus pour de très petits groupes, l'utilisation de IP Multicast n'est plus forcement intéressante, car elle implique un échange de données de contrôle visant à maintenir la structure de diffusion en place (par exemple, pour un seul récepteur, le coût d'utilisation de IP Multicast devrait être supérieur à celui de IP Unicast). Or le coût lié au maintient de l'arbre de diffusion n'a pas été pris en compte dans nos travaux (cf. 2.3.2.3). Dans ce cadre, la représentation de cette zone présentait donc peu d'intérêt.

Les résultats présentés dans la figure 2.2 ne sont présentés qu'à titre d'illustration (dans la mesure où les valeurs choisies pour les paramètres ne représentent que les ordres de grandeurs). Les seuils dépendent notamment des paramètres de coût qui ont été choisis, et en particulier de la différence de coût d'utilisation entre un lien satellite et d'une liaison terrestre. Cependant, dans tous les cas, pour des groupes vraiment importants les communications satellites sont clairement plus avantageuses que les communications terrestres. Cette zone est représentée par :

– *la zone 1'* quand un service de communication est disponible au niveau réseau, et

– *la zone 1* sinon.

C'est en particulier ces constatations qui motivent le couplage des transmissions satellites et terrestres.

2.3.2.5 Motivation d'un couplage terrestre/satellite

Lors d'une communication satellite, les pertes que subissent les récepteurs sont dépendantes des conditions climatiques. Or celles-ci sont relativement hétéroclites sur un espace géographique important, et mobiles de surcroît. Cela va entraîner une diversité et une variabilité des qualités de réception au sein du groupe. Au niveau du groupe de récepteurs, cette disparité se traduit par une diminution progressive du nombre de récepteurs qui continuent à être intéressés par la transmission au cours du temps. En effet, les récepteurs subissant de lourdes pertes doivent attendre que les conditions de réception s'améliorent afin de recevoir les informations qui leur manquent. A l'opposé les récepteurs ne subissant aucune perte peuvent quitter la session dès que les données sont intégralement transmises (sans attendre de recevoir de données retransmises). Il arrivera ainsi une période où le nombre de récepteurs sera suffisamment faible pour qu'il soit plus intéressant de diffuser les informations en utilisant le réseau terrestre. Une approche n'utilisant que la liaison satellite pour véhiculer les données doit attendre que le dernier récepteur (celui qui a subi le plus de pertes) ait totalement reçu le fichier pour arrêter la diffusion.

Ce paragraphe étudie le nombre moyen de récepteurs nécessitant encore la transmission de données (en prenant en compte les pertes survenues sur la liaison). Pour cela nous définissons une session Multicast de la manière suivante : elle commence avec le début de la diffusion des informations, et se termine lorsque tout le groupe a reçu les informations. Nous considérons qu'un récepteur est présent dans la session tant qu'il n'a pas reçu suffisamment d'informations pour reconstruire les données initiales.

En gardant les notations du paragraphe 2.3.2.3, la probabilité pour qu'un récepteur n'ai pas reçu assez de données après émission de T' paquets correspond à la probabilité qu'il ait perdu au moins T paquets. En reprenant les explications de la section 2.3.2.3, cette probabilité s'exprime comme $\sum_{K=T+1}^{T'} \binom{T'}{K} PLR_{TM}^{K} \left(1-PLR_{TM}\right)^{T'-K}$. Le nombre moyen de récepteurs n'ayant pas reçu à les données lorsque T' paquets ont été transmis peut donc être assimilé à :

$$N_{Rcv} = N \times \left[\sum_{K=T+1}^{T'} \binom{T'}{K} PLR_{SM}^{K} (1 - PLR_{SM})^{T'-K} \right] \tag{2.12}$$

A titre d'exemple, la figure 2.3 représente la moyenne du nombre de récepteurs présents dans une session avec un groupe total de 60 000 récepteurs en fonction du nombre de paquets envoyés. La session consistait ici à envoyer un bloc de 100 paquets. Des paquets de redondance ont été générés à partir de ces paquets initiaux : lorsque le nombre x de paquets indiqués en abscisses est supérieur à 100, $x - 100$ paquets de

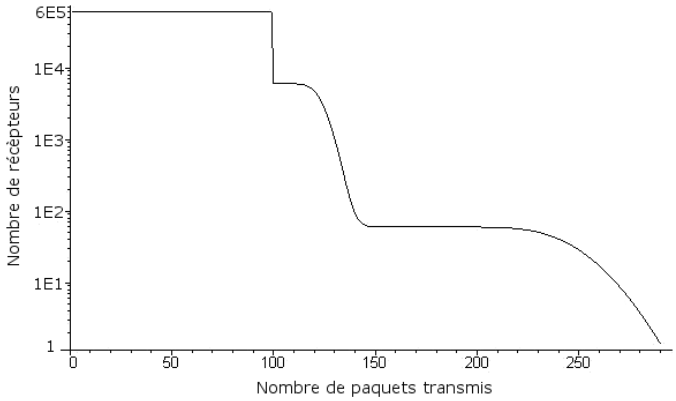


Figure 2.3 – Nombre de récepteurs présents dans la session, $N = 60000$**,** $T = 100$

redondance ont été envoyés. La taille du groupe correspond au nombre de récepteurs n'ayant pas reçu toutes les données après émission de ces x paquets.

Nous voyons sur cette courbe que le nombre de récepteurs présents dans la session diminue suffisamment pour qu'il devienne intéressant d'utiliser le réseau terrestre pour finir la transmission des données. En effet lorsque le nombre de paquets envoyés augmente, le nombre moyen de récepteurs tend vers 0 (seules les valeurs supérieures à 1 sont représentées ici). Or comme nous l'avons signalé dans le paragraphe 2.3.2.4, si le nombre de récepteurs est trop faible, il devient plus intéressant de transmettre les informations par le biais du réseau terrestre. De plus cette décroissance est relativement lente : un grand nombre de paquets doit être transféré pour faire diminuer significativement le nombre de récepteurs dans la session. Le transfert de ces paquets au moyen du satellite a un coût non négligeable. Ce constat a motivé l'étude d'un approche hybride qui utiliserait le réseau terrestre ou le système satellite en fonction de l'évolution de du nombre de récepteurs.

2.3.3 Étude d'une approche hybride satellite/terrestre

L'objectif de ce chapitre est de comparer une approche hybride utilisant à la fois les réseaux terrestres et satellites avec les approches classiques présentées ci-dessus. Le principe de l'approche hybride considérée est de choisir, en fonction de la taille du groupe, le moyen de transmission adapté, dans l'objectif de minimiser la fonction de coût définie à la section 2.3.1.

2.3.3.1 Calcul du coût d'une transmission hybride

L'approche proposée cherche à minimiser le coût global de la transmission. Nous avons supposé dans la section 2.3 que ce coût est fonction de plusieurs paramètres. Ce coût dépend, entre autre, du nombre de paquets envoyés et acheminés au sein du réseau, et de la technologie utilisée. L'optimisation consistera donc à choisir le type de liaison la moins coûteuse en fonction du nombre de récepteurs couramment intéressés par l'exploitation des paquets diffusés. Le principe général consiste ainsi :

- à transmettre les données par satellite tant qu'un grand nombre de récepteurs reste à l'écoute de la diffusion, puis,
- lorsque la plupart des récepteurs ont suffisamment de données pour reconstruire le fichier original, de compléter la transmission pour les récepteurs restant au moyen de transmissions terrestres.

Le coût global d'une telle transmission est donc naturellement défini comme la somme du coût lié à la transmission des données par satellite, et du coût des transmissions terrestres utilisées. Ce coût s'exprime donc comme suit :

$$F_H(N,T) = \alpha_{SM}\, T'_{SM} + \sum_{i=1}^{N(T'_{SM})} T'_{TU_i} \tag{2.13}$$

où

$F_H(N,T)$ correspond au coût de la transmission hybride,

T'_{SM} est le nombre de paquets transmis par satellite,

$N(T'_{SM})$ est le nombre de récepteurs n'ayant pas reçu suffisamment de données après l'envoi des T'_{SM} paquets par satellite

T'_{TU_i} est le nombre de paquets à envoyer au récepteur i (en utilisant une transmission terrestre point-à-point) pour qu'il ait suffisamment d'informations pour reconstruire les données initiales.

Il faut remarquer que la somme $S = \sum_{i=1}^{N(T'_{SM})} T'_{TU_i}$ commence à $i = 1$. Ainsi, lorsque $N(T'_{SM}) = 0$, $S = 0$ (par convention). $N(T'_{SM}) = 0$ correspond au cas où tous les récepteurs ont reçu les données après émission des T'_{SM} paquets, il ne reste donc dans ce cas aucun paquet à transmettre par voie terrestre (et S doit donc être égal à 0 pour que le calcul soit cohérent).

Avec les hypothèses exposées au 2.3.2.1, il est possible de calculer la moyenne de ces valeurs. Lorsque T'_{SM} paquets ont été envoyés par satellite vers une zone comprenant N récepteurs subissant un taux de perte PLR_{SM}, le nombre moyen de paquets à retransmettre pour ces récepteurs est donné par la formule suivante :

$$P_R = \sum_{K=T'_{SM}-T+1}^{T'_{SM}} \binom{T'_{SM}}{K} (K - T'_{SM} + T) PLR_{SM}^{K} (1 - PLR_{SM})^{T'_{SM}-K} \tag{2.14}$$

Cette formule traduit la probabilité pour chaque récepteur de recevoir $T'_{SM} - K$ paquets et de perdre les autres paquets ($0 \leq T'_{SM} - K \leq T - 1$). Il reste donc dans ce cas $(K - T'_{SM} + T)$ paquets à recevoir pour pouvoir reconstruire les données. On peut en déduire le coût moyen d'une transmission ayant utilisé le satellite pour transmettre T'_{SM} paquets.

La courbe ci-dessous montre l'évolution de ce coût en fonction du nombre moyen de récepteurs n'ayant pas encore reçu assez d'informations dans un cas particulier. Dans cet exemple, le groupe considéré est de 35000 récepteurs. Le coût est calculé de la manière suivante :

- on considère que $T' \geq T$ paquets sont transmis par satellite,

- après la transmission de ces T' paquets, le nombre moyen de récepteur n'ayant pas reçu les données est calculé,
- pour chacun de ceux-ci, le nombre moyen de paquets à retransmettre est calculé,
- les deux résultats précédents permettent de calculer le coût de la transmission terrestre complémentaire,
- le coût global est obtenu en sommant le coût de transmission des T' paquets par satellite et le coût de la transmission complémentaire associée.

Les opérations précédentes sont répétées pour différentes valeurs de T'. Dans la mesure où l'étude est focalisée sur la taille du groupe, la valeur représentée en abscisse n'est pas T', mais le nombre moyen de récepteur n'ayant pas reçu les données après émission de T' paquets (qui est directement relié à T').

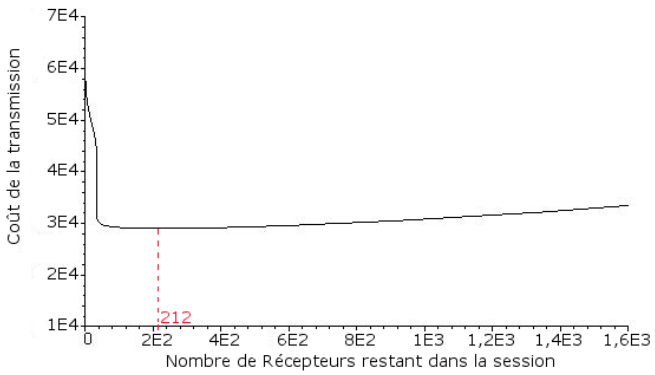


Figure 2.4 – Coût d'une transmission hybride en fonction de $N(T'_{SM})$. $N = 35000$, $T = 100$, $\alpha_{TU} = 1$, $\alpha_{TM} = 1$, $\alpha_{SM} = 200$

Les deux extrémités de cette courbe peuvent être interprétées comme suit :

- Lorsque la transmission s'est arrêtée alors qu'il restait 1 600 récepteurs, trop peu d'informations ont été envoyées par satellite. En conséquence le coût de retransmission des informations au moyen du réseau terrestre est encore relativement élevé, et il en est de même pour le coût global.
- A l'opposé, lorsque la transmission satellite s'est arrêtée alors qu'il ne restait aucun récepteur[3] toutes les informations ont été transmises avec le satellite. La transmission est dans ce cas équivalente à une transmission tout satellite, et le coût de la communication est égal à celui d'une telle communication (soit 58 840 pour un groupe de $N = 35\,000$ récepteurs selon la figure 2.2).

Entre ces deux valeurs extrêmes, la courbe présente un minimum (ce minimum est atteint ici lorsque la transmission satellite s'est arrêté alors qu'il ne restait que 212 récepteurs). Ce minimum correspond au cas où la transmission satellite a été stoppée dès que l'utilisation du réseau satellite était moins intéressante que l'utilisation du réseau terrestre. Nous avons choisi cet instant comme instant de basculement dans le calcul du coût d'une transmission hybride. Enfin, selon les résultats exposés sur

[3]En théorie, le nombre moyen de récepteurs restant dans la session n'est jamais strictement égal à 0. Il tend vers 0 lorsque $T' \to \infty$. Cependant afin de rester cohérent avec les résultats précédents, nous avons considéré que ce nombre était égal à 0 lorsque le coût de la transmission satellite dans l'approche hybride était égal au coût d'une transmission tout satellite (cf. 2.3.2.3).

cette courbe, on remarque qu'il est moins risqué d'arrêter la transmission satellite trop tôt (i.e. alors qu'il reste encore trop de récepteurs présent dans la session) que trop tard. La décroissance du coût lorsque le nombre de récepteurs diminue et se rapproche de la valeur optimale est en effet relativement lente ; tandis que l'augmentation du coût lorsque le nombre de récepteurs passe sous la valeur optimale est brusque. Le paragraphe suivant expose les résultats obtenus et les compare avec les approches classiques présentées dans la partie 2.3.

2.3.3.2 Comparaison de l'approche hybride avec les approches classiques

En choisissant le basculement tel qu'il a été présenté ci-dessus, nous avons calculé le coût d'une transmission avec les paramètres utilisés au paragraphe 2.3 : 100 paquets à transmettre, $\alpha_{TU} = \alpha_{TM} = 1$, $\alpha_{SM} = 200$, les PLR sont inchangés.

La figure 2.5 représente le coût de la transmission hybride tracé conjointement avec les coûts de la transmission en utilisant les moyens classiques. On remarque qu'un gain de coût substantiel est réalisé par rapport aux transmissions classiques (plus de 20% pour des groupes de plus de 600 récepteurs). De plus on constate que cette approche permet de diminuer la taille du groupe à partir de laquelle il est intéressant d'utiliser un lien satellite, autant pour le scénario où le service propose un service de transmission multipoint que dans le cas d'un réseau classique : le coût de l'approche hybride est inférieur au coût d'une transmission point-à-point terrestre à partir de 200 récepteurs (au lieu de 500), et coût d'une transmission multipoint terrestre à partir de 500 récepteur (au lieu de 1200).

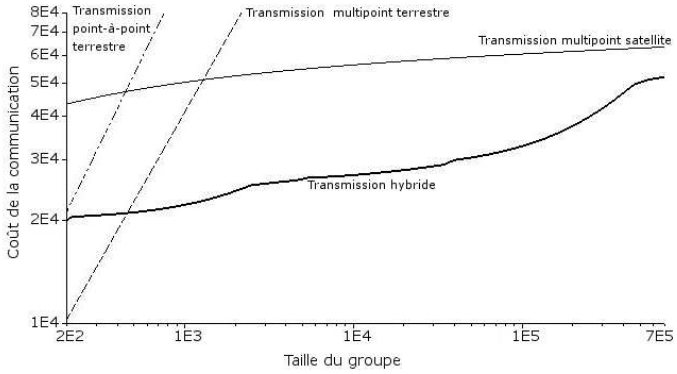


Figure 2.5 – Coût d'une transmission hybride en fonction de N. $T = 100$, $\alpha_{TU} = \alpha_{TM} = 1$, $\alpha_{SM} = 200$

Les figures 2.6 et 2.7 présentent le gain de coût obtenu avec l'approche hybride. Les résultats présentés se basent sur l'hypothèse que les retransmissions terrestres utilisent des transmissions point-à-point. Ce cas est en effet plus réaliste actuellement dans la mesure où il n'est pas commun d'avoir accès à un service terrestre de communication multipoint aujourd'hui. La courbe de coût a été comparée au minimum des deux courbes satellite et terrestre (unicast). En d'autres termes, cela revient à comparer l'approche hybride à une transmission terrestre pour des groupes de faible importance, et à la transmission satellite pour des groupes de taille plus importante. Ce gain a

été tracé pour des groupes allant de 1 à 1000 récepteurs (figure 2.6) et de 0 à 600000 récepteurs (figure 2.7), avec la fonction de coût considérée.

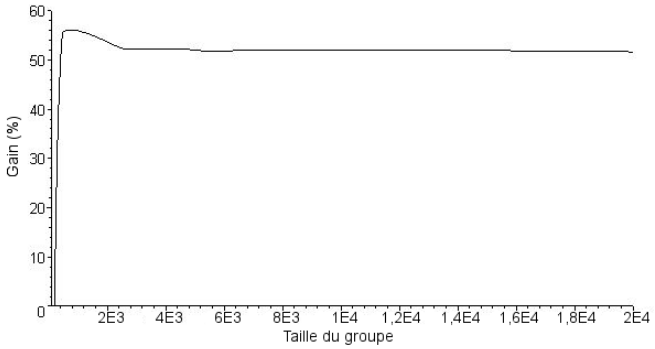


Figure 2.6 – Gain par rapport à une diffusion tout satellite ou tout terrestre, $0 \leq N \leq 20\,000$**.** $T = 100$**,** $\alpha_{TU} = \alpha_{TM} = 1$**,** $\alpha_{SM} = 200$

Nous remarquons que, avec les paramètres utilisés, l'approche hybride n'est intéressante que pour des groupes d'une taille supérieure à 200 récepteurs. De plus, avec les hypothèses effectuées pour le calcul du coût d'une transmission hybride (cf. 2.3.3.1), lorsque le groupe est trop restreint le minimum recherché correspond à une transmission purement terrestre. Le gain observé est donc de 0%. Au-delà l'utilisation de l'approche hybride devient intéressante. Il faut remarquer également que l'utilisation de cette approche permet de rendre intéressante l'emploi du satellite pour des groupes de taille plus petite que si le satellite était utilisé seul.

Pour des groupes de plus de 400 récepteurs, comme cela est visible sur les figures 2.6 et 2.7, le gain obtenu varie entre 56% et 18% environ. Nous remarquons que le gain diminue quand la taille du groupe augmente. Cette évolution est due à la définition du scénario satellite : plus le groupe augmente, plus le nombre de récepteurs subissant de lourdes pertes va augmenter. Il en résulte que la diffusion par satellite va être intéressante pendant une portion de plus en plus importante du temps total de communication, car le nombre de récepteurs couramment intéressés par la diffusion reste élevé de plus en plus longtemps.

2.3.3.3 Conclusion

Cette section a étudié le coût d'une communication hybride utilisant les réseaux terrestres et satellite pour transmettre les données. Pour ce faire, la fonction de coût présentée au chapitre 2.3 a été utilisée afin de calculer le coût d'une transmission mixte terrestre et satellite. Dans le cadre d'un scénario donné, le gain moyen induit par l'utilisation d'une telle approche a été quantifié de manière statistique. Par rapport à l'utilisation de méthodes classiques, cette approche permet d'obtenir un gain de coût allant de 17% à 50% pour la fonction de coût considérée. Cette approche particulièrement avantageuse est l'objet principal de cette thèse.

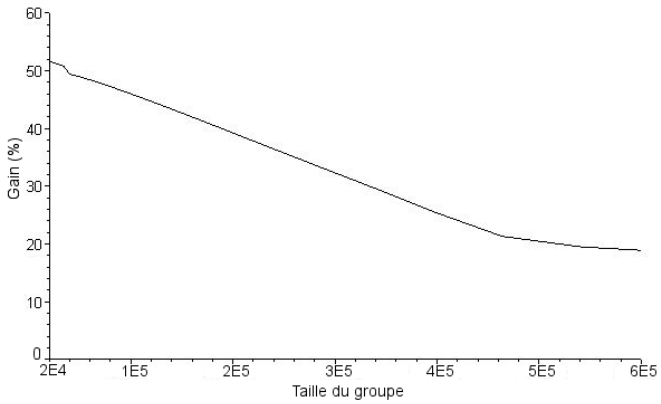


Figure 2.7 – Gain par rapport à une diffusion out satellite ou tout terrestre, $20\,000 \leq N \leq 600\,000$**.** $T = 100$**,** $\alpha_{TU} = \alpha_{TM} = 1$**,** $\alpha_{SM} = 200$

2.4 CONCLUSION : VERS UN COUPLAGE DES RÉSEAUX SATELLITES ET TERRESTRES

Dans le présent chapitre, la problématique des communications multipoints dans le contexte particulier des transmissions satellites a été étudiée. Cet environnement limite les mécanismes qui peuvent être implémentés pour rendre des services de niveau transport. Ainsi seuls des services ne requérant pas (ou peu) de contraintes temporelles peuvent être proposés. C'est en particulier le cas de la diffusion fiable de données à très grande échelle. Pour cela le satellite présente entre autres l'avantage de diffuser naturellement les informations qui lui sont transmises sur toute sa zone de couverture.

Afin de dégager l'avantage du satellite par rapport au réseau terrestre, une étude statistique étudiant le coût de communications multipoints a été réalisée. Il est apparu que les moyens de communication terrestre ou satellite peuvent tous les deux être avantageux : cela dépend de la taille du groupe de récepteurs. Pour des transmissions vers des groupes de tailles importantes (supérieures à 300 récepteurs[4]), l'utilisation des transmissions satellites est préconisée. Cependant, au cours d'une transmission satellite, le nombre de récepteurs va diminuer (à cause de la disparité des conditions de réception). Il est donc possible qu'au cours de la transmission, l'utilisation du réseau terrestre devienne plus intéressante qu'une transmission satellite.

En conséquence, nous avons imaginé et étudié une transmission qui utiliserait le médium de communication le plus avantageux en fonction de la taille du groupe. Comparée à des approches traditionnelles, cette solution s'est avérée intéressante, selon l'étude de coût réalisée dans ce chapitre. Il reste à étudier de manière plus approfondie les implications techniques de cette approche. En particulier, la nécessité de disposer de mécanismes protocolaires spécifiques apparaît. En effet de manière à être conforme aux hypothèses effectuées dans la section 2.3.2.2, cette approche requiert :

- un mécanisme permettant de gérer la fiabilité au moyen de codage FEC,
- un mécanisme permettant d'estimer la taille du groupe, et
- un mécanisme permettant de reprendre les pertes par voie terrestre.

[4] Avec les paramètres retenus pour l'étude statistique.

Une étude détaillée de ces mécanismes est présentées dans le chapitre suivant.

Chapitre 3

Conception des mécanismes spécifiques à HSTRM

3.1 INTRODUCTION : NÉCESSITÉ DE MÉCANISMES SPÉCIFIQUES À LA PROPOSITION

Comme cela a été vu dans le chapitre 2, la conception d'un protocole pose un nombre non négligeable de points durs. Cette partie se propose d'étudier plus en détail quatre mécanismes spécifiques, essentiels à l'approche hybride considérée. L'objectif est d'analyser le comportement de ces mécanismes dans l'environnement satellite. Ces travaux ont en particulier permis d'effectuer des choix lorsque plusieurs possibilités se présentaient et/ou de configurer les mécanismes choisis. Tout d'abord nous avons choisi d'utiliser des techniques de codage d'information pour gérer la fiabilité des communications. La section suivante étudie donc les différentes possibilités en la matière. Ensuite comme cela était énoncé dans le chapitre 1, le passage à l'échelle est un des problèmes les plus contraignants lors de la conception de protocoles de transport fiables. Un mécanisme permettant d'assurer le passage à l'échelle du protocole — le mécanisme de *Feedback Suppression* — est donc étudié dans la section 3.3. Le mécanisme d'estimation de la taille du groupe — de laquelle dépend toute la proposition — est étroitement liée à ce mécanisme. Le problème d'estimation de la taille du groupe est donc étudié dans cette même section. Enfin, les problèmes associés à la reprise terrestre sont abordés dans la section 3.4.

3.2 ÉTUDE DES POSSIBILITÉS EN MATIÈRE DE CODAGE AU NIVEAU TRANSPORT

3.2.1 Introduction : l'émergence des codes MDS et LDPC

Les transmissions multipoint fiables à grande échelle ont fait l'objet de nombreuses recherches [68] [98]. Parmi toutes ces recherches, l'utilisation de codes correcteurs d'erreurs est apparue comme particulièrement intéressante. Cette technique, appelée FEC (pour *Forward Error Correction*) permet en effet d'améliorer les possibilités d'utilisation à grande échelle des protocoles de transport multipoints fiables [96] [95].

L'utilisation de FEC au niveau transport a demandé une adaptation des techniques de codage d'informations traditionnellement utilisées pour la transmission de signaux [37] [41] [34](i.e. implémentées dans les couches basses). Cette adaptation est due au type de canal sous-jacent à une communication de niveau transport. Ce canal a fait l'objet d'une étude qui a abouti à la définition d'un nouveau modèle de canal de communication : le canal à effacement [35].

A l'heure actuelle, il existe deux grandes familles de codes à effacement : les codes MDS [7] (*Maximum Distance Separable* codes) et les codes LDPC [45] (*Low Density Parity Check* codes). Les codes MDS sont optimums en termes de volume de données requis par bloc pour décoder les informations reçus. Les temps de traitement de ces codes sont cependant relativement importants ce qui limite le volume de données potentiellement encodé. En réponse à ce problème, les codes LDPC on été proposés. Ces codes ne souffrent donc pas des limites inhérentes aux codes MDS, mais le volume de données requis par bloc pour décoder des informations est plus important que pour les codes MDS.

Au vu de ces caractéristiques il est légitime de s'interroger sur l'influence du choix d'un code pour une communication multipoint satellite, et en particulier pour l'approche hybride étudiée. Plusieurs aspects pratiques liés aux caractéristiques de ces codes sont donc étudiés dans les paragraphes suivants. Cette étude aboutira au paragraphe 3.2.4 à une discussion relative au choix d'un code à effacement.

3.2.2 Présentation de Codes à effacement

Cette section a pour but de décrire les codes à effacement utilisés à l'heure actuelle. Pour se faire, le modèle de canal sous-jacent à une communication de niveau transport est brièvement décrit. Ensuite quelques définitions utiles sont énoncées, ce qui permet de présenter le principe des codes à effacement au niveau transport. Enfin les sections 3.2.2.5 et 3.2.2.4 détaillent les deux familles de codes auxquelles nous nous intéressons.

3.2.2.1 Le canal à effacement

Dans les communications Internet, les protocoles des couches inférieures au niveau transport (traditionnellement appelées couches basses) détectent les erreurs de transmission au moyen de checksum (voir par exemple [9] pour la définition du checksum de

IP) ou de *Cyclic Redundancy Check* — CRC — (c'est le cas par exemple d'Ethernet [61]).

Lorsqu'une erreur est détectée, le paquet erroné est effacé. Ainsi, pour une entité de niveau transport, une communication avec une autre entité de niveau transport repose sur un canal de communication où :

- soit les paquets sont effacés dans le réseau,
- soit les paquets arrivent à destination et ne contiennent aucune erreur.

Un tel canal, nommé canal à effacement, a été introduit par Elias [35]. Il est nécessaire de bien différencier une erreur d'un effacement : une erreur correspond à un symbole faux, tandis qu'effacement correspond à un symbole manquant (aucun symbole n'a été interprété). Notons de plus que la position de l'effacement est connue (ils correspondent à des numéros de séquence manquants dans le flux de paquets reçus). En conséquence, avec une architecture classique[1], les codes placés au niveau transport n'ont pas à détecter d'erreurs : ils doivent être capables de reconstruire les paquets qui ont été effacés dans le réseau.

3.2.2.2 Quelques définitions relatives aux codes

Les définitions présentées par la suite ont pour objectif de définir clairement un vocabulaire approprié à l'étude des codes dans notre contexte. De plus le simple rappel des principes du codage permet de mieux comprendre les propriétés et les performances des codes utilisés au niveau transport.

Définition 3.1 *Tout code $\mathcal{C}$ est défini comme étant un sous-ensemble d'un ensemble $\mathcal{A}^n$.*

$\mathcal{A}$ est appelé *alphabet*, et ses éléments sont appelés les *symboles*. Les éléments de $\mathcal{C}$ sont appelés les *mots du code*. Le principe d'un code est d'associer à chaque mot de $\mathcal{C}$, un ensemble de mots de $\mathcal{A}^n$. Ainsi, lorsqu'un mot m de $\mathcal{C}$ est entaché d'erreur après avoir été transmis, il appartient à $\mathcal{A}^n$. Soit m' un mot reçu entaché d'erreurs. Si le code est bien conçu et qu'il n'y a pas trop d'erreurs, m' doit appartenir à l'ensemble des mots associés à m. La réception de m' permet dans ce cas à l'entité réceptrice d'interpréter m' comme étant le symbole m entaché d'erreurs, ou en d'autre termes de corriger les erreurs.

Tout code est caractérisé par :

- sa longueur n,
- son nombre de mots M, ou, dans le cas d'un code linéaire, sa dimension k, et
- sa distance minimale d.

Définition 3.2 *Un code de longueur n est un sous-ensemble de $\mathcal{A}^n$.*

Définition 3.3 *La dimension k d'un code linéaire est la dimension du sous-espace vectoriel formé de ses mots. Lorsque le code n'est pas linéaire, on considère à la place de k le nombre de ses mots M ou la valeur* $\log_{|\mathcal{A}|}(M)$.

[1] Plusieurs travaux étudient la possibilité d'invalider les contrôles d'intégrité des couches basses, de manière à faire remonter les erreurs bits au niveau transport ou applicatif. C'est le cas par exemple d'UDP lite [72]

Notons que ces valeurs sont liées au volume d'information contenu dans les mots du code — composés de n symboles — transmis.

La distance minimale d'un code fait appel à la notion de distance de Hamming. La distance de Hamming entre deux vecteurs de $\mathcal{A}^n$ est le nombre de composantes de ces vecteurs qui diffèrent. Notons que lorsqu'un mot m est entaché d'erreurs, si m' est le mot reçu, le nombre d'erreurs contenues dans m' est la distance de Hamming entre m et m'.

Définition 3.4 *La distance minimale d'un code est la distance de Hamming minimale entre deux mots du codes*[2]*.*

Parmi l'ensemble des codes existant, il existe un type de code particulièrement utilisé : les codes linéaires. Ces codes sont définis sur un alphabet $\mathcal{A}$ qui est un corps fini. La propriété principale de $\mathcal{A}$ est d'être un sous-espace vectoriel de $\mathcal{A}^n$. L'intérêt est de pouvoir caractériser ces codes par une matrice, dite matrice génératrice. La matrice génératrice d'un code linéaire de dimension k et de longueur n est une matrice $k \times n$ de rang k. Dans ce cas, pour k symboles initiaux $(s_{i_1}, \cdots, s_{i_n})$, on obtient les n symboles encodés $(s_{c_1}, \cdots, s_{c_n})$ en multipliant le vecteur formé des k symboles initiaux par la matrice génératrice (cf. figure 3.1). De plus, lorsque les k premiers symboles de $(s_{c_1}, \cdots, s_{c_n})$ correspondent aux symboles initiaux, ont dit que le code est *systématique*. Un autre intérêt lié à l'utilisation des codes linéaires est l'utilisation pour le décodage de la matrice de parité (ou matrice de test). Celle-ci est la matrice génératrice du sous-espace vectoriel orthogonal au sous-espace vectoriel correspondant au code.

$$\begin{bmatrix} s_{i_1} & \cdots & s_{i_k} \end{bmatrix} \times \begin{bmatrix} g_{1,1} & \cdots & g_{1,n} \\ \vdots & \ddots & \vdots \\ g_{k,1} & \cdots & g_{k,n} \end{bmatrix} = \begin{bmatrix} s_{c_1} & \cdots & s_{c_n} \end{bmatrix}$$

Figure 3.1 – Codage avec un code linéaire

3.2.2.3 Principe de l'utilisation des codes au niveau transport

Le paragraphe précédent rappelait en quelques mots comment obtenir un mot encodé lorsqu'un code linéaire est utilisé. L'utilisation de tels codes au niveau transport a demandé une adaptation particulière. L'implémentation logicielle de tels codes implique en effet certaine souplesse d'utilisation pour que les opérations de codage et de décodage ne soient pas trop contraignantes en termes de calculs. Ainsi lorsque c'est tout un ensemble de paquets qui est codé, on ne peut pas procéder en mettant les paquets ¡¡ bout-à-bout ¿¿ pour obtenir un mot (dont la taille serait très importante) à encoder. Plutôt que cela, les paquets sont regroupés en groupes de k, et le j^{ieme} symbole de chaque paquet est utilisé pour former un mot. Ce principe est illustré dans la figure 3.2 (où Λ représente le nombre de symboles compris dans un paquet de données). Dans la suite de cette section les qualificatifs ¡¡ i ¿¿ et ¡¡ c ¿¿ ont été ajoutés en indice aux symboles afin de distinguer les symboles initiaux (i) des symboles encodés (c).

[2]En d'autres termes la distance minimale d'un code est le nombre minimal de symboles qui diffèrent entre deux mots du code.

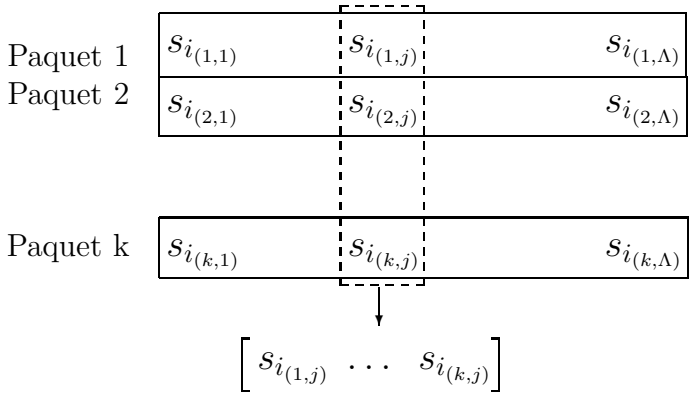


Figure 3.2 – Définition des mots à encoder

Un mot codé est obtenu en multipliant le mot initial par la matrice génératrice M_G. Un ensemble de symboles $(s_{c_{1,j}}, \cdots, s_{c_{n,j}})$ est donc obtenu. Cette opération est réitérée pour $1 \leq j \leq d$ (où d représente le nombre de symboles contenus dans le champ données d'un paquet). Ce procédé permet d'obtenir n paquets encodés, avec $k \leq n$ (voir figure 3.3). Il faut remarquer que lorsque le code est systématique, les k premiers paquets correspondent aux k paquets initiaux. L'avantage lié à l'utilisation de tels codes est donc que tout récepteur ayant reçu les k premiers paquets n'aura pas à faire appel à une technique de décodage complexe.

$$\left[si_{(1,j)} \ \cdots \ si_{(k,j)} \right] \times \left[M_G \right] = \left[sc_{(1,j)} \ \cdots \ sc_{(n,j)} \right]$$

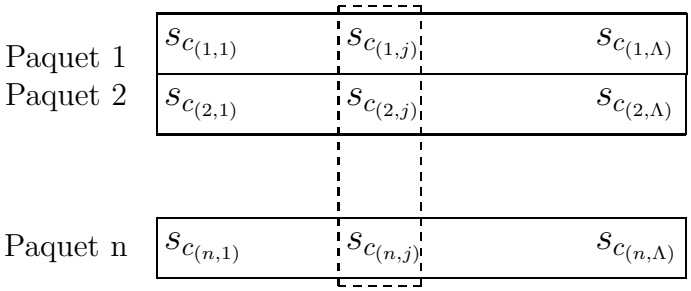


Figure 3.3 – Principe du codage par paquets

La technique exposée ci-dessus permet d'obtenir n paquets. Cependant lorsqu'un code est utilisé au niveau transport, il est rarement nécessaire de générer n paquets. Au lieu de cela, l'entité utilisant le code préfère généralement pouvoir générer x paquets sur commande avec $x \leq n$. En reprenant la technique représentée sur la figure 3.3 on remarque que pour obtenir un paquet x quelconque, avec $1 \leq x \leq n$ il suffit de multiplier les vecteurs $[s_{i_{1,j}}, \cdots, s_{i_{n,j}}]_{1 \leq j \leq d}$ par la x^{ieme} colonne de la matrice génératrice. On dit alors que le code est tronqué, puisque seulement une partie des symboles pouvant être générés est utilisée.

Cette section a présenté rapidement les adaptations des techniques de codage classiques pour un codage par paquets. En résumé, avec les techniques définies, il est

possible de générer à la demande un ensemble de paquets encodés. La réception d'un nombre suffisant de paquets encodés permet de reconstruire les paquets initiaux à partir desquels ils ont été générés.

3.2.2.4 Les codes dits ¡¡ Maximum-Distance Separable ¿¿

Les codes à effacement MDS sont une classe importante de codes car ce sont des codes optimaux en termes de capacité de correction. Les codes de Reed-Solomon (voir par exemple [99] p.138) sont un exemple très répandu de codes MDS. Les codes de Reed-Solomon sont entre autre utilisés pour fiabiliser les transmissions satellites avec le standard DVB-S [34].

Les codes MDS sont construits sur des corps finis (i.e. $\mathcal{A}$ est un corps fini). En pratique ce sont les corps de Galois à 2^p éléments, notés $\mathbb{F}_{2^p}$, qui sont utilisés. Un tel corps étant formé de 2^p éléments, tout élément de $\mathbb{F}_{2^p}$ peut être représenté par un vecteur du type $(c_1, \cdots, c_p)$ avec $\{c_i \in \{0,1\},\ 1 \leq i \leq p\}$. Cela permet donc d'associer facilement les symboles manipulés à des suites de p bits (pour plus d'informations, le lecteur intéressé est invité à regarder un ouvrage spécialisé, tel que [99]). Notons en particulier que lorsque $\mathbb{F}_{2^8}$ est utilisé, les symboles manipulés sont associés à des octets.

Un code de longueur n, de dimension k, et de distance minimale d sur un corps fini est dit MDS (Maximum Distance Separable) si $d = n - k + 1$. La valeur $n - k + 1$ est connue comme étant la *borne du singleton* : c'est la valeur maximale de d, quel que soit le code utilisé. Intuitivement ces codes correspondent donc à des codes où les mots du codes sont les ¡¡ plus espacés possibles ¿¿. Cela permet donc de maximiser les capacités de correction, dans la mesure où les mots reçus restent généralement proches des mots émis. Notons que les codes MDS sur $\mathbb{F}_{2^p}$ ont pour longueur $2^p - 1$, et permettent de générer jusqu'à $2^p - k$ symboles de redondance. La dimension du code est un paramètre que l'utilisateur peut définir selon ses besoins.

Ainsi lorsque ces codes sont utilisés au niveau transport, pour x paquets générés à partir de k paquets initiaux, la réception de tout sous-ensemble de k paquets est suffisante pour décoder les informations (et reconstruire les k paquets initiaux). Cette propriété a suscité l'intérêt de la communauté réseau, et plusieurs implémentations de ces codes ont été proposées [114] [113] [8].

Cependant le nombre maximum de paquets qu'il est possible de générer est limité, car les temps de traitement deviennent excessifs lorsque n augmente. En pratique, ces codes sont donc utilisés avec $p \leq 16$. Traditionnellement les valeurs $p = 8$ et $p = 16$ sont retenues afin d'associer les symboles manipulés à des octets ou des bi-octets. Le fait que n soit limité a un impact sur les performances des mécanismes de codage au niveau transport : comme $k < n$ et $n \leq 2^p - 1$, le nombre de paquets qu'il est possible d'encoder est limité. De plus lorsque le mécanisme HARQ2 est utilisé, il n'est possible de reconstruire les informations que si au plus $n - k$ paquets sont perdus. Si d'avantage de pertes surviennent il devient nécessaire d'utiliser un mécanisme de requête de retransmission classique car il n'est plus possible de générer des paquets qui n'ont pas été transmis. Le choix de ces paramètres doit donc être soigneusement effectué.

3.2.2.5 Les codes à matrice de parité creuse

Les codes LDPC ont été introduits en 1962 par Gallager [45]. La proposition des codes Tornado [11], qui correspond à la première introduction du principe des codes LDPC dans les communications multipoint fiables est par contre relativement récente. Cela a conduit à l'étude de nombreuses propositions, chacune visant à améliorer les performances des codes précédents. C'est le cas par exemple des *LT codes* [80] et des *Raptor codes* [124].

En pratique, les codes LDPC utilisés au niveau transport sont des codes linéaires qui utilisent de simples bits comme symboles. Les matrices génératrice et de parité de ces codes sont choisies creuses : elle est majoritairement composée de 0, et de quelques 1. Pour le symbole de redondance $s_{c_{x,j}}$, conformément à la figure 3.3, seuls les symboles dont la place correspond à un 1 sur la colonne numéro j de la matrice génératrice vont être impliqués dans la génération du symbole $s_{c_{x,j}}$. plus précisément, le symbole de contrôle va être le résultat du ¡¡ ou exclusif ¿¿ (ou *x-or*, noté $\oplus$) de tous les symboles restants. La matrice étant creuse, le calcul de $s_{c_{x,j}}$ ne va impliquer que quelques symboles. De plus le résultat de l'opération $\oplus$ est très rapide à calculer. Cela permet de comprendre pourquoi les vitesses d'encodage et de décodage de ce type de code sont très élevées.

Ces codes présentent un autre avantage : le nombre de symboles encodés qu'il est possible de générer avec k paquets est très élevé. On peut en effet associer la génération d'un symbole à une combinaison linéaire de x symboles avec $x << k$ (cette relation traduit le fait que la matrice soit creuse). Or il est possible de générer C_k^x combinaisons linéaires différentes avec x symboles choisis parmi k. Comme x peut prendre plusieurs valeurs (tant que x reste faible par rapport à k), le nombre de symbole de contrôle qu'il est possible de générer est considérable.

Par contre, le fait que la matrice soit creuse a un impact néfaste sur la capacité de correction. Considérons par exemple le cas — simple — où le symbole de redondance $s_{c_{n,j}}$ est une combinaison des symboles $s_{i_{1,j}}$, $s_{i_{3,j}}$ et $s_{i_{4,j}}$. Dans ce cas ce symbole $s_{c_{n,j}}$ sera inutile à tous les récepteurs ayant correctement reçu les symboles $s_{i_{1,j}}$, $s_{i_{3,j}}$ et $s_{i_{4,j}}$: la réception de ce symbole ne fournit aucune informations sur les autres symboles initiaux. Il faudra donc qu'ils attendent de recevoir d'avantage d'informations s'ils n'ont pas réussi à décoder les autres symboles. En conséquence, à la différence des codes précédents, les codes de type LDPC ne sont pas optimaux : pour n paquets générés à partir de k paquets, il est nécessaire de recevoir en moyenne $k(1+\epsilon)$ paquets parmi les n paquets pour reconstruire les données initiales. Bien que ϵ tende vers zéro quand k tend vers l'infini, pour des valeurs raisonnables de k, ϵ est de l'ordre de 5% [104]. C'est pour résoudre ce problème que les auteurs des codes Tornado ont proposé d'imbriquer plusieurs codes LDPC : les symboles de redondance reçus peuvent ainsi permettre de remonter jusqu'à d'autres symboles, et de décoder d'autres symboles initiaux.

3.2.2.6 Conclusion

Les objectifs qui ont conduit à la conception des codes MDS et LDPC sont donc radicalement différents. Comme la description théorique de ces deux types de code

ne permet pas de trancher en faveur de l'un ou de l'autre, une étude pratique est nécessaire. L'étude des aspects pratiques liés à l'implémentation de ces deux types de codes est présentée dans la partie suivante.

3.2.3 Étude des performances des codes MDS et LDPC

3.2.3.1 Temps de traitements

Une implémentation standard d'un code MDS [8] a été utilisée de manière à évaluer les débits obtenus en sortie d'un encodeur et d'un décodeur avec un système récent. Les tests ont été effectués en utilisant un Pentium avec une fréquence de 1 GHz, et les codes testés étaient limités à $n = 2^{16}$ (de manière à ce que le nombre de paquets potentiellement généré ne soit pas trop limité). Avec cette configuration le débit de sortie obtenu était supérieur à 1 Mo/s pour des valeurs de k allant de 16 à 256. L'utilisation de tels codes semble donc raisonnable jusqu'à $k = 256$, tant que le débit de transmission est inférieur à 1 Mo/s[3]. Comparé à cela, selon [104] les codes LDPC proposent des débits de l'ordre de plusieurs centaines de Mo/s (cet ordre de grandeur a été confirmé par une implémentation non standard d'un code LDPC).

3.2.3.2 Occupation de la mémoire

L'occupation de la mémoire a été évaluée avec la même configuration que celle du chapitre précédent. Une part de la mémoire est occupée par le stockage d'objets nécessaires au codage et au décodage. Pour les deux codes le volume de mémoire requis pour stocker ces objets a été évalué à moins de 1Mo, le stockage de ces objets est donc peu contraignant par rapport au stockage des données.

Il faut ajouter à cela le stockage tous les paquets qui doivent être encodés. Or dans le cadre de communications satellites, comme les erreurs surviennent en séquence, il est recommandé d'entrelacer plusieurs blocs de k paquets. C'est dans ce but que les blocs sont regroupés en Méta-blocs (un méta-bloc est composé de T_E blocs, cf. 4.4) lors de l'envoi de paquets. Ceci permet de répartir les pertes parmi ces blocs et d'éviter de perdre plus de paquets que le maximum acceptable (i.e. $n-k$ par bloc). Avec un débit en émission D, TDU octets envoyés par paquets, et des atténuations dont la durée cumulée est P_{Att}, cette condition s'exprime comme :

$$\frac{T_E \times TDU \times (n-k)}{D} \geq P_{Att}$$

Ce qui implique que :

$$T_E \geq \frac{D \times P_{Att}}{TDU \times (n-k)} \tag{3.1}$$

Afin d'évaluer la valeur de T_E pour des cas réalistes, nous avons utilisé les résultats présentés dans [103]. Toutes les atténuations recensées dans cet article n'excèdent pas 6 000 secondes. Avec une taille de paquet de $TDU = 1\,500$ octets, $k = 256$, $n = 2^{16}$

[3]Un code MDS est actuellement en cours de développement au Département Mathématiques et Informatique de l'ENSICA. Ce code permet d'obtenir des débits d'encodage et de décodage supérieurs à 1 Mo jusqu'à $k = 2048$ [24].

et un débit d'émission de 1Mo/s l'équation 3.1 aboutit à $T_E \geq 61,27$. Ainsi 62 blocs doivent être entrelacés pour qu'au maximum $n-k$ paquets soient perdus au cours de la communication. L'entrelacement des paquets a bien entendu un impact sur le volume de mémoire que le mécanisme de codage va utiliser. Ainsi pour T_E blocs entrelacés, chaque récepteur doit stocker en mémoire $T_E \times k \times TDU$ octets. Avec les valeurs adoptées ci-dessus, le volume de mémoire occupé est approximativement de 24Mo. En considérant qu'une atténuation de 6 000 secondes correspond au cas le pire (en termes de durée d'atténuation), un tel entrelacement permet de répartir suffisamment les pertes quel que soit l'atténuation survenant pendant la communication. De plus l'entrelacement de 62 blocs ne pose pas de problème majeur de stockage des données.

Les auteurs de [56] ont montré que l'entrelacement de plusieurs blocs de paquets pouvait entraîner un surcoût de réception. Cependant dans notre contexte le mécanisme HARQ2 implique que le nombre exact de paquets manquants est retransmis, et ce pour chaque bloc. De plus les résultats présentés dans [56] sont basés sur des pertes uniformément réparties, et non des pertes en séquence. Les hypothèses retenues dans [56] ne sont donc pas applicables au contexte satellite.

Les codes LDPC peuvent encoder d'un bloc un nombre très élevé de paquets. Cependant lorsque k devient élevé, le volume de mémoire requis pour stocker les paquets peut devenir excessif (tout dépend de l'implémentation). Pour cette raison, nous avons considéré dans la suite que k était inférieur ou égal à 50 000. Une telle valeur de k est suffisante pour que le surplus de données requis pour décoder les informations reçues ne soit pas trop élevé, et le stockage de tous les paquets correspond approximativement à 75Mo, ce qui reste une valeur raisonnable.

3.2.3.3 Données ajoutées à l'en-tête du protocole de transport

Toutes les informations qui doivent être communiquées aux récepteurs pour qu'ils puissent décoder les paquets sont définies dans la RFC 3452 [79]. Les données nécessairement ajoutées à l'en-tête tous les paquets du protocole de transport sont référencées en tant que *FEC Payload ID*. Les schémas prédéfinis 128 et 129 spécifient une structure possible pour les informations liées au codage en utilisant deux octets. Avec ces schémas, il est nécessaire d'envoyer au moins 8 octets pour identifier un paquet avec un code LDPC, et au moins 6 octets pour un code MDS.

3.2.3.4 Données générées sur la voie retour

Le nombre k de paquets qu'il est possible d'encoder dans un bloc a un impact sur le volume du trafic généré sur la voie retour. Le mécanisme HARQ2 suppose en effet que la source est informée du nombre de paquets manquants pour chaque bloc. Si le même mécanisme est utilisé pour en informer la source sans poser de problème de mise à l'échelle, le même nombre moyen N_{Back} de paquets sera transmis sur la voie retour. Avec les structures d'identification définies par le RMT-WG, pour chaque bloc il est nécessaire d'envoyer : 8 octets pour un code LDPC, et au minimum 6 octets pour un code MDS. Le rapport minimum entre le trafic généré sur la voie retour par un code

MDS et un code LDPC est donc :

$$R = \frac{B_{MDS} \times 6 \times N_{Back}}{B_{LDPC} \times 8 \times N_{Back}}$$

où B_x est le nombre de blocs envoyés (qui dépend du type de code utilisé). B_x peut être calculé en fonction de la taille du fichier transmis, des paramètres du code, et de la taille des paquets :

$$B_x = \left\lceil \frac{FS}{T_B \times TDU} \right\rceil, x \in \{LDPC, MDS\}$$

où FS est la taille du fichier, T_B le nombre de paquets par bloc. Comme cela est visible sur la figure 3.4 ce rapport est au moins égal à 70 pour de fichiers dont la taille est supérieure à 100Mo. Les grandes variations visibles sont dues à l'opérateur $\lceil - \rceil$.

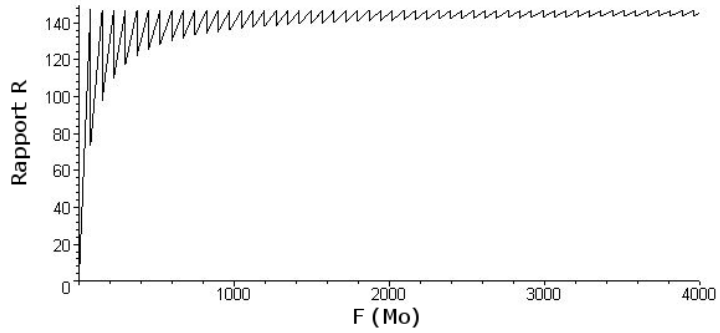


Figure 3.4 – Évolution du rapport du trafic généré sur la voie retour en fonction de la taille du fichier transmis. $k_{MDS} = 256$, $k_{LDPC} = 50\,000$, $TDU = 1500$ **octets**

3.2.4 Discussion sur le choix d'un code à effacement

Les résultats présentés ci-dessus permettent d'arriver à une conclusion simple : dans le contexte de l'étude, les codes MDS permettent d'optimiser l'utilisation de la bande passante, tandis que les codes LDPC permettent de relâcher les contraintes imposées aux utilisateurs finals. Le choix du code est donc laissé au concepteur du protocole de transport. Si les ressources réseaux ne sont pas trop coûteuses, l'utilisation de codes LDPC permet d'améliorer la satisfaction des utilisateurs : comme les temps de codage et de décodage sont rapides, les informations sont plus rapidement disponibles. Cependant dès que des pertes surviennent, l'utilisation de tels codes peut entraîner un gaspillage de bande passante. Ce gaspillage est dû à l'approche probabiliste des codes : plus le récepteur reçoit de paquets, plus la probabilité pour qu'il puisse décoder les informations augmente. Il est cependant difficile de garantir qu'après avoir envoyé x paquets les récepteurs aient décodé les informations. L'objectif de l'approche hybride présentée étant de limiter les ressources réseau utilisées, l'utilisation d'un code MDS semble préférable. Dans ce cas le débit en émission ne doit pas dépasser 1Mo/s (avec la configuration testée qui correspond à un ordinateur de capacité très moyenne aujourd'hui). La proposition se veut cependant flexible. Les valeurs adoptées pour les champs

Source Bloc Number et **Encoding Symbol ID** ont donc été choisies de manière à permettre l'utilisation des deux types de codes, tout en limitant le volume de données ajouté à l'en-tête des paquets.

Lorsque la bande passante est supérieure à 1Mo/s, une solution consiste à effectuer l'encodage avant le début de la transmission afin d'avoir tous les paquets déjà disponibles lorsque la session commence. Cependant cette technique demande de stocker tous les paquets de redondance générés. S'il n'est pas possible de générer au préalable ces paquets parce que l'espace mémoire n'est pas disponible, ou parce que la source est trop sollicitée pour effectuer le codage des paquets, il est possible par exemple de faire appel à un autre calculateur pour générer ces paquets. Si cette dernière solution n'est pas possible, seule l'utilisation de codes LDPC permet d'utiliser toute la bande passante disponible. Notons cependant qu'un débit en émission de 1 Mo/s est une valeur tout à fait acceptable au vu des débits en émission proposés à l'heure actuelle.

L'encodage des paquets est effectué par la source, lorsque cela est nécessaire, de manière à ce qu'ils soient disséminés sur tout le groupe. Toutefois, afin de connaître le volume de données à retransmettre, la source doit être en mesure de recevoir des rapports de réception (sans pour autant que le réseau terrestre soit saturé). De plus, de manière à déterminer le moment où la phase de transmission satellite doit s'arrêter, la source doit être capable d'estimer le nombre de récepteurs en attente d'informations. L'étude de ces problèmes est menée dans la partie suivante.

3.3 ESTIMATION DE TAILLE DE GROUPE ET LIMITATION DU TRAFIC RETOUR

3.3.1 Le risque lié à l'utilisation d'acquittements négatifs

La gestion des messages émis sur la voie retour est un des problèmes les plus contraignants pour les protocoles de communication multipoint. Plusieurs approches ont été proposées pour résoudre ce problème. La première idée a consisté à émettre des demandes de retransmission (ou NACK : Negative ACKnolgement) plutôt que des acquittements pour chaque message reçu. Cette idée était basée sur l'hypothèse que seule une minorité de paquets étaient perdus dans le réseau, et que les pertes subies par les récepteurs étaient décorrélées. Cependant, lorsqu'un arbre de diffusion est utilisé pour des transmissions multipoints, un risque subsiste. En effet, dans ce cas, les pertes survenant au milieu de l'arbre de diffusion sont subies par tous les récepteurs situés dans la partie l'arbre dépendant du lien où a eu lieu la perte. Le même paquet peut donc être perdu par un ensemble de récepteurs, et même par un très grand nombre de récepteurs. Il a donc fallut concevoir des mécanismes plus robustes.

3.3.2 Mécanisme de limitation du trafic retour

3.3.2.1 Mécanismes existants

Puisque l'envoi de messages sur la voie retour pose des problèmes, une idée naturelle consiste à étudier la possibilité de supprimer ces retours. Au niveau transport, cela se

traduit par l'introduction de techniques de codage. Ainsi la couche transport envoi un volume de données encodées de manière à ce que statistiquement tous le récepteurs reçoivent assez d'information pour reconstruire les données initiales. Cette technique a particulièrement été préconisée pour les transmissions satellite [66]. Cependant dans ce cas le protocole de transport n'assure pas une fiabilité totale pour tous les récepteurs. De plus cette technique implique potentiellement un gâchis de bande passante (par exemple lorsque qu'aucun récepteur n'a subi de pertes).

Plusieurs travaux ont donc étudié la possibilité d'envoyer des messages sur la voie retour sans pour autant saturer le réseau. Une des idées proposées par le groupe de travail RMT consistait à organiser l'ensemble des récepteurs en arbre logique. Cette approche était nommé TRACK pour *Tree-Based ACK protocol*. Avec cette solution, chaque récepteur transmet ses requêtes de retransmission au nœud dont il dépend. Chaque nœud satisfait la requête, ou la transmet à son propre père et ainsi de suite. Cette solution, si elle permet en théorie de réduire le nombre de messages émis simultanément vers la source pose de problèmes de délais de réponse. En effet, entre le moment où un récepteur émet une requête et le moment où il reçoit une réponse, un temps considérable peut s'écouler. Cette proposition a fini par être jugée inintéressante par le groupe de travail RMT, et les travaux ont été abandonnés.

Une autre solution a été proposée au sein de l'éditeur de texte NTE [55] qui fait partie des outils développés pour le MBone. Cette solution a été référencée dans [44] comme ¡¡ mécanisme à probabilité contrôlée ¿¿. Après l'envoie de chaque objet, la source transmet une demande de requête de retransmission. Avec ce message, la source envoie une probabilité de réponse p_r. A la réception de ce message, les récepteurs qui n'ont pas reçu l'objet en question envoient un acquittement négatif avec la probabilité p_r. Notons que comme chaque récepteur peut ne pas répondre, la source n'est pas assurée de recevoir une réponse (tant que $p_r < 1$). Si la source ne reçoit aucun message de réponse après un certain temps d'attente, elle renvoie une demande de retour en augmentant p_r. Le mécanisme continue ainsi jusqu'à ce que la source reçoit un message ou jusqu'à ce que $p_r = 1$. Notons pour ne pas risquer saturer le réseau, la probabilité de départ doit être faible. Mais cela implique que lorsque seuls quelques récepteurs veulent envoyer une requête à la source, plusieurs rounds vont s'écouler avant qu'ils envoient effectivement un message. Ce système n'est donc pas très adapté pour un système satellite où les délais de transmission sont déjà très importants, car les délais d'attente peuvent devenir prohibitifs.

La dernière proposition en la matière a consisté à introduire l'utilisation de timer. Le principe de ce mécanisme, introduit dans [93], est assez similaire au mécanisme précédent, si ce n'est que le tirage aléatoire porte sur un temps d'attente :

- la source envoie une demande de requête de retransmission,
- à la réception de ce message les récepteurs génèrent un temps d'attente selon une fonction de répartition $f(x)$ prédéfinie,
- le récepteur ayant tiré le temps d'attente le plus faible envoie un message à la source,
- dès la réception d'une réponse, la source demande aux récepteurs d'annuler l'envoie d'une réponse.

Par rapport à la solution précédente, la source n'a qu'à envoyer un seul message pour être sûre d'avoir une réponse : la définition de $f(x)$ permet de contraindre le temps maximum que les récepteurs peuvent attendre avant de répondre. Cette solution, qui semble plus adaptée au contexte satellite est décrite plus en détail dans la section suivante.

3.3.2.2 Mécanise utilisant des timers

Cette section expose les résultats présentés dans [93], utiles dans la suite de ce chapitre. Dans la suite, nous appelons :

- c le temps d'aller-retour entre la source et un récepteur, et
- **TMaxInquiry** la durée d'attente maximales avant qu'un récepteur réponde.

Avec ces notations, la source ne reçoit que les messages envoyés entre x_1^* et $x_1^* + c$, où x_1^* correspond au temps d'attente le plus faible (voir figure 3.5). Plusieurs fonctions de

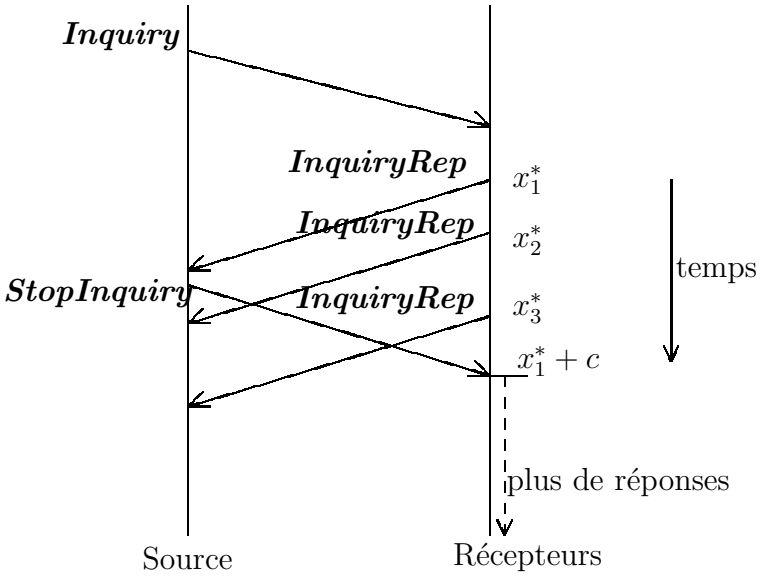


Figure 3.5 – Principe de fonctionnement du mécanisme de *Nack Suppression*

répartition on été testées pour ce mécanisme. Parmi celles-ci, la fonction de répartition exponentielle tronquée est apparue comme donnant les meilleurs résultats en termes de mise à l'échelle et de délais de retour. Dans ce cas la fonction de répartition des temps d'attente $f(x)$ est la suivante :

$$f(x) = \begin{cases} \frac{1}{e^{\lambda}-1} \frac{\lambda}{\text{TMaxInquiry}} e^{\frac{\lambda}{\text{TMaxInquiry}} x} & , 0 \leq x \leq \text{TMaxInquiry} \\ 0 & , sinon \end{cases}$$

Avec cette fonction de répartition, les paramètres λ et **TMaxInquiry** permettent de spécifier complètement le nombre moyen de messages émis sur la voie retour, ainsi que la durée maximum d'attente. Notons en particulier que, lorsque **TMaxInquiry** est fixé, le nombre moyen de messages émis passe par un minimum dépendant de λ. De

plus l'impact de **TMaxInquiry** sur la valeur de λ permettant d'atteindre ce minimum. Plus précisément, la valeur de λ optimale peut être approchée par une fonction affine du logarithme du nombre de récepteurs N :

$$\lambda_{opt} = 1.1 \ln(N) + 0.8 \tag{3.2}$$

Une fois placé à cet optimum, le choix de **TMaxInquiry** permet de spécifier le nombre moyen de messages générés. En approximant le nombre moyen de messages générés grâce à un développement limité, les auteurs de [93] ont exprimé le lien entre le nombre R de messages générés et les autres paramètres :

$$\textbf{TMaxInquiry} = \begin{cases} \frac{\lambda c}{\ln\left(R+\frac{N}{e^{\lambda}-1}\right)-\ln\left(1+\frac{N}{e^{\lambda}-1}\right)} & \text{si } R < N, \\ 0 & \text{si } R \geq N \end{cases} \tag{3.3}$$

Avec ces résultats, il possible de configurer complètement le mécanisme. Cependant, l'utilisation de ce mécanisme dans un environnement satellite implique une configuration spéciale.

En particulier, le type de pertes inhérentes aux liaisons satellites impose que ce mécanisme soit associé à une *approche explicite*. Nous entendons par là que la source doit explicitement envoyer des messages pour demander aux récepteurs d'envoyer des rapports de réception. Cette approche contraste avec une approche implicite où les récepteurs envoient régulièrement des messages sans que la source le leur demande. Avec une telle approche, les récepteurs subissant des pertes génèrent des temps d'attente alors qu'ils ne sont pas en mesure de recevoir les messages ***StopInquiry*** envoyés par la source. Tous ces récepteurs vont donc générer des messages. Dans la mesure où les pertes peuvent toucher un grand nombre de récepteurs, cela peut conduire à une saturation du réseau.

Notons que ce mécanisme permet également d'avoir des informations sur taille du groupe en étudiant les réponses reçues. En effet, le nombre de réponses générées et les temps d'attente générés dépendent grandement du nombre de récepteurs. La partie suivante se propose d'étudier le problème d'estimation basé sur l'utilisation de ce mécanisme.

3.3.3 Estimation de taille de groupe

3.3.3.1 Importance de l'estimation de la taille de groupe

Le fait de disposer d'un mécanisme permettant d'estimer la taille du groupe est un besoin majeur dans l'approche hybride proposée. Sans un tel mécanisme il n'est en effet pas possible de détecter le moment où la taille du groupe devient inférieure au seuil **MinRcv** prédéfini. Plusieurs estimateurs ont déjà été étudiés au cours des dernières années. Cependant, a notre connaissance leur efficacité dans les réseaux hybrides satellites/terrestres n'a pas été étudiée. La suite de cette section est focalisée sur les problèmes liés à l'estimation de taille de groupe dans ce contexte spécifique. Tout d'abord dans la section suivante le problème est formalisé. Plusieurs estimateurs — basés sur l'utilisation du mécanisme présenté dans la section 3.3.2 — sont ensuite

étudiés dans la partie 3.3.3.3. L'interaction entre le mécanisme d'estimation de taille de groupe et le mécanisme de limitation des retours est évalué au moyen de simulations dans la partie 3.3.3.4.

Friedman et Towsley ont fait remarquer dans [43] que l'hétérogénéité des délais de transmission avait un impact négatif sur la qualité des estimations réalisées. Ils ont proposé pour remédier à ce problème une correction de la procédure de réponse. Nous faisons l'hypothèse par la suite que cette procédure est implémentée dans le mécanisme de limitation des retours. Notons qu'il existe des propositions visant à intégrer un service de comptage des membres d'un groupe *IP Multicast* directement au niveau réseau [39]. Dans la mesure où cela n'est pas représentatif des réseaux en place actuellement, nous avons choisi de considérer le cas où un tel service n'était pas disponible.

3.3.3.2 Formulation du problème d'estimation

Caractéristiques du système

Le mécanisme d'estimation se base sur le mécanisme de limitation de retours vers la source tel qu'il a été présenté dans la section 3.3.2. En particulier la source envoie des messages ***Inquiry*** auxquels seuls les récepteurs capables de recevoir des messages répondent. Dans la suite, nous appelons récepteur *actif* tout récepteur en attente d'information et en mesure de recevoir les messages de la source. Le nombre de récepteurs actifs est noté N_A. Les autres récepteurs en attente d'information sont dit *inactifs*. Nous rappelons également que la phase de transmission satellite est divisée en deux phases :

- Elle commence par la transmission intégrale de l'objet, ce qui représente le volume d'informations minimal à transmettre. Au cours de cette phase la taille du groupe en attente d'informations ne peut diminuer (car personne ne peut avoir reçu toutes les données avant la fin de cette phase).
- Une fois cette phase terminée, commence la transmission d'informations redondantes permettant de corriger les pertes. Le nombre de récepteurs en attente d'information ne peut que diminuer au cours de cette phase puisque de plus en plus obtiennent les données qui leurs manquent.

Objectif du mécanisme d'estimation

Dans le contexte de la proposition, le principal objectif du mécanisme d'estimation est de suivre l'évolution du nombre de récepteurs en attente d'informations. Cela de manière à arrêter la transmission satellite dès que ce nombre devient trop faible. Le seul moyen permettant d'atteindre ce but sans être dangereux pour le réseau est d'estimer le nombre de récepteurs actifs. Dans ce contexte la procédure d'estimation présentée dans [76] est analysée puis adaptée à l'objectif. Les performances de cette procédure ont notamment été analysées en les confrontant à des observations. Ensuite l'étude théorique menée dans [76] a été continuée. Cela a en particulier aboutit à la proposition d'un nouvel estimateur non biaisé. Ces deux études menées, un mécanisme joignant de limitation des retours et d'estimation de taille de groupe est proposé et évalué.

3.3.3.3 Méthodes d'estimation

Notations

Dans les sections suivantes, nous appelons expérience la génération de temps d'attente causée par l'envoi d'un message ***Inquiry***. Pour la i^{eme} expérience, nous notons $(x_{1,i}, x_{2,i}, ..., x_{N_A,i})$ les valeurs générées par les N_A récepteurs. La statistique d'ordre associée est notée $(x^*_{1,i}, x^*_{2,i}, ..., x^*_{N_A,i})$. Celle-ci est telle que :

$$x^*_{1,i} = \min_{j=1,...,N_A} x_{j,i} \leq x^*_{2,i} \leq ... \leq x^*_{N_A,i} = \max_{j=1,...,N_A} x_{j,i}.$$

Nous étudions le problème d'estimation des N_A récepteurs actifs à partir de ℓ expériences dans deux cas :

- **Cas** 1 : seul les temps les plus faibles $\{x^*_{1,i}, i = 1, \ldots, \ell\}$ sont pris en compte, et
- **Cas** 2 **:** toutes les réponses reçues par la source — soit celles comprises dans $[x^*_{1,i}, x^*_{1,i} + c]$ — sont prises en compte.

Estimateur du Maximum de Vraisemblance (EMV) à partir des temps les plus faibles

Soit X une variable aléatoire, $f(x)$ sa densité de probabilité, et $F(x)$ sa distribution cumulée. La densité de probabilité de la séquence des minima $(x^*_{1,i})_{1\leq i\leq \ell}$ s'exprime alors comme (voir par exemple [51]) :

$$L(\mathbf{x^*_1}|N_A) = \prod_{i=1}^{\ell} N_A f(x^*_{1,i})[1 - F(x^*_{1,i})]^{N_A-1},$$

où $\mathbf{x^*_1} = (x^*_{1,1}, \ldots, x^*_{1,\ell})$. Le EMV de N_A (basé sur le vecteur $\mathbf{x^*_1}$) peut être obtenu en maximisant la vraisemblance $L(\mathbf{x^*_1}|N_A)$ par rapport à $N_A \in \mathbb{N}$. De plus amples calculs permettent de montrer que ce maximum est obtenu pour

$$x = \hat{N}_A(\ell) = \frac{\ell}{-\sum_{i=1}^{\ell} \ln(1 - F(x^*_{1,i}))}. \tag{3.4}$$

Cependant $\hat{N}_A(\ell)$ n'est pas le EMV de N_A car $\hat{N}_A(\ell)$ est un réel alors que N_A est un entier. L'EMV est défini à partir de $\hat{N}_A(\ell)$ par :

$$\widehat{N}_{ML}(\ell) = \arg\max \left\{ L\left(X^*_1 | \lfloor \hat{N}_A(\ell) \rfloor\right), L\left(X^*_1 | \lfloor \hat{N}_A(\ell) \rfloor + 1\right) \right\}, \tag{3.5}$$

où $\lfloor x \rfloor$ représente la partie entière de x. Les propriétés de l'estimateur $\widehat{N}_{ML}$ (en terme de biais et de variance) sont difficiles à étudier à cause de l'opérateur de maximisation ($\arg\max\{-\}$).

Les auteurs de [76] proposent d'estimer N_A en utilisant l'estimateur $\hat{N}_A(\ell)$. Il est cependant aisé de montrer que cet estimateur est biaisé. En effet, soit $Z_i = -\ln(1 - F(X^*_{1,i}))$. Z_i est distribué selon une loi exponentielle. En supposant que les

E expériences sont indépendantes, $(2N_A)\sum_{i=1}^{\ell} Z_i$ est distribué selon une loi $\chi^2_{2\ell}$. En conséquence, la moyenne de $\hat{N}_A(\ell)$ s'exprime par :

$$E\left[\hat{N}_A(\ell)\right] = \frac{\ell}{\ell-1}N, \text{ pour } \ell > 1$$

A partir de cette remarque, nous proposons l'estimateur non biaisé défini par :

$$\overset{*}{N}_A(\ell) = \frac{\ell-1}{-\sum_{i=1}^{\ell}\ln(1-F(X^*_{1,i}))}, \tag{3.6}$$

dont la variance peut être calculée pour $\ell > 2$: $\text{var}\left(\overset{*}{N}_A(\ell)\right) = \frac{N_A^2}{\ell-2}$.

Cette section a donc présenté deux estimateurs de taille de groupe : $\widehat{N}_{ML}(\ell)$ et $\overset{*}{N}_A(\ell)$. Le premier estimateur est basé sur le principe du maximum de vraisemblance. Malheureusement, les propriétés de cet estimateur sont difficiles à étudier théoriquement. Le second estimateur peut être vu comme une approximation de l'EMV. Cet estimateur ne permet en effet d'estimer les valeurs entières de N_A par des valeurs réelles. En contrepartie les performances de cet estimateur (en termes de biais et de variance) ont pu être étudiées analytiquement.

EMV obtenu à partir des observations dans $[X^*_{1,i}, X^*_{1,i}+c]$

Soit R_i le nombre de temps d'attente $x^*_{j,i}$ appartenant à $[x^*_{1,i}, x^*_{1,i}+c]$. Comme cela était avancé dans [76], le vecteur $(X^*_{1,i}, R_i)_{1\leq i\leq \ell}$ est une statistique suffisante pour l'estimation de N_A. Dans la suite, nous étudions l'EMV de N_A basé sur l'observation de $(X^*_{1,i}, R_i)$, $i = 1, \ldots, \ell$). La distribution jointe de $(X^*_{1,i}, R_i)$ est l'ensemble des densités :

$$\begin{aligned} f_{r_i}(x^*_{1,i}) &= N_A\binom{r_i-1}{N_A-1} f(x^*_{1,i})[F(x^*_{1,i}+c)-F(x^*_{1,i})]^{r_i-1} \\ &\quad \times[1-F(x^*_{1,i}+c)]^{N_A-r_i} \\ &\propto \frac{\Gamma(N_A+1)}{\Gamma(N_A-r_i+1)}[1-F(x^*_{1,i}+c)]^{N_A-r_i}, \end{aligned}$$

où $\Gamma(.)$ est la fonction Gamma et $r_i \in \{1, \ldots, N_A\}$ est le nombre d'observations dans $[x^*_{1,i}, x^*_{1,i}+c]$ ($f_p(x^*_{1,i})dx^*_{1,i}$ est la probabilité d'avoir $N_{r_i} = p$ et $X^*_{1,i} \in [x^*_{1,i}, x^*_{1,i}+dx^*_{1,i}[$). En supposant que les différentes expériences sont indépendantes l'EMV de N_A peut être obtenu à partir des observations de $(X^*_{1,i}, R_i)$, $i = 1, \ldots, \ell$ en maximisant le critère par rapport à N_A :

$$\sum_{i=1}^{\ell}\ln\left[\frac{\Gamma(N_A+1)}{\Gamma(N_A-r_i+1)}\right] + (N_A-r_i)\sum_{i=1}^{\ell}\ln[1-F(x^*_{1,i}+c)]. \tag{3.7}$$

La maximisation de ce critère peut cependant être lourde pour des applications en termes de calculs. Afin d'éviter cette procédure de maximisation, les auteurs de [76]

on proposé d'utiliser une approximation de Poisson. Cette estimation permet d'obtenir l'estimateur suivant :

$$\tilde{N}_A(\ell) = \frac{\ell + \sum_{i=1}^{\ell}(R_i - 1)}{-\sum_{i=1}^{\ell} \ln(1 - F(X^*_{1,i})) + \sum_{i=1}^{\ell} p_i}, \tag{3.8}$$

avec

$$p_i = \begin{cases} \frac{F(X^*_{1,i}+c) - F(X^*_{1,i})}{1 - F(X^*_{1,i})} & \text{si } X^*_{1,i} + c < \textbf{TMaxInquiry} \\ 1 & \text{si } X^*_{1,i} + c \geq \textbf{TMaxInquiry} \end{cases}$$

où $[0, \textbf{TMaxInquiry}]$ est le support de la densité de probabilité de $f(x^*_{1,i})$. Remarquons que l'équation (3.8) se correspond à l'équation (3.4) lorsque $p_i = 0$ et $R_i = 1$. Ce cas particulier correspond au cas où seulement la première réponse est utilisée pour l'estimation.

Notons que la maximisation du critère défini dans l'équation (3.7), revient à maximiser une somme de fonction de type $\frac{\Gamma'(x)}{\Gamma(x)}$. Cette fonction, connue sous le nom de *fonction Digamma* a déjà fait l'objet de nombreuses études. En particulier, les valeurs numériques de cette fonction ont été étudiées. Il est donc envisageable de maximiser l'expression (3.7) sans avoir à faire une approximation de Poisson. L'étude d'un estimateur basé sur la maximisation numérique sans approximation de ce critère est une voie d'amélioration de l'estimateur proposé dans cette section. A l'heure actuelle cette voie a été partiellement explorée, mais aucun résultat n'a pu être obtenu pour les simulations présentées ci-dessous.

Résultats de simulations

De nombreuses simulations ont été conduites pour valider les résultats théoriques précédents. La figure 3.6 montre la moyenne de différentes estimations de taille de groupe, obtenue à partir de 20 simulations de type Monte Carlo (les paramètres sont $N_A = 50\,000$ et $\ell = 5$). L'estimateur utilisant toutes les réponses reçues est visiblement bien plus efficace que ceux utilisant seulement la première réponse. Cette constatation est confirmée par la figure 3.7 qui montre les variances des différents estimateurs en fonction de ℓ. Chaque point a été obtenu en faisant la moyenne sur 2 000 simulations de type Monte Carlo. Notons que la valeur maximum de ℓ est $\ell_{max} = 10$. Cette valeur correspond au nombre maximum de messages ***Inquiry*** envoyés avant chaque estimation de taille de groupe. Ainsi des valeurs élevées de ℓ impliqueraient soit des délais important entre chaque estimation, soit un débit d'émission de message ***Inquiry*** élevé. De telle valeurs de ℓ sont donc inacceptables.

3.3.3.4 Interaction entre estimation et limitation des retours

Le problème d'estimation de taille de groupe est fortement lié au problème de *Nack Implosion*. En effet les paramètres λ et **TMaxInquiry** ont un impact sur le nombre de messages générés en réponse à un message ***Inquiry***. Or le nombre de messages générés a un impact sur la précision des estimations (car plus le nombre de messages reçus augmente, plus la source dispose d'informations sur la taille du groupe). On est donc en droit de se poser deux questions :

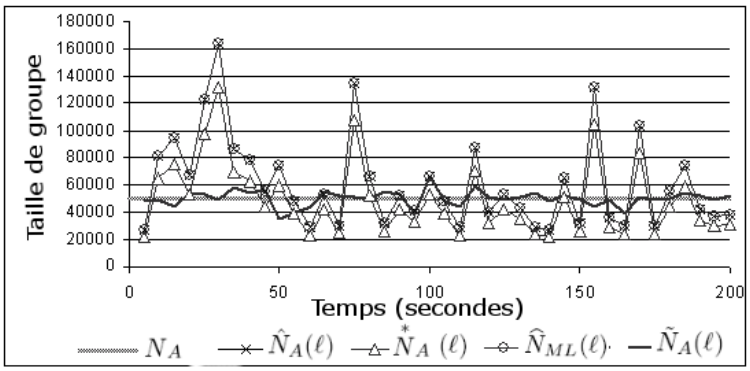


Figure 3.6 – Moyenne des estimations sur 20 simulations ($N_A = 50\,000$, $\ell = 5$)

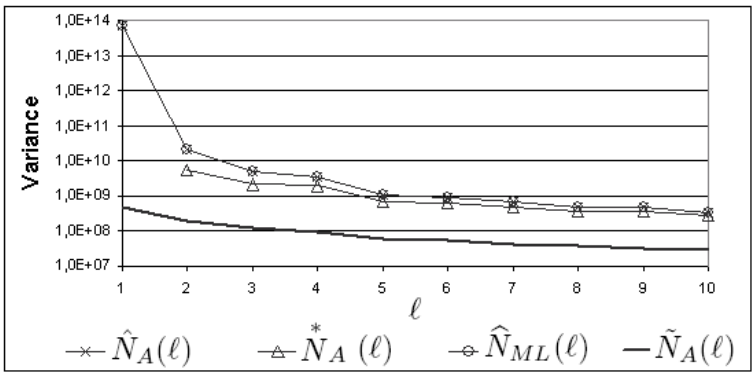


Figure 3.7 – Variance des estimateurs en fonction de ℓ)

1. Y a-t-il un risque de *Nack Implosion*?, et
2. Comment évolue la précision des estimations en fonction des paramètres λ et **TMaxInquiry** ?

De nombreuses simulations on été effectuées afin de répondre à ces questions. Dans la suite de cette section, seul l'estimateur $\tilde{N}_A(\ell)$ est étudié, car ses performances surpassent largement celles des autres estimateurs. Deux types de simulations ont été menés :

– Le premier type de simulations correspond au cas où les paramètres ne changent pas au cours du temps (i.e. les valeurs de **TMaxInquiry** et λ sont statiques) . Dans ce cas le mécanisme de *Nack Suppression* est configuré de telle sorte que en moyenne R messages soient générés lorsque la taille du groupe est de N_{max} récepteurs. N_{max} représente la taille maximale supposée du groupe de récepteurs. Les paramètres de $f(x)$ sont alors calculés comme suit (voir [93] pour plus de détails) :

$$\begin{cases} \lambda = 1.1\ln(N) + 0.8, \\ \textbf{TMaxInquiry} = \lambda c \left[\ln\left(R + \frac{N}{e^{\lambda}-1}\right) - \ln\left(1 + \frac{N}{e^{\lambda}-1}\right)\right]^{-1}, \end{cases} \tag{3.9}$$

avec $N = N_{max}$ et où R représente le nombre moyen de réponses attendues. Dans nos simulations, les paramètres étaient $N_{max} = 10^6$ et $R = 30$.

– Dans le second type de simulations, les paramètres de $f(x)$ sont mis à jour au cours du temps : après chaque simulation, ces paramètres sont calculés au moyen de l'équation 3.9 avec $R = 30$ et $N = \tilde{N}_A(\ell)$.

La figure 3.8 montre la variance de l'estimateur (obtenu à partir de 2 000 simulation simulations de type Monte Carlo) en fonction de ℓ. La mise à jour a de toute évidence un impact positif sur la qualité de l'estimation de N_A, puisque la variance de l'estimateur diminue. Notons que, selon la partie 3.3.2.2, la source doit envoyer un message ***Inquiry*** pour chaque estimation. En conséquence, l'envoi des paramètres λ et **TMaxInquiry** ne pose pas de problème dans la mesure où ils peuvent être inclus dans ces messages.

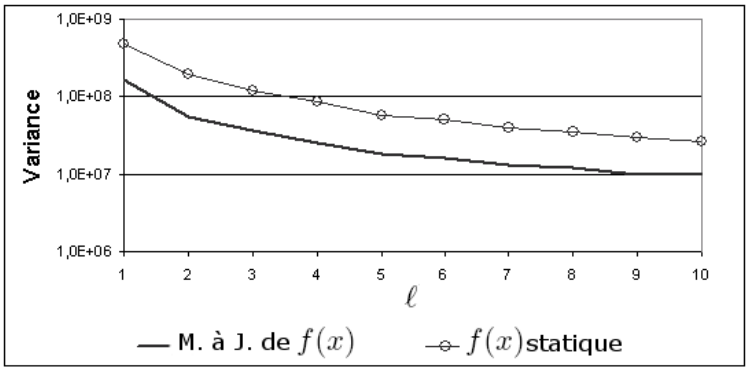


Figure 3.8 – Variance de $\tilde{N}_A(\ell)$ en fonction de ℓ ($R = 30$, $N_{max} = 10^6$, $N = 50\,000$)

La mise à jour des paramètres de $f(x)$, si elle augmente la précision des estimations, introduit un risque de *Nack Implosion*. En effet, lorsqu'une perturbation atmosphérique touche la majeure partie des récepteurs, le mécanisme d'estimation va sous-estimer la taille du groupe. Il va donc se configurer de manière à ce que l'ensemble des récepteurs visibles envoie R réponses en moyenne. Cependant, lorsque les récepteurs situés sous la perturbation vont se retrouver en mesure de recevoir les messages des récepteurs, ils vont participer à la génération de messages ***InquiryRep***, et un grand nombre de messages vont être générés. Pour éviter ce problème, une solution consiste à limiter la mise à jour de $f(x)$. Nous proposons pour cela de mettre à jour cette fonction comme suit :

– lorsque $\tilde{N}_k > N_{min}$, les paramètres sont calculés avec l'équation 3.9 avec $N = \tilde{N}_k$, et

– lorsque $\tilde{N}_k \leq N_{min}$, l'équation 3.9 est utilisée avec $N = N_{min}$

où N_{min} est un paramètre préalablement choisi en fonction de la sécurité désirée (vis-à-vis du problème d'implosion des réponses).

La détermination de N_{min} est un problème crucial pour le bon fonctionnement de la solution proposée. Une valeur trop faible de N_{min} introduit un risque important d'implosion des réponses. A l'inverse, le gain en précision d'estimation est réduit lorsque la valeur de N_{min} est trop grande. Afin de choisir une valeur acceptable pour ce paramètre, un grand nombre de simulations ont été faites. Plus précisément, le mécanisme a été configuré avec des valeurs différentes pour le seuil N_{min}. Pour chaque simulations la taille du groupe varie instantanément de $N_A = 200$ — soit une valeur située sous

le seuil — à $N_A = N_{max}$. Le test de cette situation représente le ¡¡ pire cas ¿¿ avec les hypothèses faites, dans la mesure où N_{max} représente la valeur maximale pour le groupe. Pour chaque valeur de N_{min}, l'expérience a été réalisée 20 fois et la valeur maximum du nombre de messages générés au cours de ces expériences a été retenue comme résultat. Ces valeurs sont tracées dans la figure 3.9 en fonction de N_{min} avec $R = 30$ et $N_{max} = 10^6$. Selon les résultats obtenus, le choix de $N_{min} = 6\,000$ assure que le nombre de réponses est inférieur à 1 000. Remarquons que cette opération est nécessaire, par égard pour le réseau. Cependant dans la mesure où les groupes visés sont de plus de 6 000 personnes, cette opération devrait être peu utilisée.

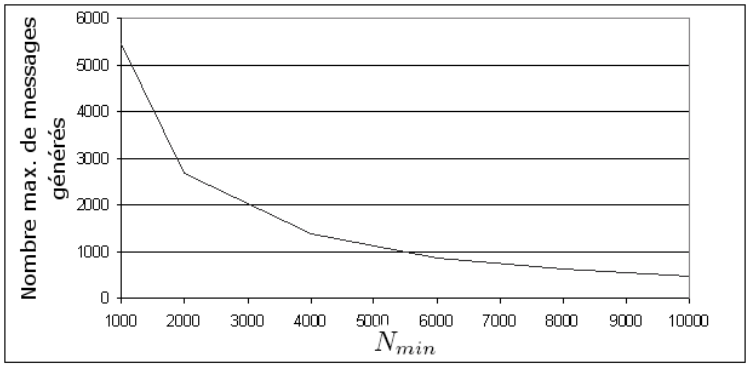


Figure 3.9 – Nombre maximum de réponses en fonction de N_{min} ($R = 30$, $N_{max} = 10^6$)

Une fois tous ces paramètres fixés, la procédure globale d'estimation de taille de groupe a été évaluée dans plusieurs cas définis à partir d'observations réelles de transmission satellites. La figure 3.10 montre un exemple de résultats d'estimation obtenus pendant une expérience. Au cours de cette expérience, $\ell = 5$ expériences étaient utilisées pour calculer une estimation de la taille du groupe. De plus, les paramètres de $f(x)$ étaient mis à jour à chaque expérience. Au cours de cette simulation, le mécanisme utilisant un seuil $N_{min} = 6\,000$ (selon la procédure décrite plus haut) a été comparé au même mécanisme n'utilisant pas de seuil.

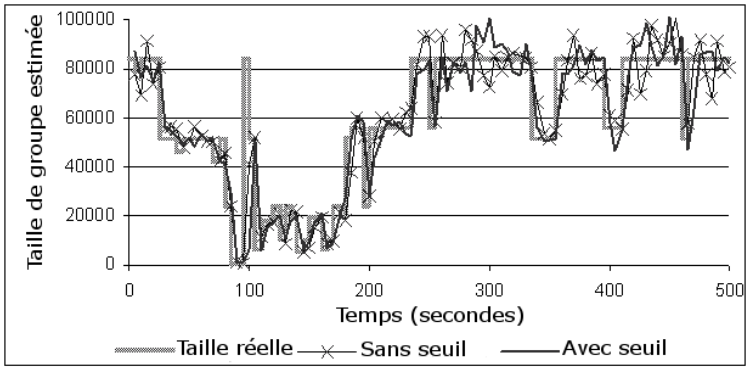


Figure 3.10 – Résultats d'estimation au cours d'une transmission $N_{min} = 6\,000$ ($R = 30$, $N_{max} = 10^6$)

Les figures 3.10 et 3.11 montrent les résultats obtenus dans les deux cas. Selon les résultats, estimations réalisées avec les deux mécanismes sont de qualité équivalente. Cependant, lorsque la procédure de seuillage n'est pas utilisée, plusieurs centaines de messages sont générés au cours de la transmission. Ce problème est entièrement gommé par le mécanisme de seuillage. L'utilisation d'un seuil semble donc être une bonne solution car elle n'influence pas la qualité des estimations de manière visible tout en éradiquant les risques d'implosion des réponses.

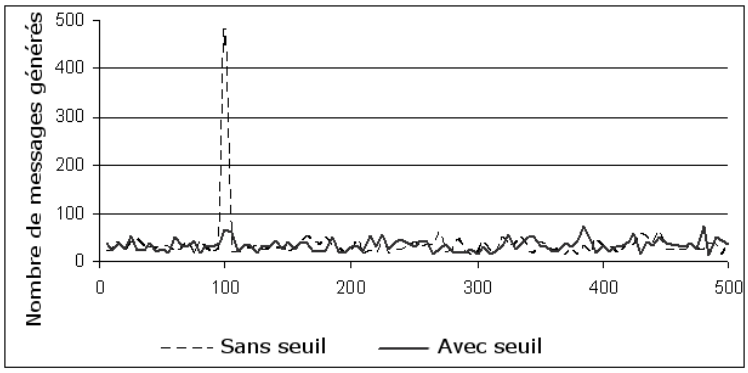


Figure 3.11 – Nombre de messages générés au cours d'une transmission $N_{min} = 6\,000$ **(**$R = 30$**,** $N_{max} = 10^6$**)**

3.3.4 Conclusion sur les performances du mécanisme

Ce chapitre a présenté les problèmes de limitation des retours à la source (*Nack Suppression*, et d'estimation de taille de groupe. Parmi les mécanismes permettant de limiter le nombre de messages émis sur la voie retour, nous proposons d'utiliser un mécanisme basé sur l'utilisation de timer, car il limite le temps écoulé entre une demande de réponses (***Inquiry***) et l'obtention de ces réponses. A partir de ce mécanisme, plusieurs estimateurs de taille de groupe ont été évalués. Ces estimateurs sont tous basés sur le principe du maximum de vraisemblance. Parmi les estimateurs analysés, les performances de l'estimateur basé sur l'ensemble des réponses reçues surpassent largement les performances des autres estimateurs.

L'objectif principal du mécanisme d'estimation est de suivre la taille du groupe afin de détecter le moment où cette taille devient inférieure au seuil **MinRcv**. L'étude des performances de cet estimateur a montré que $\tilde{N}_k$ permet d'atteindre ce but. Le risque d'implosion des retours associé à l'utilisation de cet estimateur, a été également évalué. Cela a conduit à la définition d'une procédure de mise à jour avec un mécanisme de seuillage. Celle-ci permet d'améliorer la précision des estimations réalisées sans pour autant introduire de risque de *Nack Implosion*.

Une fois que le nombre de récepteurs attendant des informations est devenu inférieur au seuil **MinRcv**, il reste à transmettre des informations au moyen du réseau terrestre. Le chapitre suivant étudie les problèmes liés à cette phase de transmission.

3.4 MÉCANISME DE REPRISE DES ERREURS PAR VOIE TERRESTRE

3.4.1 Avantages liés à l'utilisation du réseau terrestre

L'idée d'utiliser une liaison terrestre afin de compléter une transmission satellite remonte à de nombreuses années. Plusieurs travaux ont en effet montré l'intérêt de ce type de technique en termes de temps de latence et d'utilisation des ressources réseau (voir par exemple [94], [70], [20], [75]). Cette technique semble de plus particulièrement adaptée à un système hybride terrestre/satellite, dans la mesure où ces deux types de liaison offrent des services complémentaires. Nous avons de plus montré que, dans notre contexte, l'utilisation de liens terrestres était plus intéressante que l'utilisation d'un lien satellite sous certaines conditions (cf. partie 2.3). Parmi les différentes propositions de rapatriements terrestres d'informations, certains travaux récents ont démontré l'efficacité des techniques de pair-à-pairs (ou *Peer-to-peer*). Il est de plus apparu que ces techniques sont plus efficaces encore lorsqu'elles sont couplées à un codage des informations [69] [23] [22].

Outre les considérations de performances, l'utilisation d'un mécanisme terrestre pour compléter une diffusion satellite est basée l'hypothèse que la proximité géographique est différente de la proximité en terme de réseau. Deux facteurs semblent en effet prépondérant pour permettre une communication rapide entre deux utilisateurs : le fait d'être dans une ville bien desservie, et le fait d'être dépendant du même fournisseur d'accès (car dans ce cas, la communication ne transite généralement pas par un réseau autre que celui du fournisseur d'accès). Or, comme cela a été dit dans le chapitre 2 les erreurs de transmission survenant lors d'une diffusion satellite sont principalement dues à des perturbations atmosphériques. Ces dernières affectant plusieurs zones géographiques, les récepteurs touchés sont physiquement proches[4]. En conséquence, si la proximité en termes de réseau est différente de la proximité géographique, le réseau permet de rapprocher des hôtes physiquement éloignés. Il est de plus fort probable pour que ceux-ci aient subis des pertes différentes. Chacun peut dans ce cas transmettre à l'autre les paquets qu'il a reçus.

Les sections suivantes étudient les problèmes liés à la phase de reprise terrestre dans le contexte de l'étude. Tout d'abord, la section 3.4.2 présente les objectifs de ce mécanisme, ainsi que les problèmes qui se posent. La section 3.4.3 fait état des différentes propositions existantes. Les objectifs particuliers du mécanisme confrontés aux solutions déjà existantes ont conduit à une nouvelle proposition, décrite dans la partie 3.4.4, puis à son étude. Les résultats de cette étude sont présentés dans la partie 3.4.5.

3.4.2 Objectifs et hypothèses

De même que pour la définition du protocole de transport, il est primordial de définir en premier lieu les objectifs du mécanisme, ainsi que les hypothèses de fonctionnement.

[4]Selon certains experts, une perturbation atmosphérique entraîne des pertes corrélées dans un rayon de 3km en moyenne.

Cela est particulièrement important, vu la spécificité du contexte de l'étude. Ainsi lorsque la transmission satellite s'arrête, la répartition des informations dans le réseau diffère notablement des hypothèses prises dans les études citées précédemment. Les hypothèses prises sur l'état du réseau après la transmission satellite sont détaillées ci-dessous. Dans ce contexte, les différents objectifs du mécanisme de reprise terrestre des erreurs, ainsi que les problèmes techniques qu'ils posent y sont recensés.

HSTRM propose un service de communication vers des groupes de taille très importante. Compte tenu des résultats présentés dans le chapitre 2, si la transmission satellite se termine, c'est que le nombre de récepteurs n'ayant pas reçu les données est devenu relativement faible devant le nombre initial de récepteurs. Ainsi, seule une petite fraction des récepteurs ne possède pas toute l'information. On ne possède en revanche aucune information sur le volume d'information que ces récepteurs ont effectivement reçu (il est possible que certains des récepteurs n'aient pas du tout d'information). De plus en considérant que la proximité dans le réseau est indépendante de la proximité géographique, on ne possède aucune information sur la répartition des récepteurs dans le réseau. Rappelons que ces récepteurs sont connectés au réseau terrestre au moyen d'un réseau d'accès haut débit.

Dans la mesure où HSTRM est destiné à être utilisé pour des transmissions vers de grands groupes, le mécanisme doit avant tout être utilisable à grande échelle. C'est-à-dire que la mise en place du réseau de pair-à-pair et la recherche d'information ne doivent saturer ni le réseau sous-jacent (i.e. saturer les routeurs traversés), ni les récepteurs impliqués. Ce problème concerne en particulier l'établissement du réseau, car tous les membres du groupe sont impliqués dans cette phase. En effet, lors de cette phase, chaque récepteur doit au moins envoyer un message afin de se faire connaître au reste des pairs. Ainsi *tous les récepteurs* vont envoyer un message vers le réseau de pair-à-pair. Le mécanisme de création du réseau doit donc être soigneusement configuré de manière à répartir suffisamment les connexions pour que les risques de saturation ne se présentent pas.

Ensuite, de manière à ce que la proposition permette effectivement de déployer un service de communication multipoint sans modification du réseau sous-jacent, le mécanisme doit être compatible avec le réseau en place. En particulier tous les mécanismes faisant appel à des fonctions particulières du réseau ne pourront être utilisés.

La dernière contrainte — et sans doute la plus importante — porte sur le coût d'utilisation du mécanisme de reprise terrestre. En effet, pour que la proposition reste intéressante, il ne faut pas que le surcoût dû à l'utilisation du mécanisme rende la technique plus chère qu'une simple diffusion satellite ou terrestre. Le nombre de messages générés par l'utilisation de cette technique doit donc être minimisé autant que faire se peut.

Plusieurs propositions existantes ont été étudiées afin de savoir si elles permettaient de satisfaire les exigences exprimées ci-dessus. Le positionnement de nos travaux vis-à-vis de ces précédentes études est exposé dans le paragraphe suivant.

3.4.3 Analyse de propositions antérieures

Nous nous sommes intéressés aux solutions adoptées dans des domaines bien distincts. Dans l'objectif de répondre au besoin spécifique de HSTRM, nous avons analysé les techniques de ¡¡ téléchargements coopératifs ¿¿ (*cooperative downloads*), de ¡¡ multicast applicatif ¿¿, ainsi que l'utilisation d'un réseau de pair-à-pairs. Les solutions précédentes, dont l'objectif est la transmission de données de manière générale, ont ensuite été confrontées à la problématique qui nous intéressait : la reprise d'erreurs. Pour cela, des mécanismes de ¡¡ reprise locale ¿¿ (ou *Local Recovery*) ont été étudiés, afin de définir un mécanisme *ad hoc*.

Le premier mécanisme étudié est référencé sous le terme de ¡¡ téléchargements coopératifs ¿¿ (*cooperative downloads*). Dans ce cas, l'objectif est de relâcher les contraintes sur un serveur central trop sollicité. Cette situation arrive traditionnellement lorsqu'un serveur contient un contenu très populaire (comme la dernière mise à jour d'un logiciel). Ce problème a entre autre été traité par Slurpie [123] et Bit Torrent [84]. L'objectif de ces solutions est de répliquer le plus rapidement possible un fichier de taille importante sur un ensemble d'hôtes. Pour cela, les nouveaux arrivants qui cherchent à se connecter au serveur initial sont redirigés vers d'autres utilisateurs qui possèdent déjà tout ou une partie du fichier. La charge est ainsi répartie de proche en proche sur l'ensemble des utilisateurs. Les propositions citées ont en particulier montré que la charge était ainsi effectivement répartie, nous avons donc retenu l'idée de rediriger les demandes de connexion vers d'autres utilisateurs. Cependant, l'objectif principal de ces solutions est différent du notre : lors de la phase de transmission terrestre, le but n'est pas de répliquer un objet sur tout un groupe dont personne ne possède d'information. En effet à ce stade les échanges ne concernent plus que quelques récepteurs qui cherchent un volume d'information généralement faible.

Une autre solution permettant à un groupe de récepteurs de communiquer avec d'autres récepteurs consiste simplement à utiliser un service de ¡¡ multicast applicatif ¿¿. Comme cela a été dit dans le chapitre 1, ce problème a fait l'objet de nombreuses études [10], [102], [64], [17]. Toutefois, à l'instar des solutions exposées au paragraphe précédent :

- l'objectif est de transmettre un volume de données relativement important à tout un groupe, et
- les données sont transmises à partir d'une source vers le groupe.

Les objections quant à l'utilisation de cette technique sont donc identiques à celles formulées plus haut, et en conséquence cette technique s'applique mal à notre contexte.

Enfin dans la mesure où nous avons évoqué l'efficacité des techniques de pair-à-pair, nous avons analysé les techniques d'établissement de réseau de pair-à-pairs largement citées dans la littérature : CAN (pour *Content Adressable Network* [109]), et CHORD [126]. CAN et CHORD sont deux techniques proposées pour localiser rapidement des informations au sein du réseau de pair-à-pair. Elles reposent sur l'utilisation de fonctions de hachage (comme l'algorithme MD5 [112] ou SHA-1 [92]) : lorsqu'un noeud cherche une information, il passe les informations relatives à celle-ci à la fonction de hachage. Cette dernière renvoie alors l'identifiant (i.e. son ¡¡ adresse ¿¿ dans le réseau de pair-à-pair) du noeud qui a la charge de stocker les données recherchées.

Ces approches permettent ainsi de localiser des informations parmi un ensemble de récepteurs possédant une grande diversité d'informations. Notre problème est sensiblement différent : les récepteurs cherchent à localiser des données qui sont déjà stockées par un grand nombre de récepteurs. La difficulté liée à la localisation des informations est donc bien moindre : presque tous les récepteurs ont un voisin qui possède les informations qu'ils recherchent. L'utilisation de cette technique est donc peu utile. Comme cette technique introduit une charge de calcul non négligeable (due à l'utilisation de la fonction de hachage), nous avons choisi de ne pas la mettre en oeuvre.

Les mécanismes de ¡¡ reprise locale ¿¿ (ou *Local Recovery* [13], [94], [20], [59], [75]) des erreurs permettent d'effectuer une requête de retransmission à un récepteur ¡¡ proche ¿¿ pour récupérer des informations manquantes. La problématique abordée par ces mécanismes est donc voisine de celle qui nous intéresse. En analysant ce type d'approche, notre idée était d'allier les techniques retenues dans les études précédentes à des mécanismes conçus pour reprendre des erreurs. C'est en particulier le cas du mécanisme proposé dans l'article ¡¡ *Reliable Multicast via Satellites* ¿¿ [13]. Cette proposition consiste à diviser les récepteurs en plusieurs sous-groupes, chaque sous-groupe possédant un responsable. Lorsqu'un récepteur a subi des pertes, il recherche les informations qui lui manquent au sein du sous-groupe. Si personne ne possède cette information, le responsable contacte la source pour qu'elle retransmette les données par satellite. Cette approche cible un service très proche de celui qui nous intéresse. Notons toutefois que le choix d'utiliser un codage de type FEC au niveau transport change les propriétés du système à considérer. De plus celle-ci n'étudie pas du tout les problèmes liés à la mise en place du réseau de pair-à-pair, au choix des responsables, et à la recherche d'informations. Enfin les recherchent peuvent aboutir à des retransmissions par satellite, tandis que dans notre cas toutes les données doivent être transférées au moyen du réseau terrestre.

L'objectif d'un grand nombre de travaux est de minimiser le temps de rapatriement des informations. Pour satisfaire ce besoin, plusieurs mécanismes sont mis en place de manière à faire converger le réseau de pairs vers un réseau plus performant. Or parmi l'ensemble des pairs, un grand nombre de récepteurs ne va pas faire de requêtes : c'est le cas de tous ceux qui ont reçu toutes les données (plus nombreux de par le principe de l'approche adoptée). Il est donc inutile de chercher à les rapprocher d'autres pairs plus proches : ils doivent juste rester joignables. En d'autres termes, il n'est intéressant d'optimiser les connexions que pour les pairs qui cherchent des informations (et pas pour le réseau global).

Parmi tous les travaux référencés, aucun ne prend en compte de contrainte de coût liée à l'utilisation du réseau. Or la prise en compte d'une telle contrainte rend impossible par exemple l'utilisation de mécanismes permettant de faire converger le réseau de pairs (car ces mécanismes impliquent une utilisation supplémentaire non négligeable du réseau). Cela explique également le peu d'études menées sur la mise en place du réseau de pairs pour des techniques de reprise d'erreurs : le nombre de messages générés n'est pas forcément un critère pris en compte.

On peut de plus dégager un avantage non négligeable lié à l'utilisation de la solution hybride : après la diffusion satellite, c'est tout un ensemble d'informations encodées qui sont disséminées sur le réseau de pairs. En conséquence aucune politique de réplication

et de dissémination des informations (voir par exemple [71], [22] et [111]) n'est à mettre en place. Au contraire il serait dommage de ne pas exploiter cette situation propice à la reprise d'informations.

Notre approche pose un certain nombre de problèmes spécifiques qui n'ont pas été considérés dans les approches analysées. Par exemple aucun des mécanismes étudiés n'a considéré de système où la proportion des utilisateurs possédant les données était importante (le cas contraire est généralement considéré). Cette hypothèse permet entre autres de simplifier les mécanismes de recherche d'informations. Or dans le cadre de notre approche, la simplification des mécanismes utilisés est primordiale, car cela diminue le coût lié à leur utilisation. De plus, à notre connaissance, aucune proposition n'a étudié le coût lié à l'établissement du réseau logique ou à la recherche d'informations. Enfin l'hypothèse de transmission impliquant de grands groupes, alliée au fait que ce réseau soit mis en place à un instant précis (au moment où la transmission satellite s'arrête), pose des problèmes de mise à l'échelle spécifiques à l'approche hybride adoptée. L'ensemble des problèmes exposés ci-dessus nous a poussés à adapter et définir un ensemble de mécanismes adéquats à notre contexte. Ces mécanismes sont détaillés dans la section suivante.

3.4.4 Définition des mécanismes pour la phase terrestre

3.4.4.1 Présentation générale des mécanismes

Les objectifs principaux du mécanisme de reprise terrestre des erreurs découlent directement des remarques précédentes. Ils sont :

- de permettre de retrouver des informations parmi l'ensemble des récepteurs,
- de ne pas impliquer une utilisation trop coûteuse du réseau terrestre, et
- de pouvoir être utilisé à grande échelle, sans pour autant impliquer des délais inacceptables du point de vue des récepteurs.

Dans ce but, nous avons choisi d'établir un réseau de pair-à-pair de manière à ce que les récepteurs puissent communiquer entre eux. Ce réseau de pair est constitué d'un ensemble d'arbres logiques réguliers v-aire (cf. figure 3.12). Plusieurs avantages sont liés à l'organisation des récepteurs selon cette structure :

- la création du réseau est simple,
- cela permet de limiter le nombre de pairs connus les uns des autres (ce qui évite les risques d'implosions de messages et distribue la mémorisation de l'état de l'arbre), et
- le parcours de l'arbre est relativement rapide.

Le nombre d'arbres logiques sera noté τ dans la suite du document, le nombre de pairs dans chaque arbre η_i, et la hauteur des arbres h_i (avec $1 \leq i \leq \tau$). Ces mécanismes sont présentés ci-dessous.

Pour constituer ce réseau, il est nécessaire de disposer des racines (qui sont la base du réseau). Pour cela la source collecte les adresses des récepteurs qui répondent aux messages ***Inquiry*** (cf. section 3.3). Elle transmet ensuite ces adresses à tout le groupe au moyen de messages ***P2PRootList***. Ceux-ci sont envoyés toutes les **P2PRLPeriod**

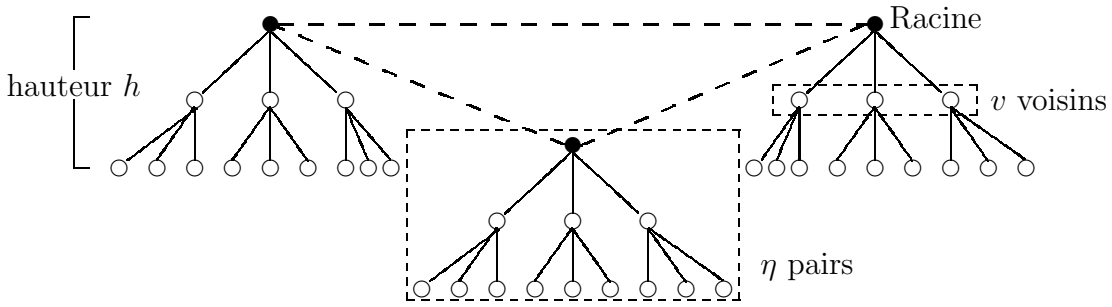


Figure 3.12 – Représentation logique du réseau de pair-à-pairs

unités de temps, pour permettre à des récepteurs subissant des pertes de les recevoir. La source envoie avec cette liste un temps de connexion **TMaxP2P** qui correspond au temps dont les récepteurs disposent pour se connecter.

A la réception du premier message ***P2PRootList***, les récepteurs tirent aléatoirement un temps d'attente compris entre 0 et **TMaxP2P**. A l'expiration du timer, les récepteurs choisissent aléatoirement une racine parmi celles connues, et lui envoient un message ***ConnectP2P*** en précisant leur adresse IP. Si la racine choisie ne répond pas, il choisit une autre racine. La première racine joignable répondra par un message ***PeerConnected***, si elle a moins de v voisins. Dans ce cas le récepteur devient un voisin direct de la racine. Dans le cas contraire, la racine renvoie un message ***Redirected*** qui précise un pair dépendant de cette racine auquel le nouvel arrivant doit se faire connaître. Ce dernier envoie alors un message ***ConnectP2P*** au pair spécifié. Celui-ci répond par un message ***PeerConnected***. Le pair que le nouvel arrivant cherche à atteindre peut cependant ne pas être joignable. Si tel est le cas, le nouveau pair recontactera sa racine qui lui retransmettra l'adresse d'un autre pair. Ces opérations sont illustrées dans la figure 3.13 pour le cas où $v = 3$ ($@_i$ représente l'adresse du récepteur i). De plus, de manière à ce que la panne d'un nœud n'entraîne pas une partition du réseau logique, les pairs gardent en mémoire un certain nombre d'adresses : chacun a connaissance de l'ensemble des racines du réseau logique qui s'ajoutent à leurs éventuels père et fils. Ainsi, si un récepteur se retrouve isolé (i.e. ses voisins sont tous indisponibles), il peut toujours contacter les autres racines et avoir accès au reste du réseau de pairs. Tout pair a donc accès à l'ensemble du réseau.

Les sections suivantes justifient les choix sur v, τ, et **TMaxP2P** qui ont été retenus en s'appuyant soit sur une étude théorique du problème, soit sur les résultats de simulations. Ainsi le coût d'établissement du réseau de pairs, le volume de mémoire utilisé pour représenter l'état du réseau, et l'espacement temporel des connexions sont étudiés au travers de calculs théorique. Ensuite le coût et l'efficacité des recherches d'informations sont étudiés grâce à des simulations. Notons que les résultats obtenus au moyen de calculs théoriques ont été vérifiés lors des simulations effectuées.

3.4.4.2 Coût d'établissement du réseau de pairs

Pour arriver au résultat escompté, chaque récepteur choisit aléatoirement une racine à laquelle il envoie une demande de connexion (message ***ConnectP2P***). Si la

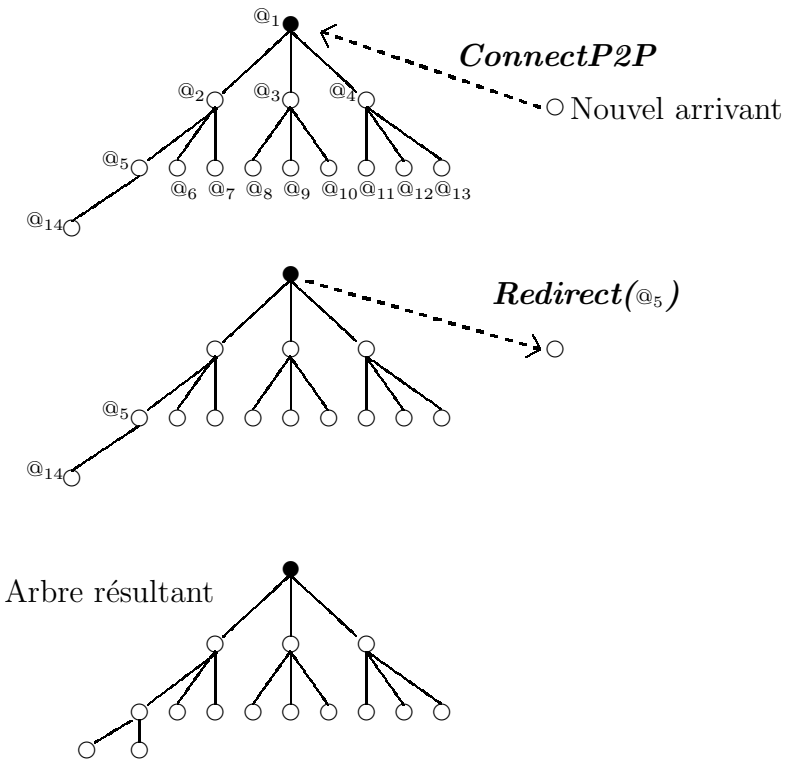


Figure 3.13 – Connexion d'un nouveau pair au réseau

racine ne possède pas v voisins, elle répond avec un message ***PeerConnected***. Dans le cas contraire, elle renvoie aux nouveaux arrivants un message de redirection indiquant un pair n'ayant pas encore v voisins. Ce dernier répondra avec un message ***PeerConnected*** pour indiquer au récepteur qu'il fait désormais partie du réseau de pair-à-pair. En considérant que tous les récepteurs peuvent communiquer les uns avec les autres (il n'y pas de récepteurs injoignables), on peut donc calculer le nombre de messages échangés lors de la phase de d'établissement du réseau :

- pour les τ racines, aucun message n'est utilisé,
- pour les $\tau \times v$ pairs connectés directement à une racine, 2 messages sont échangés, et
- pour les $N - \tau \times (v + 1)$ récepteurs restants 4 messages sont échangés.

Avec un coût d'utilisation du réseau terrestre point-à-point par paquet égal à α_{TU} (les notations sont les mêmes qu'à la section 2.3), le coût d'établissement du réseau — noté par la suite $F_{P2P}(N, v, \tau)$ — s'exprime donc comme :

$$F_{P2P}(N, v, \tau) = \alpha_{TU} \times (2 \times \tau \times v + 4 \times (N - \tau \times (v + 1)))$$

Ce coût a un impact sur la taille minimum des objets qui peuvent être envoyés pour que l'établissement du réseau de pairs soit rentable. Selon la section 2.3.2.4, le coût de transmission d'une communication :

- visant à transmettre un volume S de données,
- en utilisant un codage en bloc avec T_B paquets par blocs,

– avec un volume TDU octets de données compris dans les paquets de niveau transport,

est donné par :

$$G_{SM}(N, S, TDU, T_B) = \lceil \frac{S}{T_B \times TDU} \rceil \times F_x(N, T_B).$$

On peut alors définir une valeur maximale ρ_{max} telle que :

$$F_{P2P}(N, v, \tau)/G_{SM}(N, S, TDU, T_B) \leq \rho_{max}.$$

Cette valeur contraint la taille du fichier de manière à ce que la relation

$$\lceil \frac{S}{T_B \times TDU} \rceil \geq \frac{F_{P2P}(N, v, \tau)}{\rho_{max} \times F_x(N, T_B)} \tag{3.10}$$

soit vérifiée. Par exemple, en gardant les valeurs de la section 2.3 (c'est-à-dire $\alpha_{TU} = \alpha_{TM} = 1$, et $\alpha_{SM} = 200$), le coût de transmission de 100 paquets au moyen d'une liaison satellite est de l'ordre de $6\,10^4$ pour un groupe de 600 000 récepteurs. Pour une même taille de groupe avec $v = 10$ et $\tau = 166$ (le choix de ces valeurs est dû aux résultats obtenus dans les sections suivantes), on obtient $F_{P2P}(N, v, \tau) = 2\,396\,016$. Avec $\rho_{max} = 0,1$, la relation (3.10) est donc vérifiée pour des communications impliquant l'échange de plus de 60 Mo de données.

3.4.4.3 Évaluation de la mémoire utilisée

Pour arriver au résultat escompté, chaque récepteur choisit aléatoirement une racine à laquelle il envoie une demande de connexion (message ***ConnectP2P***). Dans le cas où la racine a déjà v voisins, elle renvoie aux nouveaux arrivant un message de redirection indiquant un pair n'ayant pas encore v voisins. Pour cela chaque racine doit garder en mémoire l'adresse de tous les récepteurs situés au bas de l'arbre. Une illustration du nombre d'adresses qu'une racine doit garder en mémoire est représentée dans la figure 3.14. Comme les groupes auxquels les communications sont destinées sont très importants, la place mémoire requise pour stocker ces adresses peut devenir contraignante. Nous avons donc cherché à évaluer la place mémoire requise par ce mécanisme.

Pour un arbre v-aire de hauteur h formé de η nœuds, on a :

$$\sum_{i=0}^{h-1} v^i < \eta \leq \sum_{i=0}^{h} v^i, \text{ avec } h \geq 1$$

En se limitant au cas où l'égalité est vérifiée (i.e. au cas des arbres où tous les étages sont complets), on obtient :

$$\eta = \frac{v^{h+1} - 1}{v - 1} \tag{3.11}$$

En isolant le terme v^{h+1} et passant au logarithme, on obtient :

$$h = \frac{\ln(\eta(v-1)+1)}{\ln(v)} - 1 \tag{3.12}$$

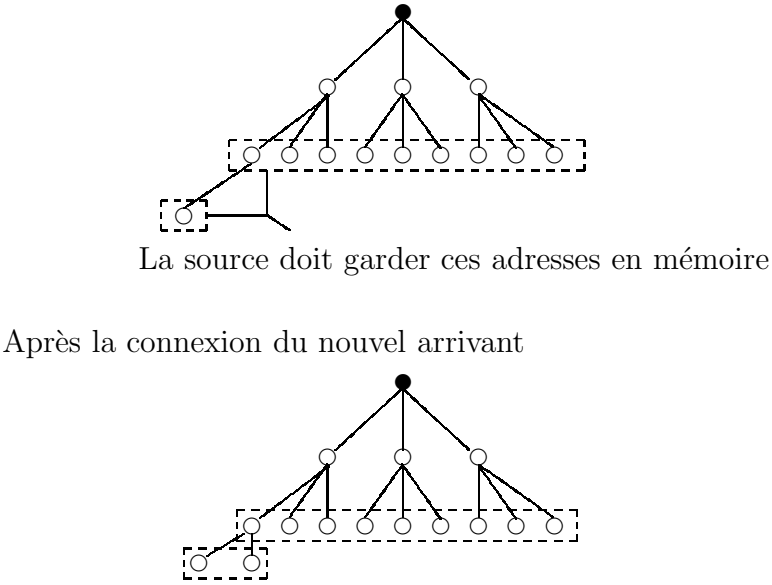


Figure 3.14 – Illustration des adresses que les racines doivent garder en mémoire. $v = 3$

Notons de plus que dans le cas où l'égalité n'est pas vérifiée, l'injection du résultat de 3.12 (dans ce cas la valeur obtenue n'est pas entière) dans l'équation 3.11 permet de retrouver la taille de l'arbre de manière approchée. De plus la valeur v^h permet d'évaluer grossièrement le nombre de nœuds situés au bas de l'arbre (v^h représente exactement ce nombre lorsque l'équation 3.11 est vérifiée).

Dans le cas qui nous intéresse, N récepteurs constituent le réseau de pairs. Ceux-ci sont partagés en τ arbres logiques. Chaque arbre contient donc en moyenne $\frac{N}{\tau}$ nœuds. En conséquence, en utilisant l'équation 3.12, chaque racine doit stocker en moyenne un nombre L d'adresses qui est de l'ordre de :

$$L \propto v^{\frac{\ln\left(\frac{N}{\tau}(v-1)+1\right)}{\ln(v)}-1}$$

Le problème consiste donc à trouver un doublet (τ, v) qui convient. Ce problème concerne notamment le choix de v : lorsque celui-ci augmente, les nœuds ont de plus en plus de voisins.

Ainsi la connectivité du réseau devient de plus en plus robuste. En contrepartie, le nombre d'adresses stockées augmente. Le choix de τ est également problématique : plus il augmente, plus le nombre d'adresses stockées par chaque racine diminue. Cependant au niveau du protocole, cela implique que la source collecte un grand nombre d'adresses (celle des récepteurs choisis pour être racine), et cela morcelle beaucoup le réseau de pairs. La courbe 3.15 expose l'évolution de l'espace mémoire utilisé pour différentes valeurs de τ et v, avec des adresses de type IPv4 (soit des adresses de 4 octets). Les résultats présentés correspondent à $N = 600\,000$, ce qui représente une taille de groupe suffisamment importante pour que l'espace mémoire utilisé puisse poser problème.

Comme cela peut se voir sur la figure 3.15, le choix de τ est primordial, car il permet de réduire de manière importante l'espace mémoire utilisé. Une fois ce paramètre fixé,

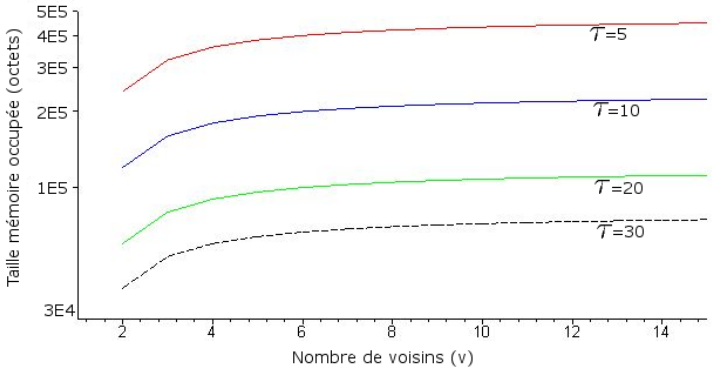


Figure 3.15 – Taille mémoire utilisée en fonction de v et τ. $N = 600\,000$

v fait peu varier l'espace mémoire utilisé dans le sens où l'ordre de grandeur de celui-ci reste ne change quasiment pas. Selon ces résultats, le choix de $\tau = 30$ lorsque $N = 600\,000$ permet de réduire le volume de mémoire occupé à moins de 80 ko par racine,ce qui semble peu contraignant pour celles-ci.

Chaque nœud des arbres doit également garder en mémoire les adresses d'un certain nombre de pairs, pour assurer une connectivité robuste. Les nœuds mémorisent ainsi :

- les adresses des racines connues,
- les adresses de tous les nœuds intermédiaires entre eux et leur racine, et
- les adresses de leurs enfants.

Le nombre d'adresses stockées est donc de l'ordre de $\tau + h + v$. Pour des valeurs de τ réalistes, soit cette grandeur est négligeable par rapport à L, soit N est suffisamment faible pour que le nombre d'adresses stockées ne soit pas contraignant. Il suffit donc de s'assurer que les contraintes sur L sont respectées.

Le choix de τ ne dépend pas uniquement des considérations sur la mémoire. En effet le choix de ce paramètre influence notablement le nombre de requêtes envoyées aux racines. Nous avons donc étudié l'impact de celui-ci sur l'espacement des connexions.

3.4.4.4 Espacement des connexions

Le principe du mécanisme d'espacement des connexions est simple : il consiste à ¡¡ étaler ¿¿ les demandes de connexion sur une plage temporelle, de manière à ce que les racines ne soient pas saturées par ces demandes.

Nous avons choisi pour cela d'adopter un mécanisme similaire au mécanisme de *Nack Suppression* (cf. 3.3.2) : à la réception du message ***P2PRootList***, chaque récepteur ne faisant pas partie de cette liste tire un temps d'attente compris entre 0 et **TMaxP2P**. Compte tenu du paragraphe précédent, le doublet (τ, **TMaxP2P**) détermine la charge de trafic généré vers les racines.

Afin de déterminer des valeurs convenables pour ces paramètres, nous avons calculé le nombre moyen de messages générés sur la voie retour en fonction de ceux-ci. Le nombre de moyen de messages de connexion — noté A_c — reçus par les racines

Si ($N < \tau_{moy} \times A_{c_{MAX}} \times \mathbf{TMaxP2P}_{MAX}$) alors
  $\mathbf{TMaxP2P} = \frac{N}{A_{c_{MAX}} \times \tau_{moy}}$
  Si ($N \leq \eta_{min} \times \tau_{moy}$) alors
    $\tau = \frac{N}{\eta_{min}}$
  Sinon
    $\tau = \tau_{moy}$
Sinon
  $\tau = \frac{N}{A_{c_{MAX}} \times \mathbf{TMaxP2P}_{MAX}}$
  $\mathbf{TMaxP2P} = \mathbf{TMaxP2P}_{MAX}$

Figure 3.16 – Algorithme de choix des paramètres τ et TMaxP2P.

s'exprime comme :

$$A_c = \frac{N}{\tau \times \mathbf{TMaxP2P}}$$

Le choix de τ et **TMaxP2P** est alors conditionné en contraignant la valeur de A_c par un seuil :

$$\frac{N}{\tau \times \mathbf{TMaxP2P}} \leq A_{c_{MAX}}$$

Il faut cependant ajouter une contrainte supplémentaire : **TMaxP2P** ne peut croître de manière trop importante. En effet, lorsque celui-ci augmente, le temps mis pour établir le réseau de pairs augmente. Cela ralenti donc le temps de réponse global du système. Nous avons donc plafonné les valeurs que peut prendre **TMaxP2P** : $\mathbf{TMaxP2P} \leq \mathbf{TMaxP2P}_{MAX}$.

Nous avons ensuite ajouté deux contraintes relatives à τ. La valeur de τ représente le nombre d'adresses que la source doit collecter avant d'envoyer un message ***ConnectP2P***. Cette opération peut demander plusieurs allers-retours, selon la configuration du mécanisme de suppression des retours. Il est donc préférable de limiter la valeur de τ, de manière à ce qu'un minimum d'allers-retours soit nécessaire. En notant τ_{moy} le nombre de réponses obtenues lors de l'envoi d'un message ***Inquiry***, il est même préférable d'avoir $\tau \leq \tau_{moy}$ tant que ce n'est pas contradictoire avec les contraintes précédentes. Enfin il semble inutile d'avoir τ_{moy} racines lorsque la taille du groupe ne le justifie pas. Nous avons donc choisi de fixer un nombre minimum de pairs par arbre : η_{min}.

Compte tenu de ces différentes contraintes, le choix de τ et **TMaxP2P** peut se faire au moyen de l'algorithme suivant décrit dans la figure 3.16.

Après avoir examiné les contraintes exprimées ci-dessus, nous avons cherché à obtenir une expression analytique des performances du réseau ainsi formé. Cette étude s'est avérée trop complexe pour aboutir, notamment à cause de problèmes de dépendances entre récepteurs. En conséquence nous avons opté pour une approche par simulation pour déterminer des valeurs convenables pour ces paramètres.

3.4.5 Résultats de simulations

L'ensemble du programme de simulation a été développé avec le langage java. La réalisation nous a permis en outre de vérifier l'efficacité des techniques décrites ci-dessus.

Les simulations réalisées ont en particulier permis de déterminer une valeur convenable pour le nombre de voisins. En effet, lorsque la valeur de v est importante, la hauteur des arbres diminue. Cela permet en particulier de parcourir rapidement tout un arbre, puisque les messages se diffusent très largement à chaque étage. Cependant cette propriété de diffusion pose un problème : selon les caractéristiques du réseau, il est possible que le nombre de messages générés ¡¡ explose ¿¿. Ce comportement est incompatible avec l'objectif de faible utilisation du réseau. A l'inverse, une valeur trop faible de v diminue la connectivité du réseau. Les recherches prennent donc davantage de temps. De plus certains récepteurs peuvent se retrouver isolés si plusieurs pairs deviennent subitement injoignables (pour cause de panne, d'arrêt volontaire de la transmission, etc.).

Pour l'ensemble des simulations réalisées, le programme était configuré comme suit :

- $\tau_{moy} = 30$. Cette valeur a été choisie car elle était cohérente avec les études réalisées dans le chapitre précédent où $R = 30$. Ainsi tant que le nombre de racine est inférieur à τ_{moy}, un seul message ***Inquiry*** permet d'obtenir toutes les adresses nécessaires.
- $A_{c_{MAX}} = 20$ messages par seconde. Cette valeur a intentionnellement été choisie faible, de manière à ce que, même avec les tirages aléatoires, les valeurs maximales restent acceptables. Notons que compte tenu de la taille des messages ***ConnectP2P***, cela représente un débit de l'ordre de 3kb/s.
- $\mathbf{TMaxP2P}_{MAX} = 180$s, de manière à ce que la valeur $\tau = \tau_{moy}$ puisse couvrir un grand nombre de cas, sans pour autant que le temps d'établissement du réseau devienne trop important.
- La taille du groupe N varie de 600 récepteurs à 600 000 récepteurs.

Dans un premier temps, nous avons cherché à évaluer l'impact des paramètres sur l'efficacité du mécanisme dans le cas où tous les récepteurs restent joignables au cours de la phase terrestre. Puis dans une deuxième phase de simulation, nous avons étudié la robustesse du mécanisme dans des cas où un certain nombre de récepteurs ayant reçu les données devenaient injoignables.

3.4.5.1 Tous les récepteurs sont joignables

Dans les simulations qui suivent, tous les récepteurs peuvent être atteints au moyen du réseau de pair-à-pair. Dans ce cas nominal, nous avons cherché à évaluer l'efficacité du mécanisme global. Nous avons ainsi étudié :

- l'espace mémoire occupé,
- le taux de sollicitation du réseau,
- le coût lié à la recherche d'informations, et
- les performances de ces recherches.

Évaluation de l'espace mémoire occupé

Selon les calculs effectués dans la section 3.4.4.3, pour un nombre de racines donné, l'espace mémoire occupé croît avec la taille du groupe. C'est donc l'occupation de l'espace mémoire dans le cas où la taille du groupe était la plus importante qui a d'abord été évaluée. La figure 3.17 expose l'évolution du nombre d'adresses stockées par les racines en fonction du nombre de voisins (v) que chaque nœud possède. Pour chaque valeur de v, l'espace mémoire maximal, l'espace moyen et l'espace minimum occupé ont été représentés. Chaque point a été obtenu à partir de 300 simulations (il est apparu que les résultats moyens étaient stables au bout de 300 simulations). Dans les résultats présentés, v varie de 2 à 10 : $v = 2$ est une valeur minimale pour établir un arbre un tant soit peu robuste à des déconnexions de ses membres, et $v = 10$ semblait une valeur suffisamment importante pour assurer la robustesse de l'arbre (ce résultat a été confirmé par les simulations réalisées). La figure 3.17 montre que l'écart entre les

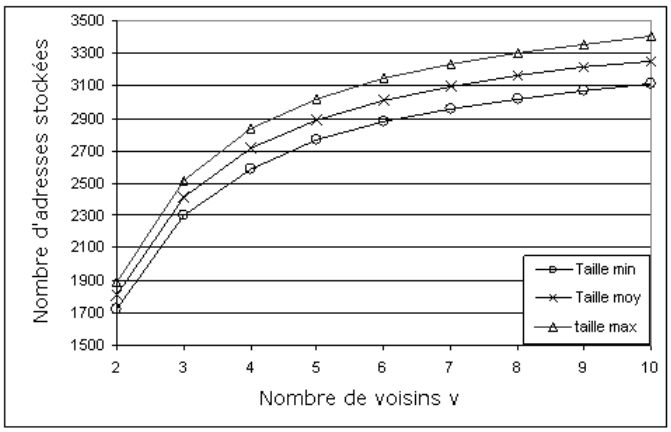


Figure 3.17 – Nombre d'adresses stockées en fonction de v. $N = 600\,000$

valeurs maximales, moyennes et minimales est suffisamment faible pour qu'il ne soit pas trop pénalisant de ne considérer que la valeur maximale. Dans la figure 3.18 le nombre maximum d'adresses stockées a été tracé en fonction de la taille du groupe, et pour différentes valeurs de v. Selon les résultats visibles sur cette figure, le nombre d'adresses stockées par les racines était inférieur à 3 500. Pour des adresses de type IPv4, cela correspond à un volume de mémoire de l'ordre de 14 ko. *A priori* le stockage des adresses ne pose donc pas de problème. Notons que, contrairement à ce à quoi nous pouvions nous attendre, le nombre maximum d'adresses stockées ne correspond pas au cas où la taille du groupe est maximale. Cela est dû au mécanisme destiné à espacer les connexions : lorsque la taille du groupe augmente jusqu'à 600 000 récepteurs, davantage de racine sont utilisées de manière à respecter les contraintes d'espacement des connexions (voir 3.4.4.4). En conséquence le ratio (nombre de récepteurs) / (nombre de racines) a diminué. Chaque racine a donc moins d'adresses à stocker.

Évaluation des risques de saturation des racines

Une autre des contraintes énoncées dans la section 3.4.4 concerne le nombre de messages envoyés simultanément dans le réseau terrestre. Nous avons donc analysé le

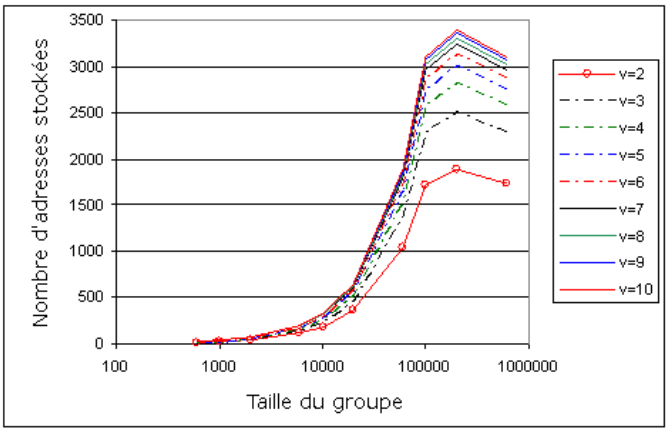


Figure 3.18 – Nombre d'adresses stockées en fonction de v.

débit du flux de messages de connexion reçus par les racines. Plus précisément, pour chaque ensemble de simulation, trois valeurs ont été mesurées :

- le nombre moyen de messages générés par seconde,
- le nombre maximum de messages générés au cours d'une seconde, et
- le nombre maximum de messages générés au cours d'un intervalle δt.

La seconde valeur rend compte du nombre maximum de messages envoyés par seconde ; tandis que la troisième est représentative du débit instantané (cette valeur est précisément le débit instantané lorsque $\delta t \to 0$). A titre d'exemple, la courbe 3.19 représente l'évolution du nombre de messages générés pendant un intervalle $\delta t = 0,2$ s durant l'établissement du réseau au cours d'une expérience particulière ($N = 600\,000$ et $v = 10$).

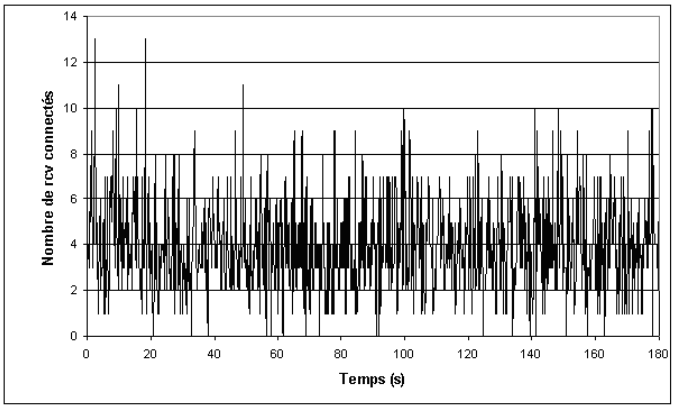


Figure 3.19 – Evolution du nombre de messages générés pendant δt secondes. $N = 600\,000$, $v = 10$.

Notons que le paramètre v n'a pas d'influence sur les grandeurs considérées : pour un nombre de racines égal, seule la taille du groupe a une influence sur celles-ci. La figure 3.20 montre l'évolution de ces grandeurs en fonction de la taille du groupe. Ces simulations étaient paramétrées avec $\delta t = 0,2$ secondes. Cette valeur a été choisie de manière à ce qu'il soit possible d'observer un ou plusieurs messages dans chaque

intervalle, pour toute taille de groupe (d'autres simulations ont été effectuées avec des valeurs de δt plus faibles mais les résultats étaient difficilement comparables lorsque la taille du groupe variait). Chaque point représenté sur la courbe correspond à la

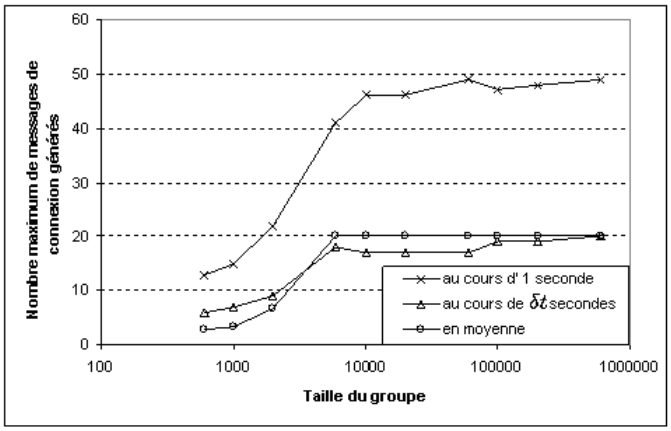


Figure 3.20 – Débits de réception de messages *P2PConnect* en fonction de la taille du groupe. $\delta t = 0,2$ et $v = 10$.

valeur maximale subie observée sur 300 expériences menées. On peut observer sur la figure que les valeurs mesurées croissent lorsque la taille du groupe augmente de 600 récepteurs jusqu'à 6 000 récepteurs. Ensuite pour des groupes dont la taille est comprise entre 6 000 et 600 000 récepteurs celles-ci restent constantes. Cela s'explique par le fait que lorsque la taille du groupe est inférieure à 6 000, il n'y a pas assez de récepteurs pour que 20 messages par seconde soient générés en moyenne. Au-delà de cette valeur, le nombre moyen de messages générés par seconde reste constant et égal à 20. Les résultats observés permettent de conclure que le mécanisme proposé pour espacer les demandes de connexion est efficace : bien que la taille du groupe augmente, le nombre moyen de messages par seconde reste constant. Toutefois, le nombre de messages de connexion générés présente des pics. Ces pics sont cependant relativement constants :

- le nombre maximum de messages générés au cours d'une seconde reste de l'ordre de 48 messages par seconde, et
- le nombre maximum de messages générés au cours de $\delta t = 0,2$ secondes est compris entre 17 messages (soit un débit de 85 messages par seconde) et 20 messages (soit un débit de 100 messages par seconde).

Bien que ces valeurs soient assez éloignées de la moyenne ciblée, elles restent cependant acceptables, la taille de ces messages étant de 14 octets.

Évaluation du coût lié à la recherche d'informations

Après nous être assurés que l'utilisation de cette technique était bien acceptable pour les racines et pour le réseau, nous avons analysé le coût lié à la recherche d'informations. L'objectif de ces mesures était de vérifier que ce mécanisme était compatible avec le principe de notre approche : la diminution du coût global de la communication.

Pour cela parmi les N récepteurs, **MinRcv** récepteurs ont été choisis aléatoirement de manière à représenter des récepteurs n'ayant pas reçu toutes les informations. Nous

avons ensuite compté combien de messages de recherche étaient générés sur le réseau de pairs. Les résultats qui suivent sont obtenus ainsi : les récepteurs ayant effectué des recherches sont numérotés de 1 à 300. Ils sont numérotés de manière à ce que le nombre de messages générés par leur recherche soit croissant. Il est apparu au cours des simulations que les cas où le nombre de messages de recherche générés était le plus important correspondait à des tailles de groupe faibles. Cela s'explique par le fait que plus la taille du groupe augmente, plus la proportion de récepteurs ayant correctement reçu les données augmente. Ainsi la probabilité pour qu'un récepteur cherchant des données se retrouve relié à un pair ayant reçu les données augmente. Les recherches s'arrêtent donc plus vite dans ce cas, et le nombre de messages générés diminue. La figure 3.21 présente les nombres de messages de recherche générés par chacun des récepteurs, pour un groupe de $N = 600$ récepteurs pour différentes valeurs de v. Chaque courbe est la moyenne de courbes identiques obtenues à partir de 300 expériences. Sur cette figure, les différents paliers visibles correspondent à des positions

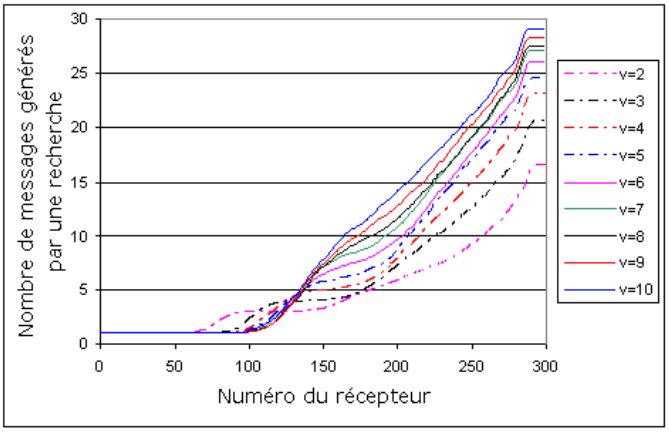


Figure 3.21 – Nombre de messages de recherche générés par chacun des récepteurs. $N = 600$**.**

différentes dans l'arbre de pairs :

- les récepteurs situés en feuille de l'arbre et dont le père a reçu les données ne génèrent qu'un message,
- les récepteurs situés au ¡¡ milieu ¿¿ de l'arbre et dont les voisins et le père possèdent les données génèrent $v + 1$ messages, et
- les autres récepteurs génèrent davantage de messages car leur recherche se propage dans le réseau.

Cette courbe laisse déjà présager qu'il va être nécessaire de trouver un compromis entre l'efficacité des recherches, et le coût de ces recherches. Par exemple lorsque l'on compare les courbes correspondant aux cas $v = 2$ et $v = 10$, on remarque que lorsque $v = 10$ davantage de récepteurs ne nécessitent qu'un message de recherche, mais que l'aire de la courbe (qui est proportionnelle à la somme des messages générés) augmente. En d'autres termes, le coût global de recherche augmente lorsque v augmente, tandis que la probabilité d'avoir un récepteur voisin possédant les informations augmente. Afin de déterminer clairement l'impact de la taille du groupe et de v sur le nombre total de messages générés, les courbes représentées dans la figure 3.22 ont été tracées.

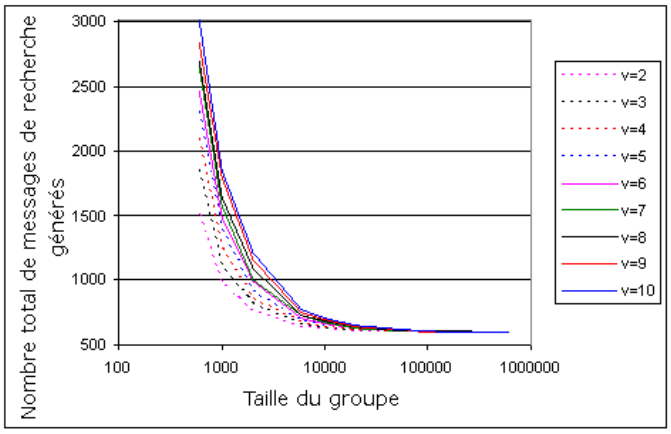


Figure 3.22 – Nombre total de messages de recherche générés.

Sur cette figure on peut voir que lorsque le paramètre v augmente, le nombre de messages globalement générés augmente. Cependant dès que la taille du groupe augmente l'influence du paramètre v diminue ; et à partir de 6 000 récepteurs, le nombre de messages générés se stabilise. Notons de plus que dans tous les cas, le coût généré par ces recherches est faible par rapport au coût de diffusion. En effet lorsque la taille des objets envoyés est de l'ordre de 30 Mo (cf. 3.4.4.2), le coût de transmission par satellite d'un bloc est de l'ordre de 6×10^4. De plus lorsque plusieurs blocs sont entrelacés, les recherches peuvent porter sur le méta-bloc complet puisque les pertes sont réparties sur tout celui-ci. Il faudrait alors comparer le coût de recherche au coût de transmission du méta-bloc complet. Le coût des recherches d'informations au sein du réseau n'est donc pas vraiment pénalisant par rapport à l'approche considérée.

Efficacité de la recherche d'informations

Après avoir évalué les coûts de recherche d'informations, nous avons cherché à obtenir une évaluation sur les performances des recherches effectuées. Pour cela, pour chaque récepteur, le nombre minimum de pairs contactés avant de trouver les informations recherchées a été mesuré. Cette valeur sera appelée par la suite nombre minimum de sauts. Comme dans le paragraphe précédent, il est apparu que c'était pour des recherches effectuées au sein de petits groupes que les résultats étaient les moins satisfaisants. La figure 3.23 montre la répartition moyenne du nombre minimum de sauts pour un groupe de $N = 600$ récepteurs, pour trois valeurs de v. Cette moyenne a été effectuée sur 300 expériences. Trois conclusions peuvent être tirées de ces courbes :

1. lorsque v augmente, le nombre de récepteurs ne trouvant les informations avec un seul saut diminue,
2. lorsque v augmente, le maximum des valeurs mesurées diminue, et
3. lorsque v augmente, le nombre de récepteurs n'ayant pas trouvé les informations au bout de deux sauts diminue.

Ce comportement peut s'expliquer de la manière suivante : plus v augmente, plus la probabilité qu'un récepteur se retrouve en position de feuille augmente. Cependant

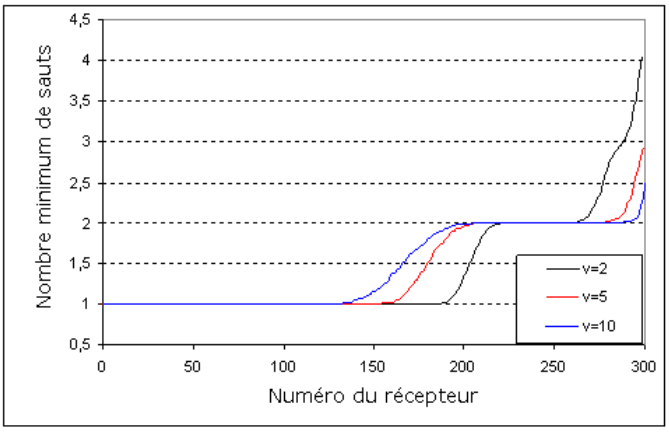


Figure 3.23 – Nombre minimum de sauts pour chaque récepteur. $N = 600$.

compte tenu de la taille du groupe il est aussi relativement probable que plusieurs récepteurs situés en feuille se retrouvent dépendant d'un père ne possédant pas les données. Par contre, comme le nombre moyen de récepteurs contactés au bout de deux sauts augmente avec v, la probabilité de trouver les informations recherchées au bout de deux sauts fait de même.

Suite à ces résultats, nous avons utilisé la valeur maximale du nombre minimum de sauts pour caractériser l'efficacité de recherche du réseau (en d'autres termes, nous avons considéré le récepteur pour qui la réponse était la plus longue à venir). Les moyennes de ces maxima sont représentées sur la figure 3.24 pour plusieurs valeurs de v. Ces moyennes ont été calculées à partir de 300 expériences. Conformément à ce qui

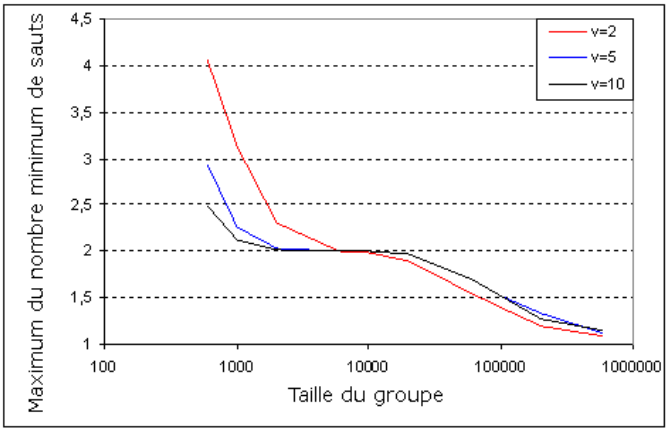


Figure 3.24 – Maximum du nombre minimum de sauts en fonction de la taille du groupe.

était annoncé, pour des groupes de moins de 10 000 récepteurs, l'augmentation de v fait diminuer le maximum des minima mesurés. En revanche, pour des groupes plus importants, la tendance s'inverse. Cependant pour des groupes de taille peu importante, l'influence de v est considérable (pour $N = 600$, la valeur du maximum passe de $4,05$ avec $v = 2$ à $2,48$ avec $v = 10$) ; tandis que pour des groupes de taille supérieure à 10 000 récepteurs, son influence est moindre (pour $N = 600\,000$, la valeur du maximum

passe de $1,09$ avec $v = 2$ à $1,15$ avec $v = 10$). Il est donc préférable de choisir une valeur de v élevée.

Au vu des résultats présentés dans cette section, le mécanisme proposé est efficace en termes de recherche, sans être pour autant trop coûteux. Toutefois, les simulations réalisées ne prenaient pas en compte la possibilité que certains récepteurs quittent prématurément le groupe et deviennent injoignables. La section suivante étudie donc le comportement du mécanisme dans une telle situation afin d'évaluer la robustesse du mécanisme.

3.4.5.2 Certains récepteurs ne sont pas joignables

Cet ensemble de simulations a été réalisé de manière à déterminer si le fait qu'un sous-ensemble de récepteurs ne soit plus joignable au moment où les récepteurs recherchent les données avait un impact vraiment négatif sur le service rendu par le mécanisme. Pour ce faire, une partie des récepteurs ayant reçu les données s'est extraite du réseau après sa constitution. Ces récepteurs ont été choisis aléatoirement parmi l'ensemble des récepteurs ayant reçu les données. Nous avons choisi de rendre ces récepteurs indisponibles *après* l'établissement du réseau car lorsque le réseau se met en place, *a priori* aucun récepteur ne sait s'il recevra toutes les données ou pas. Tous les récepteurs ont donc intérêt à se joindre au réseau. Cependant une fois les données reçues, il est envisageable que certains récepteurs décident de ne pas faire profiter les autres membres du groupe des données qu'ils ont reçues. De plus la possibilité que des récepteurs n'ayant pas reçu les données deviennent définitivement indisponibles n'a pas été prise en compte (dans la mesure où le réseau de pair est établi pour leur rendre service).

Compte tenu des indications données ci-dessus, le fait que des récepteurs deviennent indisponibles n'a aucun impact sur le volume de mémoire requis pour mettre le réseau en place, ni sur l'espacement des connexions au réseau. En revanche cela a un impact sur le coût et l'efficacité moyenne des recherches puisqu'il y a moins de récepteurs possédant les informations dans le réseau. Contrairement à ce que nous pensions de prime abord, le mécanisme est relativement robuste, quel que soit le choix du paramètre v, et quelle que soit la taille du groupe. Pour observer des cas où au moins un des récepteurs n'arrivait pas à trouver les informations qu'il cherchait, il a fallu simuler des configurations très défavorables au mécanisme. Au cours des simulations réalisées, il est apparu que les cas les plus défavorables étaient réalisés pour des groupes de petite taille. Cette situation est donc réalisée au moyen de plusieurs simulations avec un groupe de 600 récepteurs, puis l'impact des paramètres sur la qualité des recherches est analysé. Les résultats présentés correspondent à des simulations où 300 récepteurs n'avaient pas reçu les données, et 280 récepteurs parmi les 300 ayant correctement reçu les informations étaient indisponibles. Cette configuration a été testée pour différentes valeurs de v, et pour chaque valeur 1 000 simulations ont été réalisées. La figure 3.25 présente, en fonction de v, le pourcentage de simulations où au moins un récepteur n'a pas réussi à contacter un autre pair possédant les données. Conformément à ce que l'on pouvait attendre, lorsque v augmente, ce pourcentage diminue globalement. Et même, lorsque $v = 9$ ou 10, aucun cas de ce genre n'a été observé.

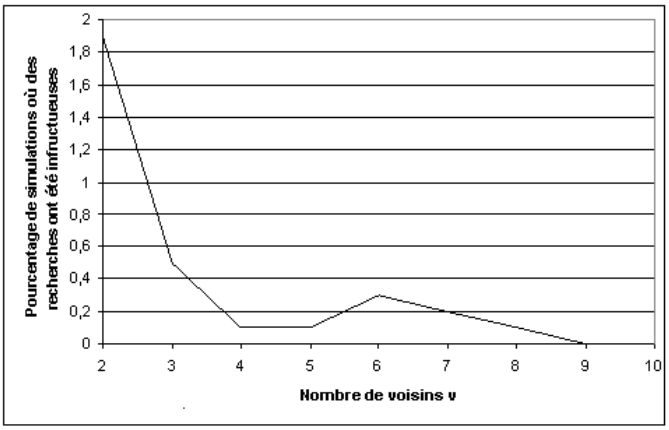


Figure 3.25 – Pourcentage de simulations où des recherches ont été infructueuses

Ensuite pour l'ensemble des simulations, le nombre maximum de récepteurs (au cours des 1 000 simulations) n'ayant pas réussi à obtenir les données a été tracé. En fait lorsque le mécanisme échoue (i.e. lorsque des récepteurs n'arrivent pas à retrouver les données), il échoue pour un grand nombre de récepteurs. Ce comportement peut s'expliquer par le fait que si un récepteur appartenant à un arbre n'arrive pas à retrouver les données, il est probable qu'un grand nombre de récepteurs dans le même arbre soit dans le même cas. La courbe 3.26 représente les valeurs obtenues. Selon ces résultats, des valeurs de v inférieures à 8 laissent la possibilité de rencontrer des cas d'échecs. Par contre, dès lors que $v = 9$ ou 10, les probabilités d'échecs s'amenuisent. En fait, aucun cas de ce genre n'a été constaté au cours de l'ensemble des simulations menées. Nous avons donc fait l'hypothèse que le choix de $v = 10$ permettait de rendre le service désiré dans la très grande majorité des cas.

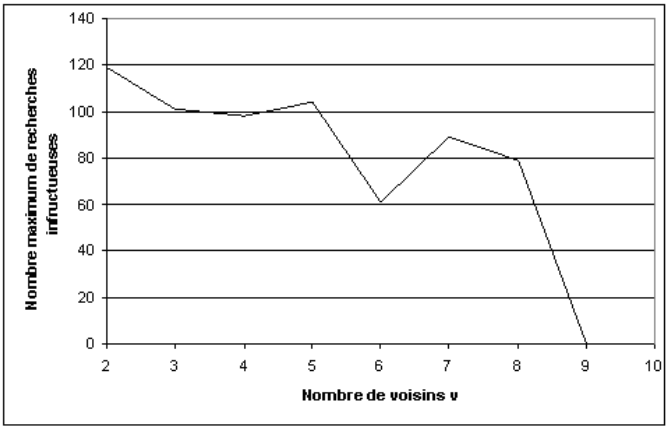


Figure 3.26 – Nombre maximum de recherches restées infructueuses au cours de l'ensemble des simulations.

Un autre critère évalué au cours de ces simulations concerne la portée des recherches. En effet, dans les simulations réalisées pour la section précédente, tous les récepteurs trouvaient les informations au sein de l'arbre v-aire auquel ils appartiennent. Avec

l'ajout de récepteurs indisponibles dans le réseau, plusieurs récepteurs ont été contraints de contacter les racines des autres arbres car ils se sont retrouvés isolés. Ce type de recherche est appelé par la suite *recherche étendue*, dans la mesure où celle-ci est étendue à tout le réseau de pairs. En évaluant ces valeurs, l'objectif était de dégager l'intérêt de donner la possibilité au pair de joindre directement les racines. La figure 3.27 présente les valeurs obtenues au cours des simulations. Selon ces résultats, un

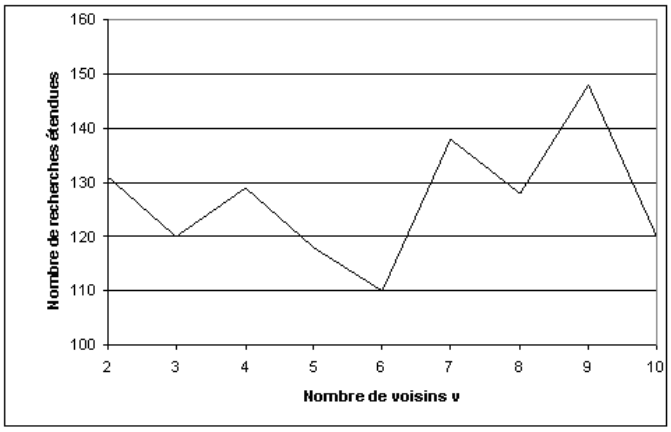


Figure 3.27 – Nombre maximum de recherches étendues effectuées au cours de l'ensemble des simulations.

grand nombre de recherches étendues a été effectué. En particulier, lorsque v avait pour valeur 9 ou 10, plus de 120 recherches de ce type ont été menées. Sans cela tous ces récepteurs n'auraient pu trouver les données. Cette précaution est donc vraiment indispensable.

Suite aux résultats précédents, le mécanisme a été configuré avec $v = 10$, de manière à ce que le taux d'échecs soit minimum. Nous avons ensuite réalisé un ensemble de simulations où, pour des groupes dont la taille était supérieure à 600 récepteurs, 50% du groupe initial était indisponible. Pour ces simulations, le coût et la qualité des recherches ont été analysés. Cependant avec cette configuration, aucun cas d'échec n'a été observé au cours de l'ensemble des simulations réalisées. Nous avons donc fait l'hypothèse que le choix de $v = 10$ permettait de rendre le service désiré dans la très grande majorité des cas. Notons toutefois que, selon la figure 3.28, le coût de recherche est réellement augmenté pour des groupes de taille peu importante. Il reste néanmoins encore acceptable, surtout lorsque le groupe est composé de 1 000 récepteurs ou plus.

Le fait que seulement 50% du groupe soit disponible a également un impact sur la rapidité à laquelle les informations sont localisées au sein du réseau. La courbe 3.29 représente cette influence au travers du nombre maximum moyen du nombre minimum de sauts : celui-ci augmente légèrement sans pour autant que cela devienne vraiment pénalisant.

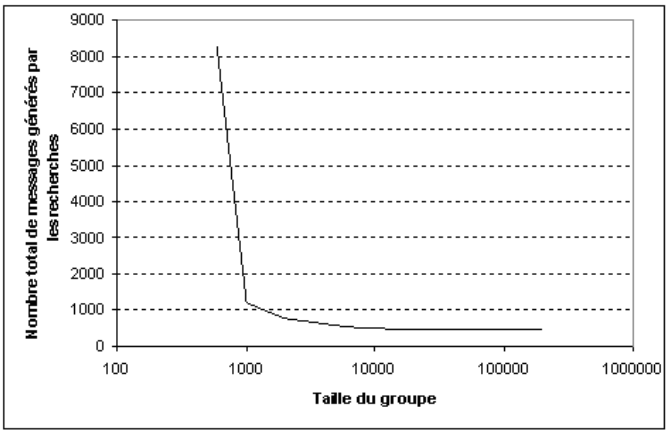


Figure 3.28 – Nombre de messages générés par les recherches.

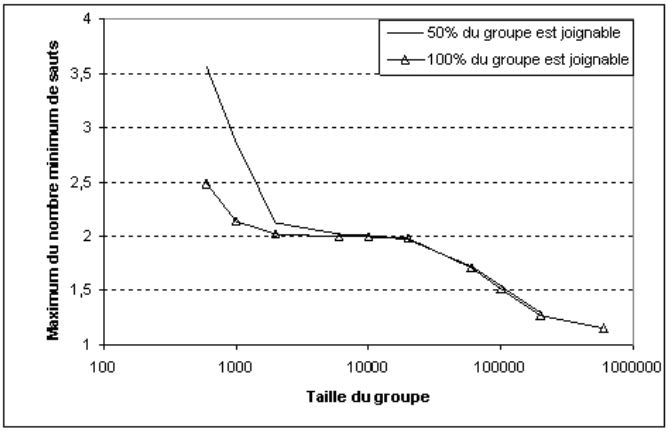


Figure 3.29 – Maximum du nombre minimum de sauts en fonction de la taille du groupe.

3.4.6 Conclusion sur les performances du mécanisme de reprise des erreurs par voie terrestre

Cette section a étudié en détail la conception d'un mécanisme visant à reprendre les erreurs par voie terrestre. Ce mécanisme se base sur la création d'un réseau logique permettant de relier les récepteurs entre eux. Selon les calculs et les simulations réalisées, le mécanisme proposé permet de rendre le service désiré : localiser rapidement des informations manquantes sans pour autant que le coût soit prohibitif. De plus, selon les résultats de simulation, cela est particulièrement vrai pour des groupes de taille importante (1 000 récepteurs ou plus).

Plusieurs améliorations pourraient cependant encore être apportées à ce mécanisme. Par exemple, lorsqu'un récepteur recherche un volume important de données, aucune optimisation n'est effectuée. Or il est possible d'optimiser le rapatriement des données codées en utilisant plusieurs sources d'informations en parallèle soigneusement choisies [23]. Une autre voie d'amélioration concerne la dissémination des informations au sein du réseau. Il est en effet envisageable d'agréger plusieurs demandes lorsque celles-ci

arrivent à un même nœud. Cela aurait pour effet de diminuer le nombre de messages envoyés, mais au prix de performances diminuées.

3.5 CONCLUSION SUR LES POINTS DURS ÉTUDIÉS

Plusieurs mécanismes indispensables à l'approche hybride satellite/terrestre envisagée dans la section 2.3.3 ont été analysés dans le présent chapitre. Le fait de disposer d'un code permettant de générer des paquets encodés est nécessaire, car sans cela chaque paquet perdu doit être géré spécifiquement. Ces codes doivent de plus avoir des vitesses d'encodage et de décodage compatibles avec les débits en émission disponibles. Dans le cadre de notre approche, l'utilisation d'un code MDS permettant de générer jusqu'à $2^{16} - 1$ paquets encodés remplit ces conditions, sans pour autant introduire de surcoût en réception.

L'estimation de la taille du groupe est incontournable car toute la proposition repose sur la connaissance de la taille du groupe. Le mécanisme que nous avons étudié est de plus relié à un mécanisme permettant de limiter le nombre de réponses émises sur la voie retour. L'efficacité de ces deux mécanismes a été évaluée au moyen de simulations, au cours desquelles ceux-là ont montré des performances suffisantes pour remplir la fonction attendue.

Enfin la reprise des informations par voie terrestre est indispensable pour garantir la fiabilité de la communication. Selon les résultats de la section 3.4, le mécanisme proposé dans ce but permet de retrouver des informations au sein du réseau sans que le nombre de messages générés soit trop important.

Maintenant ces mécanismes définis, il est possible de décrire entièrement une proposition de protocole de transport, conçu selon l'approche hybride, et utilisant ces mécanismes. Le chapitre suivant détaille donc une telle proposition, et spécifie en particulier le déroulement d'une session, les messages échangés au cours de celle-ci, ainsi que le format de ces messages.

Chapitre 4

Proposition d'un protocole de transport spécifique au service : HSTRM

4.1 DÉFINITION D'UN SERVICE DE TRANSPORT MULTIPOINT

Le chapitre 2 a, entre autre, dégagé l'impact des caractéristiques des liens satellites sur les services rendus par le niveau transport. Ce chapitre se concentre sur une proposition visant à rendre un service de communication adapté au satellite : un service de communication de groupe fiable à grande échelle et à ¡¡ moindre coût ¿¿. Tout d'abord, la section suivante précise le service visé. Ensuite la section 4.3 expose les différentes solutions techniques adoptées en réponse aux exigences liées au service. Après une brève présentation de ces solutions (qui correspondent aux études menées au chapitre 3), les comportements de la source et des récepteurs au cours d'une transmission sont décrits. Pour finir, les différents messages apparaissant lors de la transmission sont définis et détaillés.

4.1.1 Définition du service de transport considéré

Compte tenu du problème du *no-one-size-fits-all* (cf. section 1.2.4), il n'est pas possible de définir un protocole de transport répondant aux besoins de toutes les applications existantes [98]. Il est donc nécessaire de bien définir avant tout le service rendu par la couche transport, et le besoin applicatif auquel elle répond.

Le satellite représente un moyen particulièrement efficace pour transmettre des informations vers de nombreux utilisateurs, et est même d'autant plus rentable qu'il y a

de récepteurs. A cela s'ajoute le fait que le délai de transmission des systèmes satellite (de l'ordre de 300 ms) n'est pas vraiment compatible avec des applications ayant des contraintes temporelles. Le satellite est donc particulièrement indiqué pour la diffusion de données à grande échelle, pour des applications n'ayant pas (ou peu) de contraintes temporelles.

Dans ce cadre nous nous intéressons par la suite au problème des transmissions fiables et à grande échelle (plusieurs milliers de récepteurs) de données, et plus particulièrement aux applications qui doivent être assurées que tous les membres du groupe ont reçu l'information. Un exemple d'une telle application est le transfert payant de fichiers multimédia (vidéos, jeux, etc.) : comme les récepteurs payent le téléchargement des données, il est nécessaire de garantir que tout le groupe a reçu les informations à la fin de la transmission. Nous supposons également que la communication implique une unique source qui désire envoyer les données à un groupe de récepteurs. Pour finir le coût de la transmission est supposé proportionnel à l'utilisation des services réseaux, et il est réparti parmi tous les récepteurs après une transmission. Ce mode de tarification a été proposé dans [18] pour des communications multipoints.

La transmission est annoncée préalablement, et les récepteurs sont libres de s'inscrire à la session Multicast. L'application n'est pas explicitement informée des abonnements des récepteurs. Une fois inscrits à la session Multicast, les récepteurs ne peuvent pas se désabonner, sauf pour quitter définitivement la session Multicast. Il n'est pas non plus possible pour un récepteur de s'abonner à la session Multicast après le début de la transmission.

4.1.2 Principales caractéristiques du protocole de transport

Dans le cadre d'un service réseau IP Multicast *Best Effort* le service de transport présenté ci-dessus nécessite l'utilisation d'un protocole de transport fiable. Compte tenu du problème du *no-one-size-fits-all*, la proposition présentée dans ce document répond à l'objectif fixé au paragraphe précédent, et a pour seul but d'être optimisée dans ce cadre précis. Les paragraphes suivants recensent les différentes caractéristiques que le protocole de transport doit impérativement avoir.

Le protocole de transport doit avant tout garantir que tous les récepteurs ont reçu les données transmises au niveau applicatif. Il faut noter qu'il existe des protocoles de transport proposant une fiabilité statistique. Cette technique n'est cependant pas adaptée à notre objectif. En effet d'une part elle ne permet pas de garantir que tous les récepteurs ont reçu les données (bien que la probabilité pour que ce soit le cas est élevée), et d'autre part elle implique un codage systématique des données et donc une utilisation potentiellement inutile des ressources du réseau.

Un autre impératif pour le protocole de transport concerne le passage à l'échelle. En effet le service de transport visé prend en charge une diffusion de données à grande échelle, le protocole doit donc supporter sans risque une utilisation à grande échelle. Cela implique en particulier que le problème d'implosion des messages vers la source soit géré avec précision.

Une des contraintes fixées par le service désiré concerne le coût de la communication. Nous avons en effet considéré que celui-ci était proportionnel à l'utilisation des services

réseaux. Or bien que le coût d'une transmission satellite par récepteur diminue lorsque la taille du groupe augmente, l'utilisation d'un système satellite reste assez onéreuse. Ainsi l'application voudra être assurée que l'utilisation des réseaux terrestre et satellite est optimale. Le protocole de transport doit donc éviter toute utilisation du système satellite dès que cela est moins rentable que l'utilisation du réseau terrestre.

Pour finir, de manière à pouvoir utiliser le réseau terrestre, le protocole doit intégrer un mécanisme de reprise terrestre des informations. Ce mécanisme doit permettre de trouver des informations parmi tous les récepteurs abonnés à la session Multicast, et de les rapatrier afin de compléter la réception des informations. Les sections qui suivent décrivent en détails une proposition de protocole de transport permettant de rendre le service décrit dans la section précédente. La section suivante précise l'approche adoptée pour rendre ce service, ensuite la section 4.4 décrit le déroulement d'une session correspondant à l'approche adoptée. Enfin l'ensemble des messages échangés sont décrits au cours de la section 4.5.

4.2 PRÉSENTATION DE L'APPROCHE HYBRIDE SATELLITE/ TERRESTRE

4.2.1 Principe de l'approche adoptée

Nous proposons dans ce document une nouvelle approche pour le transport Multicast fiable à grande échelle. La proposition, nommée *Hybrid Satellite/Terrestrial Reliable Multicast* (HSTRM) dans la suite du document, est destinée à la transmission de données vers de grands groupes par satellite. HSTRM est destiné à fonctionner avec un réseau hybride satellite/terrestre, c'est-à-dire un réseau où les récepteurs sont connectés à un réseau terrestre, et à un réseau satellite [31]. La proposition prend en compte les particularités d'une liaison satellite pour garantir une fiabilité totale lors de la transmission à grande échelle de données applicatives. Un des principaux buts de ce protocole est d'optimiser le coût des transmissions multipoints. Ainsi, en nous basant sur les résultats de la section 2.3, nous proposons d'adopter l'approche qui consiste à choisir le mode de transmission des données — terrestre ou satellite — en fonction du nombre de récepteurs intéressés par la diffusion. Le mode de transmission est choisi de tel manière qu'il soit rentable au regard d'une fonction de coût préalablement définie (dans notre cas ce sera la fonction de coût définie dans la section 2.3). Ainsi le satellite est utilisé pour effectuer ce pour quoi il est vraiment utile : diffuser des données vers de nombreux récepteurs, tout en limitant le coût global de la transmission.

4.2.2 Hypothèses sur le réseau sous-jacent

HSTRM est conçu pour fonctionner sur un réseau hybride satellite/ terrestre, ce qui signifie que les utilisateurs finals sont connectés au réseau terrestre et au réseau satellite. Le système visé est précisément celui qui est décrit dans la section 2.2.1 :

- le satellite est géostationnaire et mono-spot,
- il opère en bande Ka, et

- la transmission est conforme au standard DVB-S : les données sont encapsulées dans des trames MPEG2-TS (au moyen de la couche MPE —*Multi Protocol Encapsulation* [89]— ou de la couche ULE —*Ultra Lightweight Encapsulation* [36]— proposée par le groupe de travail IPDVB [63]).
- les récepteurs sont connectés au réseau terrestre au moyen d'un réseau d'accès haut débit,
- les récepteurs sont connectés au réseau satellite soit directement via soit directement avec une antenne VSAT (Very Small Aperture Terminal), soit au moyen d'un réseau d'accès haut débit.

Nous faisons l'hypothèse par la suite que le réseau d'accès avec lequel la source est connectée au réseau satellite est suffisamment bien dimensionné pour éviter qu'il soit saturé par le flux de données émis par la source. Cela implique que la source ne demande pas un débit en émission trop important (i.e. supérieur à la capacité du réseau d'accès). Pour finir, la même hypothèse est faite pour les récepteurs connectés au réseau satellite au moyen d'un réseau d'accès. Pour les récepteurs connectés au moyen d'un VSAT, il ne peut y avoir de congestions de systèmes intermédiaires. Il est cependant possible que le système de l'utilisateur final soit saturé par le flux de données. Cette ¡¡ saturation ¿¿ du système de l'utilisateur final n'est pas dommageable pour le réseau, et les pertes engendrées seront récupérées au même titre que des erreurs de transmission.

Pour finir nous supposons que le système satellite est capable de déterminer si au moins un récepteur est abonné à la session Multicast. Cette contrainte, énoncée dans le projet DIPCAST, est nécessaire. En effet sans cette fonctionnalité, il serait possible de solliciter le système satellite pour transmettre des données sans qu'aucun récepteur ne soit présent, et ainsi utiliser inutilement les ressources satellites. Nous supposons également que si aucun récepteur n'est abonné à la session Multicast, le système satellite en informera la source dès qu'elle voudra émettre des informations sur l'adresse Multicast utilisée (afin que la source ne continue pas à utiliser inutilement le réseau).

4.3 PRÉSENTATION GÉNÉRALE DU PROTOCOLE

Le protocole décrit est basé sur l'approche exposée dans la section 4.2. De manière synthétique, le fonctionnement de la proposition (HSTRM) est le suivant :

- La source commence la transmission des informations au moyen du satellite. Au cours de cette phase elle estime régulièrement la taille du groupe.
- Lorsque la source a transmis toutes les informations qui lui ont été confiées par l'application, deux alternatives sont possibles :
  - Soit il ne reste pas assez de récepteurs pour que la transmission satellite soit rentable (cela correspond au cas où la qualité de réception était bonne pour la majorité des récepteurs), et dans ce cas la source arrête la transmission satellite.
  - Soit il reste assez de récepteurs pour que la transmission satellite soit rentable(un nombre important de récepteurs n'a pas reçu les données). Dans

ce cas la source continue à utiliser le lien satellite : elle envoie alors des informations encodées FEC. Ces informations sont utiles à tous les récepteurs ayant subi des pertes au cours de la première phase.

Au cours de cette phase, la source estime le nombre de récepteurs n'ayant pas encore reçu les données. Lorsque ce nombre devient trop faible pour que la transmission satellite soit rentable, la source arrête celle-ci.

- Lorsque la transmission satellite s'arrête, deux possibilités se présentent :
  - Soit tout le groupe a reçu les informations, et dans ce cas la source clôt la session.
  - Soit il reste encore des récepteurs n'ayant pas reçu les données. Ces récepteurs cherchent alors à contacter par le réseau terrestre d'autres récepteurs ayant reçu les données, pour récupérer les données qui leur manquent. Pendant cette phase la source continue à estimer le nombre de récepteurs recherchant encore des données. Lorsque ce nombre passe à 0, la source clôt la session.

Ce fonctionnement est schématisé dans la figure 4.1.

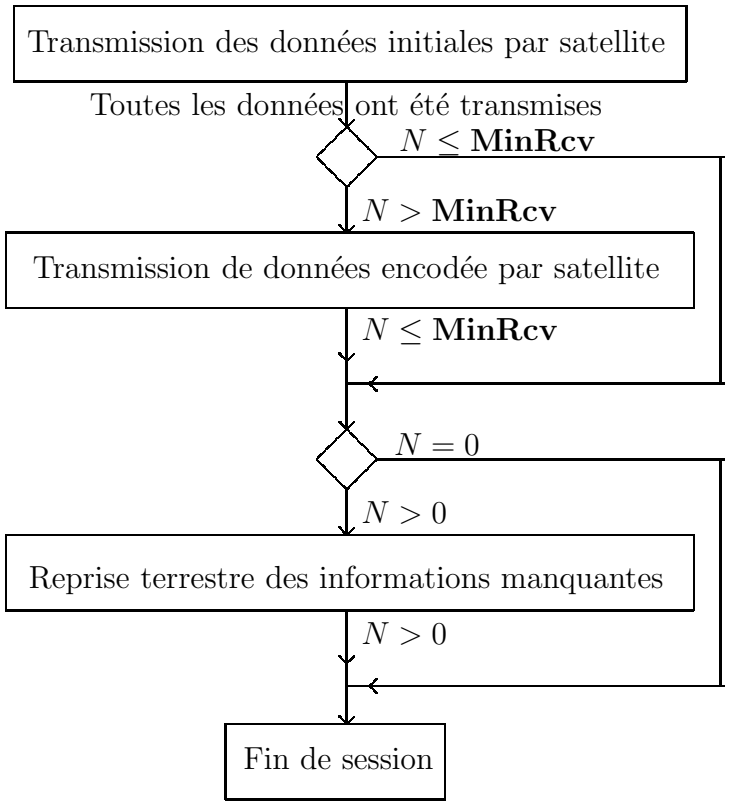
Les sections qui suivent dépeignent les principales caractéristiques techniques de HSTRM. Ces caractéristiques traduisent les exigences énoncées dans la section 4.1.2 : le service décrit demande la mise en œuvre d'un ensemble de mécanismes appropriés. Pour chaque mécanisme, des choix techniques ont été effectués. Ces choix sont présentés et justifiés dans les sections qui suivent, Ils concernent les mécanismes de fiabilisation, de limitation du trafic retour d'estimation de taille de groupe, de reprise d'erreurs, de contrôle de congestion, ainsi que l'annonce préalable des sessions Multicast. Le nom des messages échangés lors de l'utilisation de ces mécanismes sont également introduits au cours des sections qui suivent.

4.3.1 Annonce de session Multicast

Nous rappelons que le service réseau propose un service de communication multipoint. Ainsi la source peut émettre les informations vers une adresse de groupe. Cependant pour que les récepteurs puissent s'abonner au(x) groupe(s) de transmission multipoint qui l'(es) intéresse(nt), il est nécessaire d'annoncer les sessions en cours et à venir. Nous supposons pour cela qu'un outil est disponible pour informer les récepteurs des sessions disponibles, ainsi que des caractéristiques de ces sessions Multicast. Celles-ci seront détaillées au chapitre 4.4.1. Avec elles, tout utilisateur peut s'abonner à une session Multicast, et le récepteur HSTRM peut être configuré. Il est possible de se joindre à la session Multicast tant que la transmission des informations n'a pas débuté, c'est-à-dire tant que l'heure de début de session n'est pas dépassée. Ce protocole ne prend donc pas en compte le *late joining*.

4.3.2 Mécanisme de fiabilisation

Un des objectifs de HSTRM est de proposer un service de communication multipoint fiable. La proposition se base sur l'approche *Nack Oriented Reliable Multicast* [117] du



N : nombre de récepteurs

MinRcv : nombre de récepteurs en deçà duquel la transmission satellite n'est plus rentable

Figure 4.1 – Schéma de fonctionnement de HSTRM

groupe de travail *Reliable Multicast Transport* (RMT) de l'IETF pour fiabiliser les transmissions multipoints. Cette approche consiste à utiliser des requêtes de retransmission (ou ARQ : Automatic Retransmission reQuest) : les récepteurs demandent à la source des retransmissions tant qu'il leur manque des informations. Couplé à cette approche, HSTRM utilise un codage FEC des informations. Cette technique référencée sous le nom d'*Hybrid ARQ type 2* (HARQ2) permet d'améliorer grandement le passage à l'échelle des protocoles de transport multipoint [96]. Le codage adopté est un codage en bloc systématique basé sur un code MDS (Maximum Distance Separable code) à effacement [7]. Notons que ce type de code est optimal en termes de volume de données requis pour décoder des informations reçues. Ils permettent donc de tirer partie au maximum des ressources réseau.

4.3.3 Mécanisme de limitation du trafic retour

Pour pouvoir retransmettre les données perdues, la source doit être informée de l'état de réception des utilisateurs finals. Pour cela, l'utilisation de rapports de réception permet d'ajuster le volume d'information retransmis en fonction de ce qui a été réellement

perdu par les récepteurs. Cela permet donc d'optimiser l'utilisation des ressources, ainsi que de garantir la fiabilité du service. Cependant ce type de technique introduit un risque de saturation du réseau, connu sous les termes de *Nack Implosion* (cf. 1.2.2) — correspondant au cas où un grand nombre de récepteurs envoie simultanément un message vers la source.

Pour prévenir ce risque, nous proposons d'utiliser dans HSTRM le mécanisme de *Nack Suppression* car il est apparu comme étant le plus efficace pour réduire le trafic retour et le plus adaptable par rapport à la taille du groupe. De plus, selon la section 3.3, ce mécanisme permet de mettre en oeuvre un mécanisme d'estimation de taille de groupe efficace utilisant les retours générés par le groupe. Avec ce mécanisme, la source demande régulièrement aux récepteurs d'envoyer des rapports indiquant les données qu'ils ont perdues — ces demandes sont appelées par la suite messages ***Inquiry***. Ainsi, à la réception d'un message ***Inquiry***, chaque récepteur génère aléatoirement un temps d'attente. Le récepteur avec le temps d'attente le plus faible envoie alors un message ***InquiryRep*** à la source. Lorsqu'elle reçoit ce message, la source répond par un message ***StopInquiry*** qu'elle transmet au groupe. Tous les récepteurs recevant cette réponse annulent la transmission d'un message ***InquiryRep***. Avec ce mécanisme, la source reçoit donc tous les messages envoyés entre le moment où le premier message ***InquiryRep*** à été envoyé et le moment où le message ***StopInquiry*** a été reçu par le groupe.

4.3.4 Estimation de la taille du groupe

Un autre des objectifs de la proposition est de choisir le médium approprié pour transmettre les informations en fonction du nombre de récepteurs. La source doit donc suivre l'évolution de la taille du groupe au cours de la transmission. Pour ce faire, elle collecte l'ensemble des messages ***InquiryRep***, puis estime le nombre de récepteurs présents au moyen d'un estimateur basé sur le principe du maximum de vraisemblance (cf. 3.3.3).

4.3.5 Mécanisme de reprise d'erreur par voie terrestre

Lors de la phase de reprise terrestre des informations manquantes, les récepteurs cherchent à contacter d'autres récepteurs pour leur demander les informations qui leur manquent. Nous avons choisi pour cela d'établir un réseau logique permettant de relier logiquement les récepteurs entre eux (ou réseau de pair-à-pair). La définition de ce mécanisme a de plus été conduite par un critère de coût, notamment pour l'établissement de ce réseau, tout en prenant en compte la taille très importante des groupes. Cela a entre autre conduit à une simplification de mécanismes existant pour l'établissement de ce réseau. Ce mécanisme est celui décrit dans la section 3.4 : plusieurs récepteurs choisis aléatoirement jouent le rôle de racine. Le reste des récepteurs se connectent ensuite à une de ces racines afin de construire un arbre logique v-aire — chaque noeud a donc v enfants et un père, à l'exception des racines qui n'ont pas de père.

Afin de trouver les informations, les récepteurs utilisent le réseau de pair-à-pair en envoyant un message ***DataSearch*** à leur père et à leurs enfants. Chaque pair n'ayant

pas les données va transférer le message à ses propres enfants et/ou son propre père (selon que le message provient de son père ou d'un de ses fils). Après avoir trouvé un — ou plusieurs — récepteur(s) ayant les données recherchées, l'initiateur de la recherche l'utilise comme source pour reprendre ses pertes. Une session Multicast se termine dès que cette phase est terminée, c'est-à-dire dès que tout le groupe a reçu le fichier.

4.3.6 Mécanisme de contrôle de congestion

A l'heure actuelle, le choix d'un mécanisme de contrôle de congestion n'a pas été arrêté. Cela est dû au fait qu'aucun mécanisme de contrôle de congestion n'a été proposé par le groupe de travail RMT comme standard associé à l'approche NORM. De plus l'utilisation d'un lien satellite change la problématique du contrôle de congestion par rapport à l'environnement Internet terrestre (cf. 2.2.3.4). Nous faisons l'hypothèse par la suite que le réseau d'accès avec lequel la source est connectée au réseau satellite est suffisamment bien dimensionné pour éviter la saturation de celui-ci (et que la source ne demande pas un débit en émission trop important). Il existe une possibilité permettant à la fois au système satellite d'être pleinement utilisé sans saturer le réseau terrestre. Cette solution consiste à ce que la source transmette d'abord les données à un serveur de diffusion satellite. Cette transmission est une transmission terrestre classique. Le serveur de diffusion se charge ensuite de transmettre les données aux récepteurs au moyen du système satellite. Au niveau transport, cette approche est similaire à une cassure de connexion telle celle présentée dans [5] pour améliorer les performances de TCP dans les réseaux satellites. Pour finir, la même hypothèse est faite pour les récepteurs connectés au réseau satellite au moyen d'un réseau d'accès (i.e. pas directement avec un VSAT).

4.4 DÉROULEMENT D'UNE SESSION MULTICAST AVEC HSTM

Cette section décrit le déroulement d'une transmission avec la proposition. Le déroulement Global d'une session est exposé dans la figure 4.1. Les sections suivantes exposent en particulier la manière dont interagissent l'ensemble des mécanismes retenus, et les messages échangés au cours des différentes phases de la transmission.

4.4.1 Annonce préalable de session

La session est initiée par une phase d'annonce de session de transmission multipoint. Il est nécessaire de transmettre plusieurs informations lors de l'annonce de la session. Une partie de ces informations est essentielle au fonctionnement de la proposition. Ces informations peuvent être transmises au moyen d'un outil comme SDP (Session Description Protocol [53]), ou hors bande (envoi de courriers, visite de page web, etc.).

Il est tout d'abord nécessaire de transmettre la date de début de la session. Ensuite les informations suivantes sont nécessaires pour pouvoir utiliser le service de transmission multipoint :

- L'adresse de groupe IP multicast utilisée
- Le numéro de port correspondant à la session

Il peut de plus être utile de transmettre l'adresse IP de la source. Dans ce cas la source pourra servir si besoin est de point d'entrée du réseau de pair-à-pair. Cependant l'envoi de l'adresse de la source peut poser des problèmes de sécurité. Pour cette raison, l'entité gérant la source est libre de révéler son ou non son adresse lors de la phase d'annonce de la session. Il n'est toutefois possible de garantir que tout le groupe a reçu les données que si l'adresse du serveur est transmise. En effet dans le cas contraire, il reste envisageable (bien que ce soit peu probable) que des récepteurs ne reçoivent aucune des informations nécessaires à la phase de transmission terrestre[1]. Une solution pour éviter d'exposer la source à d'éventuelles attaques, est par exemple de mettre son adresse à disposition sous une forme cryptée. Les récepteurs doivent dans ce cas posséder la clef adéquate pour décrypter l'adresse.

Il faut ajouter à cela les informations relatives au codage des données transmises. Celles-ci sont identiques aux informations transmises pour chaque objet dans les messages ***MTPObjectConfig***. Il est possible de ne transmettre aucune de ces informations lors de l'annonce de la session, et de toutes les transmettre par la suite avec des messages ***MTPObjectConfig***. Cependant dans ce cas les récepteurs ne recevant pas ces messages ne pourront pas exploiter les paquets reçus. Il est donc préférable de préciser ces informations. De plus les informations transmises servent de configuration par défaut, ce qui évite de transmettre des messages ***MTPObjectConfig*** tant que les caractéristiques du codage ne changent pas.

Enfin la source peut indiquer des informations relatives aux objets envoyés. Ces informations peuvent par exemple comprendre, le nom de fichiers, la taille de fichiers, un CRC permettant de vérifier l'intégrité du fichier reçu, etc. Ces informations sont dépendantes de l'application (et indépendante de la couche transport utilisée). Pour cette raison, les informations précisées ne le sont qu'à titre d'exemple. Notons que le choix de ces paramètres est dépendant de l'application et doit donc être géré par celle-ci.

Utilisation du réseau	Adresse de classe C utilisée Numéro de port utilisé Adresse du serveur (optionnel)
Codage	FEC Encoding ID FEC Instance ID T_B (nombre de paquets par bloc)
Description du fichier	Nom Taille du fichier CRC

Tableau 4.1 – Informations transmises lors de l'annonce de session

L'ensemble des informations transmises est récapitulé dans le tableau 4.1. Dès qu'une session Multicast est annoncée, un récepteur peut, s'il le désire, s'y abonner. Si l'utilisateur décide de s'abonner à la session, cela se traduit par l'envoi d'un message (appelé

[1]La probabilité d'occurrence de ce cas est très faible puisque, dès qu'un récepteur a reçu un paquet émis par la source, il connaît son adresse (si on ne tient pas compte des éventuelles techniques de masquage d'adresses et/ou de pare-feu).

par la suite ***JoinGroup***) permettant d'informer la couche réseau qu'un récepteur désire s'abonner à l'adresse de groupe utilisée par la session Multicast.

4.4.2 Phase de transmission des données initiales par satellite

A la date prévue (définie lors de l'annonce de la session), la transmission commence. C'est-à-dire que l'application transmet à la couche transport des objets à envoyer. La source envoie les objets un à un. Ce sont d'abord des messages ***MTPObjectConfig*** qui sont envoyés. Ces messages permettent de spécifier la taille de l'objet, et au besoin des informations relatives au codage des données. Si aucune information n'est transmise à ce sujet, c'est la configuration par défaut qui sera utilisée. On peut distinguer trois tâches qui se déroulent en parallèle lors de cette phase : l'ordonnancement et l'envoi des données, l'estimation de la taille du groupe, et la mise en place du réseau de pair-à-pair.

4.4.2.1 Ordonnancement des données à envoyer

L'objet à transmettre est d'abord ¡¡ découpé ¿¿ en paquets. Les paquets sont ensuite regroupés par blocs de T_B. Les blocs sont à leur tour regroupés par ensemble de T_E. Ces groupes de T_E blocs sont appelés *Méta-blocs*. La transmission de données commence alors. La transmission est effectuée par méta-bloc, de manière à répartir les pertes survenant en séquences sur plusieurs blocs. Concrètement, le premier paquet du premier bloc est envoyé, suivi du premier paquet du deuxième bloc, et ainsi de suite jusqu'au bloc T_E (voir figure 4.2). Le fichier est d'abord transmis intégralement, sans

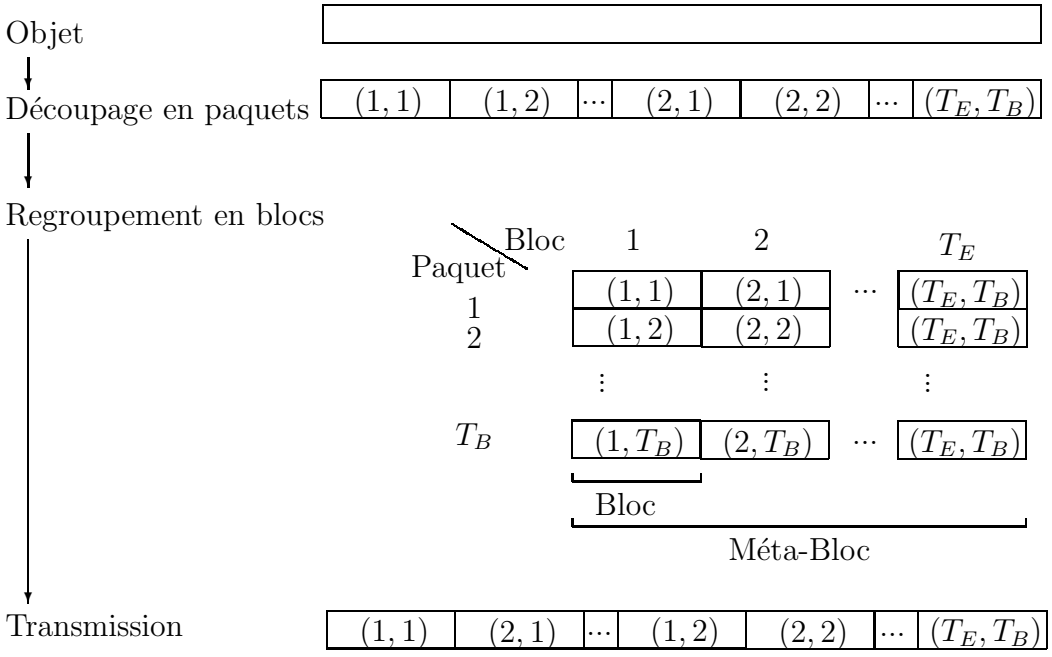


Figure 4.2 – Préparation de la transmission

codage. Ainsi les récepteurs ne subissant pas de pertes ne sont pas défavorisés par ceux qui subissent des pertes. Les premiers doivent certes garder les informations en

mémoire pour pouvoir les transmettre par voie terrestre, si besoin est, mais l'objet de la transmission est disponible à l'utilisateur plus rapidement.

4.4.2.2 Estimation de la taille du groupe

Pendant cette première phase de transmission, la source estime la taille de groupe. Notons que cette phase de transmission correspond à la transmission minimale. Ainsi, au cours de cette phase, aucun récepteur ne peut quitter la session car aucun récepteur ne peut avoir reçu la totalité des données. Le nombre de récepteurs couramment intéressés ne varie donc pas au cours de cette phase, ce qui permet d'estimer précisément le nombre de récepteurs abonnés à la session. Pour cela la source envoie des messages ***Inquiry***. Ces messages sont envoyés régulièrement au cours de la transmission avec une périodicité de **SatInqPeriod**. A la réception de ces messages, chaque récepteur génère aléatoirement un temps d'attente (selon une fonction de répartition connue). Lorsque la source reçoit la première réponse, elle transmet au groupe un message ***StopInquiry***. La source utilise ensuite le nombre de réponses obtenues, ainsi que les temps d'attente qui ont été générés pour estimer la taille du groupe. Les estimations réalisées pendant cette phase ont trois objectifs. Premièrement, elles permettent de configurer le mécanisme pour qu'il s'adapte à la taille réelle du groupe (le mécanisme est initialement configuré pour des groupes très importants de manière à éviter une saturation du réseau). Ensuite, en étudiant les variations de la taille du groupe — qui correspond en fait à l'estimation du nombre de récepteurs ne subissant pas de pertes à un instant donné — la source peut évaluer grossièrement la qualité de réception des récepteurs.

4.4.2.3 Préparation du réseau de pair-à-pair

Après l'envoi de paquets ***Inquiry***, la source collecte les adresses des récepteurs qui lui répondent. Ces récepteurs sont choisis pour jouer le rôle de racines du réseau de pair-à-pair. Pour que le groupe ait connaissance des racines du réseau, la source envoie la liste de leurs adresses dans des messages ***P2PRootList***. La source envoie avec cette liste un temps de connexion **TMaxP2P** qui correspond au temps dont les récepteurs disposent pour se connecter. Ce temps est choisi en fonction de la taille estimée du groupe, et en fonction du nombre moyen de messages par seconde autorisé lors de la constitution du réseau. Ces messages sont régulièrement envoyés au cours de la transmission — toutes les **P2PRLPeriod** unités de temps — pour permettre à des récepteurs subissant des pertes de les recevoir. Une fois le premier message ***P2PRootList*** envoyé, la construction du réseau de pair-à-pair est réalisée comme suit. A la réception de ce message, les récepteurs tirent aléatoirement un temps d'attente selon une fonction de répartition préalablement choisie. Ce temps d'attente est compris entre 0 et **TMaxP2P**. A l'expiration du timer, les récepteurs choisissent aléatoirement une racine parmi celles connues, et lui envoient un message ***ConnectP2P*** en précisant leur adresse IP. Si la racine choisie ne répond pas, il choisit une autre racine. La racine première racine joignable répondra par un message ***PeerConnected***, si elle a moins de v voisins. Dans ce cas le récepteur devient un voisin direct de la racine. Dans le cas contraire, la racine renvoie un message ***Redirected*** qui précise un pair dépendant de cette racine auquel le nouvel arrivant doit se faire connaître. Ce dernier

envoie alors un message ***ConnectP2P*** au pair spécifié. Celui-ci répond par un message ***PeerConnected***. Le pair que le nouvel arrivant cherche à atteindre peut cependant ne pas être joignable. Si tel est le cas, le nouveau pair recontactera sa racine qui lui retransmettra l'adresse d'un autre pair.

Dans certains cas extrêmes, des récepteurs peuvent ne recevoir aucun des messages ***P2PRootList***. Ces récepteurs sont alors dans l'incapacité de se connecter au réseau de pair-à-pair. La seule possibilité pour ces récepteurs est de contacter la source, dans l'éventualité où son adresse a été transmise lors de l'annonce de la session. Pour cela chaque récepteur génère aléatoirement un temps d'attente compris entre 0 et **DefaultTMaxP2P** dès que la transmission satellite se termine. La source leur envoie alors un message ***P2PRootList*** par voie terrestre. La valeur de **DefaultTMaxP2P** correspond au temps d'attente par défaut et est choisie lors de la configuration du protocole. Notons qu'il n'y a pas de risque de *Nack Implosion*. En effet les récepteurs qui en arrivent à contacter la source vont le faire au bout d'un temps aléatoire. Les messages ***ConnectP2P*** ne sont donc pas synchronisés. Il reste cependant considérer les cas où la source n'est pas joignable. Cela se produit par exemple lorsque son adresse n'a pas été communiquée, ou encore lorsque le service réseau n'est pas capable de la localiser (par exemple à cause d'une partition du réseau). Dans ces cas, le récepteur va rester en attente de réponse pendant un temps supérieur au temps maximum d'attente : **TimeoutSession**. Il va alors signifier à l'application que la réception a échouée, et arrêter la transmission.

La transmission des paquets initiaux se termine par l'envoi du dernier bloc dans lequel la valeur du champ **LastBloc** est égale à un. Cette phase est suivie par la transmission de données encodées.

4.4.3 Phase de transmission de données encodées par satellite

Lorsque cette phase commence, dès qu'un récepteur a reçu complètement les données, il devient muet pour la source : il ne répond plus aux messages ***Inquiry***. Ainsi, quand la première phase de transmission se termine, la source est en mesure de savoir s'il reste assez de récepteurs pour que la transmission satellite soit encore intéressante. Si tel est le cas, elle va commencer à transmettre des données encodées afin de compléter les données reçues par tous les récepteurs. Dès que cela est en leur pouvoir, les récepteurs commencent à décoder les informations, et les transmettent à l'application dès que le décodage a été effectué. Les récepteurs deviennent ainsi muet dès que toutes les informations ont été décodées, et la source estime à chaque demande le nombre de récepteurs n'ayant toujours pas reçu les données (et ayant reçu le message ***Inquiry*** correspondant). La source continue ainsi à utiliser la liaison satellite jusqu'à ce que le nombre estimé de récepteurs devienne inférieur au seuil **MinRcv** prédéfini. Ce seuil est défini par l'utilisateur (voir par exemple la section 2.3).

Lorsque la taille du groupe devient inférieure à la taille minimale, l'utilisation de la liaison satellite n'est plus ¡¡ rentable ¿¿ (au vu des critères pris en compte). La source envoie alors un message ***CloseSatDataTx*** afin de signifier aux récepteurs la fin de la transmission satellite. Il est possible que des récepteurs ne reçoivent pas ce message, dans ce cas ils vont rester en attente de réception satellite. Cependant cette

période d'attente va dépasser le temps maximum d'attente **TimeoutSatTx**. Ce délai dépassé, le récepteur va supposer que la transmission satellite est terminée. Avec ces paramètres, tous les récepteurs sont connectés au réseau de pair-à-pair au plus tard au bout de **TimeoutSatTx** + **DefaultTMaxP2P** unités de temps après la fin de la transmission satellite.

4.4.4 Phase de reprise terrestre des informations

4.4.4.1 Recherche d'informations dans le réseau de pair-à-pair

Lorsque la phase de transmission terrestre commence, les récepteurs n'ayant pas reçu toutes les informations envoient des messages ***DataSearch*** sur le réseau de pair-à-pair. Ces messages sont envoyés à tous les enfants (quand le récepteur en a) et au père (quand le récepteur en a un). Cette demande est propagée sur le réseau jusqu'à ce qu'elle rencontre une réponse positive, ou jusqu'à ce qu'il ne soit plus possible de la transmettre (par exemple lorsqu'elle arrive aux feuilles de l'arbre). Dans le cas où, après avoir parcouru tout l'arbre de pairs, il n'y a pas de réponse positive, le récepteur cherchant les données va envoyer un message ***DataSearch*** à toutes les autres racines (i.e. toutes celles dont il ne dépend pas). Le récepteur détecte ce cas de figure lorsqu'il n'a reçu aucune réponse au bout d'un temps **TMaxDataSearch** prédéfini. Compte tenu du nombre de récepteurs en dessous duquel la transmission satellite s'arrête et de la taille initiale du groupe, il n'est pas possible que la requête parcoure tout le groupe sans trouver de réponse. Il est de plus fortement probable que les données soient trouvées dans le sous-réseau auquel appartient l'initiateur de la demande. Elles sont dans ce cas rapidement récupérées.

Une fois une (ou plusieurs) réponse(s) trouvée(s), les récepteurs cherchant un complément d'information envoient un message ***DataReq***. Celui-ci contient la liste des informations déjà en leur possession. Chaque récepteur contacté envoie en réponse toutes les informations mémorisées qui ne font pas partie de cette liste. Après avoir récupéré toutes les informations nécessaires, l'initiateur de la recherche peut reconstruire les données initiales. La phase de transmission terrestre est terminée lorsque tous les récepteurs ont reçu les données initiales.

4.4.4.2 Clôture de session

Au cours de la phase de transmission terrestre, la source continue à envoyer, par satellite, des messages ***Inquiry*** afin de connaître la taille du groupe. Plus exactement, au cours de cette phase, tous les récepteurs actifs dans le réseau de pair-à-pair peuvent participer aux réponses (le mécanisme est le même que pour la phase satellite). Cette précaution permet d'éviter que la source ne reçoive aucune réponse lorsque tous les récepteurs cherchant les données sont situés sous une perturbation et ne reçoivent pas les transmissions satellites. Ainsi la source peut clore la session quand aucune réponse n'est reçue. En contrepartie la source estime le nombre de récepteurs participant aux transmissions terrestres et non le nombre de récepteurs à qui il manque des données. Ces messages sont envoyés avec une périodicité de **TerInqPeriod**. Dans la mesure où la source utilise ces messages que dans un but de contrôle passif, cette valeur peut être

supérieure à la périodicité utilisée lors de la transmission terrestre. Lorsque la source ne reçoit plus de réponse, elle envoie un message ***CloseMTSessionReq*** au groupe. Elle attend alors **ESTimeout** unités de temps pour laisser l'occasion à tout récepteur échangeant des données de se faire connaître. Si aucun récepteur ne contacte la source, elle envoie un message ***CloseMTSessionConf*** et la session se termine. Si un récepteur lui répond, elle envoie un message ***ResumeMTSession*** et la phase terrestre continue. Lorsque les membres du groupe reçoivent le message ***CloseMTSessionConf***, ils effacent les données encodées encore en leur possession (les données décodées ont déjà été transmises à l'application).

Alors que cette section a exposé le déroulement d'une transmission avec HSTRM, la section suivante définit les messages apparaissant dans cette transmission : les messages émis par la source d'une part, et les messages émis par les récepteurs d'autre part.

4.5 DESCRIPTION DES MESSAGES DE HSTRM

Ce paragraphe détaille les messages échangés par les différentes instances du protocole HSTRM. Cette description explicite les différents types de messages, et spécifie l'en-tête de chacun d'entre eux.

4.5.1 Partie d'en-tête commun à tous les messages

La partie de l'en-tête décrite ici est commune à tous les messages émis par les entités HSTRM. Plusieurs champs peuvent s'ajouter à cette partie commune selon le type de message transmis. Cette partie est représentée schématiquement dans la figure 4.3. Les champs de la partie commune sont les suivants :

Version : ce champ permet d'identifier la version du protocole utilisée. Ce champ est codé par un demi octet.

Session ID : ce champ permet d'identifier la session Multicast à laquelle l'échange de données est associé. Ce champ est représenté par 4 octets.

Packet size : ce champ indique la taille du paquet HSTRM qui est transmis. La taille est indiquée en octets et inclut l'en-tête. Ce champ est représenté par 2 octets, la taille maximale gérée par HSTRM d'un paquet de niveau transport est donc de 65 535 octets.

Type : permet d'identifier le type de paquets HSTRM envoyés. Les types de messages sont définis dans le tableau 4.2. Les valeurs définies incluent les messages envoyés par la source et par les récepteurs.

Options : ce champ codé sur un octet permet de spécifier les éventuelles options utilisées. Les options définies à l'heure actuelle sont référencées dans le tableau 4.3 et décrites dans le paragraphe qui suit.

IntegrityCheck (optionnel) : ce champ est utilisé pour vérifier l'intégrité du paquet reçu. La valeur inscrite dans ce champ est le résultat d'une opération arithmétique sur l'intégralité du paquet. Le résultat de cette opération est inséré à la fin du paquet de niveau transport. L'utilisation de ce champ est laissée en option.

Nom du message	Valeur du champ Type
Source Control	1
Source Data	2
Source Control & Data	3
Receiver Control	4
Receiver Data	5

Tableau 4.2 – Valeurs du champ **Type** des messages HSTRM

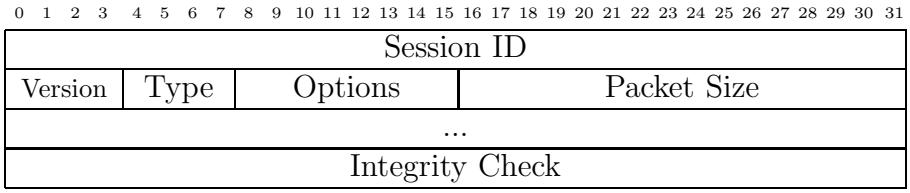


Figure 4.3 – Partie d'en-tête commun à tous les messages HSTRM

Options de HSTRM

Les différentes options définies pour HSTRM sont décrites ci-après. Lorsque le champ **Options** est laissé à 0, aucune option n'est utilisée. Les valeurs du champ **Options** sont présentées dans le tableau 4.3.

NoIntegrityCheck : si cette option est utilisée, aucun contrôle n'est effectué sur l'intégrité des paquets. L'utilisation de cette option permet de réduire le volume de données de contrôle envoyé. L'utilisation de cette option peut être intéressante dans des réseaux où le taux d'erreur est négligeable. Cette option peut être choisie également lorsque l'utilisateur considère que les couches inférieures au niveau transport effacent tous les paquets corrompus avant qu'ils n'atteignent le niveau transport.

EstimateBLoc : ce champ permet de préciser à quelle unité de données les estimations réalisées se rapportent. Lorsque le bit correspondant à cette option est mis à 1, la source estime le nombre de récepteurs n'ayant pas reçu un bloc particulier. Dans le cas contraire, qui est le cas pas défaut, la source estime le nombre de récepteurs n'ayant pas reçu chaque méta-bloc particulier.

Nom de l'option	Valeur du champ Options
NoIntegrityCheck	– – – – – – – –1
EstimateBLoc	– – – – – – – 1–

Tableau 4.3 – Valeurs du champ **Options**

Toutes les autres valeurs du champ option sont laissées libres de manière à pouvoir étendre le protocole si besoin est.

4.5.2 Messages émis par la source HSTRM

Cette section définit les messages émis spécifiquement par la source. Comme cela a été dit dans la section précédente, il y a deux types de messages : les messages de contrôle et les messages transmettant des données.

4.5.2.1 Partie d'en-tête spécifique aux messages de données

Il est nécessaire dans ce cas de rajouter à la partie commune tous les champs permettant d'identifier le paquet de données au sein du flux de données encodées. Les notations sont les mêmes qu'au paragraphe 4.3 :

- Les codes MDS demandent de regrouper les paquets par T_B pour générer des paquets encodés. Le fichier étant découpé en paquets, on appelle *bloc* un ensemble de T_B paquets à partir desquels seront générés les paquets encodés.
- A partir de ces T_B paquets, il n'est possible de générer au maximum que T_{max} paquets, soit T_B paquets non encodés, et $T_{max} - T_B$ paquets encodés.
- Selon la valeur de T_B et de T_{max}, il peut être nécessaire d'entrelacer plusieurs blocs afin de répartir les pertes sur plusieurs blocs. Dans ce cas, on appelle *Méta-Bloc* l'ensemble des blocs entrelacés.

Il faut donc ajouter les champs suivants :

Object ID : ce champ permet d'identifier l'objet qui est envoyé. Ce champ, codé sur 12 bits permet d'envoyer plusieurs objets au sein d'une même session Multicast. Au maximum 4 095 objets peuvent donc être envoyés en étant différenciés.

Source Block Number : ce champ permet d'identifier l'ensemble de T_B paquets initiaux à partir desquels les paquets ont été générés. Ce champ est codé sur 20 bits. La taille maximum en octets d'un objet transmis avec HSTRM est donc de : $2^{20} \times T_B \times TDU$, où TDU (*Transport Data Unit*) représente la taille occupée par les données, en octets, dans les paquets transport.

Encoding Symbol ID : ce champ codé sur 20 bits identifie le paquet à l'intérieur du bloc. Le nombre maximum de paquets qui peuvent être numérotés à l'intérieur d'un bloc est donc de 2^{20}. Selon l'étude menée au paragraphe 3.2, cette valeur est suffisante pour être utilisée avec les codes en blocs existants à l'heure actuelle — et même à venir. Si la valeur est inférieure à T_B, le paquet est un paquet non encodé, tandis que si la valeur est supérieure à T_B, le paquet est un paquet encodé.

Last Bloc : ce champ permet d'identifier le dernier bloc relatif à un objet. Ce champ est représenté par un bit. Lorsque ce bit est mis à 1, le bloc transmis est le dernier. Les données peuvent donc être transmises à l'application dans ce cas — si le récepteur a reçu toutes les informations nécessaires.

Offset : ce champ indique le nombre d'octets de données (sans prendre en compte les éventuels bits de bourrage) compris entre la fin de ce champ et le début des données. Les données commencent dans ce cas au prochain octet qui soit un multiple de 4 (voir figure 4.4). En d'autres termes, ce champ indique le numéro

de ligne à partir de laquelle les données commencent. Les données optionnelles d'en-tête correspondent aux messages de type 3 où des informations de contrôle sont envoyées conjointement aux données.

Notons que l'en-tête des paquets HSTRM ainsi défini n'est pas conforme aux propositions de la RFC 3452. Il est en effet spécifié que la totalité des informations comprises dans le champ *FEC Payload ID* doit être un multiple de 4 octets. Cependant, avec les autres champs de l'en-tête, cela aurait ajouté un overhead inutile : d'une part les champs tels qu'ils sont définis actuellement permettent d'encoder un volume de données très important. D'autre part il aurait fallu augmenter la taille d'autres champs de manière à ce que l'en-tête complet soit un multiple de 4 octets.

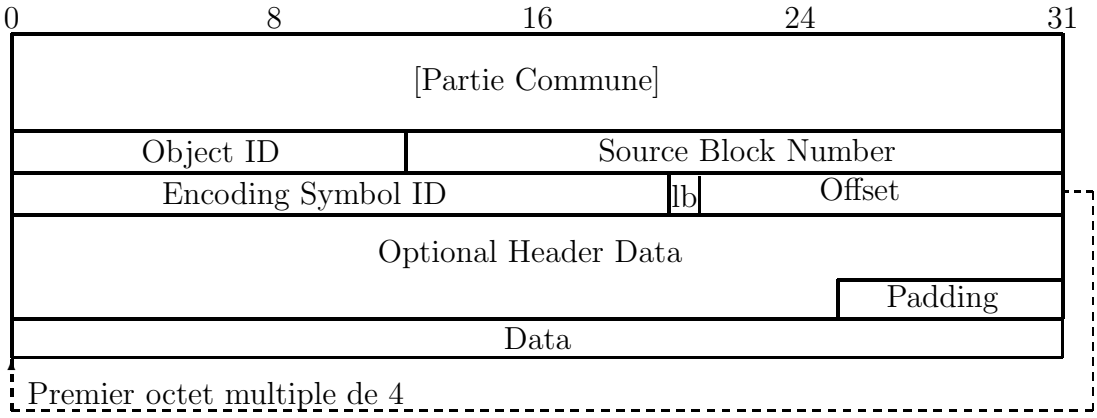


Figure 4.4 – En-tête des paquets de données

4.5.2.2 Messages de contrôle / signalisation

Il existe différents types de messages de contrôle envoyés par la source HSTRM. Ces différents messages sont référencés dans le tableau 4.4. On peut différencier deux grandes catégories de messages :

- les messages dont la valeur du champ **Message Type** permet de déterminer entièrement l'action à mener, et
- et les messages qui impliquent un échange de données pour mener l'action spécifiée.

les messages ne nécessitant pas l'ajout de données,

4.5.2.3 Descriptions des messages de contrôle sans données optionnelles

Plusieurs messages ne nécessitent pas l'envoi de données supplémentaires. C'est le cas des messages : ***OpenMTSession***, ***MTSessionAborted***, ***CloseSatDataTx***, ***ResumeMTSession***, ***CloseMTSessionReq***, et ***CloseMTSessionConf***. Cela signifie que le seul code du message permet au protocole de connaître l'action à exécuter. Dans ce cas l'en-tête des messages est constitué de deux champs :

Message Type : ce champ codé sur 15 bits permet d'identifier le type de message de contrôle envoyé.

Nom du message	Type (hexadécimal)
OpenMTSession	00 01
CloseMTSessionReq	00 02
CloseMTSessionConf	00 03
ResumeMTSession	00 04
MTSessionAborted	00 05
Inquiry	00 06 – 00 07
StopInquiry	00 08
MTPObjectConfig	00 09
CloseSatDataTx	00 0A
P2PRootList	00 0B

Tableau 4.4 – Messages de contrôle émis par la source HSTRM

M : ce bit (More) permet de savoir si une autre action de contrôle est présente dans le message de contrôle. La valeur '0' indique que l'action identifiée par le champ **Message Type** est la dernière, et la valeur '1' indique qu'une autre action suit celle qui est lue.

Les messages ne nécessitant pas un échange de données sont décrits ci-dessous :

OpenMTSession
Ce message est envoyé au début de la session Multicast pour informer les récepteurs que la transmission va commencer.

CloseMTSessionReq
Ce message est envoyé à la fin de la session Multicast pour informer les récepteurs que la session est sur le point de se terminer.

ResumeMTSession
Ce message est envoyé par la source après un message CloseMTSessionReq pour indiquer que la session continue (cela correspond au cas où il reste des récepteurs n'ayant pas reçu les données).

CloseMTSessionConf
Ce message signifie que la session est terminée.

MTSessionAborted
Ce message signifie que la session est annulée.

CloseSatDataTx
Ce message est envoyé pour informer les récepteurs que la transmission satellite est terminée, et qu'ils doivent compléter les données qu'ils ont reçues au moyen du réseau terrestre

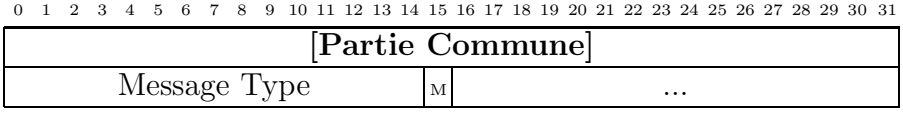


Figure 4.5 – En-tête des paquets de contrôle sans données additionnelles

4.5.2.4 Descriptions des messages de contrôle avec données supplémentaires

Messages *Inquiry*

Ces messages sont envoyés lorsque la source demande aux récepteurs d'envoyer des rapports de réception. A la réception de ce message, tous les récepteurs génèrent aléatoirement un temps d'attente (ce mécanisme est étudié plus en détail dans le paragraphe 3.3.2). La fonction de répartition de ces temps d'attente $f(x)$ est définie par deux paramètres : **TMaxInquiry** et λ, où **TMaxInquiry** représente le temps maximum de réponse. Pour ces messages, des données supplémentaires sont envoyées. Les deux types qui sont définis correspondent à deux options possibles. Le type 7 correspond à l'envoi d'un message sans mise à jour de la fonction de répartition, et le type 8 correspond à l'envoi d'un message avec mise à jour de cette fonction.

Message du type 6 : il faut ajouter le numéro de séquence de la demande. Celui-ci est représenté par 2 Octets. Ce champ permet de différencier plusieurs messages du même type lorsque plus d'un message est reçu. Cette distinction est nécessaire pour que le mécanisme d'estimation de taille de groupe fonctionne convenablement. En plus d'un numéro de séquence, le message contient l'identificateur de l'objet auquel se rapporte l'estimation, l'identificateur d'un bloc, ainsi qu'un champ **T** permettant d'indiquer si la transmission satellite est terminée. La source met ce bit à 1 lorsque la phase de transmission satellite est terminée, et à 0 sinon. Lorsque l'option EstimateBLoc est utilisée, ce champ permet de préciser le numéro de bloc auquel se rapporte le message. Dans le cas contraire, le message se rapporte au méta-bloc auquel appartient le bloc indiqué. L'en-tête de ces messages est représenté dans la figure4.6.

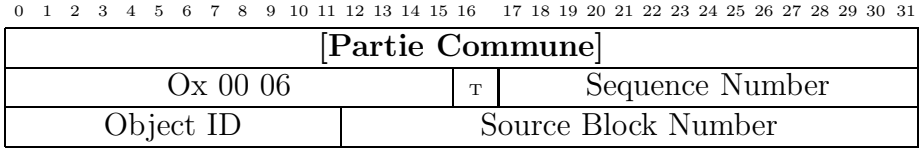


Figure 4.6 – En-tête des messages *Inquiry* sans mise à jour de la fonction de répartition.

Messages du type 7 : il faut ajouter aux informations transmises dans les messages de type 6 les paramètres **TMaxInquiry** et λ. Chacun est représenté par un réel à double précision, soit 8 octets selon le standard ANSI/IEEE Std 754 – 1985. L'en-tête de ces messages est représenté dans la figure 4.7. Les messages de type 7 permettent ainsi de mettre à jour la fonction de répartition des périodes d'attente des récepteurs, selon ce qui a été proposé dans la section 3.3.

Messages *StopInquiry*

Ce message est envoyé par la source aux récepteurs pour leur demander de ne plus répondre à un message ***Inquiry***. Les données suivantes sont envoyées dans ce message :

T : ce champ indique si la phase de transmission terrestre a débuté.

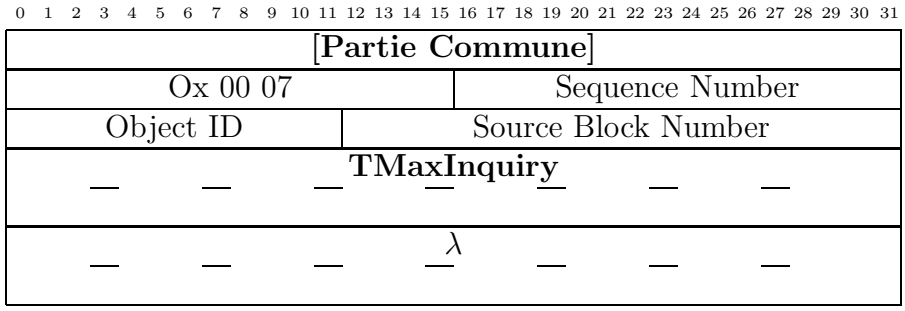


Figure 4.7 – En-tête des messages *Inquiry* sans mise à jour de la fonction de répartition.

Sequence Number : ce champ représente le numéro de séquence (codé sur deux octets) du message ***Inquiry*** auquel est associée cette demande d'annulation.

First Timeout : ce champ représente la valeur du temps d'attente le plus faible reçu par la source. L'envoi de cette valeur permet de résoudre les éventuels problèmes de desséquencements temporels dus à la traversée du réseau. Cette valeur est un réel à double précision, soit 8 octets.

Messages MTPObjectConfig

Ces messages sont envoyés de manière à mettre à jour les paramètres de la session Multicast (en cas d'erreur lors de l'annonce de celle-ci), ou à confirmer ces paramètres. Il faut noter que la réception de ce message implique que l'adresse de classe C utilisée pour la session Multicast soit correcte, ainsi que le champ **Session ID**. Ces messages sont relatifs à un objet particulier et comprennent les informations détaillées ci-dessous.

- **Object Encoding ID** : ce champ permet d'associer un identificateur qui fera référence à cet objet lors de la t transmission.
- **FEC Encoding ID** : ce champ est proposé par la RFC expérimentale 3452 [79]. Il permet d'identifier le type de codage utilisé. C'est un index numérique dont les valeurs sont comprises entre 0 et 255 (valeurs inclues). L'identification de l'encodeur en fonction des valeurs est régie par l'organisme IANA (Internet Assigned Numbers Authority).
- **FEC Instance ID** : ce champ est également proposé par la RFC expérimentale 3452 . Pour des valeurs du champ FEC Encoding ID comprises entre 0 et 127, il n'est pas nécessaire de transmettre le champ FEC Instance ID il est donc laissé à zéro.
- T_B : Le nombre de paquets par blocs. Ce nombre est codé sur 20 bits, (comme le champ Encoding Symbol ID des messages de données). Il permet de savoir combien de paquets initiaux sont encodés dans un bloc.
- **Last** T_B : ce champ permet de savoir avec combien de paquets le dernier bloc sera encodé. Selon la taille du paquet, il peut en effet être nécessaire de changer cette variable (car il ne reste plus assez de données pour former T_B paquets).
- **Size** : ce champ représente la taille de l'objet transmis. Ce champ est codé sur 7 octets de manière à rester cohérent avec les valeurs prises pour le

nombre maximum de blocs et le nombre de paquets par blocs. Notons cependant que la taille maximum ainsi obtenue devrait en pratique être bien supérieure à la taille des objets transmis. Ce champ est nécessaire pour pouvoir obtenir l'objet initialement envoyé : dans la mesure où des codes en blocs sont utilisés, il est parfois nécessaire de rajouter des bits de bourrage pour former le dernier paquet transmis. Ce champ permet donc aux récepteurs d'éliminer ce rajout une fois les données reçues.

P2PRootList : ce message est envoyé par la source lorsqu'elle désire communiquer au groupe les adresses des récepteurs qui ont été choisis comme racines du réseau de pair-à-pair. Ce message est constitué du nombre x d'adresses envoyées, codé sur 2 octets, et de la liste des adresses de ces x racines. Pour finir, suite à la liste d'adresses, la source envoie le temps **TMaxP2P** dont disposent les récepteurs pour se connecter. Ce temps est codé par un réel à double précision soit 8 octets.

4.5.3 Messages émis par les récepteurs HSTRM

Cette section spécifie les messages émis par les récepteurs HSTRM.

4.5.3.1 Messages de contrôle émis par les récepteurs

Il existe différents types de messages de contrôle envoyés par les récepteurs HSTRM. Dans la mesure où tous ces messages sont nécessairement émis séparément les uns des autres (car ils sont destinés à des instances HSTRM différentes), l'en-tête de ces paquets est légèrement différent de celui des paquets de contrôle de la source. Cet en-tête est détaillé dans la figure 4.8. Ces différents messages sont référencés dans le tableau 4.5 et sont décrits ci-dessous.

Nom du message	Type (hexadécimal)
InquiryRep	00 01
ConnectP2P	00 02
PeerConnected	00 03
Redirected	00 04
DataSearch	00 05 – 00 06
DataFound	00 07 – 00 08
OpentTerConn	00 09
CloseTerConn	00 10
DataReq	00 11
ResumeReq	00 12

Tableau 4.5 – Messages de contrôle émis par les récepteurs HSTRM

InquiryRep : ce message est envoyé en réponse à un message ***Inquiry*** émis par la source. Afin d'éviter que les réponses à un message ***Inquiry*** interfèrent avec des réponses émises suite à un autre message ***Inquiry***, il est nécessaire de préciser le numéro de séquence du message qui a entraîné cette réponse. Ce numéro de séquence est le même que celui défini dans la section 4.5.2.3 pour les messages

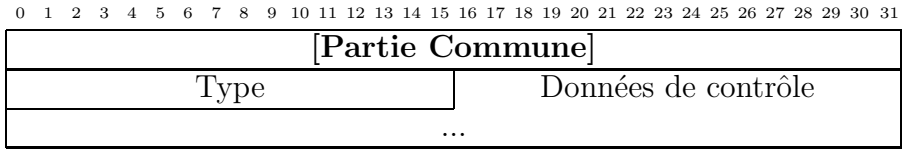


Figure 4.8 – En-tête des paquets de contrôle des récepteurs

émis par la source et est codé sur 2 octets. Afin que la source puisse correctement estimer la taille du groupe, il est nécessaire de transmettre le temps généré par le récepteur. Ce temps est codé sur un réel à double précision, soit 8 octets.

ConnectP2P : ce message est émis lorsqu'un récepteur tente de se ¡¡ connecter[2] ¿¿ au réseau de pair-à-pair. Ce message est destiné à une des racines de celui-ci.

PeerConnected : ce message est émis en réponse à une demande de connexion au réseau de pair-à-pair, pour indiquer au nouvel arrivant que sa demande a été acceptée. Avec ce message, la liste de tous les noeuds de l'arbre logique v-aire compris entre la racine et le nouveau pair est envoyée (voir la section 3.4). Le message ainsi obtenu est représenté dans la figure 4.9.

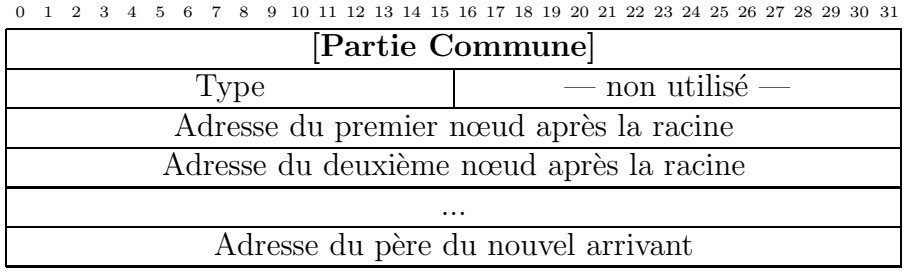


Figure 4.9 – Message *PeerConnected*

Redirected : ce message est envoyé par une racine en réponse à un message de type ***ConnectP2P***. Il est envoyé lorsque la racine contactée a déjà v voisins. Pour que le récepteur puisse se connecter, elle envoie l'adresse d'un autre pair qui ne possède pas encore v voisins. Comme dans les messages ***PeerConnected***, les 2 octets suivant le champ **Type** sont inutilisés, et l'adresse est placée sur les 4 octets suivant (soit sur la ligne de 4 octets suivante).

DataSearch : ce type de message est envoyé, au cours de la phase terrestre, lorsqu'un pair cherche des données au sein du réseau de pair-à-pair. Comme ces messages sont propagés de noeud en noeud, il est nécessaire d'inclure l'adresse du pair qui a initié la demande. Les réponses pourront ainsi atteindre celui-ci. L'identité des données recherchées doit bien sûr également être précisée. Il existe deux types de message ***DataSearch*** selon l'unité de données recherchées.

Avec les messages du **type 00 05** les récepteurs recherchent des Méta-blocs. Ceci signifie qu'ils n'ont reçu correctement aucun des blocs entrelacés dans le Méta-Bloc. Dans ce cas, le récepteur transmet le nombre de recherches effectuées, codé

[2] Il ne s'agit pas vraiment d'une connexion dans la mesure où le récepteur ne cherche qu'à obtenir des adresses d'autres pairs, et à se faire connaître : c'est une connexion logique et non une connexion au sens traditionnel du terme.

sur 2 octets, suivi d'une liste de requêtes. La liste de requêtes est constituée de doublets (**ObjectID**, **source Block Number**). Comme c'est un ensemble de blocs qui est recherché, le numéro de bloc peut correspondre à n'importe quel bloc appartenant à l'ensemble des blocs entrelacés.

Lorsque ce sont des messages du **type 00 06** qui sont transmis, ce sont des blocs qui sont recherchés. Les messages sont constitués de la même manière que pour le type précédant — si ce n'est que le numéro de blocs correspond au bloc recherché.

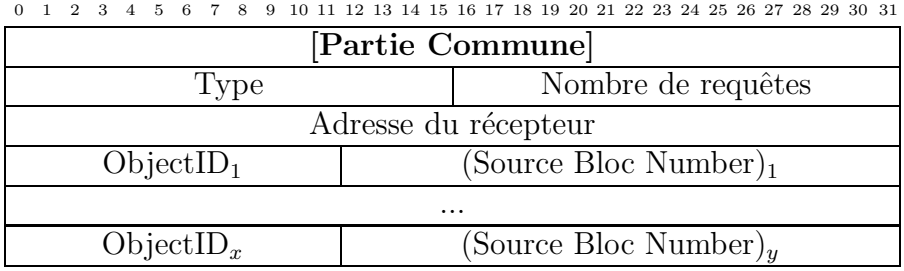


Figure 4.10 – Message *DataSearch*

DataFound : ces messages sont envoyés en réponse à un message ***DataSearch***. Comme il est possible que des récepteurs ne possèdent qu'une partie des informations recherchées, ils envoient inclue dans ce message la liste des informations qu'ils ont reçues. Cette liste est formée de la même manière que pour les messages ***DataSearch***. Le **type 00 07** correspond au cas où le récepteur possède tout le Méta-Bloc recherché, et le **type 00 08** correspond au cas où le récepteur ne possède qu'un bloc recherché.

OpenTerConn : ce message correspond à une demande d'ouverture de connexion terrestre. Ce message se traduit au niveau des deux pairs par l'utilisation du protocole de transport point-à-point terrestre implémenté dans la pile de communication — soit généralement par une requête d'ouverture de connexion TCP.

CloseTerConn : ce message correspond à une fermeture d'une connexion point-à-point terrestre. De même que pour le message précédant, cela se traduit par l'envoi d'un message associé au protocole utilisé par défaut pour les communications point-à-points terrestres.

DataReq : ce message correspond à une demande de transmission d'informations. Après avoir trouvé un récepteur qui a les informations recherchées, il reste à lui demander de transmettre toutes les informations manquantes. Ceci se traduit par l'envoi d'un message ***DataReq*** dans lequel est inclue la liste des paquets que le pair recherchant des données possède déjà. Cette liste est constituée de l'identité de l'objet, du numéro de bloc, puis de l'ensemble des paquets que le pair possède. Notons que comme les numéros de paquets sont codés sur 20 bits, un paquet de 1 500 octets permet d'envoyer plus de 500 numéros de paquets pour une requête. Ce nombre est supérieur au nombre de paquets par bloc utilisé actuellement avec des codes MDS, et est donc suffisant. Ce message est détaillé dans la figure 4.11.

ResumeReq : un message de ce type est envoyé à la source en réponse à un message ***CloseMTSessionReq*** lorsque le récepteur n'a pas reçu toutes les données. Ce

dernier indique par ce message à la source que tout le groupe n'a pas reçu les données, et que la session ne peut donc être close.

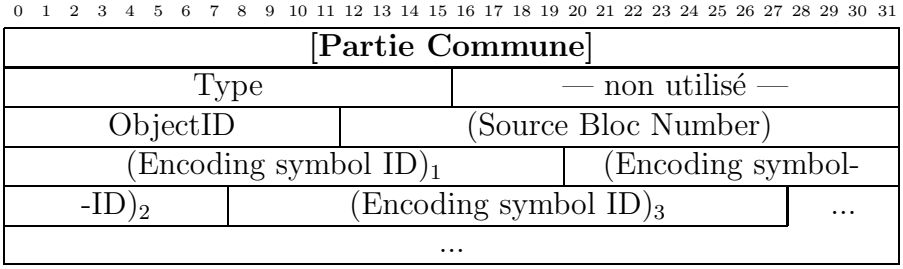


Figure 4.11 – Message *DataReq*

4.5.3.2 Messages de données émis par les récepteurs

Ces messages sont envoyés lors de la phase terrestre, après avoir recherché des informations. Une fois les informations localisées, il reste à transférer l'ensemble des paquets manquants. Ces messages sont identiques aux messages de données envoyés par la source, si ce n'est qu'il n'y a pas de partie optionnelle à l'en-tête. Ces messages sont décrits dans la figure 4.12.

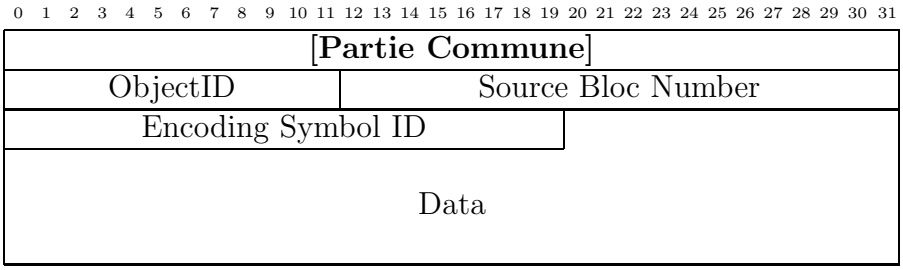


Figure 4.12 – Message de données émis par les récepteurs

Avec les hypothèses adoptées, ces paquets sont envoyés au moyen du protocole de transport point-à-point utilisé par défaut (par exemple TCP). On peut cependant remarquer que dans ce cas, l'en-tête des paquets est double : les paquets HSTRM sont insérés dans les paquets du protocole de transport point-à-point. Ainsi à l'en-tête du protocole de transport point-à-point s'ajoute l'en-tête de HSTRM.

4.6 VARIABLES DE HSTRM

Dans cette section la liste de toutes les variables utilisées dans HSTRM est faite. Chacune de ces variables sont décrites, ainsi que leur impact sur le comportement de HSTRM. Le tableau 4.6 donne l'ensemble des variables de la proposition.

MinRcv : cette grandeur représente le nombre minimum de récepteurs en dessous duquel la transmission satellite n'est pas avantageuse.

Nom de la variable	Valeurs des variables
MinRcv	300
TimeoutSession	180
TimeoutSatTx	3600
ESTimeout	5
SatInqPeriod	1
TerIndPeriod	3
P2PRLPeriod	1
TMaxP2P	180
DefaultTMaxP2P	180
TMaxDataSearch	180
TMaxInquiry	5
T_E	62

Tableau 4.6 – Variables et constantes de HSTRM

TimeoutSession : cette grandeur représente le temps maximum qu'un récepteur peut attendre sans recevoir de message d'aucune entité HSTRM. Passé ce délai, le récepteur considère que la session a échouée, et il transmet un message d'erreur à l'application.

TimeoutSatTx : cette variable représente le temps maximum que peut attendre un récepteur sans recevoir de message sur la liaison satellite. Après une telle période, le récepteur suppose que la transmission satellite est interrompue, et il entre dans la phase de transmission terrestre.

ESTimeout : cette variable correspond au temps que la source attend avant de confirmer une fin de session HSTRM.

SatInqPeriod : cette valeur correspond à la période à laquelle les messages de type ***Inquiry*** sont envoyés lors de la phase de transmission satellite.

TerIndPeriod : cette variable correspond à la période d'envoi des messages de type ***Inquiry*** lors de la phase de transmission satellite.

P2PRLPeriod : cette variable correspond à la période à laquelle sont envoyés les messages ***P2PRootList***.

TMaxP2P : ce temps correspond au temps maximum dont les récepteurs disposent pour se connecter au réseau de pair-à-pair.

DefaultTMaxP2P : ce temps est le temps maximum par défaut dont les récepteurs disposent pour se connecter au réseau de pair-à-pair. Ce temps est utilisé lorsqu'un récepteur désire se connecter au réseau de pair-à-pair et qu'il n'a pas reçu la variable **TMaxP2P**.

TMaxDataSearch : ce temps correspond au temps maximum pendant lequel un récepteur ayant envoyé un message ***DataSearch*** sur le réseau de pair-à-air attend une réponse. Passé ce délai, le récepteur considère qu'il n'y a pas les informations qu'il cherche dans l'arbre auquel il appartient. Il initie alors une recherche dans les autres arbres logiques du réseau de pair-à-pair.

TMaxInquiry : ce temps correspond au temps maximum dont disposent les récepteurs pour répondre à un message ***Inquiry***. La source est assurée de recevoir des réponses des récepteurs joignables — s'il y en a — avant que ce délai soit écoulé.

4.7 CONCLUSION

Ce paragraphe a présenté la proposition d'un protocole de transport (HSTRM pour *Hybrid Satellite / Terrestrial Reliable Multicast*) utilisant à la fois les réseaux terrestres et satellites pour transmettre des données de manière fiable vers de très nombreux récepteurs. Ce chapitre a intégré les différents choix techniques étudiés dans le chapitre 3 pour mettre en œuvre cette proposition. Ensuite, le déroulement d'une transmission avec HSTRM a été détaillé. Pour finir, les différents messages échangés par la source et les récepteurs ont été spécifiés.

Si, selon le chapitre 3 les performances de chacun des mécanismes semblent satisfaisantes, chacun d'eux a été étudié séparément. Il semble ainsi nécessaire d'étudier leur utilisation coordonnée. Le chapitre suivant propose donc une modélisation globale du protocole. L'objectif de cette modélisation est, a terme, de valider *a priori* une architecture de programmation permettant de mettre en œuvre l'approche considérée ici.

Chapitre 5

Validation fonctionnelle partielle de la proposition

5.1 OBJECTIF DE LA MODÉLISATION EN TERME DE VALIDATION

L'objectif de ce chapitre est de proposer une approche visant à valider la proposition de protocole de transport. L'objectif final de cette approche est d'obtenir une validation fonctionnelle de HSTRM, c'est-à-dire de vérifier que HSTRM rend bien le service attendu (défini au chapitre 4) : une communication multipoint fiable à grande échelle. Cependant, certains cas de figure ne permettent pas de rendre le service désiré. Par exemple, le cas où toutes les données sont perdues par tous les récepteurs dès le début de la communication aboutit nécessairement à un échec de la communication. Ainsi, un autre but de cette étude est, à terme, d'identifier les cas où la communication échoue, et pourquoi. Notre pour cela notre approche a été de valider *a priori* l'ensemble des mécanismes impliqués dans une communication utilisant HSTRM, puis la coopération entre ces mécanismes.

Pour arriver au résultat escompté, nous avons choisi d'utiliser le profil TURTLE au travers de la chaîne d'outils TTool-RTL. TTool est un outil permettant de réaliser la traduction d'une description UML particulière vers le langage formel RT-LOTOS. Nous avons choisis ce profil UML temps réel d'une part, parce qu'il capitalise dans un ¡¡ moule ¿¿ UML, un savoir-faire en matière de validation d'architecture de communication basé sur une modélisation formelle. D'autre part, son outillage permet d'aller plus loin que les outils UML du commerce, dès lors qu'il s'agit d'explorer l'espace d'état d'un système sous forme d'une analyse d'accessibilité, et d'en extraire des vues abstraites qui s'avèrent fort utiles pour caractériser le service rendu par une couche de protocole. Le profil TURTLE a de plus déjà été utilisé dans le domaine des satellites, pour modéliser

et vérifier la continuité de service lors d'une reconfiguration dynamique [3]. Toutefois, à notre connaissance, aucun travaux de ce genre n'a été mené pour valider un service de communication multipoint utilisant un réseau hybride satellite/terrestre.

Afin de comprendre le modèle réalisé, il est nécessaire de connaître le profil TURTLE, le langage RT-LOTOS, ainsi que les outils TTool et CADP. La section 5.2 présente donc brièvement ces derniers. Après avoir présenté la méthodologie du processus de validation dans la section 5.3, la section 5.4 décrit le modèle de HSTRM qui réalisé. La section 5.5 expose ensuite les résultats obtenus avec ce modèle, ce qui permet d'ouvrir une discussion sur le service rendu à l'application dans dernière la section.

5.2 PRÉSENTATION DU PROFIL TURTLE ET DES OUTILS ASSOCIÉS

5.2.1 Guide de lecture des diagrammes TURTLE

Ce paragraphe a pour but d'une part de présenter de manière générale les diagrammes utilisés dans le profil TURTLE, et d'autre part de faciliter la lecture des diagrammes présentés dans le reste du chapitre, en présentant de manière concise les différentes possibilités couramment utilisées.

L'objectif du profil UML temps réel TURTLE [2] est de combler certaines carences de la notation UML de l'OMG (Unified Modeling Language [50]) et des outils commerciaux associés. Le profil TURTLE a étendu les diagrammes de classe et d'activité définis dans UML pour décrire la structuration d'un système et les comportements des classes identifiées dans le diagramme de classe. Les classes utilisées dans les diagrammes TURTLE sont appelées *Tclasses* (pour ¡¡ TURTLE classes ¿¿). Ces Tclasses sont dotées de portes de communication par rendez-vous. A chacune des Tclasses est associé un (et un seul) diagramme d'activité qui décrit le comportement de celle-ci : cela permet d'expliciter dans quel(s) cas les portes de ces classes sont franchies.

TURTLE a emprunté à RT-LOTOS l'idée d'utiliser des opérateurs de composition, afin d'exprimer des relations de parallélisme, de séquence, de synchronisation, d'invocation et de préemption entre les tâches décrites par les classes. Ensuite, des opérateurs temporels (dont la latence) ont été intégrés. Ces derniers surpassent en pouvoir d'expression le mécanisme de temporisateur qu'UML 2.0 a hérité des Statecharts de SDL.

Les figures 5.1 et 5.2 visent à donner, au travers d'un exemple, une vue synthétiques des diagrammes TURTLE. Dans cet exemple, une classe *Émetteur*, et une classe *Récepteur* sont définies. La classe *Émetteur* peut envoyer des données et recevoir des messages d'erreurs. La classe *Récepteur* peut recevoir des données. Si ces données arrivent avant *timeout_reception* unités de temps, le récepteur utilise les données reçues (le traitement prend 5 à 20 unités de temps), sinon il envoie un message d'erreur à la source.

La figure 5.2 montre en particulier, qu'il est possible de spécifier dans le diagramme d'activité, un passage de valeurs associé à une porte de synchronisation. Notons que, à l'heure actuelle, seules des valeurs entières ou booléennes peuvent être transmises. Les valeurs sont envoyées lorsque le caractère ¡¡ ! ¿¿ est utilisé (envoi de la valeur *code_erreur*

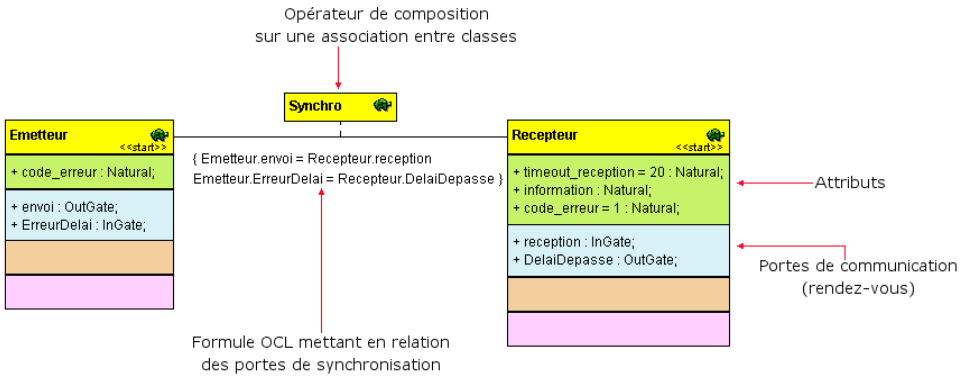


Figure 5.1 – Exemple de diagramme de classes TURTLE

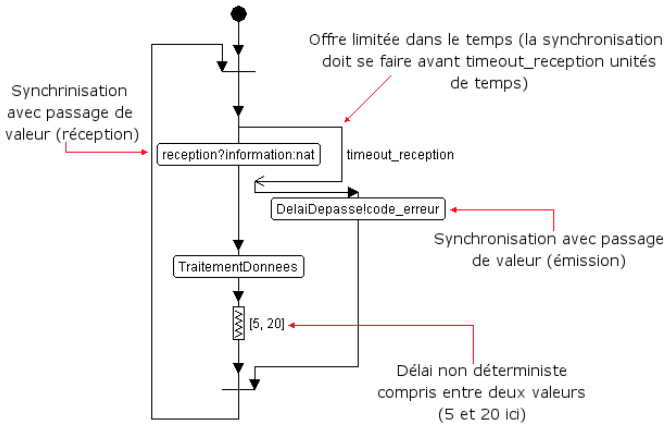


Figure 5.2 – Échange de valeurs entre deux classes TURTLE

lorsque la porte *DelaiDepasse* est franchie). La réception des valeurs est associée au caractère ¡¡? ¿¿, et le type des valeurs doit être précisé (réception d'une valeur stockée dans la variable *information* lorsque la porte *reception* est franchie).

5.2.2 La chaîne d'outils TTool-RTL

Le profil TURTLE est directement supporté par le logiciel TTool [134], développé à l'ENST. TTool inclut un éditeur graphique de diagrammes de classe TURTLE, un vérificateur de syntaxe, un générateur de code RT-LOTOS, ainsi que des outils de visualisation et d'analyse des résultats de simulation et de validation. Les simulations font ici référence à une exploration aléatoire partielle de l'espace d'état. Le processus de validation est, dans ce manuscrit, une ¡¡ validation *a priori* ¿¿ qui utilise des simulations pour observer des évolutions possibles du système, puis, lorsque le système est borné, construit l'espace d'état afin de vérifier les propriétés du système par rapport à une spécification préalable.

Le profil TURTLE est ainsi exploitable grâce à l'utilisation de RTL (Real-Time

LOTOS Laboratory [120]) : à partir d'un modèle, TTool peut générer une spécification RT-LOTOS [21]. Celle-ci peut ensuite être passée en entrée à l'outil RTL, à des fins de validation comportementale et temporelle. RTL est un outil développé au LAAS-CNRS qui, à partir d'une spécification RT-LOTOS, permet de réaliser :

- Des simulations aléatoires pour une durée donnée. Dans ce cas les résultats représentent une évolution possible du système.
- L'analyse exhaustive des états du système considéré. RTL peut alors — lorsque le système est fini — générer un graphe d'accessibilité optimisé. Les graphes d'accessibilité sont des systèmes de transitions étiquetées et temporisées. Ceux-ci mettent donc en évidence la progression du temps.

Les résultats ainsi obtenus sont ensuite récupérés par TTool pour être analysés. En particulier TTool a récemment intégré la possibilité de projeter les graphes obtenus. Il est en effet possible de minimiser des graphes d'accessibilité en déclarant non observables un certain nombre d'évènements. Il résulte de cette opération un automate quotient qui met en exergue les seuls évènements observables. Cet automate — qui peut encore contenir quelques événements internes non éliminés par l'équivalence observationnelle et ce, pour cause de préservation de l'indéterminisme — peut par exemple être manuellement comparé à un automate de service produit en phase d'analyse. Ces projections sont réalisées grâce à l'outil ALDEBARAN qui fait partie intégrante de CADP (Construction and Analysis of Distributed Processes [12]).

Le processus global de validation d'un modèle TURTLE est représenté dans la figure 5.3. L'utilisation de TTool permet de faciliter grandement ce processus car elle rend l'utilisation de RTL et CADP transparente pour l'utilisateur : celui-ci peut réaliser des simulations ou générer le graphe d'accessibilité du système modélisé, sans écrire ou examiner une seule ligne de code RT-LOTOS. Notons de plus qu'il est possible d'intégrer

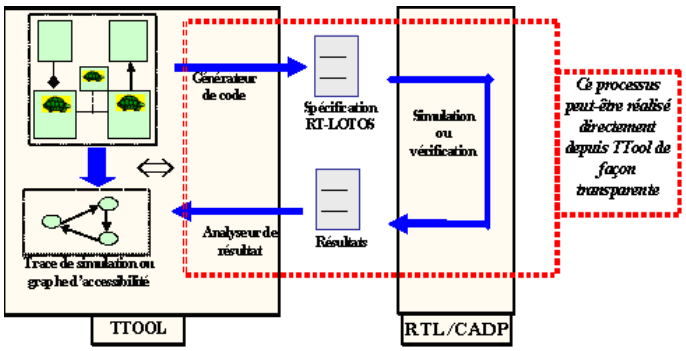


Figure 5.3 – Processus de validation de diagramme TURTLE

au modèle, des observateurs non intrusifs représentés par des classes TURTLE. L'utilisation de tels observateurs permet par exemple de rechercher des cas d'erreurs sans risquer de modifier les propriétés du système global. Enfin TTool permet d'insérer des commentaires sous forme de note, ce qui permet d'expliciter les actions et rend le modèle plus facilement compréhensible.

La proposition HSTRM a été modélisée au moyen du logiciel TURTLE, afin de vérifier les propriétés du système, et par là les services effectivement rendus à la couche

applicative. La validation de ce modèle repose sur la méthodologie présentée dans la section suivante.

5.3 MÉTHODOLOGIE

Nous nous inscrivons ici dans le cadre d'une ingénierie basée modèle et centrée sur l'architecture, avec pour objectif de montrer qu'une couche de protocole modélisée en TURTLE rend effectivement le service attendu. Nous avons appliqué au protocole HS-TRM le cycle de conception décrit par la figure 5.4. Le cycle repose sur une modélisation que nous qualifions d'¡¡ agile ¿¿ dans la mesure où, plus qu'un processus rigoureux voire rigide, nous proposons une construction incrémentale d'un modèle TURTLE par introduction progressive des exigences exprimées au niveau de la description du service au chapitre 4.

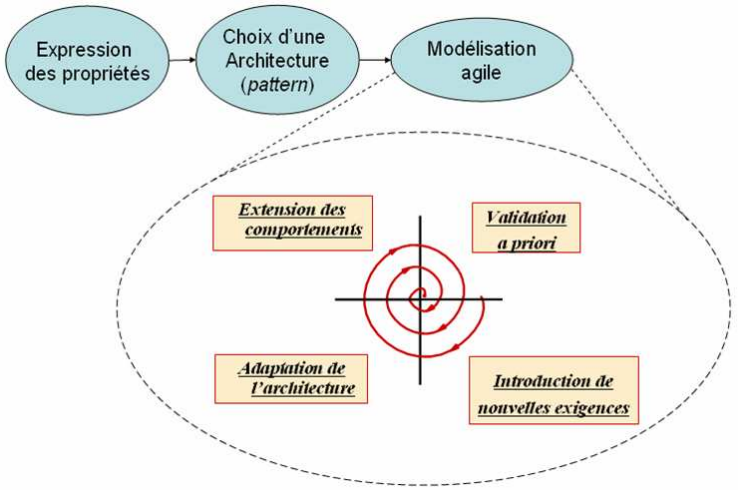


Figure 5.4 – Une méthodologie basée modèle

Pour modéliser le service attendu, nous n'utiliserons pas les diagrammes de séquences UML mais une simple description textuelle (qui pourrait par ailleurs être traduite en diagramme de séquence). Une fois le service exprimé, nous nous fixons une architecture de conception basée sur un pattern parmi les plus classiques : un modèle à trois niveaux, inspiré à l'origine par le modèle OSI de l'ISO et dans lequel deux entités de protocole s'appuient sur un service de communication existant pour rendre à leur tour un service à leurs utilisateurs. Une fois convenu d'utiliser cette architecture, elle évoluera en termes de messages échangés et de machines de protocoles à mettre en œuvre, pour traiter d'une part les messages les plus récemment introduits et d'autre part, la levée progressive d'hypothèses restrictives (par exemple le passage d'un service sous-jacent réputé fiable à un service avec perte).

Le canevas méthodologique ainsi défini n'est pas l'apanage d'un outil UML en particulier. Les outils TAU G2 de Telelogic ou Rhapsody d'Ilogix permettent sa mise enœuvre. Si l'on se penche sur la brique ¡¡ validation a priori ¿¿ de la figure 5.4, l'originalité de l'approche TURTLE réside dans la vérification par abstraction, basée sur

une analyse d'accessibilité implantée par l'outil RTL, et complémentée par une minimisation à l'aide de l'outil Aldébaran. Dès lors que le système modélisé est borné, nous construisons un graphe d'accessibilité qui prend la forme d'un système de transitions étiqueté temporisé. Les étiquettes qui apparaissent sur les transitions font référence à des identificateurs de porte (*gates*) TURTLE qui représentent chacune un message (primitive de service ou PDU). Ces échanges de messages sont traités comme des événements observables, par opposition aux événements internes de type ¡¡ progression du temps ¿¿. Une fois le graphe construit, le problème est d'en extraire une vue abstraite caractérisant le service rendu par le protocole. L'outil TTool permet à cet effet de sélectionner les portes TURTLE que l'on désire observer, les autres se ramenant alors au rang d'événements internes. Dès lors que l'on retient pour seuls événements observables les échanges de primitives de service à la frontière supérieur des entités de protocole, la minimisation du graphe d'accessibilité par rapport à une équivalence telle que l'équivalence observationnelle de Miner, donne un automate quotient caractéristique du service rendu par les entités de protocole. Cet automate — qui peut encore contenir quelques événements internes non éliminés par l'équivalence observationnelle et ce, pour cause de préservation de l'indéterminisme — peut être comparé à la description du service produit en phase d'analyse.

Plus précisément, le processus de validation s'est déroulé en plusieurs étapes :

1. La proposition a été décrite au moyen du diagramme de classe TURTLE.
2. Pour chaque classe, le diagramme d'activité a été produit.
3. Des simulations ont été menées pour vérifier que le comportement du modèle était conforme à la description du chapitre 4.
4. Le graphe d'accessibilité du système a été généré. Ce graphe a été projeté pour n'observer que les transitions correspondantes au service qui devait être validé.
5. Une fois un service validé, un autre service était ajouté au modèle pour être validé (retour au premier point).

Cet ensemble d'actions a permis de vérifier les différents états traversés par le système. En particulier cela a permis d'observer les fonctions rendues par les différents mécanismes utilisés dans la proposition, les cas d'échecs, ainsi que les conditions de ces échecs.

5.4 MODÈLE TURTLE DU COMPORTEMENT DE HSTRM

Cette section présente le modèle adopté pour vérifier les propriétés de la proposition. Ce modèle a permis de configurer correctement les valeurs des variables de HSTRM (cf. 4.6), et en particulier les valeurs des timers : grâce à l'intégration de critères de temps dans le modèle, il a été possible de détecter des incohérences dans les valeurs retenues. Notons que les unités de temps utilisées dans le modèle n'ont pas de valeur prédéfinie. Nous avons choisi de les identifier à des millisecondes car cette granularité était à la fois nécessaire et suffisante pour décrire la transmission des données. La section suivante expose le modèle général, tandis que chaque partie de ce modèle est présentée plus en détails dans les sections ultérieures.

5.4.1 Présentation générale du modèle TURTLE adopté

Pour réaliser ce modèle, nous avons adopté une approche simple : nous avons cherché à respecter les différentes couches protocolaires existantes, tout en nous focalisant sur la couche transport. Ce modèle est représenté dans la figure 5.5. On peut distinguer trois niveaux dans ce modèle : la couche ¡¡ application ¿¿, la couche transport et la couche réseau. La couche application représente en réalité l'interface transport-application. Cette classe représente donc le comportement de l'application dans la mesure où les requêtes de l'application arrivent directement à cette interface. De la même manière, la classe *NetworkService* modélise le service rendu au niveau de l'interface transport-réseau : la couche transport transmet des informations à la couche réseau pour que celles-ci soient acheminées jusqu'au destinataire. Entre les deux, la couche transport met en place les mécanismes nécessaires pour rendre le service de niveau transport décrit dans la section 4.1. Notons que, dans un objectif de simplification des diagrammes de classes, TTool confond l'instance d'une classe et la classes elle-même, dès lors qu'une seule instance de cette classe est présente. C'est le cas par exemple pour les classes représentant le niveau transport et l'interface applicative de la source.

Afin d'évaluer toutes les phases de la communication, le modèle intègre plusieurs récepteurs. Ainsi la phase d'échange d'informations entre les récepteurs est incluse dans l'évolution du système. Cette validation ne concerne cependant pas le passage à l'échelle du protocole. La validation formelle de la proposition pour l'utilisation à grande échelle n'a pas encore été abordée : cela impliquerait un modèle comprenant des milliers de classes représentant les récepteurs. Un tel modèle serait à la fois très complexe, et difficile à valider par RT-LOTOS (à cause d'un temps de calcul très important). Enfin les résultats obtenus seraient délicats à exploiter car ils impliqueraient de traiter des graphes dont le nombre d'états serait proportionnel au nombre de récepteurs[1].

Notons de plus que le modèle vérifié est un modèle simplifié du comportement de HSTRM (les simplifications sont explicitées dans les sections qui suivent). Cela est principalement dû à la complexité liée à l'expression d'un comportement sous forme explicite. Le modèle présenté doit donc être considéré comme la première étape d'une démarche de validation. Ce modèle devrait être complété lors de travaux ultérieurs. Les sections suivantes explicitent les classes représentées dans la figure 5.5 : le service réseau est détaillé dans la section 5.4.2, la section 5.4.3 décrit les interfaces applicatives, et enfin la proposition de transport est exposée dans la section 5.4.4.

5.4.2 Modèle TURTLE adopté pour le service réseau

Le modèle adopté pour le service réseau est relativement simple : le niveau transport lui donne des informations à transmettre, et elle achemine ces information jusqu'au destinataire. Plus précisément, parmi les informations transmises, se trouve un champ *ToAd* qui représente l'adresse du destinataire, et un champ *FrAd* qui représente l'adresse de l'émetteur. Afin de simplifier le modèle, les adresses ont été représentées par des entiers naturels (les variables échangées dans TURTLE ne peuvent prendre que des valeurs entières ou booléennes). De manière arbitraire, l'adresse de la source a été fixée

[1] des travaux en cours au DMI de l'ENSICA étudient la possibilité de produire des graphes formés de milliers d'états, et de les exploiter. Ces travaux ne sont pour le moment pas finalisés.

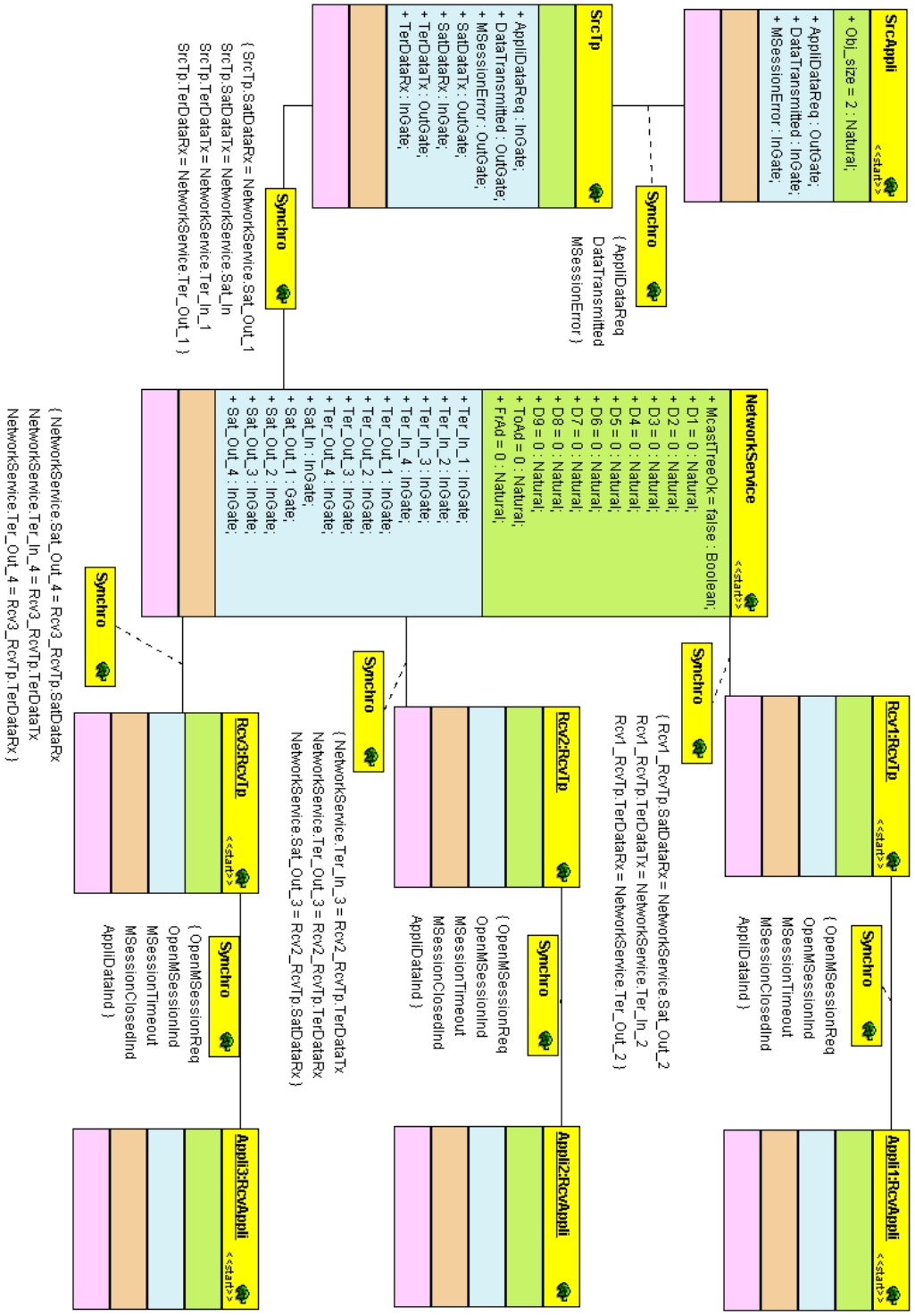


Figure 5.5 – Diagramme TURTLE représentatif du modèle général adopté

à 1, et les adresses des récepteurs à 2, 3, et 4. De plus, nous avons considéré qu'il était possible d'émettre vers une adresse de groupe. Cette adresse à été choisie égale à 10, et toutes les adresses dont la valeur est supérieure ou égale à 10 sont considérées comme

étant des adresses de groupe. En fonction de la valeur de l'adresse du destinataire, l'information est transférée de différente manière :

- Lorsque l'adresse est comprise entre 1 et 4, elle est transmise directement au destinataire. Cette communication est représentative d'une communication point-à-point terrestre. La partie du diagramme d'activité correspondant à ce type d'action est représentée dans la figure 5.6 : chaque entité peut émettre sur le réseau au moyen de la porte *Ter_In_i*.
- Lorsque l'adresse est égale à 10, l'information est diffusée à tous les récepteurs. Cette communication est représentative d'une communication utilisant un lien satellite. La différence entre les caractéristiques des transmissions point-à-point et des transmissions satellites représente le caractère hybride de la proposition : les transmissions utilisent les réseaux terrestres et satellites.

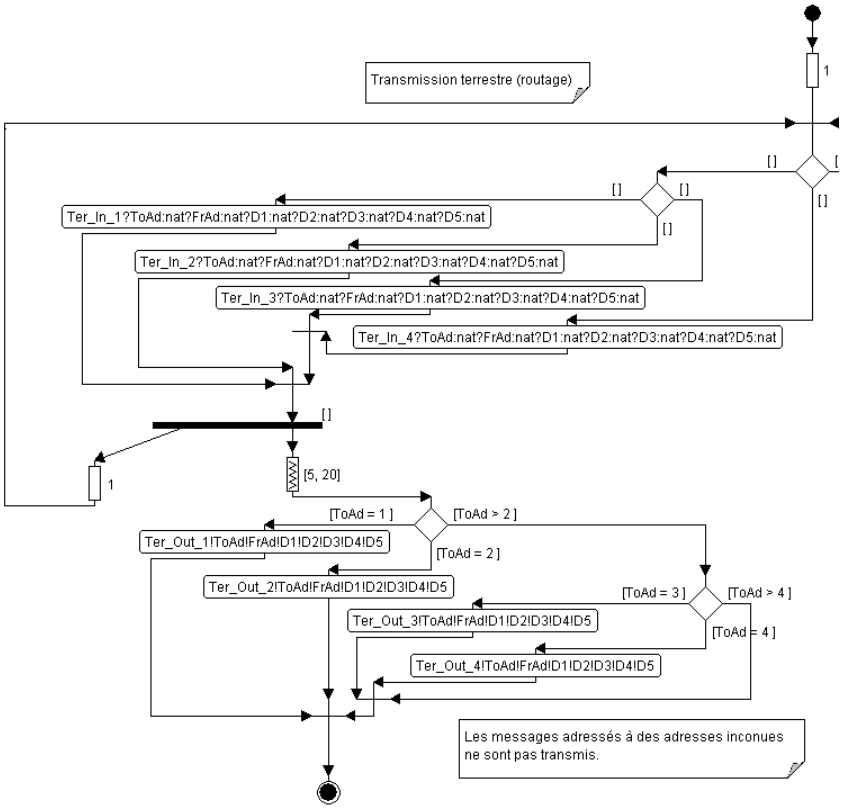


Figure 5.6 – Partie du diagramme d'activité représentant les transmissions point-à-point

En plus des adresses nécessaires à l'acheminement des données, le niveau transport confie à l'interface réseau un ensemble de données à transmettre. Ces données sont représentées par les variables $D1$, $D2$, $D3$, $D4$, et $D5$. Ces variables sont représentées par des entiers naturels, et représentent les informations contenues dans l'en-tête du protocole de transport (elles seront exploitées par la couche transport).

Le modèle ainsi obtenu permet de faire plusieurs requêtes de transmission avant que les premières requêtes soient arrivées à destination. Une des approximations de ce modèle concerne l'ordre de réception des paquets : les paquets sont transmis au

récepteur dans l'ordre où ils ont été émis. Le modèle ne prend donc pas en compte le désordonnancement possible des paquets émis. Notons de plus que le modèle de transmission point-à-point considéré ne prend pas en compte la possibilité de pertes. Ce choix a été effectué pour deux raisons : 1) il permet de simplifier les calculs de validation, et 2) nous avons supposé que toutes les communications point-à-point se faisaient avec TCP (la proposition ne couvre que les communications multipoints). Le service obtenu au niveau supérieur est donc un service, entre autre, totalement fiable. Enfin, le délai de transmission a été supposé compris entre 5 et 20 ms.

Le modèle de comportement développé pour les communications impliquant un lien satellite inclut la possibilité de perdre des paquets. Cependant, de manière à s'assurer que le système était fini, le nombre de paquets qu'il est possible de perdre est limité à un maximum : *MaxLost*. Ce maximum peut être vu comme représentant les plus mauvaises conditions de réception possibles sur une liaison satellite. La partie du diagramme d'activité relative à la transmission des données vers un récepteur est donnée dans la figure 5.7. Le délai de bout en bout a été choisi entre 300 et 400 ms. Dans le modèle, la

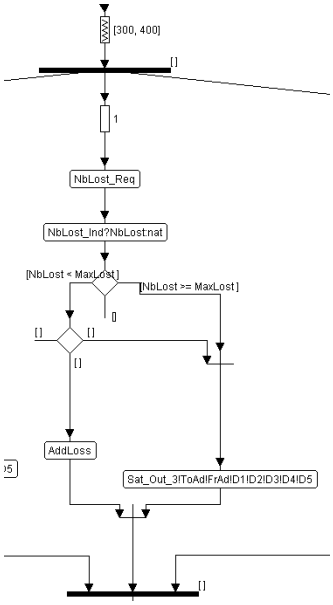


Figure 5.7 – Partie du diagramme d'activité représentant une transmission satellite

transmission des données fait appel à trois branches — identiques à celle représentée — parcourues en parallèle. Dans chaque branche les paquets peuvent être perdus (ce qui correspond à l'action *AddLoss*), ou transmis (ce qui correspond à l'action *Sat_Out*).

5.4.3 Modèles TRUTLE des interfaces applicatives de la source et des récepteurs

On peut distinguer deux types d'interface applicative : celle de la source et celle des récepteurs. Ces deux interfaces sont décrites ci-dessous. Pour les deux modèles, les problèmes liés à l'ouverture de la session n'ont pas été considérés :

– il n'y a pas de contrôle sur la date à laquelle la demande d'adhésion au groupe est effectuée, et

– le temps de mise en place au niveau réseau de l'arbre de diffusion n'est pas modélisé.

Ces deux hypothèses reviennent à supposer que la source et les récepteurs se sont déjà abonnés au groupe multicast. Le support de communication de groupe est en effet dans ce cas, déjà en place.

5.4.3.1 Interface de l'émetteur

L'interface de l'émetteur est décrite dans la figure 5.8. L'application source peut, selon cette description, demander de transmettre des données. Une fois la requête d'émission effectuée, l'application peut recevoir soit une indication de fin de transmission, soit une indication d'erreur de transmission. Ce modèle est relativement simplifié

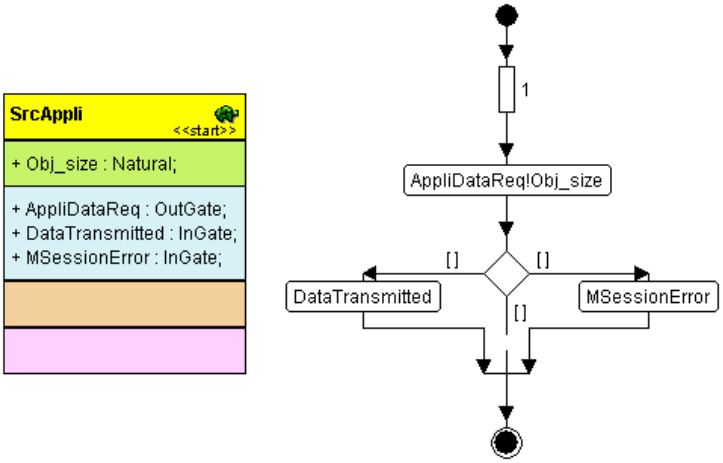


Figure 5.8 – Classe TURTLE et diagramme d'activité de l'interface applicative source

car il suppose que la transmission s'arrête après la transmission d'un seul objet. De plus ce modèle ne laisse pas la possibilité à l'application d'arrêter la transmission en cours : la fermeture de session se fait par la couche transport, une fois les données transmises, ou lorsqu'une erreur a été détectée. Notons toutefois que la possibilité d'arrêter une transmission en cours n'est pas vraiment compatible avec l'approche adoptée pour minimiser le coût de la communication : si la transmission s'arrête, le coût d'utilisation du réseau ne pourra être amorti car les utilisateurs n'auront pas reçu de données exploitables.

5.4.3.2 Interface des récepteurs

Du côté des récepteurs, le modèle considéré ne permet à l'application que de se mettre en attente de réception (*StartRx*). Une fois dans cet état d'attente, celle-ci peut recevoir un message indiquant qu'un volume de données a été reçu, un message indiquant que la session est terminée, ou un message indiquant que le timer de réception est arrivé à expiration (indication d'une erreur de réception). Ce modèle est représenté dans la figure 5.9.

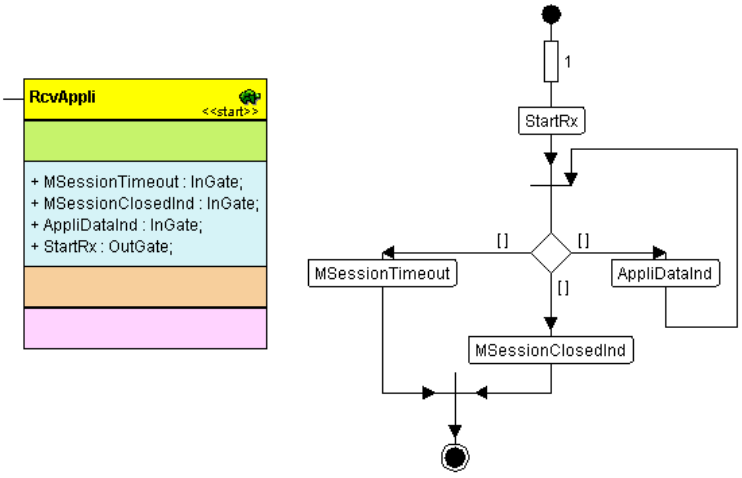


Figure 5.9 – Classe TURTLE et diagramme d'activité de l'interface applicative réceptrice

5.4.4 Modèle TURTLE de la proposition de transport

Les classes TURTLE représentant la couche transport émettrice *SrcTp* et réceptrice *RcvTp* sont en fait composées de plusieurs classes, chacune ayant une fonction particulière. Celles-ci sont détaillées dans les sections qui suivent.

5.4.4.1 Couche transport émettrice

L'approche que nous avons adoptée pour modéliser la proposition de transport a été de rester aussi fidèle que possible au ¡¡ découpage ¿¿ en mécanismes de niveau transport. Le modèle est ainsi composé de plusieurs module (chaque module a la vocation de rendre un service particulier, et est composé d'une ou plusieurs TClasses :

- d'un module chargé d'émettre les données en fonction de la taille du groupe (comme cela est décrit dans la figure 4.1 du chapitre 4),
- d'un module chargé de demander des rapports de réception et d'estimer la taille du groupe en fonction des retours,
- et d'un module chargé de gérer l'établissement du réseau de pair-à-pair.

En plus de ces trois modules, nous avons ajouté un module chargé de stocker des valeurs partagées par les différents modules, et un module permettant de faire l'interface avec le service réseau (envoi de paquets). L'ensemble de ces modules est représenté dans la figure 5.10. Les fonctions de chacun de ces modules sont présentées ci-dessous.

Transmission des données : Ce module est chargé de transmettre les données au groupe. Les données sont d'abord regroupées par blocs et méta-blocs (cf. section 3.2). Les données sont transmises entrelacées. Après l'émission des données initiales transmises, ce sont des données encodées qui sont transmises. Au cours de la première phase, la transmission se fait d'un bloc, tandis que lors de la seconde phase, le module regarde régulièrement la taille du groupe. Cela afin d'arrêter la transmission dès qu'il

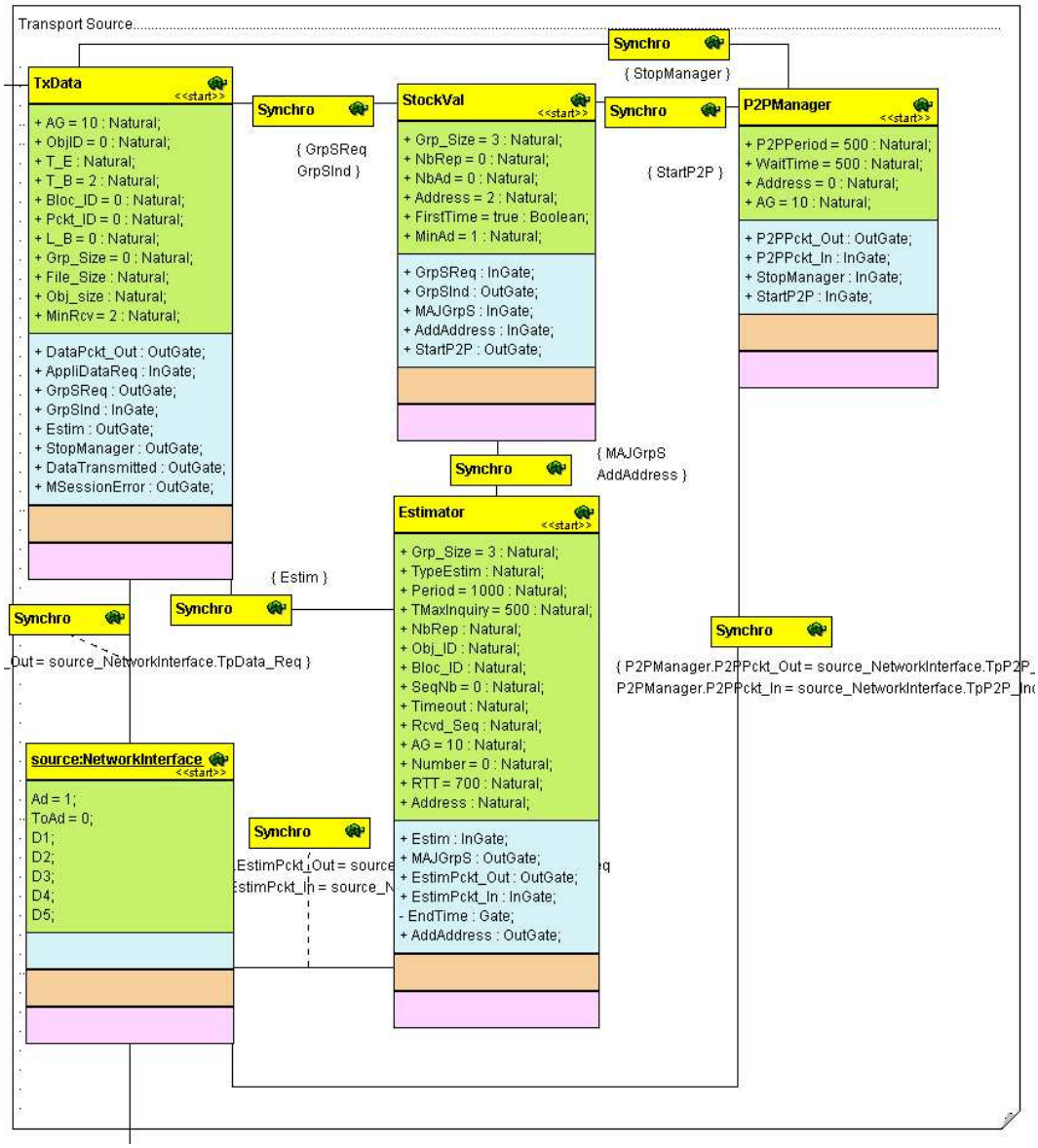


Figure 5.10 – Diagramme TURTLE de la couche transport de l'émetteur

ne reste pas assez de récepteurs pour que l'utilisation du satellite soit intéressante. Dans ce modèle, les opérations de codage n'ont pas été prises en compte. Les paquets ¡¡ encodés ¿¿ sont identifiés uniquement les champs **Bloc ID** et **Packet ID** : dès que le champ **Packet ID** est supérieur au nombre de paquets dans le bloc **Bloc ID**, cela correspond à un paquet encodé.

Estimation de la taille du groupe : Ce module a pour fonction d'envoyer les messages ***Inquiry*** lorsque le module d'émission le demande, et de collecter les réponses des récepteurs. La taille du groupe est ensuite estimée en fonction des réponses. Cependant il n'a pas été possible de modéliser l'estimateur présenté à la section 3.3. En réalité dans le modèle, tous les récepteurs répondent (il n'y en a que trois). La taille

du groupe est donc directement donnée par le nombre de réponses. Ce module collecte de plus les adresses des récepteurs ayant répondu pour qu'elles puissent être utilisées dans le réseau de pair-à-pair.

Gestion du réseau de pairs : Dans le modèle, le module de gestion du réseau de pairs a pour unique but de transmettre la liste d'adresses toutes les **P2PRLPeriod** unités de temps. Cette liste est transmise dès que suffisamment d'adresses sont connues, et ce jusqu'à ce que la transmission s'arrête. Il y a deux grandes limitations dans le modèle proposé : 1) la liste est constituée d'une seule adresse, et 2) les récepteurs ne peuvent pas se connecter à la source. Le fait que la liste ne contienne qu'une seule adresse est dû d'une part au fait que le groupe ne soit constitué que de trois récepteurs, et d'autre part au fait que TURTLE ne permette pas d'échanger des types complexes (comme un tableau d'adresses).

Stockage : Ce module permet de mettre à jour et de consulter la taille du groupe, et la liste des adresses connues.

Interface réseau : Cette classe récupère les messages émis par toutes les autres classes. Elle y ajoute l'adresse correspondant à l'hôte (adresse source), et les transmet au service réseau. Elle reçoit également les paquets transmis par le service réseau et les distribue, en fonction de l'en-tête du paquet, aux différents modules. La transmission des en-têtes des paquets a été modélisée par la transmission de 6 valeurs à l'interface réseau. Ces valeurs sont les suivantes : ToAd, D1, D2, D3, D4, D5. ToAd représente l'adresse de destination. D1 représente le type de paquet émis (cf. tableau 4.2). La signification des variables $(Di)_{2\leq i\leq 5}$ varie en fonction du paquet transmis. Pour les paquets de contrôle (paquets de type 1 ou 4), ces valeurs représentent les type de paquet de contrôle transmis, ainsi que les données transmises avec ce paquet (si le paquet transmis ne nécessite pas la transmission de données, ces valeurs sont laissées égales à 0). Pour les paquets de données (de type 2), ce sont les champs **Obj ID**, **Bloc ID**, **Packet ID** et **Last Bloc** qui sont transmis.

5.4.4.2 Couche transport réceptrice

La structure de la couche transport des récepteurs est quasiment identique à celle de l'émetteur. Elle comprend plusieurs parties :

Module de réception satellite : Celui-ci a pour rôle d'écouter les transmissions satellites, de stocker les données reçues, et d'informer l'application que les données ont été reçues.

Module de gestion des retours : Ce module a pour rôle de répondre aux messages ***Inquiry*** après avoir généré un temps d'attente. Le temps d'attente généré ne répond cependant pas à la fonction de répartition présentée dans la section 3.3 : celui-ci est uniformément réparti.

Module de gestion du réseau de pair : La fonction de ce module est de se connecter au réseau de pairs en fonction des adresses reçues, de chercher les informations manquantes lorsque cela lui est demandé, et de les télécharger.

A ces trois modules s'ajoutent un module chargé de stocker le nombre de paquets reçus, et une interface réseau (son rôle est identique à celle de l'émetteur). L'ensemble des modèles présentés a permis d'obtenir un plusieurs de résultats de simulation et de vérification. Ces résultats sont présentés et analysés dans la section suivante.

5.5 RÉSULTATS DE VALIDATION DU MODÈLE

Cette section présente les résultats de simulation et de vérification obtenus avec le modèle développé. Nous nous sommes cantonnés à la vérification de plusieurs propriétés essentielle concernant l'échange des données. Les propriétés que nous avons cherché à vérifier étaient les suivantes :

- l'absence de blocage,
- la fiabilité (absence de non réception des données injustifiée), et
- le bon fonctionnement des mécanismes liés à l'utilisation du réseau de pairs (diffusion des adresses des racines, connexion au réseau, recherche d'information, etc.).

La difficulté de l'évaluation de ces propriétés a été de déterminer si les problèmes rencontrés provenaient de la configuration du protocole ou du modèle lui-même. Notons de plus que TURTLE n'offre pas de facilité particulière à ce jour pour vérifier des propriétés : il est possible d'intégrer des observateurs, mais les diagrammes de séquence UML ne sont à ce jour pas intégrés. La portée de la validation effectuée a été croissante :

- dans un premier temps le modèle de service réseau a été validé. L'objectif était de déterminer s'il rendait un service conforme à ce qui était attendu,
- ensuite les couches transports de la source et des récepteurs ont été validées de manière unitaire.

Les étapes évoquées ci-dessus devaient initialement être complétées par la validation du modèle global. Cela s'est avéré irréalisable, pour des raisons liées aux outils utilisés : les calculs impliqués par ce processus demandaient une configuration matérielle trop importante de la machine chargée de les effectuer. Les résultats des validations effectivement réalisées sont présentés dans les sections qui suivent.

5.5.1 Validation du modèle de réseau

La première étape a consistée à s'assurer que le comportement du modèle de service réseau était conforme à ce qui était attendu. Tout d'abord les transmissions terrestres du modèle — pour lesquelles les paquets ne peuvent être perdus — ont été validées. Pour cela toutes les configurations d'envoi point-à-point ont été essayées, et le graphe d'accessibilité a été vérifié pour chacune de ces configurations. Pour cela nous avons manuellement analysé tous les chemins pouvant être parcourus dans le graphe d'accessibilité, et nous avons comparé chaque chemin à l'objectif en termes de service. Un

exemple est donné ci-dessous dans le cadre d'une transmission point-à-point. Dans ce cas l'objectif de service est la transmission systématiques des données (pas de pertes), et ce au bon destinataire. La figure 5.11 représente le cas où la source a envoyé un paquet au récepteur dont l'adresse est 2. L'envoi du paquet correspond à l'action *Ter_In_1*. Cette action est suivie par un état d'attente qui se termine par la transmission du paquet après un temps compris entre 5 unités de temps (première action *Ter_Out_2*) et 20 unités de temps (seconde action *Ter_Out_2*).

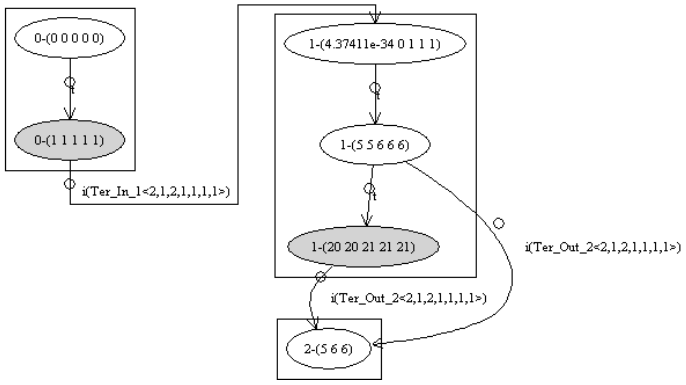


Figure 5.11 – Graphe d'accessibilité d'une transmission point-à-point avec le modèle de service réseau

Ensuite nous nous sommes intéressés aux transmissions multipoints. Dans ce cas, le paquet transmis peut être perdu pour chacun des récepteurs au cours de la transmission, c'est-à-dire que chaque récepteur peut recevoir ou non le paquet. Nous avons tout d'abord validé le modèle dans le cas où aucun paquet ne pouvait être perdu. La figure 5.12 montre le graphe d'accessibilité obtenu en ne considérant que les transitions observables (tous les états transitoires internes ont été enlevés). Ce graphe permet de conclure que dans tous les cas, les trois paquets sont transmis (action *Sat_Out_i*), et que seul l'ordre de transmission des paquets change.

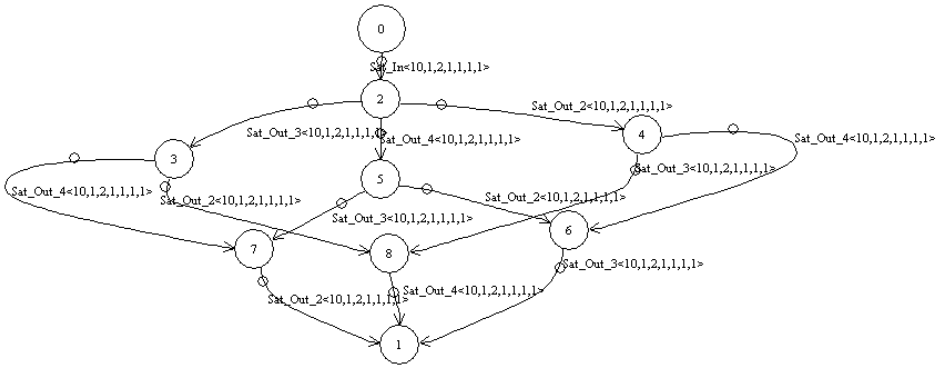


Figure 5.12 – Graphe d'accessibilité d'une transmission multipoint sans pertes avec le modèle de service réseau

Le problème rencontré lors de l'étape de validation après inclusion des pertes a été l'explosion combinatoire du nombre d'états et de transitions. A titre d'exemple, la figure 5.13 représente une fraction du graphe d'accessibilité total du modèle de transmission

satellite multipoint. Dans ce graphe on voit qu'il est possible de perdre un paquet, puis transmettre le paquet au récepteur 4 puis au récepteur 2 (itinéraire bleu), ou de perdre un paquet, puis transmettre le paquet au récepteur 3 puis au récepteur 2 (itinéraire rouge). Il a cependant été possible de vérifier le graphe obtenu pour la transmission d'un paquet, car il était composé d'une trentaine d'états.

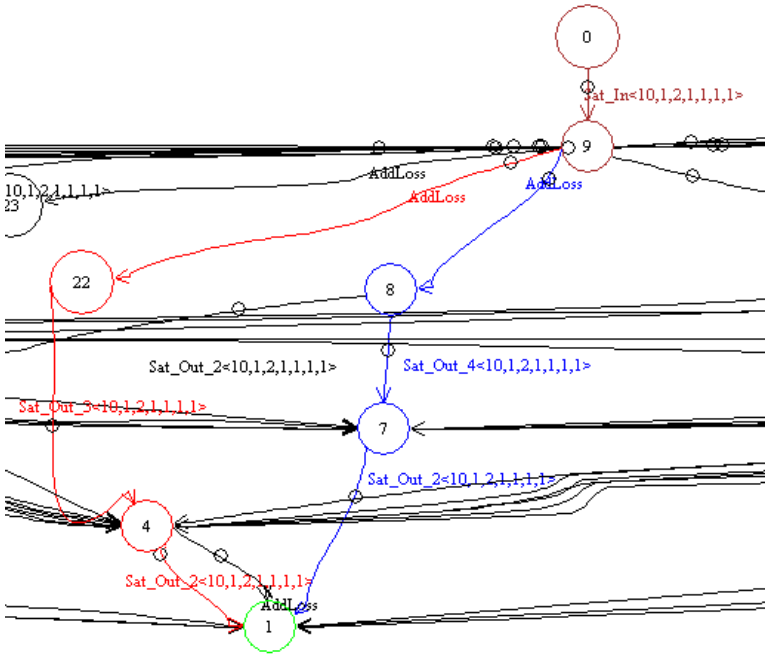


Figure 5.13 – Fraction du graphe d'accessibilité du modèle de service réseau

Comme le service réseau obtenu était conforme à l'objectif désiré, nous avons fait l'hypothèse que, si un problème se présentait par la suite, il proviendrait des modules utilisant ce service, et non du service lui-même. Nous avons donc décidé de ne plus observer les états liés au transit des paquets dans le réseau, et de nous concentrer sur les états des autres modules.

5.5.2 Validation du modèle de la couche transport de l'émetteur

Après la validation du modèle de service réseau, nous nous sommes intéressés au modèle de la couche transport de l'émetteur. Pour cela, nous avons vérifié dans un premier temps que le comportement de l'émetteur lors de la transmission des paquets était valide. Nous avons ensuite analysé le comportement du modèle à la réception de chaque paquet pouvant être émis par les récepteurs. La difficulté pour étudier le comportement en émission du modèle, est que l'émission des paquets est dépendante des messages reçus en retour. Il a donc été nécessaire de simuler — dans une certaine mesure — les messages émis par les récepteurs.

La figure 5.14 montre par exemple le graphe d'accessibilité relatif à la transmission des données dans le cas où aucun paquet n'est perdu lors de la communication. La communication commence à la demande de l'application (action *AppliDataReq*) qui

demande de transmettre 2 paquets (la taille des objets est directement donnée en nombre de paquets de manière à simplifier le modèle). Cela se traduit par l'envoi d'un message de configuration : la source envoie à l'adresse 10 un message de type 1 (message de contrôle, cf. 4.2), et dont la l'action est de type 9 (message ***MTPObjectConfig*** : voir le tableau 4.4 pour les différents messages de contrôle). Ce message précise que l'objet 1 est composé de 2 paquets, et que la transmission est effectuée avec 2 paquets par bloc. En conséquence, un seul bloc sera utilisé pour la transmission. La source lance ensuite la procédure d'estimation de la taille du groupe, et les données sont envoyées en parallèle avec l'estimation de la taille du groupe (après avoir initié le mécanisme) : tout d'abord le paquet $(0, 0)$ (bloc 0, paquet 0), puis le paquet $(0, 1)$. Pour ce dernier paquet, le champ **Last Bloc** est mis à 1 pour préciser qu'il s'agit du dernier paquet pour l'objet transmis. Après l'émission de ces paquets, la source évalue la taille du groupe (action *GrpSReq*). Comme aucun paquet n'a été perdu lors de la transmission, la taille du groupe est égale à 0 après émission de ces paquets. La source émet donc un message ***CloseMSessionReq*** (message de contrôle de type 2). Comme elle ne reçoit aucune réponse à ce message, elle indique à l'application que les données ont été transmises, et elle émet un message ***CloseMSessionConf*** (message de contrôle de type 3).

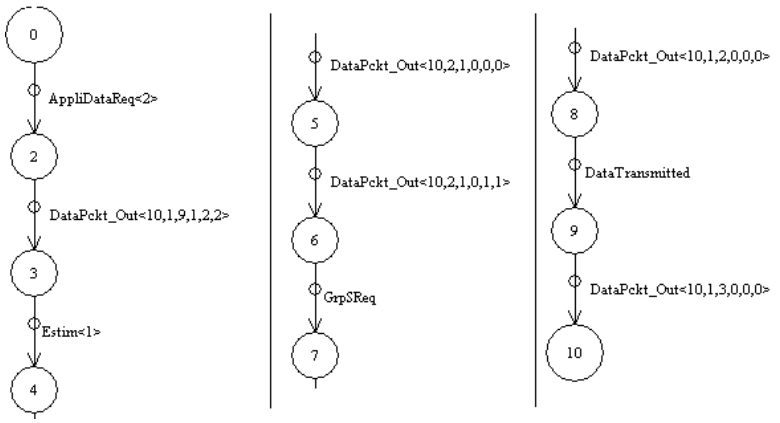


Figure 5.14 – Projection du graphe d'accessibilité de l'émetteur pour observer l'émission des données sans pertes de paquets.

Parmi les mécanismes modélisés, nous avons en particulier vérifié le comportement du mécanisme permettant de gérer les retours. Pour cela, lorsque la source envoie un message ***Inquiry*** (message de type 1, avec un contrôle de type 6) la partie réceptrice génère deux messages. Le premier est émis au bout d'un temps compris entre $0,5\,TMaxInquiry$ et $1,5\,TMaxInquiry$, et le second au bout d'un temps compris entre $0,5\,RTT$ et $1,5\,RTT$. La source attend la première réponse pendant au maximum **TMaxInquiry** unités de temps, et une fois la première réponse reçue, elle attend d'autres réponses pendant **RTT** unités de temps. La taille du groupe est ensuite ¡¡ estimée ¿¿ en fonction du nombre de réponses (conformément à ce qui a été annoncé dans la section 5.4.4, la taille du groupe est donnée par le nombre de réponses reçues). Avec ces paramètres, la source peut donc recevoir une, deux ou aucune réponse. C'est ce qu'illustre la figure 5.15 (seuls les états relatifs à ce mécanisme sont représentés) :

– la branche rouge correspond au cas où aucune réponse n'a été reçue, la taille du groupe est donc estimée à 0 (action *MAJGrpS¡0¿*),
– le chemin bleu correspond au cas où une seule réponse a été reçue, la taille du groupe est ainsi évaluée à 1 (action *MAJGrpS¡1¿*), et
– le chemin vert correspond au cas où les deux réponses ont été reçues dans les temps, la taille du groupe est ainsi estimée à 2.

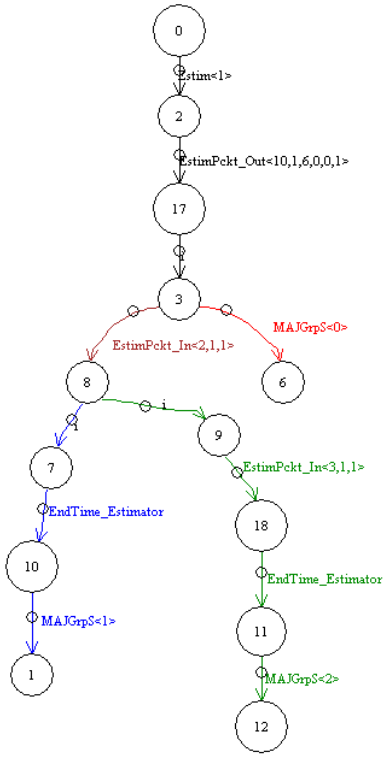


Figure 5.15 – Projection du graphe d'accessibilité de l'émetteur pour observer le mécanisme de limitation des retours.

Les différentes opérations de validation effectuées, ont permis de vérifier que le comportement du modèle de la couche transport de l'émetteur était conforme à la spécification du chapitre 4.

5.5.3 Validation du modèle de la couche transport du récepteur

Pour valider le modèle de récepteur, nous avons regardé le comportement de celui-ci après l'obtention de tous les paquets que celui-ci pouvait recevoir. Ensuite pour chaque type de paquet le graphe d'accessibilité a été généré de manière à détecter tout comportement anormal. La vérification du modèle a permis en l'occurrence de corriger des erreurs de modélisation, et d'obtenir un modèle valide. Nous avons particulièrement regardé les comportements liés à la réception de données, et à la demande de réponses. Par exemple, en ce qui concerne la transmission de données, la figure 5.16 montre le graphe d'accessibilité obtenu dans un cas simple : la communication est formée de 4 paquets. Tous ces paquets sont transmis (il n'y a pas de pertes). Ce graphe d'accessibilité montre que le récepteur reçoit et stocke les paquets reçus (actions *SatDataRx* et

StockData). Les champs passés en paramètre des actions StockData correspondent au numéro de bloc (en premier) et au numéro de paquet dans le bloc (en second). Lorsque le récepteur a reçu tous les paquets, il l'indique (ce qui correspond à l'action *DataRcvd*) et arrête la réception de données par satellite. Notons que cet arrêt ne concerne pas la réception de paquets de signalisation qui elle s'arrête lorsque la session est terminée. Ce graphe correspond donc à la spécification du comportement du récepteur pour ce qui est de la réception des données.

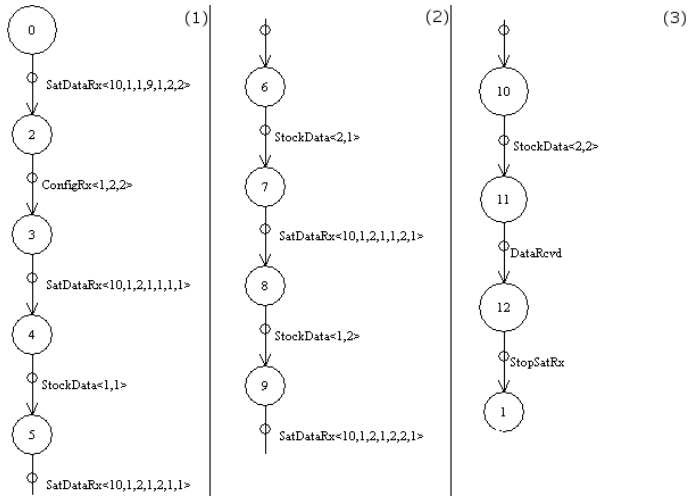


Figure 5.16 – Projection du graphe d'accessibilité d'un récepteur pour observer le comportement de celui-ci à la réception de données

La validation du mécanisme de gestion des retours (réponse aux messages ***Inquiry***) a été réalisée de la manière suivante :

- un paquet correspondant à un message ***Inquiry*** est envoyé,
- le récepteur doit répondre à ce message après 15 unités de temps,
- après un temps compris entre 5 et 20 unités de temps, la source renvoie un message ***StopInquiry***.

Cette configuration permet d'observer les cas où le récepteur répond, ainsi que les cas où le récepteur ne répond pas. La figure 5.17 présente le graphe obtenu. Ce graphe présente les deux cas nominaux :

- le chemin 0-2-8-6-7-1 : ce chemin correspond au cas où le message ***StopInquiry*** a été reçu avant les 20 unités de temps, ce qui a conduit à l'arrêt de la procédure de réponse (action *NoResponse*).
- le chemin 0-2-8-4-3-1 : dans ce cas la réponse est partie avant la réception du message ***StopInquiry***.

Entre les deux se trouve le cas limite où la source reçoit le message ***StopInquiry*** au moment où la réponse doit partir (représenté en marron). Dans ce cas soit l'envoi de la réponse est traité en priorité (chemin rouge), soit l'arrêt du processus est traité en priorité (chemin bleu). Ce cas, bien que possible, est en réalité peu probable, et le comportement du récepteur peut correspondre à l'une ou l'autre des possibilités sans que cela ait un impact fort sur le protocole. Au vu de ces résultats, le comportement du modèle ainsi défini est conforme au comportement spécifié dans le chapitre 4.

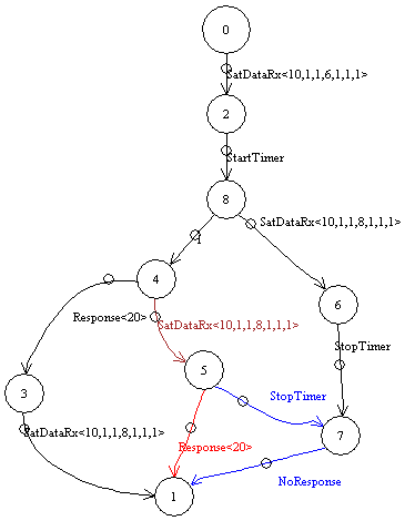


Figure 5.17 – Projection du graphe d'accessibilité d'un récepteur pour observer le comportement de celui-ci à la réception d'un message *Inquiry*

5.5.4 Conclusion sur les résultats de validation

L'ensemble des efforts de validation effectués au moyen du profil TURTLE a permis de vérifier que le comportement de l'ensemble des classes TURTLE spécifiées permettait de rendre le service pour lequel elles étaient conçues. A la suite des travaux présentés dans les sections précédentes, devait faire suite la validation du modèle global présenté dans la figure 5.5. Pour les raisons déjà évoquées (configuration matérielle insuffisante, et/ou problème logiciel non détecté), ce travail n'a pu être effectué. Cette dernière étape aurait permis de valider l'interaction de toutes les entités impliquées dans une communication HSTRM. Cette étape n'ayant pu être réalisée, la possibilité d'incohérences reste présente.

5.6 CONCLUSION SUR LE SERVICE APPLICATIF OBTENU

Le travail exposé dans ce chapitre est la suite logique de la proposition faite au chapitre 4 : il peut être vu comme une première étape nécessaire à l'implémentation de la proposition de transport. Après la description théorique des mécanismes utilisés au chapitre 4, ceux-ci ont ainsi été modélisés au moyen de diagrammes de classes et de diagrammes d'activité. Cela a permis de vérifier les fonctionnalités de chacun des mécanismes protocolaires impliqués dans une communication HSTRM. La maquette d'un protocole programmée conformément à la spécification réalisée serait donc assurée de fonctionner dans les conditions testées. Pour compléter les tests ¡¡ unitaires ¿¿ de chaque partie du modèle, nous nous sommes intéressés à la vérification du modèle complet. Il n'a toutefois pas été possible d'obtenir des résultats avec le modèle englobant tous les récepteurs, la source, et le service réseau. Ainsi, bien que chaque mécanisme fonctionne séparément, il reste possible que des dysfonctionnements apparaissent, du fait de leur utilisation conjointe.

Pour palier ce manque, le travail réalisé dans ce chapitre pourrait être utilisé comme base au développement d'une maquette de HSTRM. La réalisation d'une maquette pourrait être utilisée, par exemple, pour effectuer des tests en environnement réel. Cela permettrait de tester le fonctionnement de la proposition face à des conditions de transmission satellite réelles, et non plus avec un modèle de transmission satellite. Cependant, dans la mesure où le fait de disposer d'un système satellite n'est pas aisé, en particulier s'il doit répondre aux hypothèses adoptées, l'utilisation de techniques d'émulation semble appropriée. Ainsi, en modélisant la partie satellite du système, il serait possible d'étudier une implantation réelle de la proposition. Cette approche permettrait également de ne plus être tributaire des simplifications qui ont été effectuées lors de la spécification UML/TURTLE de la proposition. De tels travaux semblent donc inévitables pour vérifier le bon fonctionnement de la proposition.

Enfin la mise en œuvre d'une architecture de communication ne va pas sans le test de celle-ci. L'approche présentée dans ce chapitre permet de réduire les erreurs de conception, en spécifiant le comportement des classes de futures implémentations. Il reste néanmoins nécessaire de vérifier que ces dernières sont bien conformes à cette spécification. D'autres types de tests doivent également être menés avant le déploiement d'une architecture de communication : des tests de robustesse, d'interopérabilité, et de performances. Toutefois, il n'est pas question d'envisager cette phase de test de façon exhaustive [14]. Il est donc nécessaire : 1) de définir des objectifs de tests précis (étudiant des propriétés d'une implémentation), et 2) d'évaluer la couverture de ces tests. Pour le premier point, des techniques permettent de générer automatiquement des tests prenant en compte des critères temporels à partir de la spécification formelle du système à tester, et de la spécification formelle des objectifs de test [14]. Dans le cadre des réseaux mobiles, ce type de technique a par exemple été mis en œuvre sur une plateforme de validation permettant de réaliser des tests de conformité et d'interopérabilité [15]. En revanche, le calcul de la couverture des tests ainsi réalisés est encore, à notre connaissance, du domaine de la recherche.

Conclusion & perspectives

La problématique abordée au cours de ce manuscrit a été suscitée par le RNRT au travers du projet DIPCAST, et a largement dépassé le cadre de celui-ci. Cette problématique concernait les communications multipoints fiables à très grande échelle par satellite. Dans ce contexte, de nombreuses propositions de protocoles de transport multipoint ont tout d'abord été étudiées dans un cadre général. Cette étude a, entre autres, permis de dégager plusieurs mécanismes protocolaires permettant de rendre un service donné à la couche applicative. Ces mécanismes ont ensuite été confrontés à l'utilisation d'un lien satellite comme support de communication. La particularité de ce support a restreint d'une part les services qui pouvaient être rendus à l'application, et d'autre part les mécanismes qui pouvaient être utilisés de manière efficace. De plus, lors de cette seconde étude, il est apparu que l'utilisation d'un réseau hybride satellite / terrestre était particulièrement intéressante. Une étude du coût de communications multipoints, a en effet montré que l'utilisation concertée des réseaux terrestres et satellite permettait de réduire considérablement le coût de ces communications.

En nous basant sur ces résultats, nous avons donc proposé une nouvelle approche, pour la diffusion fiable multipoint à grande échelle. Cette approche, nommée HSTRM pour *Hybrid Satellite/Terrestrial Reliable Multicast*, consiste à définir un nombre minimum de récepteurs en dessous duquel la transmission satellite n'est plus intéressante (au vu d'une fonction de coût préalablement définie). En conséquence, le principe de HSTRM consiste à estimer régulièrement le nombre de récepteurs en attente d'informations, et à ¡¡ basculer ¿¿ vers l'utilisation du réseau terrestre dès que ce nombre passait sous le seuil défini. L'objectif principal de cette approche est ainsi de réduire le coût généré par des communications multipoints.

La mise en œuvre de cette approche a nécessité cependant la définition de mécanismes protocolaires spécifiques. Il faut en particulier disposer d'un mécanisme permettant de gérer la fiabilité, d'un autre permettant d'estimer la taille du groupe, et d'un troisième permettant d'utiliser le réseau terrestre.

Pour ce qui est de la gestion de la fiabilité, un mécanisme utilisant un code correcteur d'erreurs semblait recommandée. En effet cette technique, également connue sous le nom de FEC (pour *Forward Error Correction*) permet d'améliorer les possibilités d'utilisation à grande échelle des protocoles de transport multipoints fiables. L'utilisation de codes au niveau transport demande l'utilisation de codes à effacement. A l'heure actuelle, il existe deux grandes familles de codes à effacement : les codes MDS (*Maximum Distance Separable* codes) et les codes LDPC (*Low Density Parity Check*

codes). Ces deux types de codes ayant des caractéristiques très différentes, ceux-la ont été évalués, de manière à déterminer le type de code le plus adéquat pour HSTRM.

Le problème d'estimation de la taille du groupe a ensuite été étudié. Ce problème est en fait très lié à un mécanisme permettant de limiter les messages émis sur la voie retour. Plusieurs mécanismes permettant de réduire le nombre de messages émis par les récepteurs ont donc été étudiés. Parmi ceux-la, un mécanisme semblait particulièrement efficace : le mécanisme de *Nack Suppression*. Cependant, l'utilisation de ce mécanisme dans le contexte particulier d'un système hybride satellite/terrestre rendant un service de communication multipoint à très large échelle différait des utilisations déjà envisagées. Ce mécanisme a donc été évalué dans ce cadre. Il est apparu que celui-ci était très efficace une fois correctement configuré. Ce mécanisme étant choisi, le problème d'estimation de la taille du groupe a pu être considéré. Plusieurs estimateurs basés sur l'utilisation de ce mécanisme avaient déjà fait l'objet de travaux de recherche. L'étude de ces estimateurs (définis par le maximum de vraisemblance, ou par une approximation de celui-ci) a été complétée. Cela a abouti à la proposition d'un nouvel estimateur non biaisé, et à l'évaluation de tous les estimateurs étudiés dans le cadre de communications multipoints par satellite. Enfin, l'interaction entre le mécanisme d'estimation de taille de groupe et le mécanisme de limitation des retours a été évalué au moyen de simulations. Ce travail s'est achevé par la proposition d'un mécanisme complet permettant d'estimer la taille du groupe avec une précision suffisante, tout en limitant efficacement le nombre de messages émis sur la voie retour.

L'utilisation du réseau terrestre a également fait l'objet d'une étude. L'idée d'utiliser une liaison terrestre afin de compléter une transmission satellite remonte à de nombreuses années. Plusieurs travaux ont en effet montré l'intérêt de ce type de technique en termes de temps de latence et d'utilisation des ressources réseau. Un grand nombre de solutions existantes a ainsi été analysé dans le but de reprendre les pertes subies par les récepteurs. Cependant, parmi les solutions analysées, aucune ne considérait de contrainte sur le coût d'utilisation du réseau. Une solution adaptée a donc été proposée. Celle-ci a ensuite été évaluée, en termes de coût d'utilisation, de contrainte pour les hôtes, et d'efficacité des reprises d'erreurs.

Suite à la définition de ces mécanismes, la proposition complète a été décrite, en termes de comportement et de messages échangés. Cette description a permis de modéliser la proposition sous la forme des diagrammes définis par le profil TURTLE. Enfin, l'utilisation de la chaîne d'outils TTool-RTL, a rendu possible l'exploitation de ce modèle, en validant les fonctionnalités des mécanismes protocolaires modélisés.

De nombreux travaux pourraient faire suite à ceux réalisés au cours de cette thèse. Un prolongement a court terme des travaux effectués, consisterait à compléter la validation de la proposition. Il serait en effet intéressant dans un premier temps, d'étudier le comportement du modèle complet afin de valider le fonctionnement coordonné des différentes parties impliquées dans une communication avec HSTRM. Une fois le modèle global validé, il resterait encore à affiner les modèles de comportement adoptés pour chacun des modules composant la source et les récepteurs HSTRM. L'objectif d'un tel travail serait à terme, de converger vers un modèle représentant fidèlement le service effectivement rendu à l'application.

Les travaux évoqués ci-dessus restent cependant de l'ordre d'une validation *a priori* : ce sont des étapes de la phase de conception d'un programme. Ils sont donc destinés à être utilisés comme base au développement d'une maquette de protocole HSTRM. Cette maquette devrait alors être testée de manière à assurer la conformité de l'implémentation vis-à-vis de la description du service (description à la fois textuelle abstraite au travers de l'utilisation de TURTLE).

Les tests devraient de plus être menés dans un environnement réel. Toutefois, compte tenu du coût d'utilisation du système cible, et du peu de disponibilité de tels systèmes, l'utilisation de techniques d'émulation serait d'une grande utilité pour tester efficacement le protocole. Ce type de test permettrait d'observer le comportement d'une mise en œuvre de la proposition de transport, dans un environnement réel ou tout au moins proche de la réalité.

A plus long terme, plusieurs hypothèses pourraient être levées. Pour ce qui est du système satellite, la possibilité d'utiliser un satellite multispot (qui différencie plusieurs zones de couverture), ou un satellite traitant une charge ¡¡ mixte ¿¿ (i.e. une charge dont une partie est retransmise sans traitement, et dont le reste nécessite un traitement à bord du satellite) pourrait être considéré. Ces techniques ont cependant un impact important sur le service rendu par le système satellite. La mise en œuvre de HSTRM dans un tel environnement demande donc l'analyse de l'effet de ce(s) nouveau(x) service.

Une autre possibilité consisterait à envisager le changement de l'architecture en couche classique. A l'heure actuelle, de nombreux travaux analysent en effet la possibilité de rassembler des fonctionnalités des différentes couches de communication, à un même niveau (approche dite *cross layer*). Par exemple, en ce qui concerne la fiabilité des communications, la gestion des erreurs bit au niveau transport (ce qui revient à désactiver les contrôles d'intégrité des paquets des couches de niveau inférieur) peut s'avérer avantageuse. Ce type de technique change cependant les hypothèses prises sur le canal de communication, et par conséquent les besoins en termes de mécanismes de niveau transport. L'utilisation de ces techniques avec l'approche hybride définie ici, demande donc une étude approfondie.

Une possibilité d'étude concerne l'intégration du service IP Multicast dans le réseau terrestre. En effet, dans l'environnement considéré, la reprise terrestre des informations utilise un ensemble de communications point-à-point. Cependant, dans la mesure où l'objectif est de transmettre des informations à un groupe restreint de récepteurs, la transmission multipoint satellite pourrait être complétée par une transmission multipoint terrestre. Dans ce cas, un nouveau protocole permettant de choisir des communications multipoints terrestre ou satellite devrait être défini.

Enfin, l'approche proposée dans ce manuscrit ne se limite pas aux seuls scénarios évoqués. L'utilisation de cette approche à d'autres environnements pourrait également être une source de travaux intéressants. Par exemple, la diffusion de données sur des téléphones mobiles au moyen d'un satellite est un milieu relativement propice à l'application d'une telle approche.

Acronymes & Notations

Acronymes

ACK	ACKnoledgement
ADSL	Asymetric Direct Subscriber Line
API	Application Programmation Interface
ARQ	Automatic Retransmission reQuest
CDN	Content Delivery Network
CRC	Cyclic Redundancy Check
DAMA	Demand Assignement Multiple Access
DIPCAST	DVB comme support d'IP MultiCAST
DVB	Digital Video Broadcasting
EMV	Estimateur du Maximum de Vraisemblance
FEC	Forward Error Correction
FS	File Size
GEOCAST	Multicast over Geostationary EHF Satellites
HARQ2	Hybrid ARQ type 2
HSTRM	Hybrid Satellite / Terrestrial Reliable Multicast
IANA	Internet Assigned Numbers Authority
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engeenering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
LAN	Local Area Network
LDPC	Low Density Parity Check Codes
MàJ	Mise à Jour
MBone	Multicast backBone
MDS	Maximum Distance-Separable (codes)
NACK	Negative ACKnoledgement
NORM	Nack-Oriented Reliable Multicast
NTE	Network Text Editor
NTFS	New Technology File System
PLR	Packet Loss Rate
RMT-WG	Reliable Multicast Transport Working Group
RTT	Round Trip Time
SACK	Selective ACKnowledgement

SDP	Session Description Protocol
SM	scénario Satellite Multipoint
TDU	Transport Data Unit
TCP	Transport Control Protocol
TM	scénario Terrestre Multipoint
TRACK	Tree-based ACK protocol
TTL	Time To Live
TU	scénario Terrestre Unicast (point-à-point)
VSAT	Very Small Aperture Terminal

Notations

$\lfloor x \rfloor$	Partie entière inférieure de x.
$\lceil x \rceil$	Partie entière supérieure de x.
α_x	Coût de transmission d'un paquet avec la technologie x.
η	Nombre de pairs dans un arbre v-aire.
η_{min}	Nombre minimum de pairs dans un arbre v-aire.
τ	Nombre de racines dans le réseau de pair-à-pairs.
τ_{moy}	Nombre désiré de racines dans le réseau de pair-à-pairs.
χ^2_x	Loi de répartition chi-deux à x degrés de liberté.
$\mathcal{A}$	Alphabet d'un code.
A_c	Nombre moyen de messages ***ConnectP2P*** reçus par une racine.
$A_{c_{MAX}}$	Nombre maximum de messages ***ConnectP2P*** reçus par une racine.
B_x	Nombre de paquets par bloc lorsque le code x est utilisé.
c	Délai de transmission allez-retour (RTT).
$C_x(\cdot)$	Charge induite dans le réseau avec la technologie x.
D	Débit en émission au niveau applicatif.
d	Distance minimale d'un code.
$E(X)$	Espérance de la variable aléatoire X.
$f(x)$	Fonction de répartition des temps d'attente.
$F(x)$	Distribution cumulée de $f(x)$.
$F_x(N,T)$	Coût d'une transmission avec la technologie x.
$F_{P2P}(N,v,\tau)$	Coût d'établissement du réseau de pair-à-pair.
$G_x(N,S,TDU,T_B)$	Coût de transmission d'un objet de taille S.
h	Hauteur d'un arbre v-aire.
k	Dimension d'un code.
ℓ	Nombre d'expériences utilisées pour estimer N_A.
L	Dans un arbre v-aire, nombre de pairs ne possédant pas v voisins.
M	Nombre de mots d'un code.
M_G	Matrice génératrice

MinRcv	Nombre de récepteurs en dessous duquel la transmission. satellite n'est plus rentable.
N	Nombre de récepteurs dans une session Multicast.
N_A	Nombre de récepteurs actifs.
N_{Back}	Nombre de messages générés sur la voie retour.
N_{max}	Taille supposée maximale pour la taille du groupe.
N_{min}	Seuil pour la mise à jour de $f(x)$.
n	Longueur d'un code.
P_{Att}	Durée cumulée des atténuations lors d'une transmission.
p_r	Probabilité de réponse des récepteurs (mécanisme à probabilité contrôlée).
R_i	Nombre de temps d'attente générés entre $x^*_{1,i}$ et $x^*_{1,i} + c$.
S	Taille d'un objet envoyé.
s_c	Symbole encodé.
s_i	Symbole initial (non encodé).
T	Nombre de paquets à transmettre.
T'	Nombre de paquets effectivement transmis.
T_B	Nombre de paquets par bloc (avec un code en blocs).
T_E	Nombre de paquet par méta-bloc.
T_{max}	Nombre maximum de paquets générés avec un code en blocs.
v	Nombre d'enfants par noeud dans le réseau de pair-à--pair.
$x_{j,i}$	Délai d'attente généré par le récepteur i.
$\left(x^*_{j,i}\right)_{1\leq j\leq N_A}$	Statistique d'ordre associée à $(x_{j,i})_{1\leq j\leq N_A}$.
$x^*_{1,i}$	Plus faible délai générés lors de l'expérience i.

Publications

Le travaux réalisés au cours de ces trois années de thèse ont fait l'objet de plusieurs publications :

F. de Belleville, L. Dairaine, C. Fraboul, J. Lacan.
Une Approche Hybride Satellite/Terrestre pour le transport fiable multipoint à grande échelle.
Dans les actes du *Colloque Francophone sur l'Ingénierie des Protocoles*. 2003.

L. Lancerica, L. Dairaine, F. de Belleville, H. Thalmensy, C. Fraboul.
MITv - A Solution for an Interactive TV Based on IP Multicast over Satellite.
Dans les actes de *IEEE International Conference on Multimedia & Expo (ICME)*. 2004.

F. de Belleville, L. Dairaine, J. Lacan, C. Fraboul.
Reliable Multicast Transport by Satellite : a Hybrid Satellite/Terrestrial Solution with Erasure Codes.
Dans les actes de *IEEE conference on High Speed Networks and Multimedia Communications (HSNMC)*. Lecture Notes in Computer Science, Springer-Verlag. 2004.

F. de Belleville
Etude de codes à effacement dans le cadre d'un approche hybride satellite/ terrestre pour le transport multipoint fiable.
Dans les actes du *Colloque de l'EDIT*. 2004.

F. de Belleville, L. Dairaine, C. Fraboul, J.Y. Tourneret.
Group size estimation for Hybrid Satellite/Terrestrial Reliable Multicast.
Dans les actes de *IFIP World Computer Congress - Broadband Satellite Communication Systems Workshop*. 2004.

Liste des tableaux

Liste des figures

Bibliographie

[1] Adamson and Macker. MDP protocol specification version 1.6. http ://manimac.itd.nrl.navy.mil/MDP/DraftMdpSpec-1.6.txt, October 1999.

[2] L. Apvrille, J.-P. Courtiat, C. Lohr, and P de Saqui-Sannes. Turtle : A real-time uml profile supported by a formal validation toolkit. *IEEE Transactions on Software Engineering*, 30, 2004.

[3] L. Apvrille, P de Saqui-Sannes, and C. Lohr. Verifying continuity in a dynamic reconfiguration procedure : application to a satellite system. *Automated software engineering*, 11, 2004.

[4] S. Armstrong, A. Freier, and K. Marzullo. Multicast transport protocol, February 1992.

[5] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. In *IEEE-ACM Transactions on Networking*, volume 5, 1997.

[6] Fred Bauer. A transport protocol for reliable multicast over ip-multicast capable networks.

[7] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, MA, 1984.

[8] J. Blomer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D.Zuckerman. An xor-based erasure-resilient coding scheme. Technical report, International Computer Science Institute, Berkeley, California, 1995.

[9] R. Braden, D. Borman, and C. Partridge. *Computing the Internet Checksum*, 1988. Request for Comments 1071.

[10] John Byers, Jeffrey Considine, Michael Mitzenmacher, and Stanislav Rost. Informed content delivery across adaptive overlay networks. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 47–60. ACM Press, 2002.

[11] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM*, pages 56–67, 1998.

[12] CADP : Construction and Analysis of Distributed Processes. http ://www.inrialpes.fr/vasy/cadp/.

[13] Guohong Cao and Yiqiong Wu. Reliable multicast via satellites. In *International Conference on Information Technology : Coding and Computing (ITCC '01)*, 2001.

[14] R. Castanet and P. Felix. Objectifs de test pour systèmes temporisés. In *CFIP, Paris, Octobre*, 2003.

[15] R. Castanet, M. Mackaya, A. Cavalli, and et autres. Une plate-forme de validation multi-protocoles et multi-services - résultats d'expérimentation. In *CFIP, Paris*, 2003.

[16] Castify networks. http ://www.castify.com/homepage.htm.

[17] Yang-Hua Chu, Sanjay G. Rao, and Hui Zang. A case for end system multicast. In *Measurement and Modeling of Computer Systems*, 2000.

[18] John C.-I. Chuang and Marvin A. Sirbu. Pricing multicast communication : A cost-based approach. *Telecommunication Systems*, 17(3) :281–297, 2001.

[19] J. Cooperstock and S. Kotsopoulos. Exploiting group communication for reliable high volume data distribution. In *Proceedings of the IEEE Pacific Rim Conference on Communications*, 1995.

[20] Adam M. Costello and Steven McCanne. Search party : Using randomcast for reliable multicast with local recovery. In *INFOCOM (3)*, pages 1256–1264, 1999.

[21] J.-P. Courtiat, C.A.S. Santos, C. Lohr, and B. Outtaj. Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique. *Computer Communications*, 2000.

[22] F. M. Cuenca-Acuna, R. P. Martin, and T. D. Nguyen. PlanetP : Using gossiping and random replication to support reliable peer-to-peer content search and retrieval. Technical Report DCS-TR-494, Department of Computer Science, Rutgers University, 2002.

[23] L. Dairaine, L. Lancérica, and J. Lacan. Enhancing peer to peer parallel data access with peerfect. In *Proceedings of the COST264 International Workshop on Networked Group Communications, NGC 2003*. LNCS, 2003.

[24] Laurent Dairaine, Jérôme Lacan, Laurent Lancérica, and Jérôme Fimes. Content-access qos in peer-to-peer networks using a fast mds erasure code. *Computer Communications (to appear)*, 2004.

[25] S. Dawkins, J. Griner D. Glover, D. Tran, T. Henderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, and J. Semke. *Ongoing TCP Research Related to Satellites*. Request for Comments 2760.

[26] S. Deering. *Host Extensions for IP Multicasting*, 1989. Request for Comments 1112.

[27] Differentiated services. IETF working group (concluded).

[28] Digital fountain - perfect communications over imperfect networks. http ://www.digitalfountain.com/index.cfm.

[29] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14 :78–88, 2000.

[30] Christophe Diot, Walid Dabbous, and Jon Crowcroft. Multipoint communication : A survey of protocols, functions and mechanisms. *IEEE Journal on Selected Area in Communication*, 15, April 1997.

[31] DIPCAST : DVB comme support d'IP multiCAST par satellite.
RNRT project No. 67. http ://www.dipcast-satellite.com/.

[32] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang. *A Link-Layer Tunneling Mechanism for Unidirectional Links*, 2001. Request for Comments 3077.

[33] *Digital Video Broadcasting (DVB) ; Interaction channel for satellite distribution systems ;*, 2003. European Telecommunications Standards Institute, EN 301 790.

[34] *Digital Video Broadcasting (DVB) : Framing structure, channel coding and modulation for 11/12 GHz satellite services*, 1997. European Telecommunications Standards Institute, EN 300 421.

[35] P. Elias. Coding for two noisy channels. *Information Theory, 3rd London Symp.*, 1955.

[36] Gorry Fairhurst and Bernhard Collini-Nocker. *Ultra Lightweight Encapsulation (ULE) for transmission of IP datagrams over MPEG-2/DVB networks*, May 2004. Draft Intended Standards Track.

[37] M. Ferrari and S. Bellini. Low BER turbo codes for satellite communication. In *Seventh International Workshop on Digital Signal Processing Techniques for Space Communications*, 2001.

[38] F. Filali and W. Dabbous. Issues on the ip multicast service behaviour over the next-generation of satellite-terrestrial hybrid networks. In *IEEE ISCC'2001*, 2001.

[39] Fethi Filali, Hitoshi Asaeda, and Walid Dabbous. Counting the number of members in multicast communication. In *Networked Group Communication*. ACM, 2002.

[40] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5, December 1997.

[41] Jeff Foerster and John Liebetreu. Fec performance of concatenated reed-solomon and convolutional coding with interleaving, 2000. Contribution to the IEEE 802.16 Broadband Wireless Access Working Group.

[42] XTP Forum. Xpress transport protocol specification, revision 4.0, march 1995.

[43] T. Friedman and D. Towsley. Multicast session membership size estimation. In *INFOCOM 99*, 1999.

[44] Timur Friedman. *Scalable Estimation of Multicast Session Characteristics*. PhD thesis, University of Massachusetts Amherst, 2002.

[45] R.G. Gallager. Low-density parity-check codes. *IRE Transaction Information Theory*, 1962.

[46] GÉANT. http ://www.dante.net/server/show/nav.007.

[47] Jim Gemmell. Scalable reliable multicast using erasure-correcting re-sends. Technical Report Report MSR-TR-97-20, Microsoft Research, 1997.

[48] GEOCAST : Multicast over Geostationary EHF Satellites.
IST project. http ://www.ee.surrey.ac.uk/CCSR/IST/Geocast/.

[49] S. Goel, M. Chai, D. Xu, and B. Li. Efficient peer-to-peer data dissemination in mobile ad-hoc networks. In *Proc. of International Workshop on Ad Hoc Networking*, 2002.

[50] Object Management Group. Unified modeling language.

[51] E. J. Gumbel. *The statistics of extremes*. New York : Columbia Univ. Press., 1958.

[52] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, and M. Luby. *The Reliable Multicast Design Space for Bulk Data Transfer*, 2000. Informational Request for Comments 2887.

[53] M. Handley and V. Jacobson. *SDP : Session Description Protocol*, 1998. Request for Comments 2327.

[54] Mark Handley and Jon Crowcroft. Network text editor (NTE) : A scalable shared text editor for the MBone. In *SIGCOMM 97*, pages 197–208, 1997.

[55] Mark Handley and Jon Crowcroft. Network text editor (nte) : A scalable shared text editor for the mbone. In *SIGCOMM*, pages 197–208, 1997.

[56] T. Henderson and S. Bhatti. Protocol-independent multicast pricing. In *The 10th International Workshop on Network and Operating System Support for Digital Audio and Video*, 2000.

[57] T. R. Henderson and R. H. Katz. Transport protocols for internet-compatible satellite networks. *IEEE journal on selected areas of communications*, 1999.

[58] M. Hofmann. A generic concept for large scale multicast, local group concept (LGC). In *International Zurich Seminar on Digital Communication (IZS'96)*. Springer Verlag, 1996.

[59] Markus Hofmann and Manfred Rohrmuller. Impact of virtual group structure on multicast performance. In *COST 237 Workshop*, pages 165–180, 1997.

[60] Hugh W. Holbrook, Sandeep K. Singhal, and David R. Cheriton. Log-based receiver-reliable multicast for distributed interactive simulation. In *Proc. of SIGCOMM 95*, 1995.

[61] Standard IEEE. *IEEE STD 802.3-2002*. see paragraph 3.2.8 for the CRC.

[62] Integrated services. IETF working group (concluded).

[63] IP over DVB working group. IETF working group.

[64] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr. Overcast : Reliable multicasting with an overlay network. In *Fourth Symposium on Operating System Design and Implementation (OSDI)*, pages 197–212, 2000.

[65] The Java Reliable Multicast Service (JRMS).
http ://research.sun.com/techrep/1998/abstract-68.html.

[66] Matthias Jung, Jorg Nonnenmacher, and Ernst Biersack. Reliable multicast via satellite : Uni-directional versus bi-directional communication. In *Kommunikation in Verteilten Systemen*, pages 264–275, 1999.

[67] A. Koifman and S. Zabele. RAMP : A reliable adaptive multicast protocol. In *IEEE Infocom'96*, March 1996.

[68] M.W. Koyabe and G. Fairhurst. Reliable multicast via satellite : A comparison survey and taxonomy. *International Journal of Satellite Communications (IJSC)*, 24(1), 2001.

[69] J. Lacan, L. Lancérica, and L. Dairaine. When FEC speed up data access in p2p networks. In *IDMS'02 Conference (Interactive Distributed Multimedia Systems)*, 2002.

[70] Martin S. Lacher, Jörg Nonnenmacher, and Ernst W. Biersack. Performance comparison of centralized versus distributed error recovery for reliable multicast. *Transaction on Networking*, 8 :224–239, 2000.

[71] L. Lancérica, L. Dairaine, and J. Lacan. Evaluation of content-acces qos for various dissemination strategies in peer to peer networks. In *11th IEEE International Conference on Networks ICON*, 2003.

[72] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson, and G. Fairhurst. *The Lightweight User Datagram Protocol (UDP-Lite)*, 2002. Request for Comments 3450, Standards Track.

[73] D. Li and D. Cheriton. OTERS (on-tree efficient recovery using subcasting) : A reliable multicast protocol. In *Proc. IEEE International Conference on Network Protocols (ICNP'98)*, 1998.

[74] Tie Liao. Light-weight reliable multicast protocol as an extension to rtp. Inria Rocquencourt, http ://webcanal.inria.fr/lrmp/lrmp_rtp.html.

[75] Ching-Gung Liu, Deborah Estrin, Scott Shenker, and Lixia Zhang. Local error recovery in SRM : comparison of two approaches. *IEEE/ACM Trans. Netw.*, 6(6) :686–699, 1998.

[76] Chuanhai Liu and Jorg Nonnenmacher. Broadcast audience estimation. In *INFOCOM (2)*, pages 952–960, 2000.

[77] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft. *Asynchronous Layered Coding (ALC) Protocol Instantiation*, 2002. Request for Comments 3450.

[78] M. Luby and V. Goyal. *Wave and Equation Based Rate Control (WEBRC) Building Block*, 2004. Request for Comments 3738.

[79] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft. *Forward Error Correction (FEC) Building Block*, 2002. Experimental Request for Comments 3452.

[80] Michael Luby. Lt codes. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, page 271. IEEE Computer Society, 2002.

[81] L.Vicisano. RMDP : an FEC-based reliable multicast protocol for wireless environements. *ACM Mobile Computer and Communication Review, v.2 n.2*, April 1998.

[82] A. Mark, C. Hayes, H. Kruse, and S. Ostermann. Tcp performance over satellite links. In *5th International Conference on Telecommunication Systems*, March 1997.

[83] MBONED : the Multicast Backbone Deployment. IETF working group.

[84] Izal Mikel, Urvoy-Keller Guillaume, Biersack Ernst W., Felber Pascal A., Al Hamra Anwar, and Garces-Erice Luis. Dissecting BitTorrent : five months in a torrent's lifetime. In *PAM'2004, 5th annual Passive & Active Measurement Workshop,*, 2004.

[85] K. Miller, K. Robertson, A. Tweedly, and M. White. Starburst multicast file transfer protocol (MFTP) specification.
http ://globecom.net/ietf/draft/draft-miller-mftp-spec-03.html, April 1998.

[86] Yutaka Miyake, Teruyuki Hasegawa, Toru Hasegawa, and Toshihiko Kato. Acceleration of tcp throughput over satellite-based internet acces using tcp gateway. In *the fifth IEEE Symposium Computers and Communications (ISC)*, 2000.

[87] Todd Montgomery, Brian Whetten, and John R. Callahan. The reliable multicast protocol specification flow control and NACK policy. http ://research.ivv.nasa.gov/RMP/Docs/RMPflow.txt, October 1995.

[88] Robert Morris. Bulk multicast transport protocol. In *INFOCOM 97*, 1997.

[89] *Digital Video Broadcasting (DVB); DVB specification for data broadcasting*, 2004. European Telecommunications Standards Institute, EN 301 192.

[90] Moving picture experts group (MPEG). ISO/IEC working group.

[91] Multicast security. IETF working group.

[92] NIST. *Secure Hash Standard*. National Institute of Standards and Technology - Federal Information Processing Standards Publication 180-1.

[93] J. Nonnenmacher and E. W. Biersack. Scalable feedback for large groups. *IEEE/ACM Transactions on Networking*, 1999.

[94] J. Nonnenmacher, M. Lacher, M. Jung, E. W. Biersack, and Georg Carle. How bad is reliable multicast without local recovery ? In *IEEE INFOCOM'98*, 1998.

[95] Jorg Nonnenmacher and Ernst Biersack. Reliable multicast : Where to use FEC. In *Protocols for High-Speed Networks*, pages 134–148, 1996.

[96] Jörg Nonnenmacher, Ernst W. Biersack, and Don Towsley. Parity-based loss recovery for reliable multicast transmission. *IEEE/ACM Trans. Netw.*, 6(4) :349–361, 1998.

[97] NORM, PROTEAN project. http ://norm.pf.itd.nrl.navy.mil/.

[98] K. Obraczka. Multicast transport protocols : a survey and taxonomy. *IEEE Communications Magazine*, 1998.

[99] O.Papini and J.Wolfmann. *Algèbre discrète et codes correcteurs*. Springer-Verlag, 1995. Collection Mathématiques et Applications, Vol.20.

[100] V. Paxson. End-to-end internet packet dynamics. In *Proc. ACM SIGCOMM*, September 1997.

[101] Jani Peltotalo and Sami Peltotalo. MAD/TUT, tampere university of technology project. www.atm.tut.fi/mad.

[102] Dimitrios Pendarakis, Sherlia Shi, Dinesh Verma, and Marcel Waldvogel. Almi : An application level multicast infrastructure. In *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, 2001.

[103] J. Pinder, L. J. Ippolito, and S. Horan. Four Years of Experimental Results from the New Mexico ACTS Propagation Terminal at 20.185 and 27.505 GHz. *IEEE on Selected Areas in Communication*, 17, 1999.

[104] J. S. Plank and M. G. Thomason. On the practical use of ldpc erasure codes for distributed storage applications. Technical Report CS-03-510, University of Tennessee, September 2003.

[105] J. Postel. *User Datagram Protocol*, 1980. Request for Comments 768.

[106] Jon Postel. *Internet Protocol*, 1981. Request for Comments 791.

[107] Jon Postel. *Transport Control Protocol*, 1981. Request for Comments 793.

[108] K. Ramakrishnan, S. Floyd, and D. Black. *The Addition of Explicit Congestion Notification (ECN) to IP*, 2001. Request for Comments 3450, Standards Track.

[109] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *In Proc. ACM SIGCOMM, San Diego, USA*, 2001.

[110] RENATER - le réseau national de télécommunications pour la technologie, l'enseignement et la recherche. http ://www.renater.fr/.

[111] Sean Rhea, Chris Wells, Patrick Eaton, Dennis Geels, Ben Zhao, Hakim Weatherspoon, and John Kubiatowicz. Maintenace-free global data storage. *IEEE Internet Computing*, 5, 2001.

[112] R. Rivest. *The MD5 Message-Digest Algorithm*, 1992. Request for Comments 1321.

[113] L. Rizzo. The feasibility of software FEC. Technical report, DEIT LR-970131, 1997.

[114] Luigi Rizzo. Effective erasure codes for reliable computer communication protocols. *SIGCOMM Comput. Commun. Rev.*, 27(2) :24–36, 1997.

[115] Luigi Rizzo, Gianluca Iannaccone, Lorenzo Vicisano, and Mark Handley and. *PGMCC single rate multicast congestion control : Protocol Specification*, 2004. Internet draft, expire en janvier 2005.

[116] Luigi Rizzo and Lorenzo Vicisano. A reliable multicast data distribution protocol based on software FEC techniques. In *Proc. of the Fourth IEEE HPCS'97 Workshop*, 1997.

[117] Reliable Multicast Transport working group. IETF working group.

[118] V. Roca and B. Mordelet. Design of a multicast file transfer tool on top of alc. In *7th IEEE Symposium on Computers and Communications (ISCC'02)*, 2002.

[119] Luis Martin Rojas-Cardenas. *Architecture de Transport à Ordre et Fiabilité Partiels pour les applications multimédias adaptatives à temps contraint*. PhD thesis, Thèse de doctorat de l'Université Paul Sabatier de Toulouse, ENSICA, Novembre 2000.

[120] Real-Time LOTOS Laboratory. http ://www.laas.fr/RT-LOTOS/.

[121] Paul Sanjoy, Krishan K. Sabnani, john C.-H. Lin, and Supratik Bhattacharyya. Reliable multicast transport protocol (rmtp). *IEEE journal on selected areas in communications*, 15, April 1997.

[122] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *RTP : A Transport Protocol for Real-Time Applications*, 1996. Request for Comments 1889.

[123] Rob Sherwood, Ryan Braud, and Bobby Bhattacharjee. Slurpie : A cooperative bulk data transfer protocol. In *INFOCOM 2004*, 2004.

[124] M. A. Shokrollahi. Raptor codes. In *To appear in the proceedings of ISIT 2004*, 2004.

[125] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano. *PGM Reliable Transport Protocol Specification*, 2001. Experimental Request for Comments 3208.

[126] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160. ACM Press, 2001.

[127] Eiji Takahashi, Takaaki Ohara, Takumi Miyoshi, and Yoshiaki Tanaka. Cost-based pricing for multicast streaming services. In *10th International Telecommunication Network Planning Symposium (NETWORKS2002)*, 2002.

[128] T. Takahashi, M. Tammi, H. Vatiainen, R. Lehtonen, and J. Harju. Implementation and performance evaluation of multicast control protocol. In *The 9 IEEE Symposium on Computers and Communicatins (ISCC'2004)*. IEEE Press, 2004.

[129] Takeshi Takahashi and Miikka Tammi. MCOP/TUT, multicast control protocol tampere university of technology project. http ://atm.tut.fi/mcop/.

[130] Cheng Kok Tan. Reliable ip multicast services over satellite links. In *IEEE International Conference on Networks (ICON'00)*, 2000.

[131] TCP-XM : Multicast transport for grid computing, university of cambridge computer laboratory. http ://www.cl.cam.ac.uk/users/kj234, 2004.

[132] Francesco Tommasi, Simone Molendini, and Antonio Vilei. The satellite multicast distribution protocol (SMDP). In *Proceedings of SoftCOM 2002*, October 2002.

[133] Don Towsley, Jim Kurose, and Shridhar Pingali. A comparison of sender-initiated and receveir-initiated reliable multicast protocols. *IEE journal on selected areas in communications*, 15, April 1997.

[134] TTool : the TURTLE Toolkit.
http ://www.eurecom.fr/ apvrille/TURTLE/index.html.

[135] UDcast - full IP over broadcast media. http ://www.udcast.com/.

[136] Unidirectional Link Routing working group. IETF working group.

[137] The internet engineering task force working groups.

[138] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, and M. Luby. *Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer*, 2001. Informational Request for Comments 3048.

[139] Joerg Widmer and Mark Handley. *TCP-Friendly Multicast Congestion Control (TFMCC) :Protocol Specification*, 2004. Internet draft, expire en janvier 2005.

[140] X. Rex Xu, Andrew C. Myers, Hui Zhang, and Raj Yavatkar. Resilient multicast support for continuous-media applications. In *Proceedings of NOSSDAV'97*, 1997.

[141] M. Yajnik, J. Kurose, and D. Towsley. Packet loss correlation in the mbone multicast networ. In *Global Internet Conference*, November 1996.

[142] R. Yavatkar, J. Griffioen, and M. Sudan. A reliable dissemination protocol for interactive collaborative applications. In *Proceedings of the ACM Multimedia '95 Conference*, November 1995.

[143] Kenneth P. Birman Zhen Xiao. A randomized error recovery algorithm for reliable multicast. In *Proceedings of IEEE Infocom*, April 2001.

TITLE : Large scale reliable multipoint transport : taking cost into account in the hybrid satellite terrestrial Internet environment

SUMMARY

This thesis studies issues related to the proposition of large scale reliable multipoint communication services. In this context, the possibility to use a geostationary satellite, emitting in the Ka band, to deploy such a service is analysed. However, the use of the Ka band introduces a high variability of quality of reception. Thus, the use of a transport protocol, implementing specific mechanisms, is mandatory.

According to a cost function, the comparison of classical solutions, based on IP Multicast, show that a hybrid approach which uses the terrestrial and the satellite networks is advantageous. Consequently, a protocol named Hybrid Satellite Terrestrial Reliable Multicast is proposed. Its principle consists of choosing, depending on the group size, the more profitable network (i.e. terrestrial or satellite network) to transmit information. This choice is made according to a predefined cost function. A sharp description of the proposition, including the hosts' behaviours and the message set-up, is depicted.

In spite of the simplicity of the approach, several obstacles appear when one tries to design appropriate mechanisms. These issues include reliability (use of forward error correction), large group size estimation, and terrestrial error recovery (use of peer-to-peer networks). Those mechanisms are studied separately to determine satisfactory configurations, and to detect performance issues.

After the definition of those mechanisms, the proposition is globally modelized in order to start the formal validation of the proposed service. The model is realized using the real-time UML profile TURTLE, and the validation results are obtained thanks to the TTool-RTL toolkit, and to Aldebaran.

RESUME

Le travail effectué aborde la problématique des services de communication multi-points fiables à grande échelle. Dans ce contexte, la possibilité de déployer un tel service au moyen d'un satellite géostationnaire émettant en bande Ka est étudiée. L'emploi de la bande Ka introduit cependant une grande variabilité de la qualité de réception au niveau des utilisateurs finals, rendant nécessaire l'utilisation d'un protocole de transport mettant en oeuvre des mécanismes spécifiques.

Selon une fonction de coût définie, la comparaison des solutions basées sur IP Multicast classiquement utilisées montre que l'utilisation d'une approche hybride couplant l'utilisation des réseaux satellites et terrestres est avantageuse. Le principe de la proposition, nommée Hybrid Satellite Terrestrial Reliable Multicast, consiste ainsi à choisir, en fonction de la taille du groupe, le moyen de diffusion le plus rentable — au vu d'une fonction de coût définie. Une description détaillée de la proposition inclut le comportement de la source et des récepteurs, et le format des messages échangés.

Bien que le principe de cette approche soit simple, plusieurs points durs sont liés à la conception des mécanismes adéquats. Ces problèmes concernent notamment la gestion de la fiabilité (utilisation de code correcteur d'erreur ou FEC), l'estimation de taille de très grands groupes, et la reprise des erreurs par voie terrestre (utilisation de réseaux de pair-à-pairs). Ces mécanismes sont étudiés de manière unitaire afin de déterminer des configurations satisfaisantes, et pour détecter des problèmes de performances.

Ces mécanismes étant définis, la proposition de transport a été globalement modélisée, de manière à obtenir une vérification fonctionnelle du service proposé. Le protocole a été décrit au moyen du profil UML temps réel TURTLE. Les résultats de validation ont été obtenus grâce à la chaîne d'outils TTool-RTL, et à CADP.

MOTS-CLES : IP Multicast, Satellite, Transport, FEC, Estimation, Pair-à-pair.

DISCIPLINE : Réseaux et télécommunications

INTITULE ET ADRESSE DU LABORATOIRE : TéSA, 2, rue Charles Camichel F-31071 Toulouse, Cedex 7, BP 7122 FRANCE