



HAL
open science

Architecture de communication multimédia et multi-réseaux

Pascal Berthou

► **To cite this version:**

Pascal Berthou. Architecture de communication multimédia et multi-réseaux. Réseaux et télécommunications [cs.NI]. Institut National Polytechnique de Toulouse - INPT, 2001. Français. NNT : . tel-00131785

HAL Id: tel-00131785

<https://theses.hal.science/tel-00131785>

Submitted on 19 Feb 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DOCTORAT DE L'I.N.P.T.

Spécialité : **INFORMATIQUE ET TELECOMMUNICATIONS**

BERTHOU Pascal

Laboratoire : **Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)**

Titre de la thèse : **Architecture de communication multimédia et multi-réseaux**

Soutenu le :

**13 DECEMBRE 2001
LAAS-CNRS**

Directeur de recherche :

Michel Diaz

JURY :

M.	F. Lepage	Rapporteur
M.	G. Pujolle	Rapporteur
M.	O. Festor	Examineur
M.	C. Fraboul	Examineur
M.	T. Gayraud	Co encadrement
M.	D. Genty	Examineur

MOTS CLES :

Architecture de Communication
Multimédia
Multi-réseaux
Transport à Ordre Partiel

Réseaux Actifs
Qualité de Service
Modèle de Flux

RESUME :

Architecture de communication Multimédia et Multi-réseaux

Résumé :

Les progrès récents dans le domaine des technologies de communications ont ouvert la voie à de nouveaux réseaux et les applications multimédias distribuées se sont développées. Elles ont pour principales caractéristiques des débits élevés, des flux continus avec de fortes contraintes temporelles, et des besoins en synchronisation pour préserver la cohérence de leur présentation. Les réseaux de communication actuels ne pouvant tout offrir à la fois (hauts débits, qualité et respect des contraintes temporelles), la multiplicité des supports de communication et des services permet d'envisager une nouvelle voie : le choix du support de communication en fonction des contraintes de l'application. Si cette multiplicité possède des attraits, elle recèle aussi de nombreuses difficultés qu'il faut maîtriser comme l'hétérogénéité des services de bout en bout.

Ceci nous a conduit à concevoir et à réaliser une nouvelle architecture de communication « *multi-réseaux* » prenant en compte les besoins spécifiques des applications multimédias. D'une part, lorsque plusieurs réseaux sont accessibles, un protocole permettant de sélectionner la meilleure interface de communication en fonction des besoins spécifiques des flux multimédias a été défini. Afin de préserver la synchronisation multimédia, des mécanismes d'ordres partiels sont intégrés au protocole. D'autre part, lorsque des domaines très hétérogènes doivent être traversés par un flux, une solution qui permet de préserver la qualité de service des protocoles de transport est proposée. Elle est basée sur le principe de la rupture des connexions transport par des proxys situés en bordure de domaine.

Un modèle de flux associant un langage de spécification des flux multimédias à des mécanismes de traitement de ces flux par le protocole est proposé pour formaliser les spécifications de qualité de service. Enfin, cette thèse a contribué à développer des mécanismes de déploiement dynamique de protocoles de transport.

Mots-clés :

Architecture de Communication	Réseaux Actifs
Multimédia	Qualité de Service
Multi-réseaux	Modèle de Flux
Transport à Ordre Partiel	

Multimedia Multi-network Communication Architecture

Abstract :

Recent progress in communication technologies leads to new networks architectures and new developments of distributed multimedia applications. Multimedia data are characterised by high throughputs, continuous flows with temporal constraints and synchronisation requirements to ensure their presentation coherence. Actual communication networks can not provide all of these services together (high throughputs, quality and temporal constraints respect). But communication supports and services multiplicity open a new way of thinking : the choice of a communication support depending on the application constraints. If this multiplicity provides some positive aspects, several difficulties have to be coped, as end-to-end services heterogeneity.

This leads us to conceive and realise a new "multi-network" communication architecture which takes into account the specific needs of multimedia applications. On one hand, when several networks interfaces are available, a protocol allowing to select the best communication interface for a multimedia flow is defined. To preserve multimedia synchronisation, partial order mechanisms are integrated to the protocol. On the other hand, when a data flow have to cross heterogeneous network domains, a solution preserving the quality of service provided by transport protocols is proposed. This middleware solution is based on the interruption of transport level connections by proxies located on network domain borders.

A flow model linking a language dedicated to multimedia flows specification with protocol mechanisms for flows recognition is proposed to formalise the quality of service. Finally, this thesis contributes to develop mechanisms for dynamic transport protocol deployment.

Keywords :

Communication Architecture	Active Networks
Multimedia	Quality of Service
Multi-network	Stream Model
Partial Order Transport	

Table des matières

INTRODUCTION GENERALE.....	1
CONTEXTE.....	2
PLAN DE LA THESE.....	2
CHAPITRE I - LES APPLICATIONS MULTIMEDIAS DISTRIBUEES ET LES NOUVELLES TECHNOLOGIES DE COMMUNICATION	5
1 INTRODUCTION	6
2 LES APPLICATIONS MULTIMEDIAS	6
2.1 <i>Informations multimédias : caractéristiques et besoins</i>	6
2.2 <i>La synchronisation Multimédia</i>	8
3 ETAT DE L'ART SUR LES OFFRES DE RESEAUX	9
3.1 <i>Introduction</i>	9
3.2 <i>Critères de QoS</i>	9
3.3 <i>Réseaux filaires</i>	10
3.4 <i>Réseaux Hertziens par satellites</i>	13
3.5 <i>Réseaux Hertziens terrestres</i>	15
3.6 <i>Bilan</i>	18
4 LES RESEAUX ACTIFS	18
4.1 <i>Introduction</i>	18
4.2 <i>Méthodologie</i>	19
4.3 <i>Architecture</i>	19
4.4 <i>Plates-formes</i>	20
4.5 <i>Le « End-To-End Argument » et les réseaux actifs</i>	23
5 LA PROBLEMATIQUE DU MULTI-RESEAUX.....	23
CHAPITRE II - MULTI-RESEAUX PARALLELE	27
1 INTRODUCTION	28
2 ETAT DE L'ART DES PROTOCOLES MULTI-RESEAUX.....	29
2.1 <i>User Service Assistant (U.S.A.)</i>	29
2.2 <i>Composite Radio System – CRS</i>	30
2.3 <i>« Internet par satellite »</i>	33
3 PROBLEMATIQUE ET BESOINS DU PROTOCOLE.....	35
3.1 <i>Les mécanismes de sélection du réseau</i>	35
3.2 <i>Adaptation du protocole de transport au réseau sous-jacent</i>	36
3.3 <i>Les besoins en synchronisation</i>	38
3.4 <i>Les besoins en fiabilité partielle</i>	41
4 L'ARCHITECTURE DU PROTOCOLE.....	43
4.1 <i>Choix d'architecture</i>	43
4.2 <i>Les mécanismes de sélection du réseau</i>	43
4.3 <i>Adaptation du protocole de transport au réseau sous-jacent</i>	47
4.4 <i>Mécanisme de synchronisation</i>	49
4.5 <i>Le déploiement dynamique</i>	54
5 MESURES DE PERFORMANCES	56
5.1 <i>Mesures des performances observées par l'application</i>	57
5.2 <i>Mesures des performances du protocole MMPOC-MN</i>	58
5.3 <i>Bilan</i>	63
6 CONCLUSION	64

CHAPITRE III - SPECIFICATION DES CONTRAINTES D'ORDRE ET DE FIABILITE.....	65
1 INTRODUCTION & LES BESOINS	66
2 SPECIFICATION DE LA QOS DU SERVICE POC	67
2.1 <i>Notion de flux</i>	68
2.2 <i>Spécification de l'ordonnancement</i>	68
2.3 <i>Spécification de la fiabilité</i>	69
2.4 <i>Exemples</i>	71
3 MODELE DE FLUX	73
3.1 <i>Introduction</i>	73
3.2 <i>Concepts formels</i>	73
3.3 <i>Flux monomédia</i>	79
3.4 <i>Les flux multimédias</i>	83
3.5 <i>Conclusion</i>	85
4 EXEMPLES DE SPECIFICATION	86
4.1 <i>Modélisation d'un flux MPEG</i>	87
4.2 <i>Visioconférence MPEG</i>	94
5 CONCLUSION	96
CHAPITRE IV - MULTI-RESEAUX SERIE	97
1 INTRODUCTION	98
2 EXEMPLES DE SOLUTIONS MULTI-RESEAUX SERIE	99
2.1 <i>Le splitting</i>	99
2.2 <i>Le spoofing</i>	101
2.3 <i>WAP 2.0</i>	102
2.4 <i>Mécanismes de caches</i>	103
3 MNP : PROTOCOLE MULTI-RESEAUX SERIE	104
3.1 <i>Justification de l'architecture</i>	104
3.2 <i>Architecture</i>	105
3.3 <i>Couche MNP</i>	106
3.4 <i>Couche MMPOC-MN</i>	110
4 DEPLOIEMENT DE PROTOCOLE	112
4.1 <i>Déploiement manuel</i>	113
4.2 <i>Déploiement automatique</i>	114
5 PROBLEMES LIES A L'UTILISATION DE PROXYS	118
6 MESURES	119
6.1 <i>Influence de l'adaptation du transport au réseau</i>	120
6.2 <i>Influence de l'ordre et de la fiabilité partielle</i>	121
7 CONCLUSION	123
CHAPITRE V - IMPLEMENTATION.....	125
1 INTRODUCTION	126
2 INTERFACES DE PROGRAMMATION	126
2.1 <i>API du langage de flux</i>	126
2.2 <i>API du protocole</i>	130
3 PERFORMANCES DE L'IMPLEMENTATION JAVA	131
4 BILAN	132
CONCLUSION GENERALE.....	135
RAPPEL DES CONTRIBUTIONS	135
<i>Architecture de communication multi-réseaux</i>	135
<i>Modèle de flux</i>	136
<i>Déploiement dynamique de protocoles de communication</i>	136
PERSPECTIVES	136
BIBLIOGRAPHIE DE L'AUTEUR.....	139
BIBLIOGRAPHIE	140

Introduction générale

Ces dernières années ont vu l'émergence de nouvelles technologies de communication, plus rapides, de meilleure qualité, et moins chères. Elles ne sont plus seulement réservées à une élite depuis leur lieu de travail, mais sont accessibles par tous et tout le temps : à la maison, dans le train, dans la rue, etc. Un besoin de connectivité permanente est en train de naître. Pour satisfaire à cette nouvelle demande, les traditionnels réseaux câblés cèdent du terrain face au déploiement massif de la fibre optique dans les cœurs des réseaux. Que ce soit sous la forme de bornes dans les lieux publics, d'antennes à forte puissance, de satellites, de constellations de satellites, les technologies d'accès sans fil à ces réseaux de communication se multiplient.

Grâce à l'évolution concomitante des ordinateurs et à l'apparition de nouveaux périphériques à faibles coûts, tels que les microphones et les caméras numériques, les applications changent. Elles s'intègrent désormais dans une société où les médias priment. Que ce soit dans le domaine du divertissement pour la vidéo à la demande, dans le domaine professionnel pour les environnements de téléformation et de travail coopératif ou simplement dans le cadre familial avec la visioconférence, les applications de demain généraliseront l'utilisation des médias tels que l'audio et la vidéo. Le nom générique d'applications multimédias distribuées est donné à ces applications. Elles ont pour principales caractéristiques des débits élevés, des flux continus qui ont de fortes contraintes temporelles, et des besoins en synchronisation pour préserver la cohérence de leur présentation. C'est à ce type d'application que nous nous intéresserons plus particulièrement.

Ces contraintes sont bien souvent impossibles ou très difficiles à satisfaire ensemble et ceci d'autant plus que tous les médias d'une même application n'ont pas les mêmes contraintes. A l'instar des réseaux téléphoniques et des réseaux de données, un flux audio nécessite de très fortes contraintes temporelles et génère un faible débit, alors qu'inversement, un flux de données génère un fort débit sans réelle notion de temps. Entre ces deux extrêmes se situent les flux vidéo. De plus, les réseaux de communication actuels ne peuvent tout offrir à la fois : hauts débits, qualité et respect des contraintes temporelles. L'attitude face à cette situation conflictuelle consiste le plus souvent à accepter un compromis de façon à relâcher certaines de ces contraintes pour adapter globalement une application multimédia au type de réseau qu'elle emprunte.

S'il n'existe pas aujourd'hui un réseau qui offre tous ces services, la multiplicité des supports de communications et par conséquent des services, permet d'envisager une nouvelle voie : le choix du support de communication en fonction des contraintes de l'application. Nous verrons toutefois que si cette multiplicité possède des attraits, elle recèle aussi de nombreuses difficultés qu'il faudra maîtriser.

Cette constatation nous a conduit à concevoir et à réaliser une nouvelle architecture de communication prenant en compte les besoins spécifiques des applications multimédias. Cette architecture est basée sur le concept de « *multi-réseaux* ». Nous allons, dans ce mémoire, détailler cette architecture.

Contexte

Cette thèse s'inscrit dans l'ensemble des études menées au sein de deux projets : **GCAP** et **Amarrage**.

Le premier, **GCAP** (Global Communication Architecture and Protocols for new QoS services over IPv6 networks) [GCAP99] a pour but de développer un nouveau protocole de transport multimédia multicast, associé à une nouvelle architecture offrant la garantie de qualité de service aux applications multimédias multipoints multi-réseaux. Nous verrons que cette thèse apporte une contribution aux tâches WP2-3 (Spécification du protocole multimédia multipoints multi-réseaux) et WP3-4 (Principes d'implémentation du protocole multimédia multipoints multi-réseaux).

Le second, **Amarrage** (Architecture Multimédia & Administration Répartie sur un Réseau Actif à Grande Echelle) [AMA99], expérimente des technologies de réseaux actifs qui permettent une personnalisation fine du service rendu par le réseau en fonction de la nature des flux transportés. Notre contribution porte principalement sur le développement de protocoles multimédias actifs.

Les verrous levés dans cette thèse sont :

- L'adaptation des applications multimédias à la mixité des réseaux actuels. Des solutions sont proposées pour, d'une part profiter de la diversité des services réseaux et d'autre part optimiser le comportement des protocoles de transport sur les interconnexions de réseaux fortement hétérogènes.
- Le déploiement de protocoles : La diffusion de nouveaux protocoles de transport, même s'ils apportent un réel gain de performances, reste confidentielle. Des mécanismes permettant leur utilisation à grande échelle doivent être développés.
- Le développement de nouvelles méthodes de spécification de la qualité de service (QoS) : Celles ci restent aujourd'hui trop abstraites pour être mises en œuvre par les développeurs d'applications. D'autres méthodes plus concrètes doivent être proposées.

Plan de la thèse

Tout d'abord le premier chapitre expose la double problématique associée à nos travaux, à savoir :

- la problématique liée à la manipulation et au traitement de données multimédias qui nécessitent de concevoir des mécanismes pour gérer la qualité de service et pour garantir les contraintes de synchronisation.
- la problématique soulevée par l'émergence de nouvelles technologies de communication offrant des qualités de services très disparates qui nécessitent de ne plus considérer, comme cela était fait auparavant, les réseaux comme étant des entités homogènes de bout en bout.

A cela se rajoutent de nouvelles possibilités de gestion des réseaux rendues possibles par l'utilisation de réseaux dits actifs qui perturbent d'autant plus la vision traditionnelle des réseaux. Dans cet état de l'art se profile la problématique que nous appelons « multi-réseaux ». Ce premier chapitre se termine par la présentation de celle ci. Deux classes de solutions y sont clairement identifiées : le multi-réseaux parallèle et le multi-réseaux série. Ces deux solutions sont les principaux challenges que cette thèse se propose de résoudre.

Le second chapitre détaille la solution permettant de tirer profit des configurations multi-réseaux parallèle, c'est à dire dans lesquelles le moyen de communication entre deux utilisateurs n'est pas unique. Le protocole que nous avons défini se propose d'associer les différents flux émanant d'une application multimédia distribuée aux réseaux sous-jacents les plus appropriés. Ce protocole repose sur l'adaptation de la qualité de service requise par ces applications aux différentes qualités de services offertes par l'accès à de multiples réseaux indépendants. Afin de garantir les contraintes de synchronisation imposées par le multimédia, les principes d'un transport à ordre partiel sont mis en œuvre. Nous verrons comment les mécanismes de fiabilité partielle et de notification des pertes peuvent être avantageusement utilisés.

Enfin, une nouvelle approche de déploiement dynamique de protocoles, reposant sur des principes de mobilité de code, est mise en œuvre pour le déploiement du protocole multi-réseaux.

Le troisième chapitre met en avant les insuffisances des méthodes actuelles de spécification de la qualité de service pour les protocoles à ordre partiel. Les limites des méthodes reposant sur les formalismes de réseaux de Pétri sont détaillées. Nous proposons, en remplacement, un modèle de flux associant un langage de spécification des flux multimédias à des mécanismes de traitement de ces flux par le protocole. Le langage permet la description de la qualité de service requise par un flux multimédia en termes de fiabilité et de synchronisation. Les traitements sont exprimés sous la forme d'automates, à intégrer dans le protocole, qui permettent de garantir les contraintes spécifiées par le langage.

Une série d'exemples détaillera l'utilisation de ce formalisme pour la spécification d'applications multimédias distribuées, dont une application de visioconférence MPEG.

Le chapitre 4 se consacre au deuxième volet de la problématique multi-réseaux qui est celle du multi-réseaux série. Lorsque des réseaux à caractéristiques très hétérogènes sont traversés par un flux, il est fréquent que le protocole de transport qui assure la qualité du flux soit mis en défaut. Pour pallier à ce problème, nous proposons une architecture de communication basée sur la rupture des connexions transport. Sur chaque portion homogène est déployée une connexion transport adaptée au réseau traversé. Une couche logicielle supplémentaire est utilisée pour garantir la sémantique de bout en bout.

Le déploiement dans le réseau de cette architecture est étudié et une solution innovante basée sur le nouveau paradigme des réseaux actifs est proposée.

Ces mécanismes ainsi que l'évaluation des performances de l'architecture sont décrits dans ce chapitre.

Enfin, le cinquième et dernier chapitre présente les principes de l'implémentation réalisée. D'une part, une interface de programmation déduite du langage de spécification de la qualité de service y est détaillée. D'autre part, les mesures de performance du protocole multi-réseaux sont présentées et discutées.

Chapitre I

Les Applications Multimédias Distribuées et les Nouvelles Technologies de Communication

1 Introduction

La vision de l'informatique distribuée a beaucoup évolué depuis les 10 dernières années avec l'apparition de nouvelles applications et de nouveaux types de réseaux.

Les applications multimédias distribuées, beaucoup plus riches, qui utilisent conjointement différents types de données (textuelles, sonores, graphiques, ...), représentent la principale évolution dans le domaine des applications.

Mais leur développement n'aurait pas été rendu possible, sans une profonde évolution des supports de communication dans leurs caractéristiques physiques ainsi que dans leur administration :

- De nombreux réseaux sont apparus avec des techniques particulières, certains étant spécialisés dans le haut débit, d'autres dans l'offre de services évolués comme la mobilité ou la diffusion, etc.
- Pour faciliter l'administration de ces réseaux très évolués, des concepts novateurs qui améliorent le déploiement de nouveaux services ont vu le jour. Les réseaux actifs en sont un exemple abouti.

Toutefois, il résulte de toutes ces évolutions une très (trop) grande diversité des applications et des supports de communication. La coexistence de ces nouvelles technologies devient très difficile. Aussi, l'objectif de ce chapitre est de dégager la problématique inhérente au déploiement des applications multimédias distribuées sur ces nouveaux supports de communication.

Ce chapitre est structuré en quatre parties :

Tout d'abord, comme introduit ci-dessus, la problématique est double puisqu'elle intègre les aspects émanant de l'évolution des applications et des réseaux. Aussi dans la première partie, les caractéristiques et les exigences du multimédia seront décrites.

La seconde partie fera un point sur les différentes évolutions des supports de communication. La troisième partie présentera les nouvelles perspectives en matière de gestion des réseaux amenées par le concept de réseaux programmables.

Enfin, la dernière partie présentera une vue qui clarifie les problèmes soulevés par l'évolution conjointe des applications multimédias et des nombreux supports de communication. Ce dernier point est ce que nous appellerons « problématique multi-réseaux ».

2 Les applications multimédias

2.1 Informations multimédias : caractéristiques et besoins

2.1.1 Notion de flux, flux continus, flux discrets

La première caractéristique qui distingue les données multimédias des données informatiques classiques (texte, données binaires, ...) est leur unité de traitement : les données multimédias se manipulent par *flux*, alors que les textes et les données binaires se manipulent par fichiers. En effet, les données audio et vidéo sont des séquences d'images ou d'échantillons sonores qui se succèdent à une cadence constante ou non. Ces données ne sont pas incompatibles avec

la notion de fichier informatique et les fichiers sont toujours utilisés pour stocker des films ou des documents sonores ; mais les traitements se font unité d'information par unité d'information (image par image, par exemple) ; ces unités d'information, mises bout à bout, forment un flux. Il faut noter toutefois que les textes graphiques et données binaires s'accrochent de ce mode de traitement.

Cependant, les flux se caractérisent aussi par les relations temporelles qui existent entre les différentes unités d'information qui les composent. Par exemple, il n'existe pas de relation temporelle entre les caractères qui composent un flux textuel. De même, pour une image unique qui peut être vue comme un flux de bits, il n'existe aucune relation temporelle entre les différentes unités du flux. Ce sont typiquement des *flux discrets*.

Par contre, pour la vidéo ou le son, les images ou les échantillons sonores doivent être produits, traités ou présentés avec une cadence régulière. On parle dans ce cas là de *flux continus*. Si le temps entre deux unités de base d'un flux est constant, on parle de flux *isochrone* ; toutefois, une certaine variabilité sur ces temps peut être tolérée : cette variabilité est appelée *gigue autorisée*.

2.1.2 Notion de qualité de service

Dans la présentation de la notion de flux qui vient d'être faite sont déjà apparues des notions de **contraintes temporelles**. Ces contraintes sont de deux ordres : le *décal* et la *gigue*.

Du décal de bout en bout, qui est le temps qui s'écoule entre l'émission d'une donnée et sa présentation, dépend l'interactivité de l'application. Il est communément admis [VEG96] que pour un dialogue agréable, un décal maximal de 400 ms est acceptable. Pour un dialogue confortable, 250 ms sont conseillés. Bien sûr, jusqu'à 600 ms, la discussion reste possible mais au prix d'une certaine tolérance.

En fait, les flux multimédias se caractérisent par leur qualité de service (QoS en Anglais). Une des familles de paramètres de qualité de service concerne la **qualité de restitution** des flux multimédias.

Par exemple, les données binaires ou textuelles ne tolèrent aucune perte (elles nécessitent une *fiabilité* totale). De manière identique, les graphiques supportent très mal les pertes ou les erreurs. En revanche, la vidéo animée est le médium le moins contraignant en terme d'erreur et de perte : la perte d'une image n'est pas perceptible par l'utilisateur terminal.

Finalement, nous appellerons **contraintes de transmission** les paramètres qui définissent la qualité des flux émis. Parmi cette famille de paramètres, le plus important est le *débit* d'informations générées par flux. Le débit d'un flux vidéo peut varier de quelques dizaines de kb/s pour une vidéo de faible qualité (40 kb/s pour un format QCIF¹), à plusieurs dizaines de Mb/s pour une vidéo haute définition. D'ailleurs, des efforts importants ont été effectués dans le secteur de la compression vidéo avec les algorithmes H261, MPEG, etc. Toutefois, les flux vidéo, même compressés, restent extrêmement gourmands en place mémoire et en bande passante.

La Table I-1 donne des exemples de contraintes de qualité de service communément admises pour les principaux flux d'une application multimédia.

¹ Quarter Common Intermediate Format : Recommandation H.261 ITU-T

	Délai de bout en bout	Gigue tolérée	Fiabilité requise	Bande passante
Flux audio	< 400 ms	< 20 ms	> 90 %	Jusqu'à 128 kbit/s pour une qualité CD
Flux vidéo	< 400 ms	< 20 ms	> 80 %	Dépend de la qualité demandée
Images fixe	Pas de contrainte	Grande	> 80 %	minimum
Flux de données	Pas de contrainte	Grande	100 %	minimum

Table I-1 - Qualité de Service des principaux flux multimédias

Ce paragraphe vient de donner un petit inventaire des contraintes de qualité de service des différents médias. Même si cet inventaire n'est pas exhaustif, et si chaque application possède ses propres besoins, les principales familles de critères de qualité de service des applications multimédias ont été mises en évidence.

2.2 La synchronisation Multimédia

La partie précédente vient de soulever la notion de QoS multimédia et l'a illustrée en présentant les principaux paramètres de qualité de présentation associés aux flux multimédias. Si les contraintes temporelles sont primordiales pour la définition d'un flux, d'autres contraintes apparaissent lors de la présentation de plusieurs flux. Ces contraintes sont de plusieurs ordres et expriment les relations spatiales et temporelles existant entre ces flux. Le nom de contraintes de synchronisation multimédia est généralement employé.

2.2.1 Synchronisation spatiale

La synchronisation spatiale exprime les contraintes d'ordonnement des objets multimédias lors de leur présentation. Elle définit par exemple la position, la taille ou la superposition des zones de présentation des objets.

2.2.2 Synchronisation temporelle

La synchronisation temporelle constitue l'un des points les plus délicats dans la gestion des systèmes multimédias. Elle consiste à définir les contraintes temporelles qui existent entre les différents objets multimédias. Elle se présente sous deux formes :

- La *synchronisation intra-flux* spécifie les contraintes temporelles existant entre les différentes unités d'information d'un même flux. Elle consiste à assurer leur présentation selon la cadence à laquelle ces données ont été produites (par exemple, à afficher sur l'écran les images d'une vidéo au même rythme que celui auquel la capture a été effectuée). Ce type de synchronisation consiste principalement à gommer la *gigue* accumulée par les données au cours de la traversée du réseau.
- La *synchronisation inter-flux* spécifie les contraintes temporelles existant entre deux flux (ou plus). Son objectif est d'assurer la cohésion de présentation de plusieurs flux, en supprimant un éventuel décalage qui pourrait être induit par la gestion séparée des flux dans le réseau et le système. Par exemple, lorsqu'il s'agit d'assurer la synchronisation

d'un flux vidéo et d'un flux audio, c'est à dire quand il faut assurer qu'un son émis correspond au mouvement des lèvres, on parle souvent de *lip synchronisation*.

Ce type de synchronisation a un impact psychologique très important car il est difficile de suivre une présentation dont l'image et le son sont décalés, alors que, séparément, ces flux sont parfaitement compréhensibles.

Cette synchronisation temporelle peut être garantie de plusieurs manières :

- Lorsque cela est nécessaire en rattrapant le décalage devenu trop important, en supprimant la présentation de quelques unités de données. On parlera alors de *synchronisation discrète*.
- Régulièrement, en introduisant à intervalle constant des points de synchronisation entre les flux. On parle de *synchronisation continue*.

2.2.3 Synchronisation hypermédia

Ce type de synchronisation exprime les contraintes d'ordonnement des objets multimédias dans le temps. Les objets sont liés entre eux par des liens hypermédias. Ces liens hypermédias permettent de décrire des informations sémantiques sous forme de relations entre éléments de documents qui ne sont pas des relations hiérarchiques. La synchronisation hypermédia ajoute des notions d'interactivité à la présentation. Ce type de synchronisation ne sera pas abordé dans cette thèse.

3 Etat de l'art sur les offres de réseaux

3.1 Introduction

Après avoir passé en revue les caractéristiques et les besoins des applications multimédias distribuées, attachons nous maintenant à décrire quelles sont les caractéristiques et qualités de services offertes par les réseaux aujourd'hui disponibles. Cette section n'a pas pour vocation d'être exhaustive, ni d'effectuer un état de l'art technique sur l'ensemble des technologies de communication, ce qui représenterait une quantité d'informations colossale. L'objectif est plutôt de présenter ces différents réseaux en fonction de critères liés à la qualité de service qu'ils offrent, ce afin d'établir une correspondance entre les besoins des applications multimédias distribuées et l'offre en QoS des différentes technologies de réseaux. L'idée sous-jacente à cette étude est, en fonction des besoins spécifiques d'une application, d'être capable de choisir parmi tous les moyens de communication actuellement disponibles, celui qui présentera le service le plus adapté et au meilleur prix.

Cette comparaison se veut pragmatique et se consacre à l'étude des réseaux aujourd'hui déployés ou en bonne voie de l'être. Il ne sera pas discuté de solutions comme « la fibre optique à la maison » qui ne sont encore à l'heure actuelle pas implantées en Europe ou n'ayant pas fait la preuve de leur validité économique. Les valeurs des critères de qualité de service sont mesurées sur des systèmes déployés, ou bien tirées d'études de dimensionnement de ces systèmes. Ces valeurs sont parfois difficiles à obtenir car souvent confidentielles, en particulier pour les réseaux sans fil qui sont en plein essor.

3.2 Critères de QoS

Parmi les nombreux critères qui définissent la qualité de service offerte par un réseau, quatre sont principalement utilisés :

- le **délai** entre l'envoi d'un paquet et sa réception par le destinataire,
- la **gigue** : variation du délai de bout en bout,
- la **bande passante** ou débit maximum,
- la **disponibilité** : taux moyen d'erreurs d'une liaison.

La qualité de service est gérée à différents niveaux : support, contrôle d'accès, réseau, transport, et application. L'étude est focalisée sur la QoS fournie au niveau support. Elle est mise en oeuvre physiquement ou logiquement, soit à l'aide de liens de communication dédiés (RNIS, liaison louée), soit à l'aide de liens logiques dédiés (circuit virtuel FR, ATM, MPLS, VPN) ou de priorités (Ethernet ou autres).

Parmi les réseaux étudiés, certains se distinguent par un rôle spécifique. Ce sont les réseaux d'accès. Ils n'ont pas pour vocation d'offrir un moyen de communication de bout en bout, mais seulement un moyen d'accéder à un autre réseau de communication. La qualité de service offerte aux utilisateurs dépendra alors du résultat de la combinaison des QoS des réseaux traversés. Ce point particulier sera abordé en détail dans le chapitre 4.

Les réseaux seront groupés selon leurs similitudes et seront présentés dans l'ordre suivant : D'abord les réseaux filaires, ensuite les réseaux par satellites, puis les réseaux sans fil terrestres.

3.3 Réseaux filaires

Dans cette première partie, sont présentées les principales solutions de communication basées sur des réseaux filaires. Le premier paragraphe traite des réseaux offrant un service de bout en bout. Le second paragraphe abordera les réseaux d'accès.

3.3.1 Réseaux longues distances

L'héritage issu du développement des réseaux téléphoniques a favorisé l'émergence de réseaux de télécommunication à capacité symétrique et à qualité de service garantie. Parmi ce type de réseaux, les solutions disponibles pour déployer une architecture distribuée sont assez nombreuses. Toutes ne sont pas présentées, comme X.25 qui n'est, par exemple, pas adaptée aux communications multimédias. Un classement chronologique peut être effectué en présentant d'abord la solution basée sur un réseau téléphonique, puis les architectures orientées voix telles que RNIS ou ATM.

Modem/RTC² :

Moyen de communication le moins coûteux, les réseaux téléphoniques offrent néanmoins une solution de bout en bout à qualité de service garantie. Originellement basés sur des liaisons analogiques ce réseau est désormais presque numérique. Le cœur du réseau téléphonique est aujourd'hui entièrement numérique et seul l'accès reste analogique. La comparaison de ces réseaux avec les réseaux numériques se justifie toutefois dans le sens où la qualité de service perçue par l'utilisateur est quand même celle du réseau d'accès analogique. La bande passante garantie par l'opérateur s'étale de 300 Hz à 3400 KHz, selon les recommandations M.1020 à M.1040. L'utilisation d'un modem permet le transfert de données à des débits, qui ont évolués

² Réseau Téléphonique Commuté

au fil des années, de 2400 Bauds/s jusqu'à 56 Kbits/s. Le standard V90 offre un débit de 56 Kbits/s qui est toutefois très sensible à la qualité de la ligne et à la qualité des informations transférées du fait d'un fort taux de compression.

Ce réseau étant dimensionné pour le transfert de la voix, le délai de bout en bout, ainsi que la gigue restent très faibles. Cela en ferait un réseau idéal pour des applications de type visioconférence, si ce n'était sans compter sur sa très faible bande passante rendant impossible la transmission de vidéo de qualité.

RNIS :

Numéris est le réseau téléphonique de France Télécom basé sur la technologie RNIS ("Réseau Numérique à Intégration de Services", en anglais ISDN). Le RNIS est un réseau aux infrastructures flexibles dédié à l'intégration de voix et de données. Il a été pensé pour remplacer les lignes téléphoniques analogiques actuelles par du « tout numérique ». Sa fiabilité est bien supérieure au réseau téléphonique.

Le débit est de 64 Kbps (128 en utilisant deux canaux) au lieu de 56 Kbps avec les modems les plus rapides. Le standard RNIS bande étroite (Narrowband ISDN) permet l'intégration de services pour des débits de 56 Kbps à 2 Mbps. Les solutions les plus courantes de visioconférence utilisent le service RNIS, de 128Kbit/s à 384 Kbit/s, qui est fortement déployé et offre un coût d'utilisation relativement faible. Toutefois, ce genre de système ne fournit pas une solution hétérogène.

Liaison terrestre filaire louée :

Les débits supérieurs sont offerts par des solutions utilisant des « lignes louées ».

Ces lignes louées permettent la transmission de données à moyens et hauts débits (2,4 Kbps à 140 Mbps) en liaison point à point ou multipoints (service Transfix). Les 3 lignes les plus répandues sont les T1 (1.5Mbps), les T2 (6 Mbps), et les T3 (45Mbps). Il existe aussi des lignes nettement plus rapides : ce sont les E1 (2Mbps), E2 (8Mbps), E3 (34Mbps) et E4 (140Mbps), etc. Les protocoles utilisés pour l'utilisation de ce type de liaison sont généralement Frame Relay, pour les plus anciennes et de plus en plus ATM. L'ATM permet de transférer des données à une vitesse allant de 25Mbps à plus de 622Mbps (il est même prévu d'offrir plus de 2Gbps sur fibre optique). La notion de qualité de service est intégrée par l'offre de 4 classes de service garanties : CBR, VBR, ABR, UBR. Le taux d'erreur bit généralement constaté pour de tels réseaux avoisine 10^{-12} , et les délais sont généralement très courts, légèrement supérieurs à ceux induits par un transport à la vitesse de la lumière, lorsqu'un réseau à fibre optique est utilisé.

Les équipements nécessaires pour ce type d'infrastructure étant chers, ceux-ci sont essentiellement utilisés par les opérateurs de télécommunication sur des lignes longue distance. Ils ne sont donc généralement pas utilisés de bout en bout, mais plutôt après l'interconnexion d'un réseau d'accès. Les garanties de QoS ne sont généralement pas assurées au delà du réseau de type ATM, sauf si des solutions telles que MPLS ou Diffserv [BLA98] sont mises en place.

3.3.2 Réseaux d'accès téléphoniques

Les réseaux déployés à l'heure actuelle n'offrent pas un service à large bande et à faible coût, bien que cette demande provenant des besoins applicatifs soit de plus en plus forte. La « fibre optique à la maison » n'étant pas non plus une réalité économique, des solutions permettant

l'accès à haut débit à ces réseaux se sont développées, en utilisant l'infrastructure disponible. La technologie ADSL, qui offre un accès haut débit en utilisant l'infrastructure du réseau téléphonique, en est un exemple.

ADSL/xDSL :

l'ADSL (Asymmetrical Digital Subscriber Line) est une technologie de liaison asymétrique dont le débit se situe entre les débits de la ligne de type Numéris et du câble. Dédiés à l'accès, ce protocole utilise l'infrastructure d'accès téléphonique constituée de fils de cuivre dont on a toujours pensé qu'ils ne pouvaient pas supporter des vitesses de communication de plus de 10 Kbit/s par seconde.

En fait, le réseau téléphonique a été conçu à la base pour transporter des voix, c'est-à-dire que les infrastructures téléphoniques étaient conçues pour utiliser une bande passante de l'ordre de 3 KHz. Pourtant, ces lignes peuvent supporter physiquement des bandes passantes allant jusqu'à 1 Mhz [Table I-2]. Il est donc possible, en utilisant les lignes de téléphone, d'optimiser les taux de transfert sur de très courtes distances. En fonction de la distance entre l'utilisateur et le central téléphonique, les débits peuvent s'échelonner entre 1,5 Mbit/s et 10 Mbit/s, offrant des possibilités beaucoup plus grandes que les possibilités actuelles (64 Kbit/s à 128 Kbit/s pour les lignes téléphoniques). Ces chiffres doivent toutefois être modulés car, par exemple, l'offre actuelle de France Télécom est de 1 Mbit/s pour le débit maximal sur le canal opérateur - utilisateur, et de 256 Kbit/s sur le canal retour !

Ces solutions sont toutefois réservées à l'accès de réseaux du fait qu'elles ne sont déployables que sur de courtes distances.

Bande haute (1 Mhz)	Canal descendant - 8 Mbit/s
Bande médiane (500 khz)	Canal bidirectionnel
Bande basse	Téléphonie (0 à 4 KHz)- RNIS (0 à 80 KHz)

Table I-2- ADSL

Il existe différentes technologies basées sur ce principe, nommées "xDSL" (ADSL, HDSL, SDSL et VDSL). Elles correspondent chacune à une utilisation particulière. C'est l'ADSL qui semble être la plus au point commercialement; elle est déjà en expérimentation en France. Une particularité de cette technologie est d'avoir été adaptée au mode client/serveur de l'Internet, grâce à un débit supérieur sur le canal de réception (1,5 à 10 Mbit/s en réception 640 Kbit/s en émission).

Modem	56 Kbit/s
RNIS	64 ou 128 Kbit/s jusqu'à 2 Mbit/s
ADSL	1,5 à 10 Mbit/s en réception 640 Kbit/s en émission
Câble	de 500 Kbit/s à 10 Mbit/s

Table I-3- Type de liaison / taux de transfert théorique

3.3.3 Réseaux Locaux

Les réseaux locaux sont un cas un peu particulier, car ils offrent généralement des services haut débit, mais uniquement sur de courtes distances. Lorsque les communications sortent du réseau local, ces réseaux offrent la même fonctionnalité qu'un réseaux d'accès, en assurant la connectivité à un réseau longue distance.

Le monopole d’Ethernet sur ce segment est une réalité, même s’il n’offre pas de garanties suffisantes en termes de QoS. Les très hauts débits qu’offre ce protocole (de 10 Mbit/s au Gbit/s) permettent, entre autre, de pallier au manque de QoS en permettant facilement le déploiement de réseaux sur dimensionnés. Le fait qu’un réseau ne soit pas trop chargé permet d’offrir un service de bonne qualité, même si aucun service n’est garanti. Par contre, dès que le réseau est chargé, ces propriétés disparaissent. Des solutions permettent toutefois d’assurer la garantie de QoS sur les réseaux locaux de type Ethernet, comme BMRP (Bandwidth Management and Reservation Protocol) [HOR00].

Quoiqu’il en soit, la limitation de la QoS du réseau est rarement engendrée par le réseau local, mais plutôt par le réseau longue distance accédé.

3.4 Réseaux Hertziens par satellites

La majorité du réseau cœur de l’Internet est constitué de réseaux terrestres câblés, avec des bandes passantes allant de quelques Mbit/s (pour les réseaux cuivrés) à plusieurs centaines de Mbit/s (pour les réseaux à fibre optique). L’Internet nouvelle génération devra offrir des services large bande, ainsi que proposer des solutions étendues de mobilité, par l’unification des différents types de réseaux sans fil (LAN, MAN, Satellite). Sur ce dernier point, les réseaux satellites sont très bien placés, grâce à leur large spectre de diffusion et leur bande passante importante, pour offrir un service de connectivité global.

3.4.1 Caractéristiques des réseaux par satellites GEO

Les systèmes de communication utilisant des satellites géostationnaires représentent la grande majorité des systèmes de communication par satellites. Ils sont caractérisés physiquement par une distance à la terre et une couverture fixe, qui se traduit en service rendu par un délai de traversée constant et une couverture géographique importante. Il est difficile d’exprimer des valeurs de bande passante, de débit, de taux d’erreur sur de tels réseaux car tout dépend des choix de conception du système. A titre d’exemple la Table I-4 compare les services offerts par deux types de satellites de communication géostationnaires : SAGAM et Europe star.

	SAGAM	Europe star	
Altitude (km)	35786	35786	
Bande passante	80 Mb/s	30*36Mhz	
Application	multimédia	Télécom/data	
Délai de propagation (ms)	120	120 (240/2)	
Variation du délai de propagation (ms)	0	0	
Bit error rate (BER)	10-7	10-10*	10-6*
Disponibilité	>0.990	95.9*	99.6*

Table I-4 - Caractéristiques spécifiques de réseaux satellite GEO

*Le taux d’erreur est en relation avec la disponibilité.

La comparaison entre SAGAM et Europe Star (E*) doit être faite en considérant que SAGAM est un satellite avec régénération (traitements à bord) alors que E* est un satellite transparent. Cela implique, par exemple, que le délai de bout en bout pour SAGAM sera globalement plus important (du fait du traitement à bord). Toutefois ceci n’apparaît pas dans les tables de

caractéristiques, du fait que seul le délai de transmission est considéré. D'autre part, les caractéristiques sont considérées de bout en bout, ce qui, du fait de la régénération à bord de SAGAM, réduit d'un facteur de 2 la valeur du délai mesuré pour SAGAM (c'est dans ce cas le trajet sol – satellite qui est mesuré et non plus le trajet sol – satellite - sol).

3.4.2 Caractéristiques des réseaux par satellites LEO

Les réseaux par « satellite basse orbite » LEO ont la propriété d'observer des délais plus courts du fait de leur altitude plus basse, mais en revanche présentent des variations de délai beaucoup plus importantes induites par leur mobilité et les changements de cellules.

Bien sûr, il y a un prix à payer pour atteindre des délais comparables à ceux des réseaux terrestres, qui est la complexité de l'architecture des passerelles s'occupant de la gestion des commutations de cellules.

Les données suivantes (Table I-5) reflètent ce que sont (ou devraient être) les performances des futurs réseaux par satellites LEO d'un point de vue utilisateur.

	Globalstar	Skybridge
Altitude (km)	1410	1469
Bande passante utilisateur	2.4 – 9.6 kb/s 2800 voix circuits	60 Mb/s descendant 2 Mb/s montant
Application	Téléphonie, fax, donnée	Multimédia
Délai de propagation minimum (ms)	4.63	5
Délai de propagation maximum (ms)	11.5	13.8
Variation des délai de propagation (ms)	6.87	8.8
CLR	10^{-6}	10^{-6}
Bit error rate	10^{-7}	10^{-8}

Table I-5 - Caractéristiques des réseaux par satellites LEO

3.4.3 Avantages et inconvénients des réseaux satellites

Les réseaux satellites, et particulièrement ceux utilisant des satellites géostationnaires, possèdent les avantages suivants :

Large bande

Une bande Ka (20-30 GHz) peut permettre la transmission de Giga bits par secondes.

Faible coût

Les systèmes de communication par satellite offrent des solutions de déploiement de réseaux grande distance à faible coût, comparativement aux solutions câblés, grâce à leur large spectre de diffusion.

Mobilité

La mobilité est un des autres atouts des réseaux satellites du fait de sa simplicité de mise en œuvre sur les réseaux GEO. L'utilisateur peut être mobile sur toute la surface couverte par le satellite.

Diffusion / multi-points

Les services de diffusion et de diffusion multi-points sont naturellement déployables sur ce type de réseaux. Le Mbone utilise déjà ce type de service [ALM98].

D'un autre côté, les délais de communication entre deux stations sol interconnectées par un satellite suscitent d'autres problèmes de performance.

Pour un réseau utilisant des satellites GEO, le délai nécessaire pour un « saut », c'est à dire le trajet station sol – satellite – station sol, nécessite au moins 250 ms (pour le délai physique minimal) et peut atteindre 400 ms (lorsque des mécanismes complexes de traitement à bord sont utilisés). Si l'influence de ce délai n'est pas primordiale pour de simples transferts de données, l'impact sur le fonctionnement des protocoles nécessitant de fréquents échanges entre les entités émettrice et réceptrice est grand. Ce qui est malheureusement le cas du protocole TCP qui n'est pas prévu pour de tels délais. Une fiabilité très variable des liaisons, principalement dépendante des conditions météo, est aussi à signaler comme caractéristique principale de ces systèmes.

Dans le cas des réseaux LEO, le délai est moindre, mais les satellites n'observant pas une position fixe, des constellations de satellites sont nécessaires pour assurer une couverture permanente. La complexité de gestion d'un tel système induit des risques de coupure de service réseau particulièrement pénalisant pour certains protocoles. La variation périodique des délais est un autre paramètre caractéristique de ces systèmes qui doit rentrer en compte dans l'élaboration des mécanismes des nouveaux protocoles.

3.5 Réseaux Hertziens terrestres

3.5.1 Réseaux cellulaires mobiles

Actuellement, trois générations de réseaux sans fil et cellulaires ont vu le jour, la seconde étant aujourd'hui en activité et la troisième en plein développement [Table I-6].

La première génération des réseaux mobiles ne fournit quasiment aucun service en dehors du téléphone et présente donc peu d'intérêt pour notre étude. Elle reposait sur une communication analogique et n'a connu qu'un succès restreint en raison du coup des équipements.

Le numérique a été adopté avec l'apparition de la seconde génération et la normalisation de peu d'interfaces air a permis la fabrication en grande série puis la distribution vers le grand public.

La troisième génération se veut révolutionnaire en matière de services fournis. Elle a pour objectif de devenir la continuité parfaite des réseaux filaires.

Type de génération	Réseau sans fil	Réseau cellulaire
1 ^{re} génération	CT0,CT1	NMT,R2000,AMPS,TACS
2 ^e génération	CT2,DECT,PHS	GSM, D-AMPS, PDC, PCS1800/1900, IS95/IS41 et IS136/IS41
3 ^e génération	-	IMT2000, UMTS, IS95B

Table I-6 - Les différentes générations de réseaux sans fil et cellulaires

Avec une bande passante de 9,6Kbit/s et, surtout, une architecture de réseau fondée sur la commutation de circuit, le GSM n'est pas réellement adapté à la transmission des données. La facturation se fait au temps de connexion, et non au volume de données transmises.

Théoriquement, l'UMTS devrait régler ces problèmes, puisqu'il utilise la commutation de paquets et offrira 384 Kbit/s de bande passante au début, et 2 Mbit/s plus tard (dans le meilleur des cas). La Table I-7 représente les caractéristiques prévues pour les futurs réseaux UMTS.

Malheureusement, pour passer à l'UMTS, il faut pratiquement faire table rase du GSM : en particulier l'UMTS ne fonctionne pas dans les mêmes bandes de fréquences. D'où l'idée de faire une étape intermédiaire qui améliore le GSM, sans tout remplacer : c'est le GPRS. Celui-ci fonctionne en mode paquets et offre une bande passante de 171Kbit/s (dans des conditions idéales).

Condition de test	En intérieur	De l'extérieur vers l'intérieur et pour un marcheur	Véhicule à 120 km/h	Véhicule à 500 km/h
Services testés	Débit BER Délai Activité du canal	Débit BER Délai Activité du canal	Débit BER Délai Activité du canal	Débit BER Délai Activité du canal
Données à faible délai (audio)	8 kb/s $\leq 10^{-3}$ 20 ms 50 %	8 kb/s $\leq 10^{-3}$ 20 ms 50 %	8 kb/s $\leq 10^{-3}$ 20 ms 50 %	8 kb/s $\leq 10^{-3}$ 20 ms 50 %
Commutation de circuit	144-384-2048 kb/s $\leq 10^{-6}$	64-144-384 kb/s $\leq 10^{-6}$	32-144-384 kb/s $\leq 10^{-6}$	32-384 kb/s $\leq 10^{-6}$
Données à faible délai	100 ms 100%	50 ms 100%	50 ms 100%	50 ms 100%
Commutation de circuit	144-384-2048 kb/s $\leq 10^{-6}$	64-144-384 kb/s $\leq 10^{-6}$	32-144-384 kb/s $\leq 10^{-6}$	32-384 kb/s $\leq 10^{-6}$
Données contraintes	300 ms 100%	300 ms 100%	300 ms 100%	300 ms 100%

Table I-7 - Caractéristiques des différentes configurations avec UMTS

L'intégration de classes de service dans la norme UMTS est une très bonne chose. Des services tels que la voix, la vidéo, et le transfert de données devraient être supportés. La qualité d'écoute de la voix se veut similaire à celle fournie par les réseaux fixes RNIS bien que, en matière de pertes, RNIS soit bien plus performant. Cependant les améliorations dans le domaine du codage de la voix peuvent permettre d'espérer une telle qualité.

3.5.2 Réseaux locaux et métropolitains sans fil

Réseaux métropolitains sans fil (MAN)

La Boucle Locale Radio (B.L.R) est un nouveau moyen d'accéder à l'Internet haut débit mais également à des services de téléphonie. Traditionnellement, la boucle locale est le dernier lien entre les prises téléphoniques et les serveurs centraux de France Télécom, appelé également "le dernier kilomètre". La boucle locale radio permet donc à un opérateur de relier l'abonné à ses centraux sans passer par les fils de cuivre et en utilisant une liaison radio hertzienne. Pour l'utiliser, le client a besoin d'une antenne et d'un câble qui relie l'antenne à un boîtier d'accès

Internet. De son côté l'opérateur doit disposer d'antennes éparpillées sur le territoire, sachant qu'elles doivent être au maximum à 15 kilomètres du client final. Cette technologie autorise au final un débit compris entre 512 Kbits/s et 2Mbits/s.

La technologie permet en outre d'améliorer sensiblement l'accès mais elle se heurtera à quelques contraintes, notamment les conditions météorologiques qui pourraient affecter le fonctionnement des antennes et donc les débits.

Réseaux locaux sans fil (WLAN)

Ethernet 802.11

Le LAN sans fil (WLAN) est un système de transmission des données conçu pour assurer une liaison indépendante de l'emplacement des périphériques informatiques qui composent le réseau et utilisant les ondes radio plutôt qu'une infrastructure câblée. Dans l'entreprise, les LAN sans fil sont généralement implémentés comme le lien final entre le réseau câblé existant et un groupe d'ordinateurs clients. Ils offrent aux utilisateurs de ces machines un accès sans fil à l'ensemble des ressources et des services du réseau de l'entreprise, sur un ou plusieurs bâtiments. Le standard 802.11, norme régissant les réseaux locaux sans fil créé en 1997, s'appliquait à des débits de 1 et 2 Mbit/s et définissait les règles fondamentales de la signalisation et des services sans fil. Conscient de la nécessité d'augmenter ce débit, l'IEEE a ratifié dernièrement la spécification 802.11b (également baptisé 802.11 Haut Débit) qui entérine des transmissions à 11 Mbit/s maximum dans la bande des 2,4 GHz. Le groupe 802.11a travaille actuellement à la transmission sur la bande des 5GHz pour atteindre des débits de 20/25 Mbit/s en crête. Ces systèmes sont plutôt destinés aux liaisons points à points. Les caractéristiques générales des systèmes basés sur la norme 802.11b sont données dans la Table I-8.

Bande de fréquences RF	2,4 GHz (2400-2500 MHz)			
Taux d'erreurs sur les bits (BER)	Meilleur que 10^{-5}			
Puissance nominale	15 dBm			
Portée / Vitesse de transfert	Vitesse élevée 11 Mbits/s	Vitesse moyenne 5,5 Mbits/s	Vitesse standard 2 Mbits/s	Basse vitesse 1 Mbit/s
Environnement de travail ouvert	160 m	270 m	400 m	550 m
Environnement de travail semiouvert	50 m	70 m	90 m	115 m
Environnement de travail fermé	25 m	35 m	40 m	50 m
Sensibilité d'un récepteur	-83 dBm	-87 dBm	-91 dBm	-94 dBm
Étalement du temps de propagation (avec un FER < 1 %)	65 ns	225 ns	400 ns	500 ns

Table I-8 - Caractéristiques des réseaux 802.11

D'autres normes, dont Hyperlan [HYP], sont en concurrence avec la norme 802.11. Toutefois, leurs caractéristiques générales étant sensiblement équivalentes, elles ne seront pas présentées dans cette section.

Il est particulièrement intéressant de noter que le taux d'erreur annoncé est inférieur à 10^{-5} . Dans les pires cas, un tel taux d'erreur est très pénalisant pour les applications nécessitant de la qualité de service, et ce d'autant plus que le délai séparant les entités communicantes est grand. Les principes de retransmission de bout en bout des protocoles de transport imposent des délais de retransmission, pour le recouvrement des données perdues, proportionnels au

délai de bout en bout. Si le taux d'erreur et le délai d'une connexion sont grands, le comportement de certains protocoles de transport, comme TCP, est fortement perturbé [PAR97, RFC3155].

3.6 Bilan

Si nous devons effectuer un bilan sur la diversité des réseaux actuellement disponibles, la principale remarque qui pourrait être faite est que l'hétérogénéité des services offerts par les différents supports n'est pas un mythe.

Parmi les critères de QoS que nous avons évalués, de très fortes variations sont visibles. Les délais de transmission sont compris entre quelques milli secondes à près d'une demi seconde, les taux d'erreur les plus élevés sont 10 millions de fois supérieurs à ceux des réseaux les plus sûrs et les bandes passantes disponibles varient du Kbit/s au Gbits.

Les modèles de réseaux en couche, comme le modèle OSI ou le modèle Internet, ont pour principe de masquer aux couches supérieures cette hétérogénéité par un principe d'homogénéisation fourni par le niveau 2.

Pourtant, comment et pourquoi cacher ce qui ne peut être réellement caché ? Pourquoi ne pas essayer de tirer profit de cette diversité de services pour l'adapter aux besoins des applications ?

C'est indéniablement, un des points clé qui permettra l'offre de qualité de service dans les futures architectures de communications.

La couche d'interconnexion de réseaux n'offre aujourd'hui aucune fonctionnalité pour une telle gestion. Les routeurs de l'Internet n'ont pour rôle que le relayage de paquets IP d'un lien vers un autre, et ne possèdent aucun mécanisme de haut niveau pour la gestion de la qualité de service.

Les réseaux actifs ouvrent une nouvelle voie en permettant le déploiement dynamique de nouveaux services sur les nœuds du réseau. Cette voie semble prometteuse quant au déploiement de services de gestion de la QoS dans les réseaux actuels.

4 Les réseaux actifs

4.1 Introduction

La capacité de créer, déployer et gérer de nouveaux services en réponse à des demandes des utilisateurs est le facteur clé de l'apparition des réseaux actifs. Ce type de réseaux offre plusieurs avantages pour les fournisseurs de services et les vendeurs des équipements de télécommunication. La compétition pour les fournisseurs de services va se faire sur leur capacité d'offrir différents services en réponse aux besoins de leurs clients. La compétition pour les constructeurs d'équipements va se faire sur les caractéristiques de leurs équipements et sur leur degré de programmabilité.

Les réseaux actifs [TEN96] offrent une ouverture au niveau des réseaux en implémentant des fonctions spécifiques pour le traitement des données au niveau des nœuds.

L'idée est de pouvoir ouvrir les réseaux et accélérer leur programmation de façon contrôlée et sécurisée pour le déploiement de nouveaux protocoles, services et architectures.

La séparation des équipements matériels (commutateurs, routeurs...) et des logiciels de contrôle est nécessaire pour rendre le réseau plus ouvert. Une telle séparation est difficile à réaliser ; la raison est que les commutateurs et les routeurs sont intégrés de façon verticale. Ainsi, les fournisseurs de services réseau n'ont pas accès aux mécanismes de contrôle des équipements (exemple: les systèmes opératoires des routeurs), ni à leurs algorithmes (exemple : protocoles de routage), ni à leurs états (exemple : tables de routage). Cela rend le déploiement de nouveaux services difficile à cause de la nature fermée des nœuds.

C'est pour cette raison que les services sur les réseaux actuels sont uniquement des services de bout en bout. Les réseaux actifs visent à supprimer cette contrainte pour implémenter tout type de service.

4.2 Méthodologie

Actuellement, il y a une grande demande pour l'ajout de nouveaux services aux réseaux ou pour leur accommodation pour répondre aux besoins de nouvelles applications. Seulement, l'introduction de ces nouveaux services se fait toujours de façon manuelle (intervention sur chaque nœud et reprogrammation dans le langage du constructeur) et c'est un processus qui consomme du temps et de l'argent.

L'intérêt des réseaux actifs est de pouvoir justement offrir un déploiement dynamique des services. Deux approches existent alors pour la construction des réseaux actifs. La première approche est celle proposée par la communauté Opensig et propose une offre d'interfaces de programmation pour les équipements du réseau pour donner accès aux nœuds du réseau (commutateurs et routeurs), permettant ainsi de réaliser de nouveaux services et architectures (par exemple des réseaux virtuels). La deuxième approche est celle du DARPA et permet le déploiement de nouveaux services de façon dynamique à l'intérieur des réseaux. Le niveau d'exécution dynamique est plus flexible en considérant la transmission, le routage et l'exécution des "paquets actifs". Une autre approche orientée vers le déploiement de services aux extrémités du réseau, est celle des code mobiles. Ce cas particulier de réseau actif sera détaillé dans le chapitre suivant.

Finalement on peut dire que l'approche Opensig sépare le contrôle du réseau de l'information transportée et se focalise spécialement sur les commutateurs pour offrir un certain niveau de qualité de service, alors que l'approche DARPA se concentre sur les réseaux IP où le contrôle et le transfert des données sont combinés.

4.3 Architecture

Tous les nœuds du réseau actif offrent des fonctionnalités de base communes, qui sont définies par l'architecture du nœud. L'architecture concerne le traitement des paquets et la gestion des ressources, ainsi que des problèmes généraux d'adressage et de gestion des ressources de bout en bout, etc.

4.3.1 Les composants du nœud actif

L'architecture du nœud actif est divisée entre les environnements d'exécution EEs (Execution Environments) et le système d'exploitation du nœud (NodeOS).

L'environnement d'exécution définit pour une machine virtuelle des interfaces de programmation qui sont offertes aux utilisateurs du réseau actif. Les utilisateurs contrôlent cette machine virtuelle en envoyant du code à l'EE dans des paquets. L'exécution des

instructions du code change en général l'état de l'EE et du nœud et définit la transmission des paquets (envoi immédiat ou différé).

Le NodeOS est la couche opératoire entre les environnements d'exécution et les ressources physiques (bande de transmission, cycles de processeur, mémoire de stockage). Son existence est justifiée à la fois par le besoin de pouvoir supporter plusieurs EEs et d'offrir un ensemble de fonctionnalités communes pour tous les nœuds actifs.

Il y a trois types de ressources de base : les threads, la mémoire et les canaux. Les deux premiers se comportent de la même manière que dans un OS classique. Les canaux sont de deux types : des canaux qui acheminent les paquets d'un point d'entrée à un point de sortie sans être interceptés par l'EE et des canaux qui remontent les paquets à l'EE pour les traiter.

4.3.2 Le protocole ANEP

ANEP (Active Network Encapsulation Protocol) [ALE97] est le protocole standard d'échanges d'information et de code dans les réseaux actifs. Les paquets actifs sont encapsulés dans un paquet ANEP. Son utilisation est entre autre nécessaire si plusieurs environnements d'exécution sont installés sur le même nœud actif. Dans ce cas, l'information *type* contenue dans l'en-tête des paquets ANEP permet la redirection du paquet vers le bon environnement d'exécution. Le protocole ANEP est implanté originellement au dessus d'UDP. De récents travaux ont proposé son implantation directement au dessus d'IPv6 afin d'accroître les performances des réseaux actifs [DAL00]. Il existe un pendant à ce protocole, défini pour le réseau ANON [WOL00], s'appelant SAPF (Simple Active Packet Format).

4.4 Plates-formes

Plusieurs publications dressent un état de l'art et énumèrent les différents travaux en cours dans le domaine des réseaux actifs [FES00,TEN97,CAL98]. Ce paragraphe présente une vue de quelques réalisations de prototype de réseaux actifs.

4.4.1 Active IP

Active IP est un prototype de réseau actif développé en 1996 au MIT [WET96]. Ce prototype est basé sur l'utilisation des extensions IP pour le transport du code à appliquer à chaque paquet actif. Tout routeur supportant la fonction de routeur actif déclenche sur réception d'un paquet actif le traitement de l'option IP qui contenait un code TCL (Tool Command Language) étendu avec des primitives de manipulation de paquets IP ainsi que des fonctions de base d'accès aux ressources du nœud (adresse, horloge, ...). Ce code peut modifier le contenu des données du paquet IP, reconstruire une ou plusieurs trames et les émettre. Les routines actives sont transportées dans une nouvelle option d'IP proposée par les auteurs.

4.4.2 Smart Packets

Smart Packets [SCH99] est un projet de plate-forme active sponsorisé par le DARPA. L'objectif de ce projet est d'appliquer les technologies actives à l'administration et à la supervision de réseaux. Utilisant le protocole ANEP, l'architecture Smart Packets définit un nouveau format de paquets et deux langages spécifiques (*Sprocket* et *Spanner*) pour le transport de programmes au comportement « sécurisé » dans les paquets de données. Pour la première fois, l'option « router alert » de l'entête IP est utilisée pour indiquer à un routeur IP qu'un paquet est actif et qu'il doit - être traité en conséquence.

4.4.3 Switchware

Switchware [ALE98A] est un prototype de réseaux actifs qui regroupe les activités de l'Université de Pennsylvanie. Les travaux menés dans ce projet visent à définir une architecture homogène incluant différents paradigmes tels que le nœud actif, le paquet actif ainsi que la sécurisation d'un réseau actif. Les aspects langage de programmation de réseau actif sont également au cœur des travaux de Switchware.

L'architecture Switchware repose sur 3 principaux composants. Ces composants sont : ALIEN, PLAN et SANE.

ALIEN [ALE98C] explore l'architecture d'un nœud actif comportant à la fois des extensions actives chargées par l'opérateur du réseau et un support pour des paquets actifs chacun muni de code associé à leur traitement. Le langage de programmation des extensions ainsi que des paquets actifs est OCAML (Objective CAML) étendu avec des primitives de manipulation de paquets actifs (accès, envoi, réception).

PLAN (*Packet Language for Active Networks*) [HIC98] s'appuie sur l'architecture d'ALIEN pour la construction d'un nœud actif mais n'exploite pas un langage de programmation générique. En effet, il définit son propre langage inspiré du langage fonctionnel *Scheme* et CAML.

SANE (*Secure Active Network Environment*) [ALE98B,ALE99] a pour objectif la sécurisation d'un nœud ainsi que des paquets et extensions actives. Un mécanisme de contrôle d'accès à des extensions actives privilégiées avec authentification des paquets entrants permet de restreindre le champ d'accès de chaque paquet. Sur les paquets actifs, PLAN fournit des garanties importantes pour la sécurité. Notamment, un mécanisme assure que tout programme PLAN se termine (pas de boucles infinies). La limite de ressources garantit cette terminaison à l'échelle du réseau. Ces fonctions sont offertes dans ALIEN.

4.4.4 Netscript

Le projet de NetScript [DAS] concerne des réseaux flexibles pour le déploiement dynamique du code sur les nœuds intermédiaires. Le code de NetScript est empaqueté en tant qu'agents mobiles qui peuvent être acheminés aux nœuds du réseau où ils vont être exécutés. Des messages de protocole sont définis et encodés en tant qu'objets à niveau élevé de NetScript. NetScript permet de définir des filtres pour les paquets transitant sur le réseau, des algorithmes de routage, des analyseurs de paquets, etc...

4.4.5 ANTS

ANTS [WET98,WET99] offre un environnement de réseau programmable pour la définition d'une architecture basée sur les capsules (des paquets qui transportent des données et du code). Les capsules représentent des unités atomiques des interfaces de traitement et d'acheminement des informations. Les nœuds actifs traitent les capsules, assurent leur routage et supportent le mécanisme de distribution du code pour le déploiement autonome de nouveaux services.

Un réseau ANTS est constitué de nœuds interconnectés supportant l'environnement d'exécution ANTS. Cet ensemble de nœuds définit un réseau actif virtuel. Chaque nœud possède une table de routage et la possibilité d'interpréter les paquets actifs.

L'unité de transmission sur un réseau ANTS est la capsule [WET]. La capsule la plus basique comporte le code des méthodes qui permettent sa transmission et son routage dans le réseau actif. Un service ANTS est représenté par un ensemble de capsules définies par l'utilisateur. Ce service est aussi appelé *protocole ANTS*. Les capsules qui permettent le déploiement de nouveaux services possèdent, en plus, les informations relatives au traitement défini par l'application émettrice. Ces informations ne sont pas le code du programme du traitement lui

même, mais une référence qui permet de le télécharger dans les nœuds du réseau selon le schéma suivant :

Lorsqu'un nœud reçoit une capsule, il consulte son cache de code, si le code nécessaire au traitement de la capsule est présent, la capsule est exécutée tout de suite. S'il n'est pas présent, le nœud met en attente la capsule et envoie une capsule de demande de chargement de code au nœud précédent (i.e. celui dont il vient de recevoir la capsule).

Le nœud recevant la capsule de requête répond en envoyant le code demandé, le code est ainsi incorporé dans le cache du nœud en ayant fait la demande, la capsule est réveillée et traitée.

Le déploiement de code sur les nœuds du réseau est alors fait de proche en proche, puisque par ce mécanisme d'interrogation du nœud précédant, lors de l'introduction de la capsule dans le réseau, le premier nœud rencontré interrogera le nœud où se trouve l'application émettrice qui connaît forcément le code du service. La capsule continuera alors son chemin suivie du code à déployer. La Figure I-1 écrit le principe de déploiement

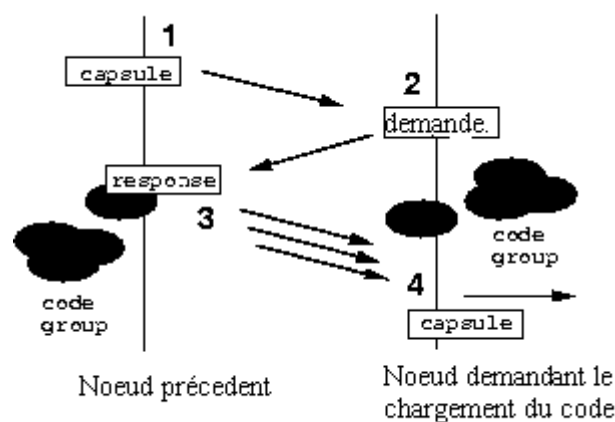


Figure I-1 - Mécanisme de déploiement de code

Avantages et inconvénients de la plate-forme ANTS :

Différents modèles de réseaux actifs existent actuellement. ANTS utilise le modèle de réseau actif par capsules. Ce modèle est celui qui permet le niveau de programmabilité le plus élevé. Le déploiement de nouveaux protocoles peut s'effectuer de façon rapide et non coordonnée.

C'est un modèle qui donne le maximum de liberté aux concepteurs d'applications.

Distribuer le code par l'intermédiaire des capsules est le moyen le plus dynamique de déployer les services sur le réseau, puisque la propagation d'un nouveau service se fait par la même voie que la transmission des données. ANTS fournit des services de base tels que le transport du code mobile, le chargement de ce code à la demande et les techniques de cache associées. Ces services de base suffisent à introduire de nouveaux protocoles ou à étendre des protocoles existants.

Parmi les inconvénients du modèle par capsules face à d'autres modèles de réseaux actifs, il y a la complexité de l'environnement de programmation. En effet, bien que très flexible, la programmation par capsule est délicate, car le code est attaché non plus à un nœud qui traite des paquets, mais à des capsules qui décident elles-mêmes du traitement qu'elles doivent subir, et des autres capsules éventuelles à créer. Les problèmes de sécurité sont également importants car il faut garantir la stabilité du système quelles que soient les capsules à traiter.

4.5 Le « *End-To-End Argument* » et les réseaux actifs

Le « *end-to-end argument* » a été l'un des principes d'architecture du début de l'Internet [RFC1958]. Que recouvre-t-il ? Il souligne que certaines fonctions de bout-en-bout ne peuvent être réalisées que par les systèmes d'extrémités eux-mêmes et que, de plus, des fonctionnalités ne doivent être placées dans le réseau que si et seulement si elles peuvent y être implémentées de manière simple et efficace. Même si, à première vue, les modèles de réseaux actifs semblent aller à l'encontre de ces principes (en autorisant les applications à programmer les nœuds du réseau), une version « revisitée » sur l'argumentation du bout-en-bout est présentée dans [BHA98]. Les auteurs y montrent que les réseaux actifs en sont une conséquence naturelle, puisque certaines fonctions peuvent être implémentées de manière plus efficace en utilisant des informations qui ne sont disponibles que dans la couche réseau : leur exploitation peut permettre une amélioration du service vu par l'application. A titre d'illustration, la localisation d'une congestion dans le réseau, ou encore la localisation du routeur responsable de pertes dans un arbre de diffusion, sont des informations qui ne sont accessibles qu'au niveau réseau. Par ailleurs, l'application peut posséder des informations qui permettraient d'optimiser les performances du service réseau. Par exemple, seule l'application sait si, parmi les données qu'elle envoie, certaines sont plus importantes que d'autres. Si cette connaissance était disponible au niveau réseau elle permettrait d'en déduire les paquets « prioritaires » en cas de congestion (i.e. ceux qu'il ne faut pas perdre). La dépendance entre unités de données (paquets), adéquation de la Qualité de Service demandée et traitements effectués dans le réseau sur les paquets, montre la pertinence de l'approche réseaux actifs : combiner l'application et le réseau constitue un moyen d'améliorer les performances.

Un des autres arguments du [RFC1958] est de n'offrir un service dans la couche réseau que seulement s'il est utilisé par toutes les applications et s'il ne présente pas de redondance avec d'autres services de niveau supérieur. Cette précaution tend à éviter les pertes de performances dues à des calculs redondants entre différentes couches. Ce problème n'a pas lieu d'être dans les réseaux actifs, puisque chaque application peut déployer les services selon ses propres besoins. Ainsi un service ne doit pas influencer sur le comportement global d'autres applications. Toutefois, la gestion d'une plate-forme active induit un certain *overhead* de calcul, qui n'est transparent pour aucune application, même celles utilisant un service minimum.

5 La Problématique du multi-réseaux

Les applications multimédias nécessitant différents types de QoS en fonction des types de données utilisés, cette QoS comporte divers paramètres tels que la fiabilité, le débit ou encore le délai de bout en bout.

Prenons l'exemple d'une application de visioconférence qui a ouvert 3 connexions : une pour la transmission de l'audio, une pour la vidéo et une pour la signalisation. Typiquement, le flux audio est un flux synchrone à débit constant avec une fiabilité presque totale et un délai de bout en bout faible pour favoriser un fort niveau d'interactivité entre participants. Le flux vidéo correspond pour sa part à un flux presque synchrone à débit variable avec un débit crête et un débit moyen. Un niveau de fiabilité lié à la qualité vidéo demandée par les utilisateurs (exemple : haute qualité pour de l'imagerie médicale ou une qualité médiocre pour la téléformation), et un faible délai de bout en bout, le caractérisent. La connexion de signalisation doit être totalement fiable, avec un débit faible, et pouvoir travailler avec un délai de bout en bout compatible avec celui des liaisons audio et vidéo.

En réponse à ces différents critères de QoS, de nombreuses infrastructures réseaux basées sur des technologies très diverses sont apparues et peuvent alors être sélectionnées pour faire correspondre au mieux les besoins applicatifs avec la QoS fournie par chacun de ces réseaux. Il est désormais courant de voir les ordinateurs posséder plusieurs interfaces réseau et on peut ainsi utiliser des réseaux différents tels qu'Internet, ATM, GPRS/UMTS, le satellite (LEO, GEO), RNIS, etc. En ce qui concerne la visioconférence par exemple, le flux audio peut tirer parti d'une connexion RNIS, le flux vidéo d'un canal VBR ATM et la signalisation d'une connexion classique TCP/IP à travers Internet.

Ces réseaux correspondent bien aux besoins en QoS d'une vidéoconférence : RNIS est un réseau synchrone offrant un débit fixe, une fiabilité totale et un délai de bout en bout sans gigue ; ATM est un réseau large bande proposant un haut débit, avec un mode VBR, un faible délai de bout en bout et une bonne fiabilité ; TCP/IP dans Internet fournit, quant à lui, un service fiable sans réservation de bande passante ni contrôle du délai de bout en bout.

Pour bénéficier simultanément des services de tous les réseaux accessibles et de leurs caractéristiques variées, un protocole de transport doit être capable de séparer les différents flux de données de l'application en fonction de leur type. Il doit, de plus, les envoyer sur le réseau disponible le plus adéquat pour le flux considéré.

Par exemple, si l'on considère une application de VoD (de caractéristiques très voisines de celles de la visioconférence, hormis un délai de bout en bout moins fortement contraint), la Figure I-2 montre que le flux audio utilise une liaison RNIS, la vidéo un lien satellite ATM tandis que la connexion de signalisation utilise Internet.

Nous proposons une nouvelle couche de protocole capable de gérer dynamiquement cette séparation et répartition des flux en fonction des besoins en QoS de l'application et des caractéristiques des réseaux accessibles. Excepté le choix du réseau, un autre problème important que doit résoudre ce protocole est la re-synchronisation des 3 flux, envoyés sur des réseaux différents tant physiquement que logiquement. On parlera dans ce cas de « *multi-réseaux parallèles* ».

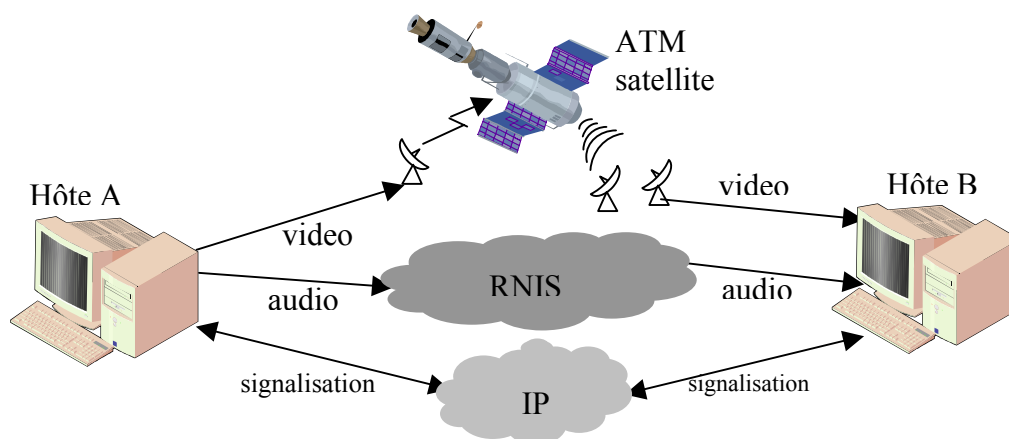


Figure I-2 - Multi-réseaux parallèle

Mais d'autres cas topologiques peuvent se présenter. En effet, il est possible que, pour atteindre un utilisateur distant, les flux de données doivent traverser successivement et en séquence plusieurs réseaux, chacun ayant ses propres caractéristiques de QoS. Suivant le type

de réseaux traversés, les implémentations du protocole utilisé (par exemple TCP) peuvent avoir des configurations différentes, établies spécialement pour chaque réseau.

Par exemple, en reprenant l'application de VoD sur un service TCP traversant un réseau terrestre ATM, un réseau ATM satellite (GEO) et enfin, un réseau MAN (Metropolitan Area Network) terrestre (Figure I-3), les implémentations et/ou configurations de TCP peuvent beaucoup différer sur chaque liaison. Il est alors difficile de garantir la QoS à travers tous ces réseaux. Ceci constitue typiquement du « *multi-réseaux en série* ».

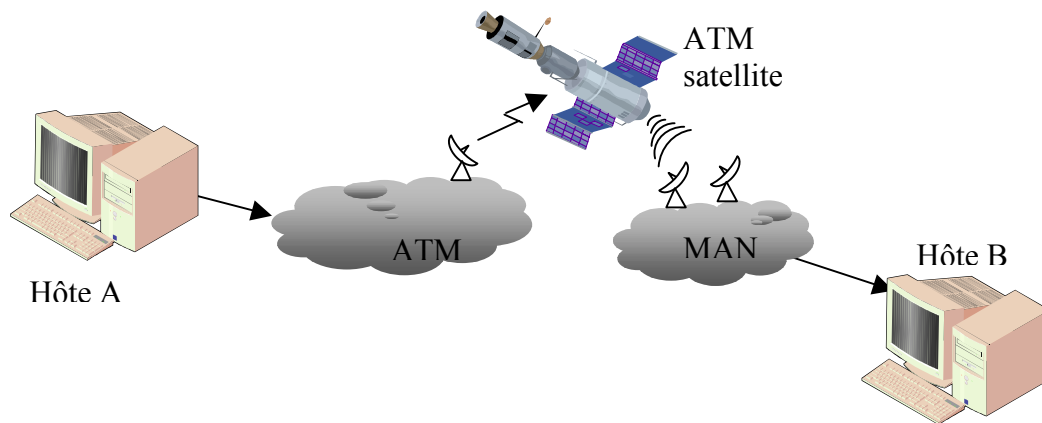


Figure I-3 - Multi-réseaux série

Ce type de problème multi-réseaux est sûrement l'un des problèmes demandant le plus d'effort dans la communauté Internet aujourd'hui. Le protocole IP a été conçu pour unifier toutes les techniques de communications, entre tous les types de réseaux physiques et leurs couches liaisons. De ce fait, il s'avère que l'Internet a été perçu et géré comme un réseau « à plat », la couche IP et les protocoles associés masquant au utilisateurs l'hétérogénéité du réseau (pour l'adressage, la QoS, etc.). Cette gestion à plat du réseau est sans aucun doute efficace tant que le début de la construction de l'Internet n'unifiait que des réseaux dont les propriétés étaient similaires (typiquement des réseaux câblés), ou tout du moins pas aussi différentes que ce qu'il est possible de voir depuis peu. Aujourd'hui, les réseaux à câbles et à fibres optiques ne sont plus les seuls que l'on peut rencontrer dans l'Internet. Les réseaux mobiles sans fil, et dans une moindre mesure, les réseaux par satellite, sont dans le domaine des liens utilisables. Les réseaux mobiles et satellites sont assez différents des réseaux câblés terrestres du fait de délais de transmission, de capacités et de taux de pertes différents. En conséquence, les protocoles, et en particulier les protocoles de transport doivent être repensés, ou tout au moins modifiés pour optimiser leur fonctionnement sur de tels types de liaisons. Il existe, par exemple, des versions de TCP dédiées aux réseaux satellites [ALL98] et aux réseaux mobiles [WAP].

Dans notre exemple de VoD, dans l'optique d'optimiser l'utilisation du segment satellite, la meilleure solution serait de convertir la souche du protocole de transport utilisé sur le réseau filaire de bordure, en une souche spécialisée pour les communications satellites afin de fournir la qualité de service requise par l'application. De plus en plus, ces applications de conversion, ou tout au moins de modification du trafic sont insérés dans le réseau. Ces applications sont souvent appelées « proxys », à l'instar des composants qui réalisent des opérations de cache sur les trafics asynchrones. Ces « proxys », ou mandataires, ont en charge, de nos jours, la translation de protocoles ou le détournement du trafic. Mais le véritable challenge est de les rendre capables de fonctionner sur des flux de données et/ou du trafic temps réel [MCC00].

Adresser ce que nous appelons le multi-réseaux série a pour objectif de palier à ce genre de problèmes qui nous obligent à ne plus considérer les réseaux comme aussi plats qu'avant.

Bien sûr, l'application de VoD peut aussi traverser des compositions série et parallèle de différents réseaux comme sur la Figure I-4. Sur cette figure, le flux audio bénéficie d'une liaison RNIS, le flux vidéo traverse une liaison ATM terrestre, un lien satellite géostationnaire puis un intranet métropolitain pendant que la signalisation utilise une liaison Internet de base.

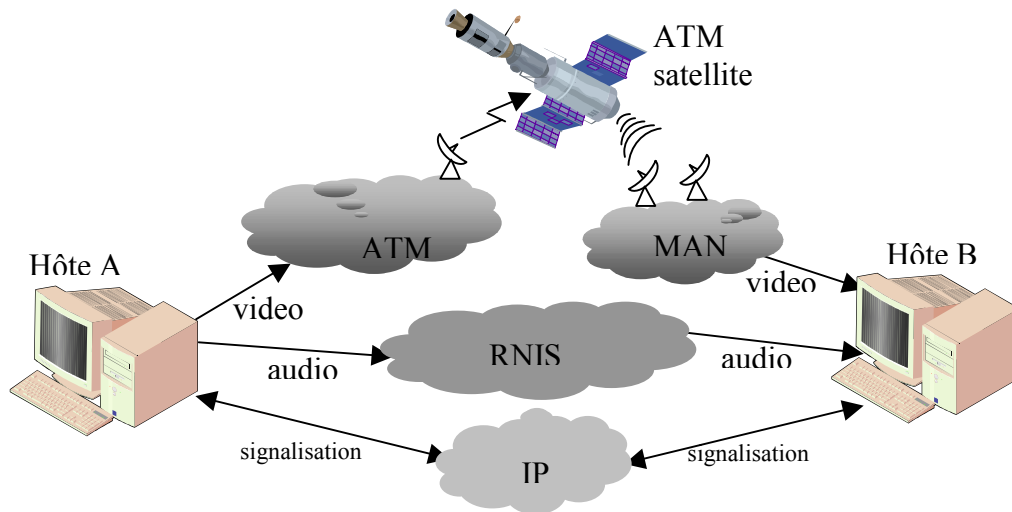


Figure I-4 - Multi-réseaux parallèle et sérié

Chapitre II Multi-réseaux Parallèle

1 Introduction

Malgré la multiplicité des technologies d'accès des réseaux, aujourd'hui peu d'architectures offrent de manière intégrée la possibilité de profiter « intelligemment » des différences de qualité de service offerte par ces réseaux.

D'un côté, de nombreux réseaux offrent une qualité de service adaptée à un certain type de transfert : les réseaux téléphoniques dédiés à l'audio, les réseaux par satellite adaptés à la vidéo ainsi que les interconnexions de réseaux conçues pour le transfert de données.

D'un autre côté, les flux générés par les applications multimédias nécessitent une qualité de service adaptée aux médias qu'ils transportent : les besoins d'un flux audio sont différents de ceux d'un flux vidéo.

Lorsque deux utilisateurs peuvent communiquer à l'aide de réseaux indépendants, c'est à dire s'ils possèdent une adresse distincte sur chacun des réseaux, la configuration est appelée multi-réseaux parallèle. Dans ce cas, plutôt que d'utiliser un seul des réseaux accessibles pour transférer l'ensemble des flux d'une application multimédia, il peut être plus efficace de choisir le réseau adapté à chaque type de flux.

Peu de systèmes de communication offrent de manière intégrée une telle possibilité. Face à ce vide, l'architecture d'un protocole multimédia multi-réseaux est proposée dans ce chapitre.

Ce chapitre est structuré en quatre parties :

- Un état de l'art est tout d'abord réalisé, présentant les principaux systèmes de communication existant aujourd'hui et classables dans le cadre des architectures multi-réseaux.
- La seconde partie se consacrera à la définition des caractéristiques qui doivent être offertes par un protocole multimédia multi-réseaux.
- La troisième partie détaillera l'architecture du protocole.
- Enfin, la quatrième partie illustrera, à l'aide de simulations, les performances obtenues dans une configuration particulière de multi-réseaux parallèle.

De nombreuses références seront faites au protocole TCP. Même si les mécanismes qui sont mis en œuvre dans ce protocole sont souvent contestés, TCP reste un très bon cas d'étude et d'expérimentations. Aujourd'hui, dans l'Internet, 80% des flux sont transportés par TCP. Il est clair que tout protocole qui veut être déployé doit être compatible avec TCP. La terminologie anglo-saxonne de « TCP Friendly » est employée pour désigner un protocole dit « compatible » avec TCP, c'est à dire qu'il implémente des mécanismes permettant l'équité de ses flux avec les autres flux TCP partageant les mêmes liaisons. Ces mécanismes sont typiquement, dans TCP, le contrôle de flux et de congestion par fenêtre glissante, avec ses phases de démarrage (*slow start*) et ses phases d'évitement de congestion (*congestion avoidance*).

2 Etat de l'art des protocoles multi-réseaux

2.1 User Service Assistant (U.S.A.)

L'architecture proposée par B. Landfeldt, A. Seneviratne, et C. Diot, nommée U.S.A. [LAN98] offre une vision nouvelle concernant l'adéquation des besoins en qualité de service des applications avec les services offerts par le système de communication de manière générale. Ce système offre à l'utilisateur un *assistant* permettant de choisir et de configurer son système de communication en fonction du type d'application qu'il utilise. Dans cette approche généraliste, le terme « système de communication » comprend aussi bien le système d'exploitation, que l'ensemble des réseaux de communication directement accessibles par l'ordinateur de l'utilisateur.

Souhaitant éviter les écueils d'une approche trop complexe, U.S.A. propose une approche de niveau applicatif, qui, à chaque lancement d'application, indique à l'utilisateur la configuration de son système qui correspondrait le mieux à ses attentes. L'analyse des besoins de l'utilisateur est basée sur une étude de son niveau de satisfaction lors des précédentes utilisations de l'application. Ainsi, contrairement à d'autres environnements de gestion de la QoS [BES94, CAM94, NAH95] qui évaluent les besoins de l'application pour déterminer le niveau de ressources réseau et le système nécessaire, c'est l'utilisateur qui, au travers d'une interface graphique, indiquera s'il est satisfait du déroulement de l'application, et si par conséquent la configuration du système doit être améliorée, ou non.

Description de l'architecture

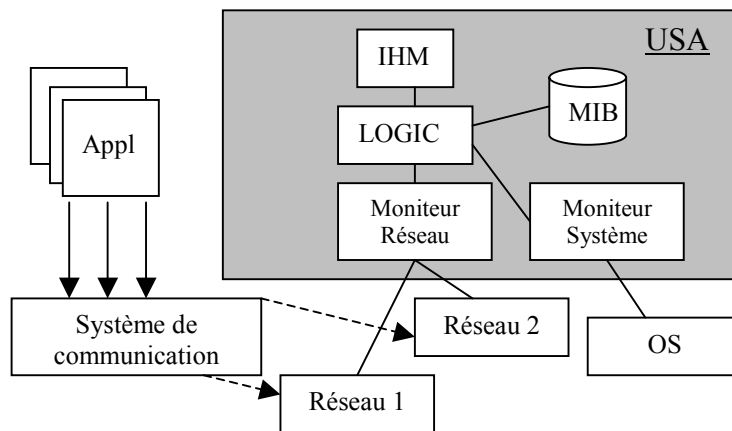


Figure II-1 - Architecture du système U.S.A.

Comme indiqué dans la Figure II-1, le système est composé de plusieurs modules permettant une architecture flexible.

- (a) La base de données (MIB). Elle correspond à l'ensemble des connaissances acquises par le système. Elle rassemble :
 - Des informations concernant les applications du système :
 - nom de l'application
 - le type de service demandé par l'utilisateur : Best Effort, Service Statistique, Service Garanti, type d'application : adaptative ou non).
 - Des informations sur les réseaux accessibles :
 - nom du réseau, bande passante, délai, BER, coût, possibilité de réservation

- (b) Des modules de détermination des ressources du système de communication (Moniteurs). Ils ont pour but d'évaluer et de déterminer les valeurs des paramètres contenus dans la base de données. Peu de détails sont fournis sur ces moniteurs ; toutefois l'utilisation d'une extension de l'application *ping* est proposée.
- (c) Une interface graphique (IHM). Elle permet d'obtenir le niveau de satisfaction de l'utilisateur pendant l'utilisation d'une application. Si le bouton « non satisfait » est pressé, l'information est sauvegardée, et la configuration du système réactivée avec de nouveaux paramètres afin de proposer une solution « satisfaisante » à l'utilisateur. C'est grâce à ce mécanisme, (LOGIC), que l'adéquation entre les ressources du système de communication et les besoins en qualité de service de l'application est évaluée.

L'architecture U.S.A. explore une nouvelle façon de gérer la qualité de service, en étant *réactive* plutôt que *prédictive*. Une assistance à la décision est offerte à l'utilisateur, contrairement aux autres approches qui tentent d'automatiser le processus de réservation de ressources et de renégociation. Ainsi, l'utilisateur aura toujours à prendre la décision finale concernant le niveau de service et le coût qu'il souhaite obtenir. La plate-forme de test est écrite en *Java* et en *C* (pour certains modules de mesures).

USA est une des premières plates-formes à proposer une sélection automatique de la meilleure interface réseau disponible en fonction des besoins d'une application. La construction de la base de connaissance est faite, entre autre, à l'aide d'envois de paquets permettant de tester les réseaux accessibles. L'auteur décrit les techniques utilisables pour évaluer les paramètres de ces réseaux tel que le délai, la bande passante disponible, le taux d'erreur, etc. Cette plate-forme est compatible avec les applications classiques puisque la redirection des connexions se fait au niveau IP. Toutefois aucune technique concernant l'adressage des paquets n'est détaillée. Les auteurs ont proposé ultérieurement une évolution de l'architecture nommée TOMTEN (Total Management of Transmissions for the End-User) [ARD98] qui prend en charge des considérations d'adaptation des applications.

Le principal point faible de cette architecture est que, parmi les différentes interfaces réseau disponibles, une seule est activable à la fois. Ainsi, les applications multimédias ne peuvent tirer profit des différents réseaux disponibles pour le transport de leurs flux. La problématique des applications multimédias n'est pas donc couverte par cette architecture.

2.2 Composite Radio System – CRS

L'architecture « Composite Radio System » de Motorola est l'une des premières architectures multi-réseaux définies dans le monde industriel. A ce titre, ce paragraphe s'attache à décrire le système en détails. Toutefois, étant encore aux prémices de son développement, de nombreuses informations sur son fonctionnement interne restent confidentielles, et ne peuvent être divulguées.

2.2.1 Motivations

Spécialisée pour les communications mobiles, l'architecture CRS vise à intégrer les nouvelles technologies, de plus en plus nombreuses, de communications sans fil avec des interfaces air et des architectures spécifiques (réseaux cellulaires, diffusion, ...).

Chaque réseau offre des services spécialisés :

- Le GSM (Global System for Mobile Communications) est adapté aux transport de la voix pour les mobiles,
- le GPRS (Global Packet Radio Service) est adapté aux services mobiles de données,
- le DVB-T (Terrestrial Digital Video Broadcasting) à la diffusion,
- Hiperlan pour les réseaux locaux sans fil, ...

L'architecture « Composite Radio System » permet d'utiliser plusieurs réseaux sans fil hétérogènes disponibles sur un terminal. Ce système devrait permettre d'optimiser l'utilisation de la bande passante en choisissant le réseau d'accès approprié en fonction des besoins de l'application, des capacités du terminal de réception, et de la disponibilité du réseau. L'utilisation concomitante de plusieurs interfaces est possible.

Comme ces différents réseaux ne possèdent pas les mêmes caractéristiques, en termes de qualité de service, de fiabilité, et de coût, le système devrait permettre de router les flux à destination d'un terminal composite vers le réseau le plus adapté. Les critères de sélection sont l'état de congestion du réseau, le coût, les préférences de l'utilisateur et le type d'application utilisé. De plus, un routage dynamique étant prévu, la technologie de transmission pourrait changer durant la communication, sans toutefois provoquer d'interruption du service.

2.2.2 Technologies radio candidates à l'architecture CRS

Les principaux réseaux mobiles et leurs caractéristiques accessibles en Europe sont résumés ci dessous :

- GSM destiné principalement aux communications vocales mobiles offre une bande passante de 9,6 kbit/s pour les données.
- GPRS adapté pour les communications de données. Le débit maximum annoncé est de 171 kbit/s, avec une bande passante partagée par le nombre d'utilisateurs connectés.
- DVB-T est un système de diffusion numérique (DBS), initialement conçu pour la diffusion de la télévision numérique. Il offre une bande passante maximum de 12Mbit/s aux utilisateurs nomades qui doit être partagé entre les applications.
- UMTS est la base des réseaux mobiles de 3^{ème} génération. Il offre une bande passante de 8kbit/s à 2Mbit/s au maximum pour les utilisateurs nomades.
- Hiperlan2 : Technologie radio (arrivée à sa version 2) de communication à large bande spécifié par l'ETSI. Hiperlan émet dans la bande des 5 GHz et permet d'atteindre un débit de 54 Mbit/s. Réservé aux réseaux locaux, il peut être déployé dans les lieux publics (comme les aéroports), ou plus généralement dans les bureaux en remplacement des réseaux filaires.

Les chiffres fournis ci dessus sont très dépendants du dimensionnement des réseaux opérationnels, du nombre d'utilisateurs du service, de leur mobilité, et des besoins des applications.

2.2.3 Architecture CRS

« Composite Radio System » permet d'utiliser plusieurs réseaux d'accès à la fois et assure la continuité du service dans la limite, bien sûr, de chacun des réseaux d'accès. Dans ce but, l'architecture contient une entité appelée « Composite Access Server » (CAS), dont la fonctionnalité est de router les données d'une application sur le réseau le plus approprié. Le choix de la technologie employée est fait après négociation entre le CAS et le terminal mobile. Ce choix dépend de nombreux critères, comme l'état de congestion des réseaux sous-

jacents, les besoins applicatifs, les capacités du terminal, les coût d'accès, ... La figure ci-dessous donne une vue d'ensemble de l'architecture CRS.

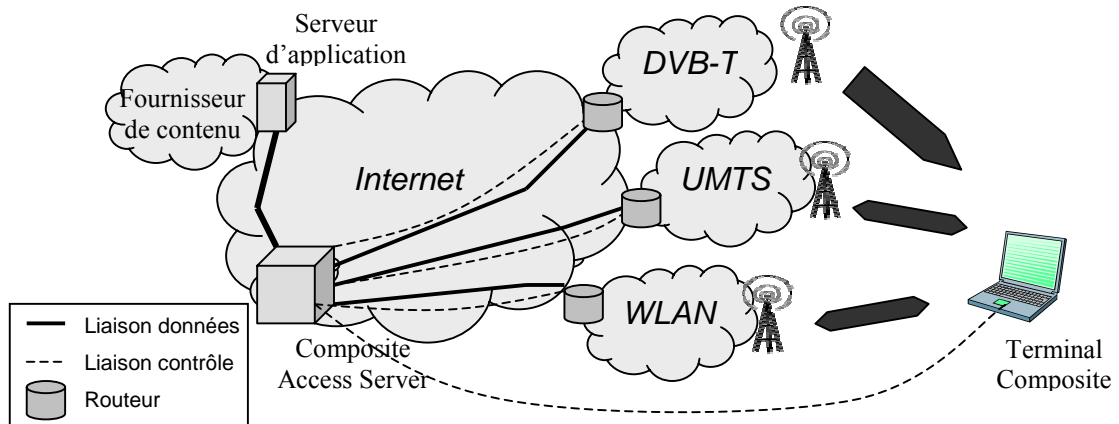


Figure II-2 - Architecture globale de CRS

Comme indiqué dans la Figure II-2, le routage des données entre l'Internet et les réseaux d'accès des opérateurs est assuré par le CAS. A cette fin, une extension de *IP mobile*, appelée « Composite Mobile IP » (CMIP), est utilisée.

Routage :

Aujourd'hui la mobilité dans les réseaux IPv4 est rendue possible par l'utilisation de « mobile IP » [PER96]. Les données à destination d'un mobile, envoyées à l'adresse qu'il possède dans son réseau d'appartenance (home network), sont interceptées par le « Home Agent » du réseau d'appartenance. Ce dernier, connaissant l'adresse temporaire obtenue par le mobile dans le réseau visité, encapsule ces données et les redirige vers cette nouvelle adresse. Une fois reçues, les données sont désencapsulées et délivrées à l'application du terminal mobile.

Dans l'architecture CRS, les données sont envoyées à destination d'une adresse IP fixe. Ces paquets sont interceptés par le CAS qui joue le rôle de « Home Agent ». Il possède une intelligence particulière, lui permettant de choisir la technologie d'accès la mieux adaptée. Les paquets sont alors, à l'instar de IP mobile, encapsulés (plus exactement : tunnelés) et envoyés à l'adresse IP possédée dans le réseau d'accès correspondant. La Figure II-3 décrit la procédure de routage des paquets à destination de l'utilisateur nomade.

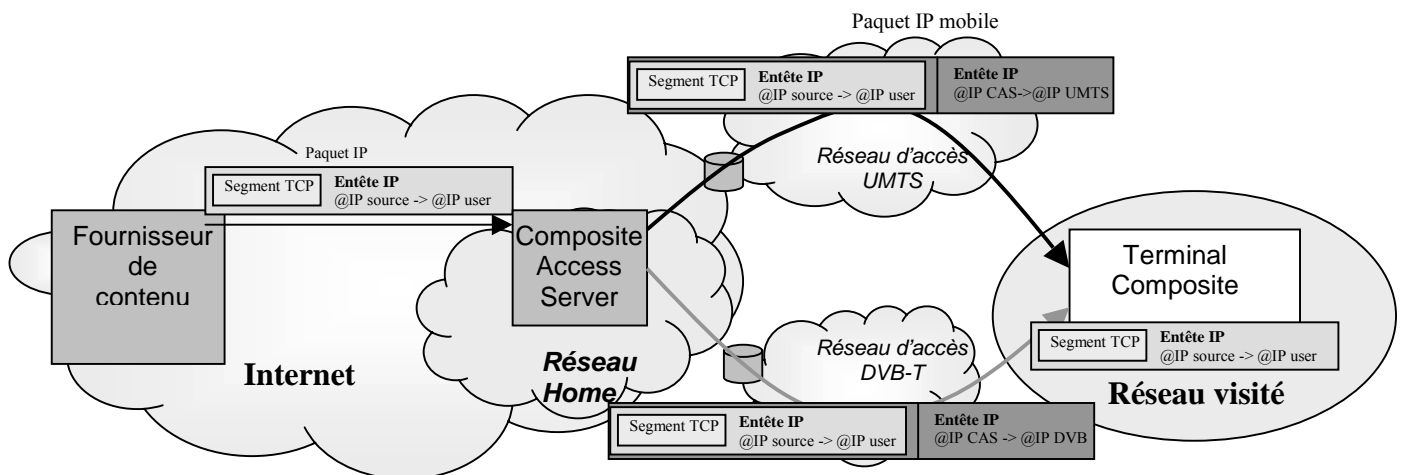


Figure II-3 - Routage dans « composite mobile IP »

Du point de vue de la source, le terminal mobile apparaît comme un terminal classique, et sur toute la connexion les segments TCP ne sont pas modifiés. Le choix du réseau d'accès est totalement transparent pour les couches application et transport. L'ensemble des protocoles de transport classiques (TCP, UDP, ...) sont supportés, ainsi que les applications qui les utilisent.

Les applications peuvent ainsi supporter les services offerts par CRS, ou au contraire, les ignorer totalement, ce de la même manière qu'une application Internet classique. La couche transport n'étant pas modifiée, n'importe quel protocole de transport peut-être utilisé. Cette architecture est totalement compatible avec le monde Internet, car seul le « serveur d'accès composite » et son protocole de routage associé sont propriétaires. Une simple installation des composants, par l'administrateur du domaine, est alors suffisante pour offrir ce service multi-réseaux.

2.2.4 Conclusion

Cette architecture étant en cours de développement, de nombreuses fonctionnalités ne sont pas dévoilées, comme par exemple les mécanismes de sélection du réseau, ou d'expression des besoins des applications. La gestion du multi-réseaux est faite au niveau 3 du modèle OSI, par l'utilisation de « tunneling » intelligent. Le masquage des réseaux d'accès rend ainsi le système compatible avec l'ensemble des applications Internet. En contrepartie de cette transparence, le comportement des protocoles de transport peut être perturbé par le changement de qualité de service induit par l'utilisation d'un accès radio. Les coupures de service induites par les changements de cellules, ou bien les rafales d'erreurs dues à un brouillage électromagnétique, rendent, par exemple, inopérant les mécanismes de contrôle de congestion et d'erreur de TCP. Afin de pallier ces problèmes, les protocoles de transport doivent être adaptés à ce type de liens.

Tant que de telles modifications ne seront pas implantées dans l'ensemble des souches des protocoles de transport de l'Internet, cette architecture ne nous semble pas des plus adaptée ; en effet, rien ne garantit qu'un utilisateur souhaitant communiquer avec un utilisateur mobile implantera une version modifiée de TCP, et que par conséquent la gestion de ses flux sera efficace. C'est pour tout cela, que nous préconiserons plutôt des solutions de niveau transport.

2.3 « Internet par satellite »

Ce tour d'horizon des systèmes classables dans la catégorie multi-réseaux ne serait pas complet sans parler des systèmes hybrides d'accès à l'Internet par satellite. Bien que techniquement possible, l'accès à l'Internet par l'intermédiaire d'un accès satellite unique n'est guère développé du fait du prix de la bande passante montante de ces systèmes. Afin de palier à cette contrainte, une technique classique consiste à utiliser une ligne de retour terrestre, généralement une ligne téléphonique. L'équipement d'un utilisateur se compose d'une carte de réception DVB-MPEG2, d'une parabole, et d'un modem RTC.

L'accès par satellite à l'Internet (Figure II-4), bien que concurrencé par les offres « hauts débits » ADSL, apporte une alternative comparable en termes de débits et de prix. Ce type de réseaux remplace avantageusement l'ADSL lorsque ce dernier ne peut être déployé pour des raisons techniques. De plus, la diffusion naturelle du Satellite permet d'être la solution qualité/prix la plus performante pour les applications « point à multi-points » (les premiers services de diffusion de télévision numérique personnels sont en train de se mettre en place). Le mode de transmission de données permet de recevoir de 500 Kbps à 45 Mbps (dans la réalité, les débits sont de l'ordre de 400 kbit/s à 2 Mbit/s en réception selon le service contracté). Bien que peu adapté aux applications bidirectionnelles interactives, du fait du

temps de latence élevé imposé par ce mode de transmission, ce type d'accès possède un potentiel important de développement dans les prochaines années.

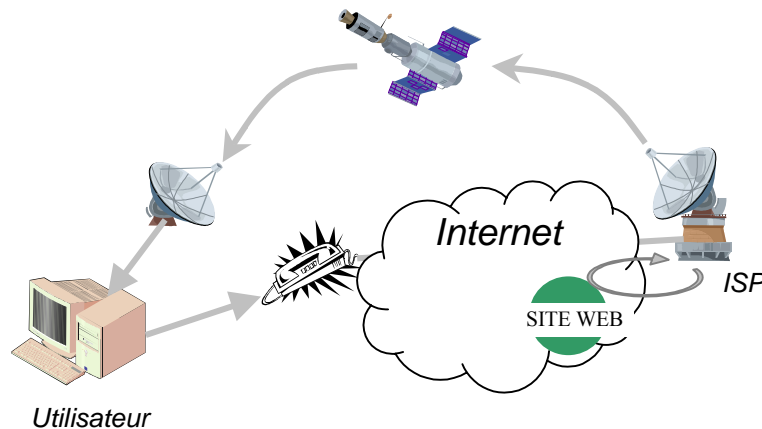


Figure II-4 - Architecture Internet par satellite

La configuration d'un système d'accès hybride « satellite – terrestre » peut être faite sans modification des protocoles de transport, en intervenant uniquement au niveau du routage :

- Côté utilisateur, une route par défaut est dirigée vers le routeur du fournisseur de service par l'interface reliée au modem.
- Côté fournisseur de service, les paquets à destination de l'utilisateur sont routés vers l'accès satellite.

Tous les paquets sortants (en provenance de l'utilisateur) sont dirigés vers la passerelle satellite du fournisseur de service par l'intermédiaire du réseau terrestre ; tous les paquets entrant (en provenance de l'Internet) sont reçus sur la liaison satellite.

Grâce à ce type de configuration, l'asymétrie de la liaison est masquée aux couches protocolaires supérieures, et les protocoles de transport n'ont pas à être modifiés, permettant une compatibilité totale avec le reste des utilisateurs.

Bien que séduisante, cette solution n'est pas la plus performante, en particulier concernant les connexions TCP qui pâtissent des contraintes spécifiques à la liaison TCP ainsi qu'à l'asymétrie de la liaison. Le contrôle de congestion et d'erreur de TCP est fortement perturbé du fait qu'il dépend du délai de la liaison (augmentation lente de la fenêtre de congestion, taille de fenêtre d'émission insuffisante limitant le débit des connexions, ...), et de ce fait TCP doit être optimisé en utilisant des options spécifiques définies dans ce but [RFC2488]. D'autre part, si le flux de données de la liaison satellite haut débit est trop important, le canal de retour terrestre pour les acquittements peut saturer et du coup limiter le débit de la connexion. Des mécanismes de réduction du nombre d'acquittements [BAL99], comme celui des acquittements sélectifs : un paquet acquittant plusieurs segments, diminuent le risque de congestion du canal de retour. Pour ces raisons, les protocoles de transport, et en particulier TCP, doivent être adaptés pour ce type d'accès en implantant les algorithmes précités.

En fin de compte, l'accès à un réseau par une liaison hybride satellite et terrestre possède l'avantage d'offrir une bande passante descendante à faible coût, mais au prix d'une interactivité considérablement dégradée par le délai de la liaison satellite. Les communications de type téléphonique ou bien les applications de jeux distribués sont ainsi fortement pénalisées par ce type de système (pour la voie de retour en tout cas). La liaison terrestre utilisée comme canal de retour, offre quant à elle, une qualité de service totalement

différente, avec un faible débit pour un court délai. Utilisés intelligemment, les flux à fortes contraintes temporelles (tels les flux audio ou les notifications de position d'un jeu distribué) pourraient être redirigés par la passerelle satellite vers la liaison terrestre, plutôt que vers la liaison satellite, ceci afin de palier aux contraintes précédemment citées. Toutefois, cette solution n'est pas si simple, et suppose de repenser l'architecture des protocoles de transport en leur permettant de choisir le réseau le plus adapté aux contraintes des flux transportés.

3 Problématique et besoins du protocole

3.1 Les mécanismes de sélection du réseau

Aujourd'hui il est possible de posséder plusieurs interfaces réseaux sur son ordinateur, et ce cas devient même courant, avec l'émergence des nouveaux réseaux Hertziens offrant mobilité pour la plupart et une large bande passante dans certains cas.

Citons par exemple le cas des ordinateurs portables qui, au fil des années, se sont vu équipés de nouvelles technologies de communications. L'évolution commence avec l'ère des prises de connexion directe, de type *câbles parallèles*, d'un ordinateur portable avec une station de travail pour y télécharger des données. L'ère de l'explosion des réseaux locaux offrira aux portables de nouvelles interfaces, type *cartes ethernet*, lui permettant de se connecter directement au réseau d'entreprise pour accéder à toutes les ressources disponibles. Peu de temps après, le télétravail et le besoin de connectivité en dehors de l'entreprise commençant à se développer, il n'en faudra pas plus pour voir fleurir des *modems* permettant une connexion par la ligne téléphonique de n'importe quel foyer au réseau global qui se construit. Finalement, l'ère nouvelle des communications *mobiles*, ouvrira la voie à la connectivité en déplacement et par conséquent de nouvelles interfaces (GPRS/UMTS, 802.11, etc.) permettant le travail en train ou en bus apparaîtront. Au final, même si les connexions directes par câble ne sont plus utilisées, ce sont souvent trois interfaces qui se retrouvent sur la même machine, avec chacune ses caractéristiques et ses applications propres. Le téléchargement de données, par FTP par exemple, sera plutôt destiné à être employé lors d'une connexion au réseau local à cause des débits nécessaires, la lecture des mails pourrait être faite aussi à distance par modem, et la consultation de son agenda n'importe où, avec n'importe quelle interface.

Il est ainsi possible de configurer son environnement de travail pour indiquer aux différents types d'applications quelle est l'interface qui lui est réservée. A l'heure actuelle, ces configurations nécessitent un savoir faire et une bonne maîtrise des environnements pour parvenir à ses fins. Dans les environnements courants, tels que *Windows* ou *Linux*, il n'est pas rare de devoir dupliquer un logiciel pour en obtenir deux souches capables d'accéder à des interfaces différentes !

La première spécificité d'un protocole multi-réseaux parallèle est tout d'abord de gérer *l'accès concurrent* aux interfaces réseau disponibles sur un même ordinateur, *sans nécessiter de configurations particulières*. Cette première considération, bien qu'en apparence simple implique de profonds changements dans la façon de gérer les accès réseaux des systèmes actuels.

La plupart des systèmes actuels sont prévus pour accepter de multiples interfaces réseaux, et leur utilisation concomitante, dans la mesure où ces systèmes sont parfois utilisés comme routeurs. Toutefois, cette fonctionnalité n'est généralement pas directement fonctionnelle,

puisque les applications utilisent la pile de protocole activée par défaut à l'initialisation du système.

Hormis le fait de laisser à l'utilisateur la responsabilité de configurer son système, une solution idéale pourrait être de proposer une pile de protocoles génériques, activée par défaut, qui se chargerait d'associer dynamiquement aux applications les interfaces dont elles ont besoin. Ce choix est celui fait pour l'architecture USA [LAN98] décrite précédemment.

Deux optiques différentes peuvent être prises dans la réalisation de cette association des interfaces de communication et des besoins applicatifs :

- La première consiste à laisser à l'utilisateur le soin de configurer de manière figée les redirections que devra faire le protocole vers les interfaces en fonction de l'application utilisée.

Par exemple, en reprenant le cas précédent, le protocole multi-réseaux sera configuré pour, dans le cas de l'application FTP : initialiser l'interface de la carte Ethernet et y rediriger les paquets qui en émanent ; Dans le cas de l'application de mail : initialiser la carte Ethernet si possible, sinon le modem et y rediriger les paquets ; etc ...

Cette solution, bien que robuste, nécessite une configuration manuelle et exhaustive de tous les logiciels communiquant de l'utilisateur, tâche qui peut s'avérer lourde et techniquement complexe.

- La seconde, est une solution entièrement automatisée, comme nous le proposerons dans la suite de ce document. Le choix peut être opéré de plusieurs manières, soit en fonction des réseaux accessibles, soit en fonction des besoins en qualité de service des applications, soit en fonction des deux.

La Figure II-5 résume ces différentes possibilités, de la configuration manuelle (en haut à gauche), à la configuration entièrement automatique (en bas à droite).

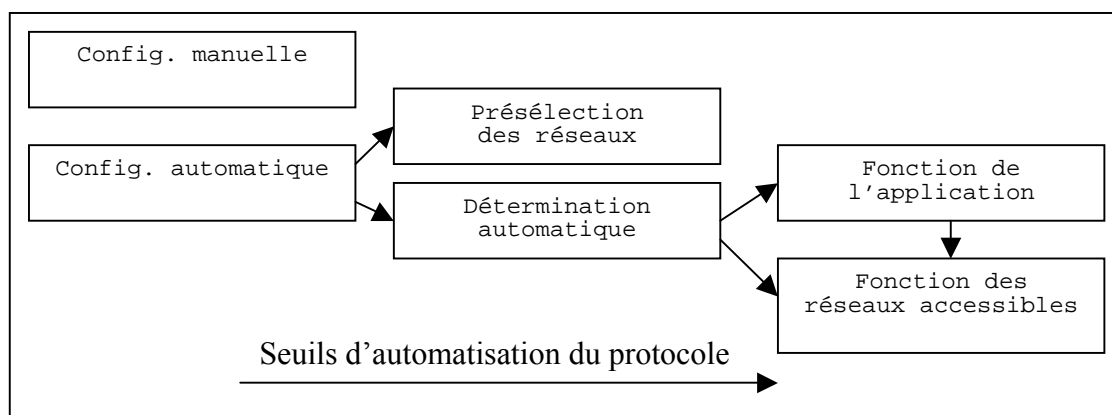


Figure II-5 - Niveaux d'automatisation d'un protocole multi-réseaux

3.2 Adaptation du protocole de transport au réseau sous-jacent

L'un des principes fondamentaux des architectures de communication en couches, tel le modèle OSI, ou bien sa version simplifiée définissant l'Internet, est que chaque couche fournit un service rendant transparent à la couche supérieure les services rendus par les couches inférieures. En d'autres termes, dans le cas de la couche réseau, le service fourni a

pour but de banaliser toutes les technologies de communication sous-jacentes pour offrir une vision uniforme à la couche transport.

Ce principe a jusqu'ici trouvé sa justification dans la possibilité d'interconnexion de réseaux de nature très différente, dont le meilleur exemple est aujourd'hui l'Internet.

Mais ceci était sans compter avec les besoins naissants de qualité de service. Si effectivement les protocoles de transport fonctionnent sur tous les types de réseaux actuels, que ce soient des réseaux câblés, optiques ou radios, dans certains environnements, le fonctionnement de certains protocoles de transport – nous pensons en particulier à TCP – est limité par les caractéristiques des liaisons de communication empruntées. Par exemple, le délai important de propagation d'une liaison satellite perturbe le fonctionnement de certains mécanismes de TCP, et n'est malheureusement pas compensable par l'utilisation d'autres techniques de niveau liaison. Dans un autre cas, c'est le taux d'erreur totalement variable et de nature imprévisible des réseaux mobiles qui dérouté les mécanismes de ce protocole originellement conçu pour emprunter des réseaux filaires, c'est à dire à délai et taux d'erreur faibles. On pourrait encore citer le cas des réseaux formés par des constellations de satellites basse orbite, tel que les projets Skybridge ou Teledesic, qui ont la particularité d'offrir des délais très variables induits par le mouvement des satellites, ou bien les options de TCP permettant de choisir la taille de paquet la plus adaptée au réseau sous-jacent ; mais ceci ne ferait qu'agrandir la liste déjà longue des raisons prônant des solutions allant à l'inverse de ce principe fondateur.

Les mesures suivantes (Figure II-6) montrent le comportement du protocole TCP lorsqu'il est utilisé dans ces configurations particulières de réseau. La première figure mesure le retard emmagasiné par un flux MPEG3 de 128 kbit/s (qualité CD), transporté par TCP, lorsqu'il emprunte les différentes configurations de réseaux. Ce retard est à comparer avec le délai physique du réseau, qui dans un cas optimal, devrait être le seul retard subi par les paquets. Les simulations ont une durée de 2 minutes, temps suffisant pour obtenir un comportement stable du protocole dans ces conditions. La seconde figure montre l'évolution de la fenêtre de congestion de TCP, en partie responsable des mauvaises performances, durant la simulation.

Le RTT long de la liaison satellite crée une limitation en terme de bande passante. Sa valeur est de 520 ms compte tenu de toutes les interfaces traversées. La bande passante théorique peut être approximée par la formule suivante :

$$\text{Taille maximum de fenêtre} = \text{Bande Passante} \times \text{RTT}$$

Avec une taille de fenêtre de 64 kbit/s, valeur par défaut des implémentations de TCP, nous obtenons une bande passante théorique de 126 kbit/s inférieure à celle générée par l'application. De ce fait un certain retard, visible sur la première figure (courbe de gauche) est emmagasiné par le flux lors de sa transmission sur la liaison satellite. Au contraire, aucun retard n'est à signaler lorsque le réseau terrestre est utilisé : la courbe des retards est confondue avec l'axe des ordonnées.

En ce qui concerne la liaison du réseau mobile, le taux d'erreur/bits avoisinant les 10^{-6} perturbe le fonctionnement du mécanisme d'évitement de congestion [RFC3155]. Les réductions successives de la fenêtre imposent une taille de fenêtre si petite que le débit généré est bien en deçà du débit du flux MP3. L'importance du retard accumulé par ce flux est visible sur la première figure (courbe de droite). La deuxième figure montre le comportement chaotique de la fenêtre dans le cas des réseaux hertziens. Sur un réseau terrestre (courbe du haut), la fenêtre croît régulièrement, sur un réseau satellite de grosses variations imposées par

le délai sont subis, et enfin, dans le cas du réseau mobile (courbe du bas), la fenêtre n'arrive pas à croître.

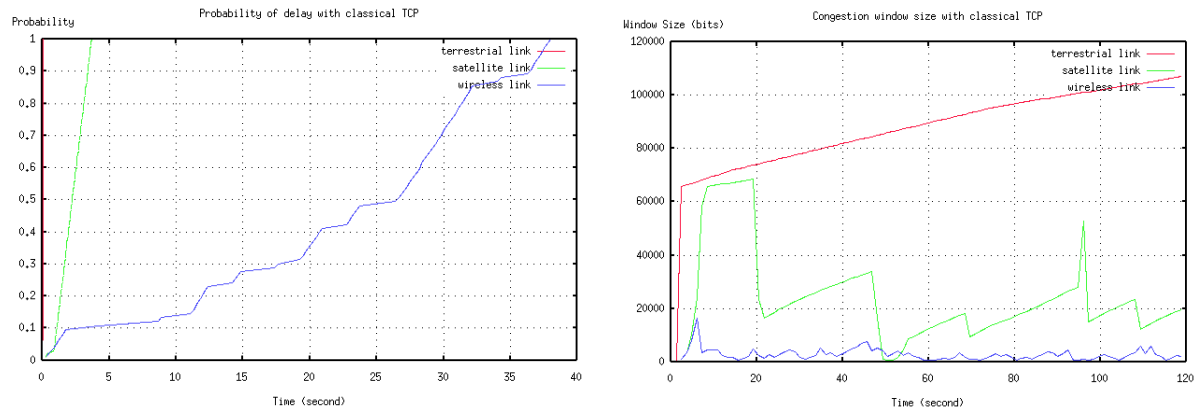


Figure II-6 - Comportement d'une souche TCP sur des réseaux Hertiens

Un comportement adapté du protocole de transport aux caractéristiques de la liaison, comme par exemple une augmentation de la taille maximum de la fenêtre TCP, ou bien l'utilisation de mécanismes d'acquittement différents pour les liaisons satellite, voire même celle d'un contrôle de flux de type *rate based* [HEN97], sont alors préconisés pour obtenir un fonctionnement plus efficace.

3.3 Les besoins en synchronisation

Malgré la forte croissance du trafic multimédia au sein des informations transportées par l'Internet, les protocoles de transport actuels n'intègrent pas la notion de flux multiples. Cette notion nous paraît essentielle, car sans elle il est impossible d'offrir de mécanismes de synchronisation de flux nécessaires au transport de ces données particulières que sont les données multimédia.

Evoquée dans le chapitre d'introduction, la synchronisation reste un des points essentiels du point de vue de la qualité finale d'une présentation multimédia. La synchronisation temporelle est bien évidemment la plus délicate à garantir de part la nécessité de son contrôle tout au long de la présentation, les synchronisations spatiale et hypermédia restant principalement des problèmes de signalisation.

Dans un environnement multi-réseaux, bien plus qu'ailleurs, la gestion de ce type de synchronisation revêt un aspect particulier compte tenu des différences de qualité de service offertes par l'ensemble des technologies de réseaux existantes.

3.3.1 Synchronisation intra - flux

La synchronisation intra-flux, première composante de synchronisation temporelle, consistant principalement à contrôler la gigue occasionnée par la traversée du réseau, est généralement contrôlée par l'application elle-même.

La gestion d'un stockage intermédiaire de données est principalement employée pour absorber le retard ou l'avance d'une donnée délivrée par le protocole. Les données sont stockées dans un tampon, entre leur livraison par le protocole et leur délivrance aux périphériques de présentation (carte vidéo, carte son, ...) . De cette manière, une donnée délivrée trop tôt peut « perdre » un peu de temps dans ce buffer, ou bien inversement, le retard d'une donnée peut être compensé par l'avance accumulée des autres données présentes dans

le buffer. En contre partie, ce mécanisme engendre du délai, dont la quantité est proportionnelle à la taille du buffer.

Les mécanismes mis en place pour ce type de synchronisation sont bien connus et largement détaillés dans [OWE96]. Ils n'incombent généralement pas au protocole de transport, puisqu'ils nécessitent une précision dans les délais de livraison aux périphériques de présentations des données, qu'il est impossible d'atteindre dans un système d'exploitation asynchrone classique. En effet, lors de la traversée des nombreuses couches logicielles entre le protocole de transport et les pilotes des périphériques, les données emmagasinent des retards très variables selon l'état du système, ce qui par conséquent induit de la gigue. Il serait inutile de supprimer la gigue au niveau du protocole de transport et de devoir le faire à nouveau au niveau de l'application.

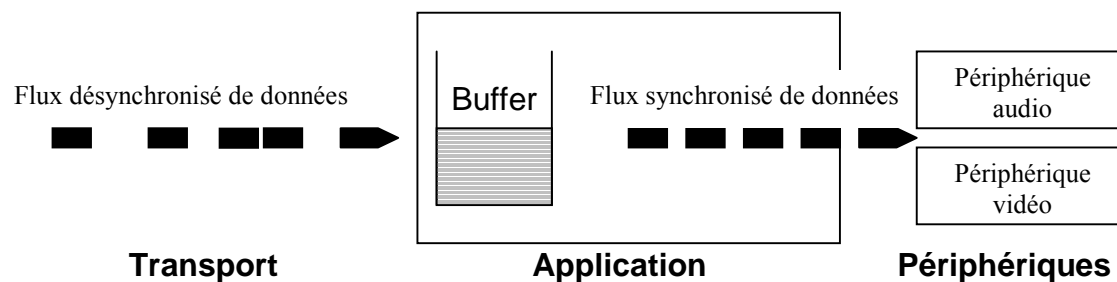


Figure II-7- Synchronisation intra-flux réalisée par l'application

3.3.2 Synchronisation inter - flux

Deuxième aspect de la synchronisation temporelle : la synchronisation inter-flux. En l'état actuel des développements, il n'existe pas de moyens simples permettant de synchroniser plusieurs flux à destination d'un même ordinateur. Ce constat, bien que surprenant, n'est en fait pas si étrange, puisqu'à l'heure actuelle, les transmissions des données, qu'elles soient multi-flux ou non, sont faites sur des réseaux à qualité de service uniforme. En effet, dans le meilleur des cas un réseau garantissant une certaine qualité de service, un VPN par exemple, est utilisé et tous les flux observent un comportement similaire du fait qu'ils empruntent le même chemin, et subissent en conséquence la même qualité de service. La désynchronisation est alors très faible à l'arrivée, et peut être corrigée par l'application. Dans le cas le plus courant, tous les flux empruntent un réseau sans garantie de QoS, tel l'Internet, en suivant généralement le même chemin. Du fait de la maigre qualité des flux à l'arrivée, la synchronisation n'est pas le problème le plus important, et les flux peuvent être présentés tels quels. Généralement, lorsqu'aucune QoS n'est garantie un codage mixant les flux est privilégié, afin de supprimer les problèmes de synchronisation inter-flux.

Motifs de décalage

Lorsque les flux prennent des chemins différents, ou traversent des réseaux différents, voire utilisent des services à qualité de service différenciée [RFC2475], les problèmes de synchronisation sont aggravés. Si, comme cela est préconisé par notre approche multi-réseaux, plusieurs réseaux sont empruntés en fonction de la qualité de service requise par chacun des flux, les risques de désynchronisation sont encore plus importants.

Les motifs du décalage sont de plusieurs ordres, et sont induits par :

Le matériel utilisé :

Les cartes d'acquisition peuvent posséder des tampons qui mettent plus ou moins de temps à se remplir, engendrant ainsi un peu de délai au départ de l'application.

Les réseaux traversés :

Les différents réseaux traversés ne fournissent pas tous la même qualité de service, et le délai peut être engendré par plusieurs paramètres :

- délai physique du réseau : la différence peut être grande entre un réseau terrestre et satellite,
- La bande passante disponible,
- Le délai d'établissement de connexion (uniquement pour certains réseaux).

La différence de QoS requise par les flux :

Si une fiabilité totale type TCP est requise alors que le réseau peut perdre des paquets, un délai supplémentaire sera engendré par la reprise des pertes par le protocole.

Des différences d'implantation des mécanismes de contrôle de flux et de congestion peuvent d'autre part influencer sur le délai de récupération des erreurs. Les implantations de TCP original, TCP Tahoe, TCP Reno n'offrent pas les mêmes délais de récupération des données.

Il apparaît ainsi, qu'un mécanisme de synchronisation est indispensable à une telle configuration.

Synchronisation par l'application ou le transport ?

De la même manière que pour la synchronisation intra-flux, on pourrait imaginer laisser cette tâche à l'application.

Ceci ne nous paraît pas judicieux, et ce pour plusieurs raisons:

- Premièrement, afin de faciliter le développement des applications multimédias, déléguer la synchronisation inter-flux à une autre entité ne peut être que souhaité.
- Deuxièmement, l'application n'est pas la mieux placée ! En effet, le décalage engendré par le retard systématique d'un flux, ne peut être compensé qu'en ralentissant à son tour les autres flux, sous peine de voir saturer les buffers de l'application très rapidement. Lorsqu'un flux d'une application multimédia est en retard, soit il est ignoré et la présentation est présentée de manière incomplète, soit c'est l'ensemble de la présentation qui se trouve retardée pour préserver la qualité globale. Dans ce dernier cas, les données provenant des autres flux doivent être stockées tant qu'elles ne peuvent être présentées. L'émetteur n'ayant pas moyen de connaître l'état des buffers de l'application réceptrice, le débit des flux les plus avancés ne sera pas modifié ayant pour conséquence de provoquer la saturation de l'application.
La réduction du débit d'émission s'imposant, quel autre mécanisme est le mieux placé que le contrôle de flux, rôle inhérent au protocole de transport, pour réaliser cette action ?

Pour ces raisons diverses il apparaît indispensable d'implémenter un mécanisme de synchronisation au sein du protocole de transport multi-réseaux lui-même.

Synchronisation logique ou temporelle ?

Pour les mêmes raisons que citées précédemment, une synchronisation temporelle réalisée au niveau du protocole de transport ne serait pas judicieuse, puisque :

- D'une part, aucun réseau n'offre actuellement de garantie forte sur les délais. Seuls des délais bornés peuvent dans certains cas être obtenus.
- D'autre part, les systèmes d'exploitation asynchrone ne garantissant pas non plus des contraintes temporelles fortes, les perturbations introduites par la traversée du système par les données conduiront nécessairement à une synchronisation finale, comme cela est proposé dans [OWE96].

3.4 Les besoins en fiabilité partielle

La qualité de service offerte par les protocoles actuels en termes de qualité de service est limitée aux services offerts par TCP et UDP, où TCP garantit une fiabilité totale des transmissions, et UDP ne garantit rien. Ce constat n'a rien de surprenant en considérant que d'une part la majorité des réseaux utilisés actuellement sont des réseaux filaires possédant un taux d'erreur très faible, et que d'autre part, avant l'émergence toute récente du multimédia, les données transportées sur les réseaux étaient principalement des fichiers nécessitant une fiabilité totale.

Mais aujourd'hui les choses ont changé, la tendance est en train de s'inverser. Les technologies des réseaux mobiles de 3^{ème} génération, tel UMTS [UMT], semblent être l'une des voies de développement les plus prometteuses pour les futurs réseaux d'accès. D'autre part, le rapport existant entre le trafic de type données et le trafic de type multimédia est en train de s'inverser, preuve en est l'enthousiasme suscité par la diffusion de musique sur l'Internet.

D'un côté, ces nouveaux réseaux d'accès Hertzien, du fait de la technologie employée possèdent certaines particularités en terme de fiabilité. Tout d'abord la fiabilité de ces réseaux est très variable, le taux d'erreur par bit variant de 10^{-3} à 10^{-6} [Table I-4] selon le service demandé, la mobilité de l'utilisateur, le lieu d'accès et les perturbations avoisinantes. De telles valeurs ne sont pas classiques pour les réseaux terrestres, et les protocoles ne sont généralement pas prévus pour ces conditions extrêmes. Toutefois, les liaisons par satellite, bien que moins problématiques du fait de la plus grande largeur de bande disponible, avoisinent, dans certaines conditions, ces valeurs.

D'un autre côté, la nouvelle problématique soulevée par la transmission de données multimédia n'est pas intégrée dans les protocoles actuels. Auparavant, pour la transmission de données, tels les fichiers, le paramètre le plus important était la fiabilité. Peu importe la durée d'un transfert, pourvu que l'intégrité du fichier soit respectée.

Pour les applications multimédia, temps réel ou non, de part la quantité de données échangées, la problématique n'est plus seulement de transférer l'intégralité des données, mais aussi plutôt d'effectuer ce transfert rapidement !

Comme les flux multimédia peuvent tolérer un certain seuil de pertes, comme cela est montré pour MPEG [roj98a], l'idée sous-jacente est d'éviter de perdre trop de temps à récupérer des données perdues, si elle ne nuisent pas à une présentation correcte du flux. Les délais de bout en bout peuvent être réduit de manière considérable.

La Figure II-8 présente le retard emmagasiné lors de la présentation de flux vidéos diffusés sur un lien satellite possédant un taux d'erreur avoisinant les 10^{-7} . Les formats d'encodage utilisés sont MJPEG et H263 et la durée des extraits est d'une minute. Le retard est mesuré pour chaque paquet dès sa livraison à l'application. Un retard d'une seconde indique que ce paquet a attendu une seconde (moins le délai de transmission) dans le buffer du protocole avant d'être délivré à l'application.

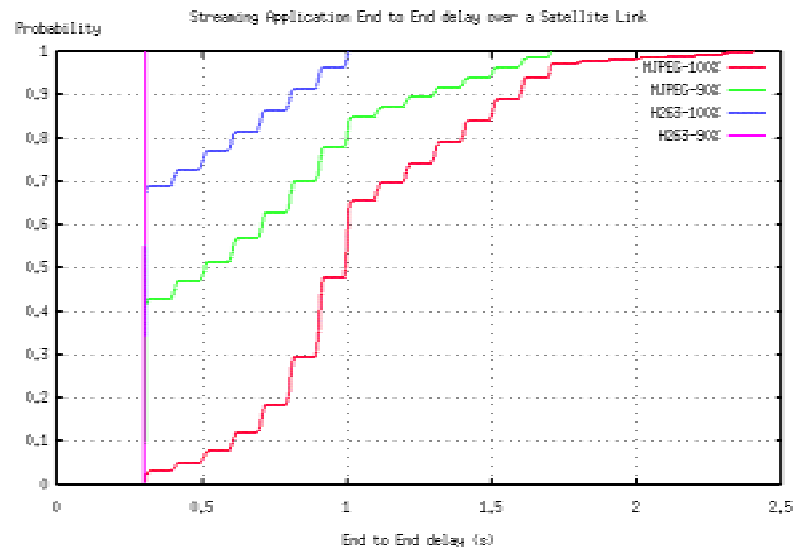


Figure II-8 - Retards emmagasinés par la transmission de plusieurs flux audio sur une liaison satellite

Pour chaque type d'encodage utilisé, le délai engendré par la garantie d'une fiabilité totale est comparé à celui généré par la garantie de recevoir 90% des données correctement. Si le retard pris par le flux MJPEG est supérieur à celui pris par le flux H263, cela s'explique par le fait que la taille des paquets générés est supérieure à celle générée par H263, et que par conséquent la probabilité d'erreur sur un paquet augmente. Par contre, on peut s'apercevoir que dans le cas d'une transmission fiable (courbe de droite) seulement 50 % des paquets du flux MJPEG ont été délivrés dans un délai inférieur à une seconde, contre que 80 % des paquets dans le cas de la fiabilité à 90%. De plus 50% des paquets du flux partiellement fiable sont délivrés avant une $\frac{1}{2}$ seconde, contre 5% pour le flux fiable. Le flux H263, ne subissant pas un taux de perte inférieur à 90 % peut être délivré directement par le protocole, et ne subit donc que le délai physique de la liaison.

On pourrait ainsi imaginer réduire les délais jusqu'à la barrière du délai physique en utilisant un protocole sans garantie, tel UDP. Toutefois, cette hypothèse n'est pas justifiée, car si le nombre de pertes est trop important, le décodage du flux peut ne pas pouvoir être fait, ou bien sa présentation risque d'être complètement perturbée. Par exemple, la diffusion d'un flux MP3, avec le protocole UDP, sur un réseau mobile expérimentant un taux de perte de 10^{-4} n'a aucun intérêt car le flux serait impossible à écouter.

Si l'on veut réduire le délai de bout en bout, il est donc nécessaire d'offrir un mécanisme qui permette d'accepter un certain seuil de pertes, mais qui permette aussi d'assurer la fiabilité du flux si le taux d'erreur observé descend en dessous de ce seuil. C'est ce que nous appellerons la *fiabilité partielle*.

4 L'architecture du protocole

4.1 Choix d'architecture

Le paragraphe précédent a énuméré un ensemble de contraintes auquel devra répondre une architecture de protocole de communication multimédia multi-réseaux. Aucun protocole multi-réseaux, tel que nous l'entendons, n'existe actuellement, et la complexité des mécanismes à mettre en œuvre n'est sans doute pas étrangère à ce constat. En effet, de nombreuses problématiques soulevées par cette définition du multi-réseaux sont encore à l'étude au sein de projets de recherche [DIA01], ou bien en cours de définition au sein d'organismes officiels [RFC2488, RFC3155].

L'architecture que nous proposons s'attache à donner une solution pour chacun des points abordés précédemment, et respecte une vision modulaire du protocole. La compatibilité des « briques » constituant le système a été, bien évidemment, l'un des principaux éléments moteur de la définition de cette architecture.

Comme évoqué dans l'introduction, cette architecture propose la définition d'un protocole de transport, qui par conséquent est déployé aux extrémités du réseau, c'est à dire sur les entités communicantes. L'une des hypothèses qui sera faite tout le long de la définition de l'architecture est que les utilisateurs peuvent communiquer par l'intermédiaire de plusieurs réseaux séparés, comme par exemple l'Internet, une liaison téléphonique, et une liaison satellite. Par conséquent, plusieurs interfaces réseaux sont disponibles sur les machines.

4.2 Les mécanismes de sélection du réseau

Dans l'optique d'offrir une solution capable de sélectionner dynamiquement le réseau le plus approprié aux paramètres de qualité de service demandés par une application multimédia, une architecture de communication est proposée Figure II-9. Elle permet, en fonction de la QoS de chaque flux de l'application et de la connaissance des réseaux accessibles (aussi bien au niveau de l'émetteur que du récepteur) de sélectionner le réseau le plus approprié pour chaque flux de données.

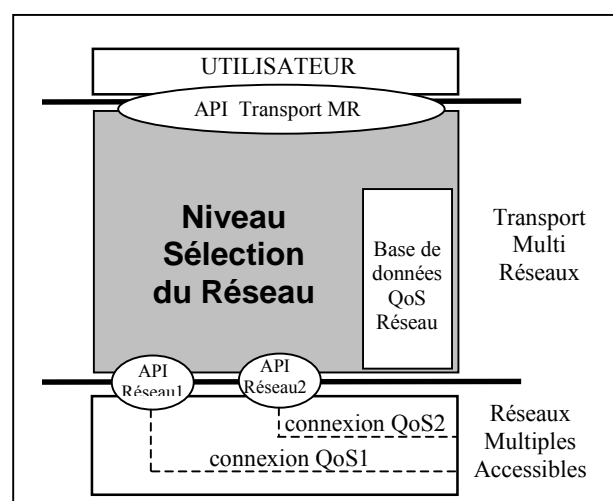


Figure II-9 - Architecture du protocole multi-réseaux

Le principal problème consiste alors à choisir « le bon » réseau. Pour cela, les paramètres de qualité de service de chaque réseau accessible doivent être déterminés de manière dynamique. Au préalable, une étude sur les réseaux de communication, et en particulier ceux reposant sur de nouvelles techniques, comme les réseaux satellitaires (GEO / LEO), mobiles, etc. a été réalisée. Il apparaît ainsi, qu'un ensemble de paramètres peut être isolé afin de caractériser la QoS de ces réseaux [Table II-1].

	Débit	RTT	Gigue	BER	Sécurité	Diffusion	Coût
ATM/CBR	Garanti	Faible	Faible	Faible	Oui	Non	Elevé
ATM/UBR	*	Faible	Faible	Faible	Oui	Non	Elevé
RNIS	64 Kb	Faible	Faible	Faible	Oui	Non	Bas
Satellite	> 64 Kb	Elevé	Faible	Elevé	Non	Oui	Moyen
Mobile	< 64 Kb	Faible	Faible	Très Elevé	Non	Non	Moyen
Internet	*	*	Elevé	Elevé	Non	Non	Bas

Table II-1 - QoS offerte par certains services réseaux

Il est clair que certains paramètres n'ont pas de sens pour certains services réseaux comme par exemple la bande passante pour un service « best effort » fourni par un réseau IP. Toutefois, cette information reste intéressante et peut quand même influencer le choix du réseau. En effet, un réseau IP n'étant pas approprié au transfert d'un flux vidéo ou audio fortement contraint temporellement, et qui nécessite un débit garanti, ne devra pas être choisi.

Tous ces paramètres permettent de construire une base de connaissances contenant les caractéristiques des réseaux accessibles.

Il faut noter qu'à l'opposé de ce qui est décrit dans la Table II-1, la base de connaissances ne comportera pas de données qualitatives sur l'état des réseaux sous-jacents, mais bien les valeurs quantitatives de ces différents paramètres. Ces caractéristiques appartiennent à deux catégories différentes : Les statiques dépendant directement des drivers réseaux et fournies par le système, et les dynamiques qui doivent être mesurées (comme du délai de bout en bout). Finalement le choix du réseau peut être fait en fonction de tous ces paramètres, mais aussi en fonction de l'état du réseau à un moment donné.

Le protocole multi-réseaux effectue le choix du réseau pour chaque flux, en fonction d'une métrique établie suite à une campagne de mesures expérimentales des différents paramètres de QoS sur les réseaux actuels. Cette métrique évalue la distance entre les paramètres de qualité de service requis par l'application, et ceux fournis par chaque réseau accessible, la plus petite distance étant sélectionnée.

4.2.1 Méthode de sélection du réseau

Définir une méthode de sélection de l'interface réseau répondant aux besoins de l'application n'est pas trivial. Après de nombreux essais, nous avons opté pour une méthode simple qui, même si ce n'est pas la plus efficace parmi les méthodes envisagées, effectue quand même des choix pertinents. Cette méthode est décrite ci dessous [Figure II-10].

Application = définie par (Débit, RTT, gigue, fiabilité, Sécurité, Diffusion, Coût)

Sélection() : Soit une procédure de sélection des interfaces qui prend en paramètre

- un ensemble d'interfaces,
 - les critères de QoS à fournir,
- renvoi la liste des interfaces garantissant les critères.

Critères_QoS = (Coût, Sécurité, Diffusion, Débit, RTT, fiabilité, gigue)

Interfaces_sélectionnées = Interfaces disponibles

Répéter

Interfaces_testées = Interfaces_sélectionnées

Interfaces_sélectionnées = Sélection (Interfaces_testées, Critères_QoS)

Changer_de (Critères_QoS)

Tantque(|Interfaces_sélectionnées| > 0)

Activer parmi « Interface_testées » l'interface de moindre coût.

Figure II-10 - Méthode de sélection du réseau

L'application possède les valeurs des critères de qualité de service qui définissent le flux qu'elle va transmettre parmi :

- Le débit du flux généré,
- Le retard et la gigue tolérables,
- Le niveau de fiabilité acceptable,
- Un ensemble de services que doit fournir le réseau, comme la sécurité ou la diffusion,
- Et finalement, le coût maximal toléré pour la communication.

Une procédure de sélection des interfaces réseau est définie. Elle reçoit en paramètre une *liste d'interface réseau* et un *critère de QoS* à fournir. En retour, la *liste des interfaces* capables d'offrir ce service est renvoyée.

Par la suite, cette procédure est appliquée consécutivement à la liste des interfaces réseaux disponibles au moment du choix, afin de déterminer, par élimination, une liste d'interfaces capables d'assurer le service requis. L'interface du réseau offrant les coûts de communication les moins importants est ensuite sélectionnée parmi cette liste.

Un des paramètres importants de cet algorithme est l'ordre des critères de QoS utilisés lors des appels successifs de la procédure de sélection.

L'ordre suivant à été choisi : Coût, Sécurité, Diffusion, Débit, RTT, gigue, fiabilité

Les réseaux dont les coûts de fonctionnement sont trop élevés doivent être éliminés en premier. Les réseaux n'offrant pas les services requis par l'application (besoins en sécurité, diffusion, etc.) sont à leur tour éliminés de la sélection. Ensuite, afin de limiter le délai de bout en bout, doivent être sélectionnés les réseaux offrant un débit supérieur à celui requis pour le flux (sinon du retard sera accumulé par le flux). Le délai, la fiabilité et la gigue du réseau sont pris finalement en considération.

4.2.2 Méthode d'évaluation des caractéristiques du réseau

La méthode de sélection du réseau est basée sur la connaissance des caractéristiques des réseaux accessibles. Cette connaissance est stockée dans une base contenant des paramètres statiques (généralement les services offerts par le réseau) et dynamiques (délai, taux de perte, etc.). Deux techniques sont utilisées pour évaluer ces paramètres.

La première est passive, elle se base sur des caractéristiques garanties par chacun des réseaux. Ces valeurs sont généralement connues par le système de communication.

La seconde est dynamique. Elle se base sur des mesures effectuées régulièrement sur chacune des interfaces.

Statique :

Les services offerts par un réseau sont généralement connus du système et/ou de l'utilisateur qui a souscrit l'abonnement. Idéalement ils peuvent être obtenus par consultation des « drivers » du matériel utilisé. Toutefois, ceci demanderait une uniformisation de ces derniers, ce qui n'est actuellement pas le cas. Il est difficile d'obtenir ce type d'informations du système et nous avons préféré, au moins comme solution provisoire, compléter la base de connaissances de manière manuelle. Elle se trouve sous forme d'un fichier de configuration qui doit être mis à jour à chaque installation d'une nouvelle carte de communication.

Dynamique :

Les critères généralement utilisés tels que la disponibilité, le délai, la gigue et le taux de pertes de paquets sont difficiles à prévoir, du fait du caractère aléatoire des pertes de paquets et du temps indéterminé passé dans les files d'attente sur tous les routeurs qui constituent le chemin entre une source et une destination. Pour établir une vérification de la QoS ou pouvoir comparer les QoS fournies par différents réseaux, il faut pouvoir comparer ces observations.

Dans cette optique, le groupe de travail IPPM de l'IETF a défini préalablement une structure formelle de méthodologie des mesures [RFC2330].

Des « drafts » définissent les métriques de délai dans un seul sens [RFC2679], de pertes de paquets [RFC2680] et de connectivité [RFC2678], ainsi que la variation instantanée du délai [RFC2681].

Pour mesurer les performances, deux types de méthode existent :

- La méthode active consiste à injecter du trafic dans le réseau de manière contrôlée et à analyser les paquets retournés (*ping*, *traceroute*, *pathcar*, *treno*).

L'inconvénient de ces outils est de fausser les résultats, en introduisant du trafic supplémentaire pouvant influencer les métriques de la Qualité de service qu'on souhaite précisément mesurer.

- La méthode passive consiste à observer et analyser les paquets reçus sur un système terminal.

Dans le protocole multi-réseaux, les deux solutions ont été mises en œuvre :

- La méthode active pour mesurer les critères de RTT, gigue et fiabilité
Pour cela, l'application *ping* a été utilisée. Elle fournit des renseignements pour ces trois paramètres.

- La méthode passive pour mesurer la bande passante
Puisque la bande passante maximale d'un réseau ne peut être mesurée qu'en injectant des paquets dans le réseau, cette mesure est faite pendant des communications. La valeur maximale du débit instantané y est calculée et conservée pendant. C'est cette valeur qui détermine le critère bande passante.

D'autres outils de mesure [JIN01], plus précis, ont été testés, mais leur complexité et leur lourdeur n'a pas semblé encore une approche adéquate pour ce type d'application.

4.3 Adaptation du protocole de transport au réseau sous-jacent

Comme cela a été montré dans le paragraphe 3.2, les performances d'un même protocole de transport ne sont pas équivalentes en fonction des réseaux sous-jacents utilisés. Les gains de performances apportés par l'utilisation d'un protocole adapté suffisent bien souvent à justifier cette approche.

Toutefois, dans le contexte actuel d'une très forte hétérogénéité de réseaux interconnectés, il est difficile de créer des architectures de communication multi-protocoles, et les solutions préconisant la recherche d'un compromis dans la configuration d'un protocole de transport prédominent. Les principaux travaux de recherches actuels sont axés sur la définition d'une configuration généraliste de TCP, même si cela est de plus en plus souvent remis en question car généralement « compromis » ne rime pas avec « performances ».

Ceci étant dit, l'architecture multi-réseaux parallèle échappe à ce problème d'hétérogénéité, car le principe est de profiter des réseaux accessibles « de bout en bout », afin que ces derniers garantissent naturellement une certaine qualité de service.

En effet dans le cas d'une visioconférence utilisant le réseau téléphonique pour l'audio, et le réseau Internet pour la vidéo, pourquoi utiliser le même protocole, avec les mêmes paramètres, alors qu'il serait plus efficace d'utiliser des souches différentes ? Un TCP classique bien adapté au monde de l'Internet, avec ses mécanismes de contrôle de flux et de congestion pourrait être utilisé pour la vidéo, alors qu'une souche modifiée de TCP, adaptée à une liaison à faible débit mais à qualité de service constante, serait plus performante sur la liaison téléphonique.

La Table II-2 récapitule les configurations apportées à certaines souches de TCP, expérimentales ou bien commerciales, afin de fournir un comportement optimal sur certains types de réseaux.

Options	Réseaux		Satellite		Mobile
	Filaires	Faible débit/haut débit	LEO / GEO		
Echelle de Fenêtre	✗	✓	✗	✓	✗
SACKs	✗	✗	✗	✓	✗
Mesure du RTT	-	-	✓	-	✓
PAWS	-	-	✓	✓	✗
FEC	✗	✗	✗	✗	✓
Persistence	✗	✗	✓	✗	✓

✓ recommandé

✗ inutile, voire
problématique

– sans objet

Table II-2 - Options recommandées pour l'utilisation de TCP sur différents types de réseaux

Echelle de fenêtre (Scaling Window) : Permet d'accroître la taille de la fenêtre d'émission de TCP, le débit maximal théorique est alors augmenté.

Acquittements sélectifs (Selective ACKnowledgements) : Optimise la retransmission des pertes lorsque plusieurs segments sont perdus dans la même fenêtre.

Mesure du RTT : Mesure plus précise du délai pour les liaisons à long délai ou délai variable. Basé sur des estampilles temporelles.

Protection des numéros de séquences (Protection against wrapped sequence numbers) : Utilise des estampilles temporelles pour accroître de manière artificielle la taille de fenêtre.

Redondance de codage (Forward Error Correction) : Ajoute de manière logicielle de la redondance d'information pour protéger les segments d'erreurs de transmissions.

Persistance des connections (Time Out Persistence) : Améliore la résistance des connections TCP à des coupures du service réseau.

Cette table [Table II-2] confirme l'incompatibilité des différentes souches optimisées pour tel ou tel type de réseau. Par exemple l'option agrandissant la taille de fenêtre, généralement couplée à une taille initiale de fenêtre plus grande, améliore les performances du contrôle de flux de TCP pour les réseaux dits « Long Fat Network », c'est à dire dont le produit du délai par le débit est grand (cas des réseaux très haut débits, et des réseaux satellites du fait du délai). En revanche cette option provoque des saturations des liaisons à faibles débits, qu'elles soient terrestres ou bien hertziennes.

Du fait de cette constatation, et de la particularité de l'architecture multi-réseaux parallèle, une architecture **multi-protocolaire** s'impose. A chaque réseau disponible dans la base de donnée est associée une souche du protocole le mieux adapté pour son utilisation. Ainsi, en même temps qu'est sélectionné un réseau, en fonction des besoins de l'application, la couche protocolaire associée est activée. L'architecture multi-réseaux, multi-protocoles est présentée à la Figure II-11. Chaque réseau est alors utilisé de manière optimale.

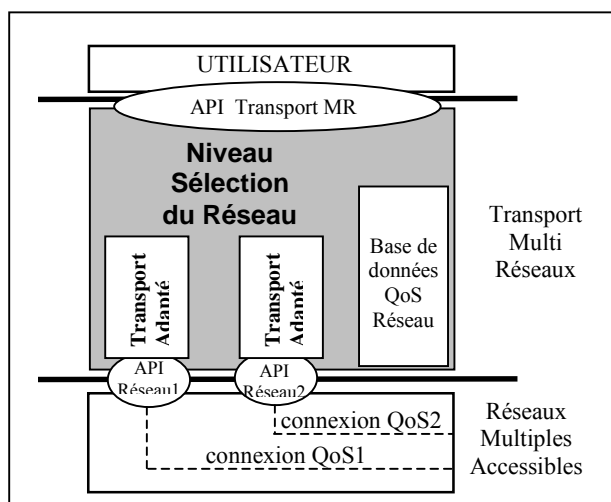


Figure II-11 - Architecture du protocole multi-réseaux

4.4 Mécanisme de synchronisation

4.4.1 Protocole multimédia à ordre et fiabilité partiels

Il est désormais bien connu que les protocoles de transport classiques d'Internet ne sont pas réellement adaptés aux flux de données multimédias des applications distribuées, parce qu'ils ne peuvent pas assurer divers types de QoS multimédia. En effet, TCP fournit seulement un service totalement ordonné et fiable, et UDP, un service sans ordre ni fiabilité, alors que les flux de données multimédias nécessitent un service partiellement fiable (il n'est pas gênant qu'une image d'un flux vidéo de 25 images/s vidéo soit perdue ou altérée) et partiellement ordonné (puisque les applications multimédias comportent plusieurs compositions série et/ou parallèle de différents flux de données). Apportant une solution à ce problème, un transport à ordre partiel [AME94][CHA95][DIA95] est un transport qui délivre les unités de données envoyées sur une ou plusieurs connexions, suivant un ordre (partiel) donné. Cet ordre est situé entre l'ordre total (TCP) et aucun ordre (UDP) et peut s'exprimer comme une composition série/parallèle d'objets ou d'unités de données. Notons que cet ordre peut, par exemple, être décrit par un Réseau de Pétri à Flux Temporels (Time Stream Petri Net – TSPN, une extension temporelle du modèle de Réseau de Pétri général [DIA95]), comme illustré Figure II-13 dans le cas de la composition série/parallèle des flux son et vidéo d'une application de visioconférence. Dans ce cas, l'ordre de délivrance peut être vu comme la synchronisation logique d'objets multimédias, la synchronisation étant l'un des points-clés les plus importants des systèmes multimédias distribués [OWE98].

De plus, la possibilité de pertes dans le réseau amène à l'intéressante notion de fiabilité partielle. Cette notion est étroitement liée à la QoS du transport : elle définit une QoS nominale et une QoS minimale en dessous de laquelle le service demandé par l'utilisateur n'est plus assuré. Cette QoS minimale peut être exprimée en définissant un ensemble de pertes acceptables, par exemple un nombre maximum de pertes à l'intérieur d'une séquence, un nombre maximum de pertes consécutives, etc. Quand une perte acceptable est détectée (c'est-à-dire quand un objet reçu a une numérotation plus élevée que prévu), l'objet manquant peut instantanément être déclaré perdu (indication de perte au plus tôt), et les données suivantes déjà reçues peuvent être immédiatement délivrées à l'application (délivrance au plus tôt). Si la perte ne peut pas être acceptée par rapport à la fiabilité demandée, une retransmission devra être effectuée.

La qualité de service que garantit le protocole est représentée par deux composantes : la fiabilité et l'ordre de la livraison des données. Avec ces deux paramètres, l'application peut « construire » n'importe quel service selon ses besoins. Pour schématiser les capacités d'un service POC³, [AME93] et [DIA94] comparent le service offert par POC à ceux de TCP et UDP, à l'aide d'un plan cartésien (Figure II-12).

La fiabilité partielle exigée, définie dans le service de transport à ordre partiel et ayant pour résultat des indications de pertes et de délivrances au plus tôt, se déduit des besoins de l'application [OWE98].

³ Partial Order Connection

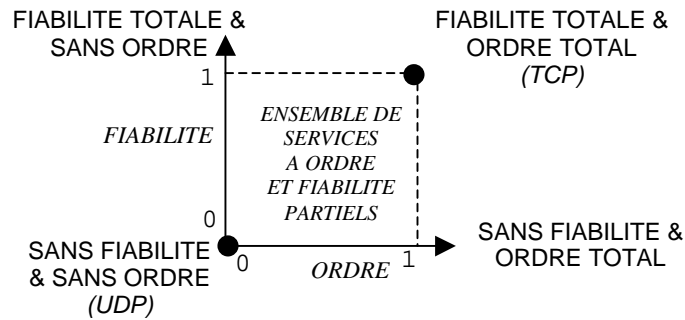


Figure II-12 - Représentation d'un service POC dans le plan

En fait, deux approches existent pour contrôler la fiabilité partielle : média par média ou par groupe de médias [FOU96]. Média par média signifie que l'entité de réception du flux peut seulement gérer la fiabilité partielle sur le flux qu'elle contrôle, et pas sur les autres flux de la connexion multimédia. Dans une gestion par groupe de médias, l'entité de réception peut déclarer des pertes sur d'autres flux de la même connexion multimédia, ce qui mène à un comportement plus interactif, comme nous allons le voir maintenant.

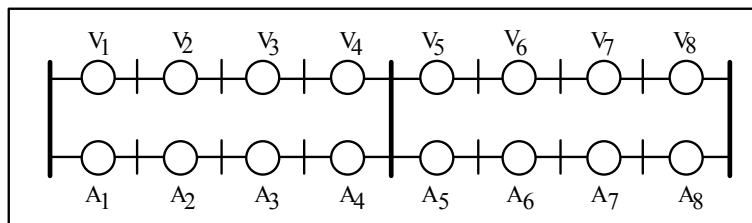


Figure II-13 - Exemple d'ordre partiel

Considérons la Figure II-13 qui représente un réseau de Petri de composition série/parallèle d'une connexion multimédia (son et vidéo d'une application de vidéoconférence par exemple). Supposons que le nombre maximum de pertes autorisées sur chaque flux pendant une période de synchronisation (V_1 - V_4 , A_1 - A_4) est de un . Supposons également que les données V_1 , V_2 , A_1 , A_2 , et A_3 ont été reçues par l'entité de réception et délivrées à l'application.

Puis supposons que l'entité distante reçoive V_4 ; le respect fort de l'ordre partiel sur la connexion vidéo indique que V_4 ne peut pas être délivrée à l'application : c'est parce qu'il est logiquement localisé après V_3 et que V_3 n'a pas encore été reçu. Maintenant, V_4 pourrait être délivrée si V_3 avait été perdue, en raison de la fiabilité choisie (une perte de donnée autorisée par période) : c'est le principe de la fiabilité partielle. Une perte est acceptable sur chaque période pour chaque flux, et afin de délivrer V_4 dès que possible, V_3 sera déclarée comme perdue, V_4 étant alors délivrée. C'est le principe des pertes et des délivrances au plus tôt.

Maintenant supposons que V_5 soit reçue. Concernant l'ordre partiel, cet objet ne peut pas être délivré, parce qu'il est logiquement après A_4 (en raison de la synchronisation inter-flux après A_4 et V_4).

Avec une gestion média par média de l'ordre partiel, V_5 doit être stockée jusqu'à ce que A_4 soit reçue ou déclarée perdue.

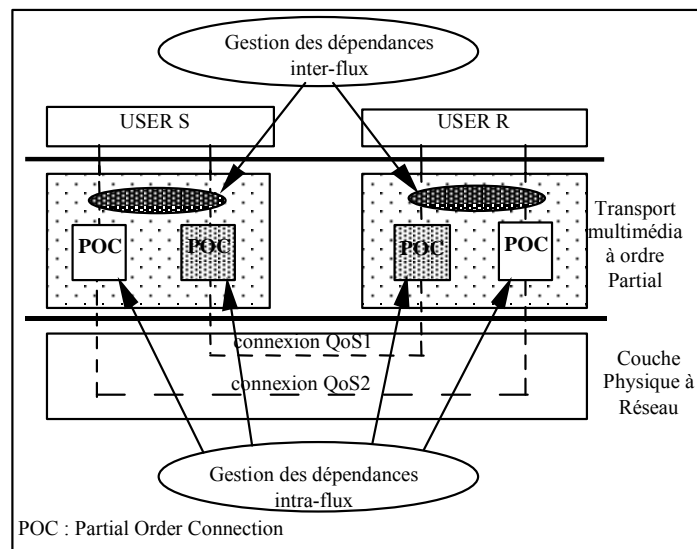


Figure II-14 - Architecture d'un transport à ordre partiel

Ainsi, avec la solution par groupe de média, le gestionnaire de la connexion vidéo devient capable d'assumer des pertes sur la connexion son. Dans ce cas, comme une perte est acceptable sur le flux son (sur une période), on déclare A4 perdue et on délivre V5 à l'application dès qu'elle est reçue. Cette gestion de la fiabilité partielle par groupe de média mène à la conception de l'architecture fortement interactive donnée sur la Figure II-14, qui montre également la gestion de multi-connexions [OWE98].

A chaque média est associée une connexion à ordre et fiabilité partielle, nommée POC (Partial Order Connection), garantissant la qualité de service du flux en termes d'ordre et de fiabilité. Jamais un flux ne sera délivré à l'application avec un taux de pertes supérieur à celui spécifié par l'application.

Assurant la synchronisation logique, un module appelé POM (Partial Order Manager) délivre à l'application les données provenant des différents POCs selon l'ordre partiel préalablement spécifié. L'application n'a plus alors qu'à :

- Présenter les données à l'utilisateur si la synchronisation requise est relativement faible.
- Ou bien, si une synchronisation plus fine est nécessaire, à les délivrer à une couche de présentation qui, à l'aide de mécanismes tels que des estampilles temporelles, optimisera la synchronisation intra-flux.

Ce mécanisme d'ordonnement combiné avec la gestion de la fiabilité de chacun des flux, diminue les risques de blocages liés à des attentes inutiles de données perdues, alors que leur non présentation pourrait être acceptable à la vue de la qualité de service demandée.

Ce protocole multimédia est nommé MMPOC (MultiMédia Partial Order Connection).

4.4.2 Intégration des notions de multimédias et multi-réseaux

L'architecture du protocole multimédia à ordre et fiabilité partiels MMPOC détaillé ci-dessus présente des propriétés intéressantes, créant le lien entre les besoins en qualité de service des

applications multimédias, et les différentes qualités de services offertes par un environnement multi-réseaux.

Tout d'abord, comme cela a été souligné dans le paragraphe 3.4, la notion de fiabilité partielle est une notion indispensable lorsque sont utilisés des réseaux à forts taux de pertes comme les réseaux mobiles ou satellites. Il est dans ce cas inutile de lutter contre les pertes, surtout lorsque, comme c'est le cas pour une application multimédia, c'est le délai d'interaction qui est prioritaire.

D'autre part, la notion de synchronisation logique induite par la gestion d'un ordre partiel inter-flux apporte une solution aux désynchronisations induites par l'utilisation simultanée de réseaux distincts, désynchronisations intolérables pour les applications multimédias.

Finalement, la définition des modules de « connexion à ordre partiel » supporte les principes de configuration multiple de protocoles évoqués dans le paragraphe 4.2.1. Chacun des POCs peut ainsi être configuré pour être adapté au réseau qu'il utilise comme support de transmission.

La Figure II-15 présente l'architecture finale du protocole multimédia multi-réseaux, nommé MMPOC-MN (MultiMédia Partial Order Connexion – Multi-Network)

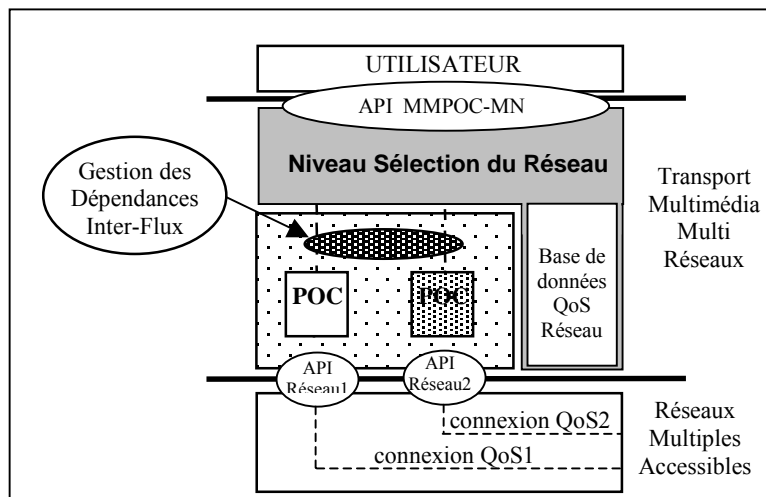


Figure II-15 - Architecture du protocole multimédia multi-réseaux

4.4.3 Gestion des tampons d'émission et de réception

La synchronisation de plusieurs flux empruntant des réseaux différents peut toutefois poser de nouveaux problèmes qui n'ont pas lieu d'être dans un protocole classique.

Un protocole classique, type TCP, possède des mécanismes protégeant la souche réceptrice du protocole des dépassements de capacité mémoire, qui peuvent avoir lieu lorsque que le débit de production des données de l'application émettrice est supérieur à celui de consommation de l'application réceptrice. Cette situation conduit au remplissage progressif des buffers du protocole récepteur qui n'arrive pas à « écouler » les données reçues, et à terme conduit à un débordement si rien n'est fait.

Dans TCP, lorsque les buffers de réception sont pleins, les nouvelles données arrivantes sont alors refusées, ce qui a pour effet de provoquer une détection de congestion et entraîne une

réduction du débit d'émission (en réduisant la taille de fenêtre). Le débit d'émission ainsi réduit, la taille du buffer de réception peut alors diminuer. De son côté, l'émission repasse en phase de « slow start » qui aura pour but d'augmenter à nouveau progressivement le débit.

Le protocole MMPOC-MN est bien évidemment confronté à ce facteur de congestion des tampons de réception, mais un autre facteur aggrave ce risque. En effet, comme cela est expliqué ci dessous, la synchronisation inter-flux est source de remplissage des tampons lorsque la désynchronisation des flux augmente.

Supposons que le protocole MMPOC-MN ait en charge la synchronisation de deux flux, et que suite à des perturbations réseau (congestion du réseau, brouillage, ...) le débit d'un des deux flux chute brusquement. La proportion des données reçues change alors en faveur du flux n'étant pas ralenti. Toutefois, du fait de la synchronisation requise, le protocole devra continuer à fournir à l'application les données dans la même proportion qu'auparavant. Pour ce faire, toutes les données reçues du flux « prenant de l'avance » sont stockées jusqu'à ce qu'elles puissent être délivrées conformément à la synchronisation requise.

A terme, si la situation n'évolue pas, comme l'indique la Figure II-16 (qui présente l'évolution du nombre de données contenue dans les différents buffers de réception du protocole MMPOC-MN) un débordement des tampons du protocole est à craindre.

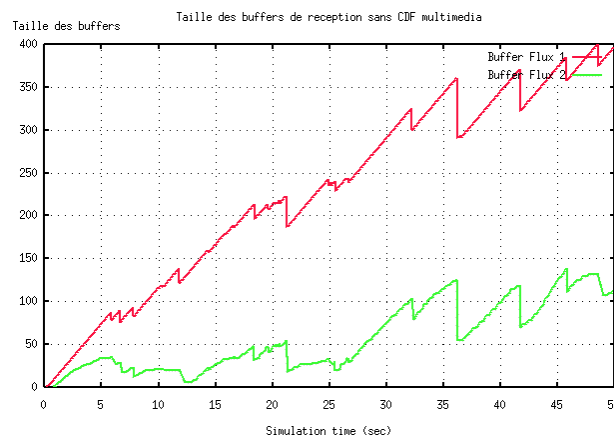


Figure II-16 - Evolution des Buffers de Réception

A l'instar de TCP, une solution provoquant la réduction du débit des connexions les plus en avance a dû être mise en place.

Parmi les différentes options possibles pour l'implantation d'un tel mécanisme de contrôle de flux, un mécanisme *auto adaptatif* semblable à celui de TCP a semblé le plus approprié. Dès qu'un tampon de réception dépasse un seuil proche de la saturation, les acquittements pour cette connexion ne sont plus renvoyés à l'émetteur, ce qui aura pour effet de lui faire réduire la taille de sa fenêtre d'émission. La taille de cette fenêtre étant directement liée au débit de la connexion, le débit sera réduit. Dès que le buffer descendra en dessous du seuil, les acquittements seront à nouveau émis. Enfin, pour que ce contrôle de flux puisse réagir à des baisses passagères de performance du réseau, un mécanisme d'augmentation de la fenêtre d'émission est lui aussi mis en place. Ce dernier permet une augmentation progressive du débit de la connexion.

4.5 Le déploiement dynamique

Implanter un protocole de transport n'est vraiment intéressant que s'il peut être largement et facilement utilisé. A la lumière des travaux réalisés précédemment sur l'implémentation du protocole MMPOC sur une plate-forme Sun Solaris 2 [FOU96], à l'aide du mécanisme des streams, il est apparu qu'en dépit des performances apportées par l'utilisation d'un tel protocole, son déploiement sur des environnements hétérogènes s'avère très difficile. Ce constat nous a amené à en abandonner la maintenance qui était trop coûteuse et délicate.

Pourtant les projets menés en parallèle sur le travail coopératif comme CANET [OWE97] ou l'enseignement à distance comme TOPASE [VIL98], auraient dû tirer profit de ce protocole. Mais la complexité de déploiement de MMPOC sur chaque plate-forme (Sun, HP, SGI, PC, etc.) nous a contraint à utiliser les services classiques comme TCP/IP ou AAL5 d'ATM. Il faut donc noter que même si un protocole apporte un gain de performances intéressant, son succès est étroitement lié à son aptitude à être facilement portable et largement déployable.

4.5.1 Solutions existantes

Cette constatation a amené à approfondir de nouvelles solutions ainsi que de nouveaux langages, facilitant la mobilité des programmes dans des environnements hétérogènes. Ces fonctionnalités sont offertes, entre autres, par le nouveau domaine de recherche qui est celui des « réseaux actifs ». Ceci permet d'explorer un nouveau type de protocoles spécialisés et facilement déployables entre des entités communicantes [TEN97].

Les réseaux actifs sont une réponse au manque de flexibilité des réseaux classiques maintenus par les opérateurs traditionnels. Les réponses complémentaires, avec des niveaux de granularité différents, sont :

- l'approche des routeurs programmables (1) [ALE98A] qui offre un mécanisme de téléchargement manuel de programmes (dans les routeurs), mais qui maintient le format classique des paquets.
- Par opposition, l'approche *capsule* (2) [WET98, SCH99] va plus loin en ajoutant à l'entête des paquets traditionnels, des programmes miniatures qui peuvent être exécutés dans tous les nœuds du réseau traversé.
- Finalement l'approche code mobile (3) [HAR96, JAV98, JAV99] offre le mécanisme de plus haut niveau qui permet la mobilité et l'exécution de programmes au travers d'un environnement hétérogène.

Cette dernière solution a été choisie pour les raisons qui sont évoquées dans la section suivante.

4.5.2 HotJava

Les environnements offrant les mécanismes de mobilité de code semblent être bien adaptés aux problèmes de distribution et de déploiement de protocoles de transport. De nombreux travaux sont en cours dans ce domaine, mais peu de plates-formes sont accessibles [JAV98, JAV99]. Les plates-formes basées sur le langage de programmation Java sont les plus avancées en termes de performances et de portabilité. La solution proposée par l'environnement *HotJava* semble la plus appropriée, au moins pour un certain temps, pour distribuer notre protocole MMPOC-MN.

HotJava est le navigateur web développé par *Sun* dans le but de démontrer les possibilités du langage *Java*. Sa capacité à intégrer dynamiquement de nouveaux comportements en fait sa principale différence avec les autres navigateurs web.

Grâce à sa capacité de téléchargement automatique de code Java, des programmes de visualisation ainsi que de nouveaux protocoles peuvent être facilement intégrés et offerts de manière transparente aux utilisateurs. D'ailleurs les protocoles et les applications déjà présents dans le navigateur sont implémentés suivant le même principe.

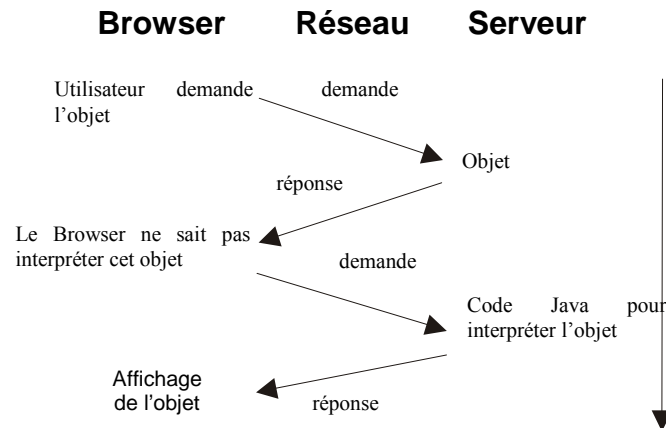


Figure II-17 - Transfert du code de l'application

Un document *HotJava* est référencé par un URL et son protocole associé. Ceci permet son accès au travers d'une station distante. Si le protocole n'est pas implémenté dans le navigateur qui effectue la requête, le mécanisme de distribution de code offert par HotJava, télécharge le « byte code Java » du protocole sur le serveur où se trouve le document [Figure II-17]. L'exécution du protocole permet ainsi la récupération du document, qui pourra être ensuite présenté par une application dédiée pour ce type de documents. De nouveau, si l'application de présentation n'est pas présente dans le navigateur, comme pour le protocole, son téléchargement aura lieu. Afin d'éviter des retransmissions inutiles de code, un mécanisme de cache est intégré. Cet environnement permet ainsi de déployer le protocole MMPOC-MN ainsi qu'une application de démonstration de « video on demand » (VoD).

Une application de VoD a été choisie pour illustrer les bénéfices du protocole MMPOC-MN et son adéquation aux contraintes multimédias. Par ailleurs, ce type d'application respecte le paradigme client/serveur imposé par *HotJava*. Cette architecture est découpée en deux parties :

- Une partie mobile joint le protocole MMPOC-MN et l'application de VoD
- L'autre partie statique comporte serveur de fichiers et le protocole MMPOC-MN.

D'un point de vue programmation, écrire du code mobile dans l'environnement *HotJava*, ne présente pas de difficultés spécifiques, puisque cela consiste à respecter des interfaces *Java* fournies avec l'environnement.

Néanmoins, les mécanismes de présentation automatique des objets, bien adaptés au mode transactionnel, imposent de recevoir entièrement un document avant de le présenter. Cette spécificité étant opposée à la notion de flux multimédias, un artifice de programmation a été nécessaire. Ainsi, dans notre solution, un document référencé par un URL ne contient pas de données multimédias, mais seulement les paramètres de qualité de service requis par l'application, et les URL des différents médias. Ensuite, le protocole MMPOC-MN doit être utilisé par l'application pour le rapatriement des différents fichiers. La Figure II-18 représente les différents échanges d'informations.

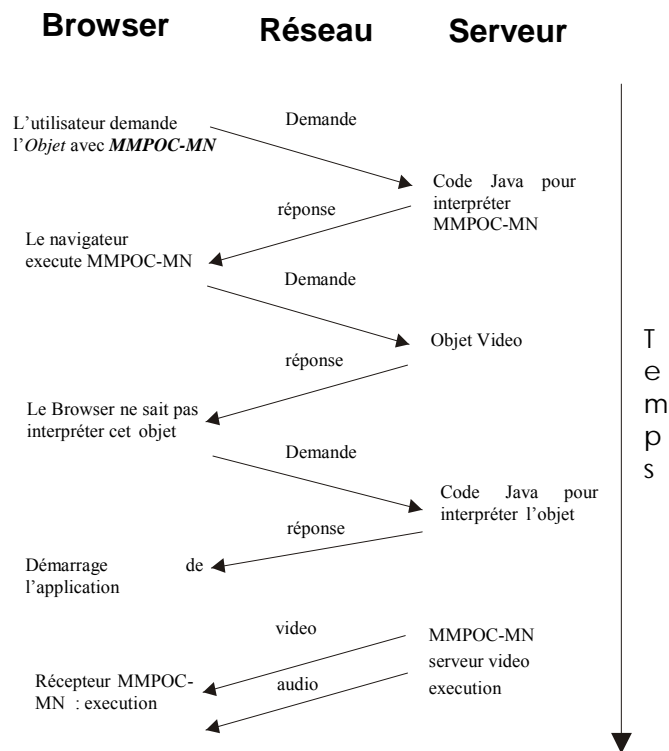


Figure II-18 - MMPOC-MN Architecture de déploiement

5 Mesures de performances

Afin d'évaluer les performances d'une telle architecture de communication, des simulations multi-réseaux ont été réalisées à l'aide du simulateur de réseaux OPNET. En effet les expérimentations multi-réseaux ont été limitées par le fait que nous n'avons pas accès à des réseaux satellites ou mobiles, mais seulement au réseau Internet classique, ainsi qu'au réseau ATM national RENATER 2. La diversité des configurations multi-réseaux est telle qu'il n'était pas possible de présenter les performances du protocole multi-réseaux dans tout les cas de figures. Il a alors été choisi de présenter une configuration particulièrement défavorable, combinant l'utilisation d'un réseau terrestre à celle d'un réseau satellite, afin de montrer les gains et les limites d'un tel protocole.

Toutefois, dans le but de confirmer les résultats obtenus par simulation, ainsi que montrer l'impact du protocole multimédia sur les performances de l'application, les mesures qui sont présentées ci-dessous ont été réalisées sur l'implémentation réelle déployée sur un réseau ATM.

Une application de visioconférence est utilisée comme référence car le déploiement de tels applications reste un problème d'actualité. Les contraintes sont nombreuses :

- Le délai existant entre le moment de capture d'une image ou d'un échantillon sonore et sa présentation ne doit pas être trop grand sous peine de pénaliser l'interactivité du système.
- La synchronisation des deux flux doit être maintenue dans des limites raisonnables afin de préserver la « lip synchronisation ».
- La qualité du flux audio original doit être conservée afin de préserver son intelligibilité. La fiabilité et la gigue du flux doivent être maîtrisés en priorité.

- Le débit généré par le flux vidéo est un problème en soi, car peu de réseaux garantissent à faible coût des hauts débits de transmission.

En fin de compte, à la vue de ces nombreuses contraintes, une application de visioconférence peut-être caractérisée de « killing application » pour les réseaux actuellement déployés.

Deux flux sont générés, un de 64 Kbit/s pour la transmission de l'audio, et un de 128 kbit/s pour la vidéo. La taille des paquets est respectivement de 512 octets pour l'audio, et de 1024 octets pour la vidéo.

5.1 Mesures des performances observées par l'application

Tout d'abord, l'utilisation du transport à ordre et fiabilité partiels conduit à réduire le taux d'occupation moyen des tampons de réception ainsi que le délai de bout en bout, ceci grâce aux mécanismes de délivrance et de déclaration de pertes au plus tôt. D'autre part, la fiabilité partielle permet de réduire le débit généré en limitant les retransmissions.

Il est important de noter que l'application de VoD respecte les principes de synchronisation audio / vidéo cités dans [OWE98]. Les pertes de temps qui peuvent avoir lieu durant l'attente de données perdues, peuvent être évitées (par les mécanismes de délivrance et de déclaration de pertes au plus tôt). Ces pertes de temps sont très pénalisantes car pour respecter les contraintes de synchronisation, les objets (images ou échantillons sonores) qui arrivent trop tard sont automatiquement écartés par l'application.

Dans cette section, le taux de pertes au niveau application est mesuré dans deux cas : (1) avec un transport à ordre partiel et (2) avec un transport sans connexion classique (UDP). Un simulateur de réseau permettant de simuler des pertes a été utilisé. Ce simulateur peut être placé sur la machine émettrice afin de retarder et détruire quelques paquets.

Les résultats des expérimentations sont décrits sur la Figure II-19. Le taux moyen de pertes au niveau de l'application est mesuré dans deux cas (POC et UDP) en fonction du taux de pertes sur le réseau. Afin de mettre en avant l'impact de POC sur les pertes dues aux mécanismes de synchronisation, les retransmissions ne sont pas demandées. Le support est un réseau ATM à 155 Mbit/s.

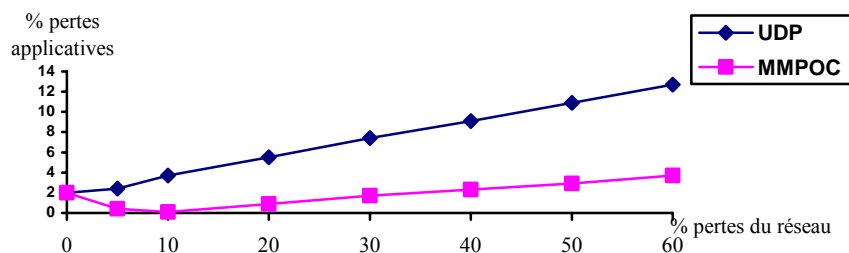


Figure II-19 - QoS VoD sur POC vs UDP

Cette figure montre que la QoS obtenue en utilisant POC est meilleure que celle engendrée par l'utilisation d'UDP. En fait lorsqu'une indication de perte est reçue du transport à ordre partiel, l'application peut anticiper et continuer le traitement des données présentes dans le tampon. Dans le cas d'UDP, l'application reste en attente puisqu'elle ne reçoit aucune indication de perte. Par conséquent, la courbe présentant les résultats de l'application de VoD utilisant UDP augmente linéairement. Les pertes réseau étant plus nombreuses, les pertes de temps sont plus fréquentes, et donc le nombre de données devant être perdues pour respecter les contraintes de synchronisation inter-flux augmente. Dans le cas de MMPOC, le taux de

pertes est toujours inférieur à celui d'UDP, et plus il y a d'erreurs, plus la différence de performances entre les deux approches est importante.

5.2 Mesures des performances du protocole MMPOC-MN

Comme évoqué dans l'introduction, l'étude porte sur une configuration multi-réseaux particulièrement contraignante du fait de la distance existant entre les qualités de services offertes par les deux réseaux. Un réseau terrestre est utilisé pour la transmission des données audio pour la qualité de service garantie qu'il offre : Une bande passante de 64 Kb/s garantie, une gigue faible, un délai de bout en bout de 50 ms, et un taux d'erreur très faible. Pour le flux vidéo, c'est une liaison fournie par un satellite géostationnaire qui est utilisée. La bande passante minimum garantie est de 128 Kbit/s, le délai de l'ordre de 260 ms avec une gigue faible, et un taux d'erreur/bit élevé compris en 10^{-6} et 10^{-7} , accompagné de rares pertes groupées de l'ordre d'une dizaine de paquets. La Figure II-20 résume la configuration simulée.

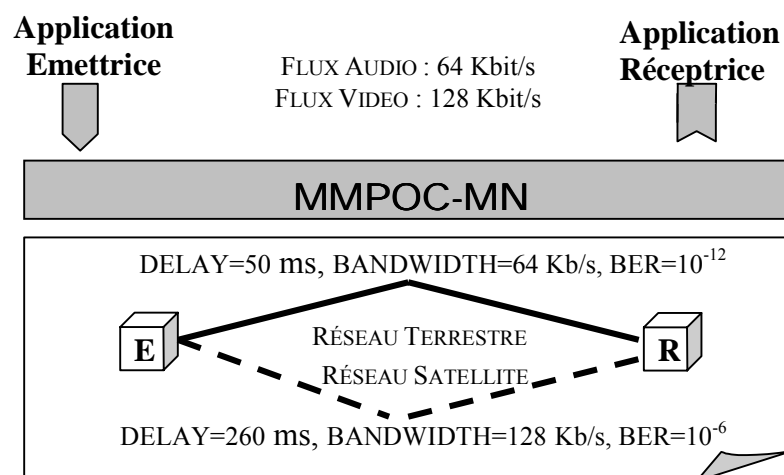


Figure II-20 - Conditions d'expérimentation

En premier lieu est présenté le gain de performances obtenu en utilisant un protocole adapté au réseau sous-jacent utilisé. Dans les simulations suivantes, les protocoles seront par défaut adaptés aux réseaux sous-jacents. La synchronisation inter-flux sera ensuite évaluée, puis l'impact de la gestion d'une fiabilité partielle sur les délais de bout en bout sera mesurée. Finalement, le rôle du contrôle de flux multimédia est montré en suivant l'évolution des buffers de réception du protocole.

5.2.1 Utilisation d'un protocole de transport adapté

Le premier groupe de simulation rappelle la problématique de l'adaptation du protocole de transport MMPOC-MN au réseau sous-jacent. Ces mesures ont été effectuées sur le protocole TCP, puisqu'elles ont eu pour rôle d'indiquer quelles étaient les options à implémenter dans le protocole MMPOC-MN afin d'être rendu adaptable à tout type de réseau. Les résultats obtenus avec le protocole MMPOC-MN sont tout à fait similaires à ceux obtenus à l'aide des souches modifiées de TCP, puisque les mêmes mécanismes ont été mis en place.

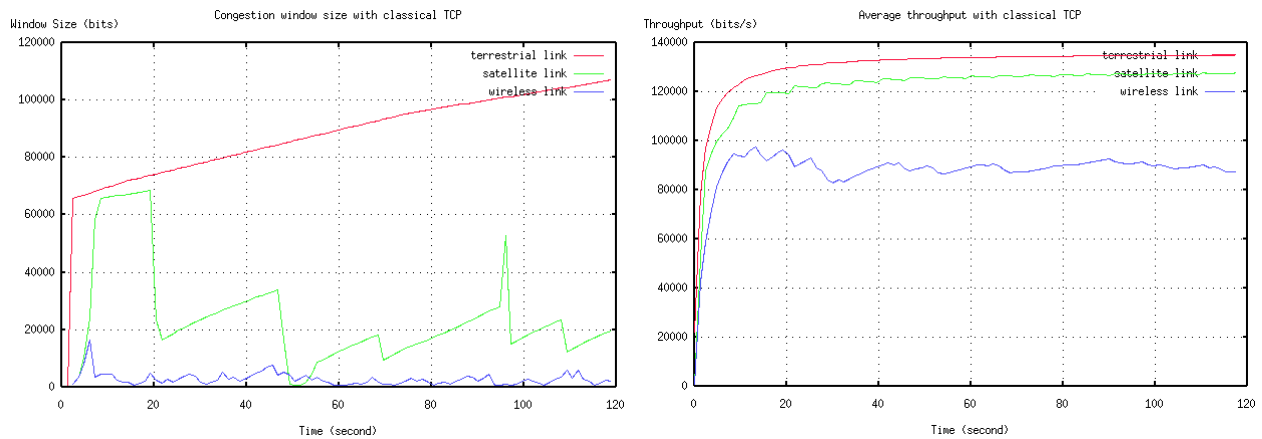


Figure II-21 - Performances de TCP classique sur divers réseaux

Les Figure II-21 rappellent le comportement de TCP détaillé dans le paragraphe 3.2. On observe une claire limitation de la bande passante (courbes de droite) sur les réseaux hertziens quand une souche non modifiée de TCP est utilisée.

Sur les figures suivantes [Figure II-22], des souches utilisant les options recommandées pour un comportement optimal de TCP ont été testées pour les réseaux satellite et mobile. Nous appellerons ces souches « TCP Sat », et « TCP Mobile ». Ces options sont décrites dans le paragraphe 4.2.1. On observe un comportement beaucoup moins chaotique de la fenêtre de congestion dans tous les cas. L'utilisation d'une fenêtre de plus grande taille permet de générer un débit supérieur à celui requis. On aperçoit que hormis la phase de *slow-start*, où le comportement des TCP Sat reste moins efficace que sur la liaison terrestre, le débit généré est ensuite équivalent. En revanche, les pertes très nombreuses sur la liaison mobile réduisent très souvent la taille de la fenêtre, ce qui fait chuter le débit. L'utilisation de codes correcteurs compenserait ici ces déficits, mais cette option n'est implantée ni dans le simulateur OPNET, ni dans la souche du protocole MMPOC-MN pour des raisons évidentes de performances.

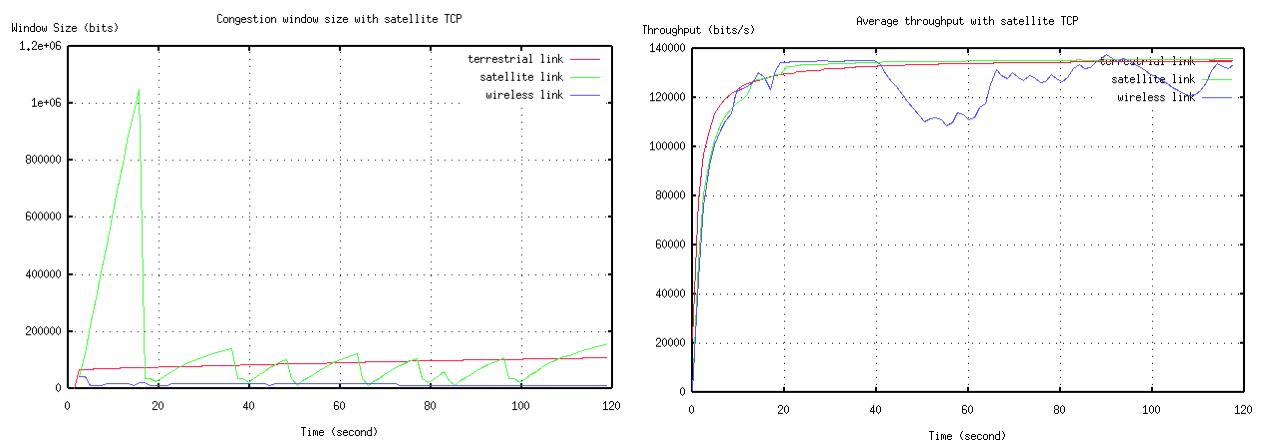


Figure II-22 - Performances de TCP optimisé sur divers réseaux

5.2.2 Rôle de la synchronisation

Comme cela a été précédemment signalé, la probabilité que deux flux se trouvent décalés est plus grande dans un contexte multi-réseaux. Nous appellerons « décalage » le fait que des

données appartenant à des flux différents, produites à la même date, ne soient pas délivrées en même temps. Cette différence des dates de livraison est mesurée. Elle quantifie la qualité de la synchronisation fournie par le protocole. Le scénario étudié est celui décrit précédemment. Le décalage est mesuré pour des flux fiables (type TCP), le paramètre fiabilité étant fixé.

La Figure II-23 présente la valeur mesurée au cours des simulations du décalage entre le flux audio et vidéo dans deux cas : sans synchronisation et avec synchronisation.

Lorsqu'aucune synchronisation n'est requise, les deux flux sont délivrés indépendamment à l'application réceptrice, de la même façon que si deux connexions TCP avaient été utilisées. La courbe sombre de la figure de gauche indique un déphasage oscillant entre 0,5 et 2,5 secondes dont la valeur moyenne, donnée à la Table II-3, est de 1,5 secondes. Pour l'application, ce déphasage se traduirait par un retard de 1,5 secondes de l'image sur le son.

Les raisons d'un tel décalage ont déjà été évoquées : on peut citer dans ce cas, deux causes principales :

- Le temps d'initialisation de la connexion et la phase de « slow-start » moins rapide sur la liaison satellite que sur la liaison terrestre. Ceci explique le décalage constant d'une seconde, qui additionné à la différence de délai physique des deux réseaux produit ce décalage moyen.
- Les nombreuses reprises de pertes ayant lieu sur la connexion satellite pour garantir l'intégrité des données. Certaines données subissent un retard supplémentaire induit par le recouvrement de leur perte. Ceci explique les variations de délai.

En revanche, dans le cas où une synchronisation est spécifiée au protocole MMPOC-MN, la valeur de la synchronisation oscille vers une valeur proche de zéro (courbe claire, figure de gauche).

Cette synchronisation est spécifiée au protocole par un ordre partiel. La taille de l'ordre partiel, appelé aussi période de synchronisation constitue la granularité de la synchronisation. Plus la période de synchronisation est faible, moins le décalage entre les flux peut être grand. Les « pics » indiquant un décalage de quelques centièmes de secondes pour des données isolées correspondent à des pertes dont la retransmission a été attendue avant la délivrance des données suivantes. Une période de synchronisation de 8 paquets a été spécifiée pour la mesure, c'est à dire qu'un point de synchronisation est introduit tous les 8 paquets.

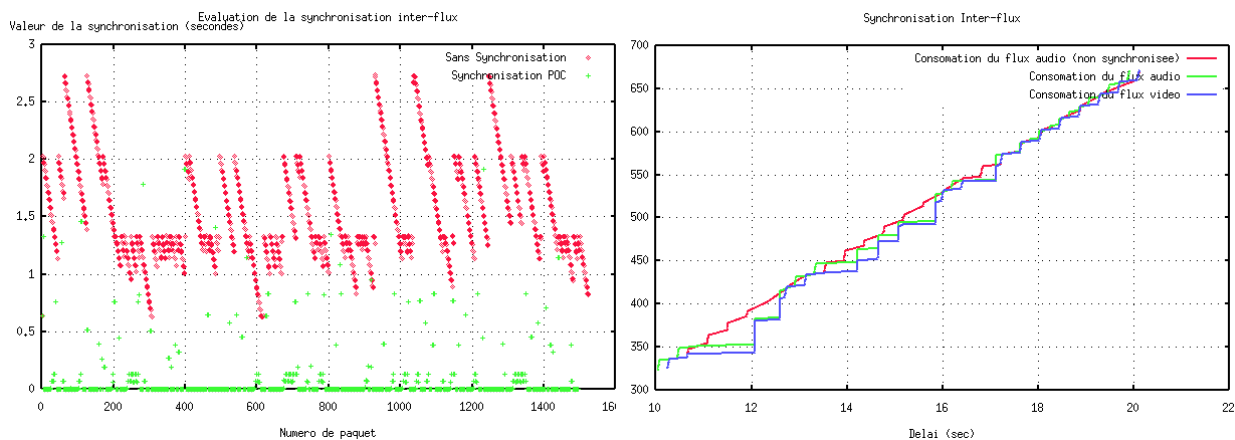


Figure II-23 - Valeur du décalage entre les flux

La Table II-3 indique les valeurs moyennes du décalage en fonction du type de synchronisation utilisé. On remarque que plus la synchronisation spécifiée est lâche, donc avec une période de synchronisation plus grande, plus le déphasage moyen augmente. Ceci s'explique, dans notre cas, par le fait que le flux empruntant la liaison satellite est constamment en retard par rapport à l'autre flux (du fait de la différence de délai physique). Les données du flux audio étant stockées dans un tampon, la livraison des données synchronisées à l'application est déclenchée par l'arrivée de données du flux vidéo. L'ensemble des données d'une période du flux audio est délivré dès la réception de la dernière donnée du flux vidéo de la période précédente. La valeur du décalage est calculée à partir de cet instant, jusqu'à la réception totale des données du flux vidéo (pour la période en cours).

Décalage (secondes)	Pas de Synchro	Synchro 2 paquets	Synchro 8 paquets
Moyenne	1,5434	0,096316	0,177066
Maximum	2,734166	1,912063	2,607345

Table II-3 - Valeur du déphasage en fonction de la synchronisation

5.2.3 Influence de la fiabilité sur des flux synchronisés

Afin de réduire les délais de bout en bout, critère de QoS important pour les applications temps réel, le protocole MMPOC-MN propose une méthode innovante de gestion de la fiabilité. Lors de la définition de QoS de chaque flux, il est possible d'indiquer un taux moyen de pertes acceptables sur ce flux afin d'éviter des retransmissions qui ne seraient pas strictement nécessaires.

La liaison satellite provoquant de nombreuses pertes et la vidéo n'étant pas un média qui requiert une fiabilité totale pour être présenté correctement, l'impact de la gestion pour ce flux d'une fiabilité partielle sur le délai de l'application est présenté dans la Figure II-24. L'audio tolérant mal les pertes, une fiabilité totale a été conservée pour ce flux.

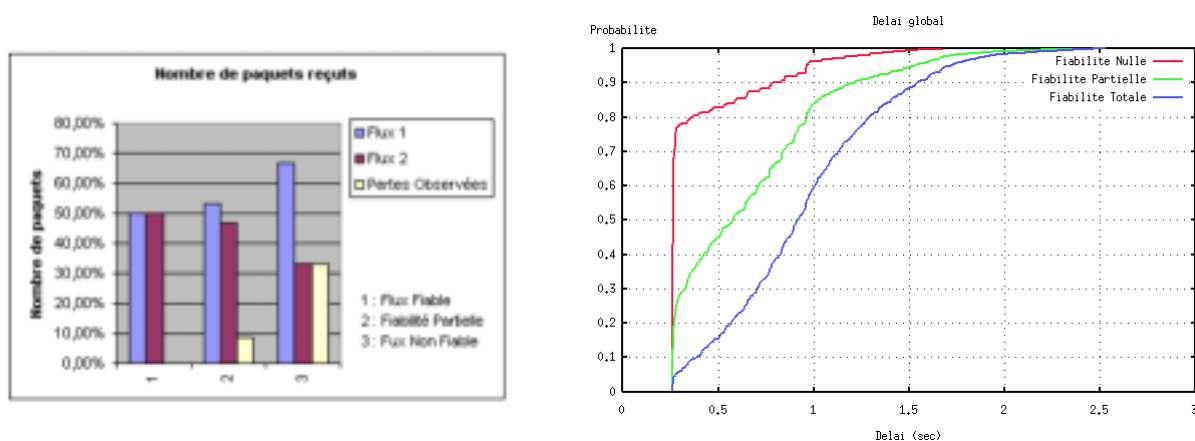


Figure II-24 - Influence de la fiabilité partielle sur le délai de bout en bout

Le délai de l'application a été mesuré dans trois cas :

- Quand une fiabilité totale est requise pour les deux flux
- Quand la fiabilité du flux vidéo est fixée à 90%
- Quand aucune fiabilité n'est requise sur le flux vidéo

Sur la courbe de droite, on peut apercevoir que dans le premier cas, 50 % des données sont délivrées avec un délai inférieur à une seconde après avoir été produites. Dans le second cas (courbe du milieu), ce délai devient quasiment inférieur à une demi seconde pour la moitié des données. Non seulement les données du flux vidéo peuvent être délivrées plus rapidement du fait qu'il ne soit plus nécessaire d'attendre toutes les données perdues (gestion intra-flux de la fiabilité), mais aussi accélère la livraison des données du flux audio, qui peuvent anticiper un point de synchronisation en déclarant perdues les données manquantes du flux vidéo (gestion inter-flux de la fiabilité). Sur la figure de gauche on remarque que le seuil des 10% des pertes acceptables est bien respecté. Dans le cas où aucune fiabilité n'est requise, plus de 30% des données du flux vidéo sont perdues, et 80 % des données sont reçues dans un délai équivalent au délai physique du réseau.

La Figure II-25 présente l'effet de la gestion de la fiabilité partielle spécifiée pour le flux vidéo, sur la délivrance des données audio à l'application. Puisque des pertes sont acceptées sur le flux vidéo, le flux audio peut déclarer perdues des images en retard, dans la limite de la fiabilité requise, et ce afin de permettre une délivrance des données au plus tôt. Le rythme de livraison des données audio est présenté dans trois cas : pour un flux vidéo non fiable, pour un flux vidéo partiellement fiable, et pour un flux vidéo totalement fiable.

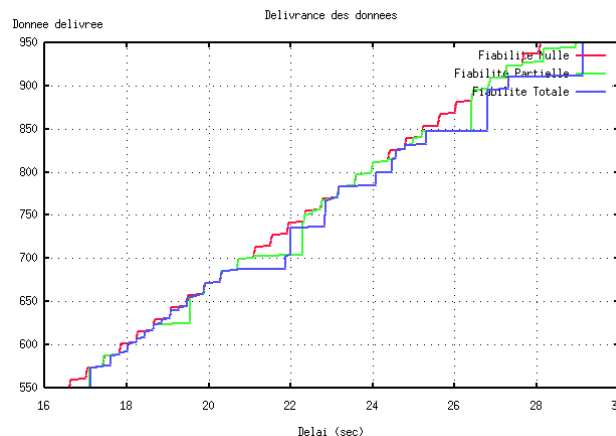


Figure II-25 - Effet de la fiabilité partielle sur la consommation du flux audio

On remarque la présence de « marches » dans la livraison des données, le comportement optimal étant une droite. Ces « marches » indiquent que des données ont été attendues et ont été délivrées en retard. Lorsque la fiabilité partielle est utilisée on aperçoit que ces ruptures de QoS sont moins nombreuses et de moins grande taille que lorsqu'une fiabilité totale est spécifiée.

5.2.4 Influence du contrôle de flux multimédia

Afin de montrer l'impact du contrôle de flux multimédia, les caractéristiques de la liaison hertzienne ont été dégradées, ce qui a eu pour effet de réduire le débit de la connexion transportant le flux vidéo (flux 2). Sur le couple de mesures composant la Figure II-26, la croissance rapide du buffer de réception de la connexion audio (flux 1) peut être observée sur le schéma de gauche. Le contrôle de flux multimédia évoqué au paragraphe 4.4.3 a été activé pour les mesures présentées dans le schéma de droite. Dès que la taille du tampon de

réception dépasse la valeur seuil, les acquittements ne sont plus retournés et rapidement, le débit de la connexion chute, entraînant ainsi une baisse progressive du buffer de réception. Le mécanisme d'augmentation de la fenêtre permet ensuite de stabiliser le débit de la connexion audio à une valeur proche de celui de la connexion vidéo.

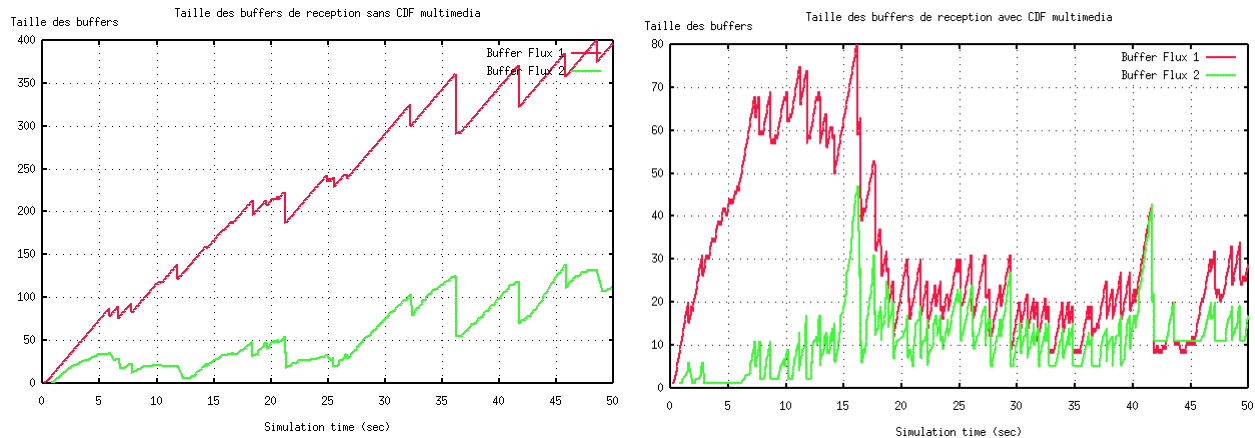


Figure II-26 - Evolution de la taille des tampons de réception

5.3 Bilan

Comme dit précédemment, l'évaluation de performances « multi-réseaux » a été réalisée à l'aide du simulateur OPNET, par manque d'accessibilité à des réseaux à caractéristiques diverses (i.e. mobiles ou satellitaires). Toutefois, les simulations de l'application de visioconférence ont porté sur des couples de réseaux intrinsèquement différents (ATM terrestre et satellite géostationnaire), (ATM terrestre et Internet), (RNIS et Internet). Les résultats sont à comparer avec le cas où les deux flux sont émis sur le même réseau.

Dans le cas où un seul réseau est utilisé, en l'occurrence un réseau par satellite, il apparaît qu'une dégradation du service réseau a un impact sur la qualité de service des deux flux, ce qui peut rendre la présentation difficile à suivre. Dans le cas où plusieurs réseaux sont utilisés, des améliorations ont pu être montrées lors des simulations. Transmettre les données audio sur un réseau RNIS, et la vidéo sur un réseau par satellite, a permis d'assurer la qualité du flux audio, alors que le flux vidéo a été dégradé. Par contre le délai de bout en bout a augmenté, du fait de la synchronisation à respecter entre les données vidéo et audio. Les mêmes remarques peuvent être faites dans le cas de la comparaison d'un simple réseau ATM surchargé, et du couple de réseaux (ATM terrestre et service UBR, RNIS), la vidéo empruntant la liaison ATM, et l'audio la liaison RNIS.

Il apparaît également que le RTT (Round Trip Time) influence fortement les performances, en particulier lorsqu'il est nécessaire de faire des retransmissions. Par exemple, dans la configuration (RNIS, satellite GEO), des pertes sur le lien RNIS sont moins pénalisantes que des pertes sur le lien satellite. Il reste donc important de bien adapter la taille des tampons du protocole aux deux RTT observés sur les deux réseaux, pour pouvoir garantir la synchronisation audio/vidéo entre les flux. Néanmoins, même si ce problème n'est pas vital pour une application de vidéo à la demande par exemple, qui peut toujours avoir de très gros tampons, elle est extrêmement stratégique pour des applications plus interactives, comme une application de visioconférence.

6 Conclusion

Ce chapitre a proposé un nouveau protocole de transport, qui permet de tirer profit des réseaux accessibles, en accordant au mieux les besoins spécifiques des applications, aux caractéristiques particulières de chacun des réseaux accessibles. Ce protocole multi-réseaux, est basé sur une extension du protocole MMPOC, dont l'intérêt principal est d'être adapté à la transmission de flux multimédia. Nous montrons que l'intérêt des mécanismes d'ordre et de fiabilité partielle est accru dans un contexte multi-réseaux. D'une part, par la fiabilité moindre des réseaux hertziens, d'autre part par l'augmentation des risques de désynchronisation induits par l'utilisation simultanée de plusieurs réseaux.

D'autre part, il a été choisi d'implémenter ce protocole en utilisant le concept nouveau de réseaux actifs, qui permet, entre autre, un déploiement facile du protocole. Une solution utilisant les capacités de téléchargement dynamique de protocole du navigateur Web *HotJava* a été décrite.

Des évaluations et des mesures ont été faites sur les protocoles MMPOC et MMPOC-MN, montrant les bénéfices apportés par l'utilisation de tels protocoles. L'étude s'est focalisée sur un cas particulièrement défavorable afin de montrer les limites d'un tel protocole.

L'application spécifique au protocole MMPOC-MN ses besoins en qualité de service, d'une part sous forme d'un ensemble de critères permettant la sélection automatique du réseau, et d'autre part sous la forme d'un ordre et d'une fiabilité partielle pour les données des flux. Depuis la conception du protocole MMPOC [CHA95], jusqu'aux premières implantations [FOUR97], l'ordonnancement des données était exprimé sous la forme d'un Réseau de Pétri, et la fiabilité partielle sous la forme d'un ensemble de valeurs exprimant le pourcentage de pertes acceptables par flux. Le chapitre suivant, montre les limites d'une telle approche en terme de facilité d'utilisation et de pouvoir d'expression. Une approche basée sur la définition d'un langage de flux est alors proposée.

Chapitre III
Spécification des Contraintes
d'Ordre et de Fiabilité

1 Introduction & Les besoins

Le chapitre précédent détaille les mécanismes du protocole POC utilisé à des fins de synchronisation inter flux. Basé sur la gestion d'un ordre partiel, décrivant finement les contraintes d'ordonnancement existant entre les données d'un même flux, ou bien entre les données appartenant à des flux différents, le protocole permet d'optimiser leur délivrance à l'application, en autorisant des déséquences de données non pénalisant pour l'application.

De nombreux travaux ont adressé directement ou indirectement le protocole POC. On peut citer, par exemple, les auteurs suivants [AME94] [CHAS95] [OWE96] [FOUR97] [ROJ00] qui présentent respectivement :

- les principes du protocole,
- les mécanismes protocolaires mis en œuvre,
- des exemples de réalisation de service de communication pour des applications particulières : visioconférence, vidéo à la demande (MPEG).

Un des points communs existant entre tous ces travaux est l'utilisation de formalismes basés sur les réseaux de Pétri pour la spécification des ordres partiels, sans qu'aucune autre méthode de spécification n'ait été explorée. Seuls les travaux de [LEC01] proposent une méthode alternative de spécification « à la volée » de la fiabilité pour des flux monomédias.

Les réseaux de Pétri permettent de modéliser d'une façon graphique des processus concurrents et asynchrone, et sous certaines conditions, d'appliquer des méthodes de validation du modèle. Toutefois, si ce formalisme présente des qualités concernant la représentation d'un ordre partiel, il n'intègre pas directement la possibilité de spécifier la fiabilité, comme cela est requis pour la définition de la qualité de service garantie par le protocole POC. De plus, l'utilisation d'une méthode graphique pour la représentation de la qualité de service, lors de la programmation d'une application, nécessite l'utilisation d'un logiciel « ad hoc » permettant d'intégrer la spécification au programme.

Pour toutes ces raisons, nous proposons un autre formalisme pour la modélisation des flux à ordre partiel et à fiabilité partielle. Suivant une pratique héritée des algèbres de processus, nous définissons une algèbre de flux par des opérateurs, ceux-ci étant ensuite interprétés dans les domaines sémantiques choisis. Le principal domaine est celui des ordres partiels étiquetés, et hérite des définitions et résultats du domaine du « vrai parallélisme ». Nous pouvons distinguer :

- Une syntaxe de flux comportant des opérateurs d'ordre et des opérateurs de fiabilité.
- Une modélisation par des (langages d') ordres partiels étiquetés, ou les opérateurs mentionnés sont interprétés par des opérations sur les langages d'ordres partiels.
- Une modélisation par des automates synchronisés capables de reconnaître et d'accepter des langages d'ordres partiels étiquetés. Ces automates sont également structurés par des opérateurs reflétant la syntaxe des flux.

Les opérateurs spécifiant l'ordre partiel entre les données conduisent à la définition d'un ordre partiel sans perte appelé le langage émis. Les opérateurs spécifiant la fiabilité, c'est à dire les pertes admissibles, se modélisent par l'ensemble des sous flux du langage émis qui satisfont les contraintes de fiabilité attendues, et appelé le langage délivrable.

Les définitions existantes [DIAZ93] [AME93] sont basées sur les ensembles des linéarisations des ordres partiels et induites par la définition des séquences de données délivrables a

l'application. Par rapport à celles-ci, l'utilisation formelle des ordres partiels s'avère déterminante par la concision et la maniabilité des définitions, à la fois des flux et des opérateurs. La définition des sous flux satisfaisant les contraintes de fiabilité bénéficie largement de ces caractéristiques. Il est plus facile de reconnaître une linéarisation à partir d'une définition de l'ordre partiel, que de reconstituer ce dernier à partir de ses linéarisations. De plus, les comportements des protocoles capables de reconnaître ces flux peuvent être définis par des compositions d'automates, chacun capable de traiter un sous-flux linéarisé particulier.

Le chapitre est structuré ainsi :

D'abord une brève présentation des formalismes utilisés pour la spécification de qualité de service dans les études antérieures du protocole POC est faite.

Le modèle que nous avons développé est ensuite détaillé, où la description des concepts mis en œuvre est suivie par la définition de la grammaire de flux.

Parallèlement à la méthode de spécification, la définition des automates de flux est fournie.

Une série d'exemples illustrent enfin le modèle. Des exemples simples sont d'abord donnés pour présenter la méthode de spécification de qualité de service. Des exemples complexes sont ensuite abordés dans l'optique de montrer le pouvoir d'expression de la grammaire.

2 Spécification de la QoS du service POC

Ce paragraphe a pour but de présenter les différents formalismes utilisés pour la spécification de la qualité de service lors des précédents travaux sur le protocole POC. Comme indiqué lors de la présentation du protocole, la qualité de service qu'il garantit est représentée par deux composantes : la fiabilité et l'ordre de la livraison des données.

Les performances du service rendu par le protocole, en terme de délai de livraison, sont entièrement dépendantes de la qualité de service demandée. Pour un même système de communication, entre deux services spécifiés, le service le moins contraignant (le moins ordonné et/ ou le moins fiable) sera rendu avant (voire dans le même temps si le système de communication est idéal) que l'autre service, probablement en attente de données perdues ou retardées.

Malgré la simplicité apparente de la représentation, la spécification de la qualité de service requise par une application est une chose délicate et dont la performance générale de l'application est totalement dépendante. La méthode de spécification de la QoS doit être suffisamment souple pour permettre l'expression la plus fine des contraintes d'une application (modéliser les contraintes d'ordonnancement d'un flux MPEG par exemple). Mais d'un autre côté son expression doit pouvoir donner lieu à des modules qui pourront garantir cette QoS.

Le pouvoir d'expression de chaque formalisme étant généralement différent, l'implémentation du protocole qui permet de garantir cette QoS doit être spécifique au formalisme employé.

La section suivante se propose d'examiner différents formalismes permettant de spécifier un ordre partiel ainsi que les notions de fiabilité associées. Ensuite, les méthodes de spécification employées pour les différentes souches des protocoles MMPOC [FOUR97] [ROJ00] seront présentées.

2.1 Notion de flux

La technique de modélisation de flux devant être abordée en priorité, les formalismes de modélisation sont décrits en suivant. Les applications multimédias génèrent le plus souvent des flux de données continus, tels que la vidéo ou l'audio, qui possèdent une structure répétitive. La durée, et par conséquent le nombre de données échangées d'un flux n'étant à priori pas connus lors de la conception d'une application multimédia, il est difficile d'explicitement les contraintes existantes entre toutes les données constituant le flux.

Toutefois, la représentation d'une période de temps contient suffisamment d'informations pour exprimer l'ensemble des besoins en termes d'ordre et de synchronisation. Pour une application multimédia de type visioconférence par exemple, la qualité de service requise pour être définie pour une période de base (par exemple 1 seconde) sera répétée jusqu'à la fin de la transmission des données. Pour un flux MPEG, la qualité sera définie pour un groupe d'images, puis répétée jusqu'à la fin du flux.

Cette technique permet par conséquent, de simplifier la modélisation d'un flux qui peut être vu comme un ensemble de données de taille infinie, par un ensemble fini sur lequel on exprime les contraintes d'ordre et de fiabilité. Toutes les données sont numérotées, ce qui permet d'exprimer l'ordre et la fiabilité sur des ensembles finis d'entiers.

2.2 Spécification de l'ordonnement

2.2.1 Ensembles de séquences

La première façon de représenter un ordre partiel sur un ensemble fini d'entiers est de spécifier toutes les séquences qui respectent cet ordre partiel. Ainsi, toutes les séquences admissibles de livraison des données, qui respectent la qualité de service désirée, sont énumérées. Si la QoS demandée spécifie un ordre total (cas du service TCP par exemple), une seule séquence sera admissible et sera celle d'émission. Si aucun ordre n'est souhaité (UDP), les séquences acceptables seront toutes les permutations de la séquence d'entiers constitués des numéros des données émises.

A titre d'exemple, supposons l'émission d'un ensemble de 5 données numérotées $E = (1,2,3,4,5)$. Considérons que la définition informelle du service souhaité soit : « La donnée 1 doit être délivrée la première, et celle numérotée par 5 en dernier ». En utilisant une notation séquentielle, cette qualité de service pourrait être définie par l'ensemble des séquences admissibles, c'est à dire $R = \{ (1,2,3,4,5), (1,2,4,3,5), (1,3,2,4,5), (1,3,4,2,5), (1,4,2,3,5), (1,4,3,2,5) \}$. La délivrance des données à l'application par le protocole se fera alors selon l'ordre défini par l'une de ces séquences.

La notation séquentielle utilisée pour la spécification d'ordres partiels est un moyen efficace pour les petites séquences de données, mais pour des séquences de taille importante, l'utilisation de cette notation peut induire des problèmes d'explosion combinatoire. Il est donc clair que du fait de la lourdeur de son écriture, la notation séquentielle est à exclure pour la spécification de la qualité de service par l'application.

2.2.2 Réseaux de Pétri et extensions

Les réseaux de Pétri sont largement utilisés pour représenter les systèmes discrets à évolution simultanée ; en cela ils sont à priori adaptés pour modéliser les flux d'information parallèles

qui prennent part à la présentation d'un document multimédia. En outre, le caractère graphique des réseaux de Pétri permet de facilement mettre en œuvre des paradigmes tels que le paradigme de régie numérique [SEN94].

Ces dernières années de nombreux modèles de représentation de la synchronisation reposant sur le formalisme de réseau de Pétri ont vu le jour. Dans tous ces modèles, une place est associée à la présentation d'une donnée (image, séquence vidéo, échantillon sonore, ...), et à la structure du réseau et les transitions exprimant les relations de dépendance intra et inter-flux. Il définit en l'occurrence l'ordre partiel de délivrance des données.

Un exemple de réseau de Pétri spécifiant les contraintes de dépendance pour une période d'une application audio/vidéo est présenté Figure III-1. Chacune des places correspond à la présentation d'une donnée sonore ou vidéo. Le graphe d'accessibilité de ce réseau de Pétri définit l'ensemble des ordonnancements possibles de délivrance des données.

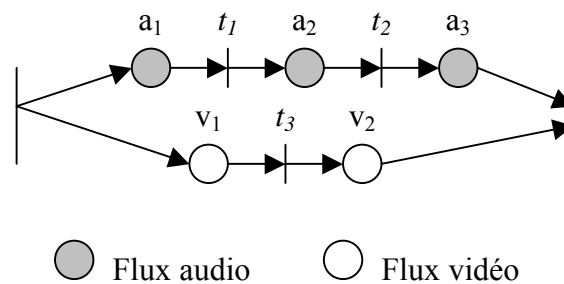


Figure III-1 - Exemple de spécification d'ordre à l'aide d'un Réseau de Pétri

2.3 Spécification de la fiabilité

2.3.1 Ensembles de séquences

De même que pour la définition de l'ordre partiel, la définition d'un service à fiabilité partielle peut se faire à l'aide de la notation séquentielle en spécifiant les pertes acceptables sur la période d'un flux.

Prenons l'exemple précédent (2.2.1) décrivant l'émission d'un ensemble E de 5 données numérotées. Si la perte d'un de ces éléments est acceptable par l'application, alors la définition du service par une notation séquentielle sera l'énumération de toutes les parties de l'ensemble E contenant au moins 4 éléments : $P = \{ (1,2,3,4,5), (2,3,4,5), (1,3,4,5), (1,2,4,5), (1,2,3,5), (1,2,3,4) \}$.

Un fois de plus, cette notation ne peut être utilisée directement pour la spécification de la qualité de service requise par une application du fait de sa lourdeur d'écriture. D'autant plus que l'ensemble des séquences de données acceptables est augmenté lorsque l'on combine la définition d'un service partiellement ordonné avec celle d'un service partiellement fiable pour définir un service S à fiabilité et ordre partiels.

2.3.2 Taux de pertes acceptables

Une des solutions utilisée pour la définition de la fiabilité dans l'implantation du protocole MMPOC [FOU97] consiste à spécifier un taux de pertes acceptables pour la période décrivant un flux. Ce taux peut être défini comme étant un nombre de pertes acceptables sur cette

période, ou bien un pourcentage évalué sur plusieurs périodes. Par exemple, une application peut décider d'accepter la perte de 25 % des données composant un flux sans qu'il n'y ait besoin d'effectuer de retransmissions pour en améliorer la qualité.

Cette méthode a l'avantage d'exprimer de manière concise et simple la qualité globale qu'une application est prête à accepter pour un de ses flux. Mais en revanche, elle ne permet pas une gestion précise des pertes. En effet spécifier un taux de perte globale sur un flux ne peut pas prendre en compte les caractéristiques intrinsèques du flux, comme par exemple l'importance de certaines données pour un flux possédant une structure hiérarchique (comme c'est le cas dans MPEG). Rien ne garantit alors qu'un objet perdu, essentiel au décodage du flux, sera récupéré, ce qui aurait pour conséquence d'être particulièrement nuisible à la présentation. D'autre part si le pourcentage de données reçues n'est pas atteint, la récupération de données peu importantes peut être pénalisante en augmentant le délai, sans pour autant apporter d'amélioration particulière à la qualité de restitution du flux.

2.3.3 Extension des réseaux de Pétri

[ROJ00] propose l'utilisation d'une extension des réseaux de Pétri, nommée Réseaux de Pétri à Flux Temporel (RdPFT), proposée par [DIA93] et [SEN96], pour la spécification des pertes tolérées.

Hormis le fait que ce modèle est temporisé, de nouvelles transitions possédant des sémantiques particulières sont définies. Les transitions du RdPFT permettent de spécifier les données dont la livraison est impérative et celles dont la perte est tolérée. Par exemple, une sémantique de type « *et* » induit la délivrance complète des données à l'entité utilisatrice. A l'opposé, une synchronisation de type « *maître* » permet, quant à elle, de définir explicitement l'élément qui doit être délivré, et par là même, les éléments dont la livraison est indispensable. Selon les règles de tir du RdPFT, une transition maître ne peut être tirée que lorsque le processus maître associé est terminé, les autres processus, appelés processus esclaves seront alors interrompus afin de tirer la transition au plus vite.

L'exemple suivant étend l'exemple présenté en 1.2.2, en ajoutant à la notion d'ordre partiel défini par le réseau de Pétri, la notion de flux prioritaire. Si l'on considère le flux audio comme étant le plus contraint, l'interruption de sa délivrance à l'application lors des points de synchronisation avec le flux vidéo peut rendre le flux audio incompréhensible, et enlever ainsi tout intérêt à l'application. Rendre le flux audio prioritaire peut être exprimé par l'utilisation d'une transition de type *maître* au point de synchronisation des deux flux (transition t_4). Comme le montre la Figure III-2, à l'arc associé à la place a_3 une transition de type *maître* est indiquée par un trait plus épais. Lorsque l'élément a_3 est reçu, la transition t_4 peut être tirée, que l'élément v_2 ai été reçu ou non.

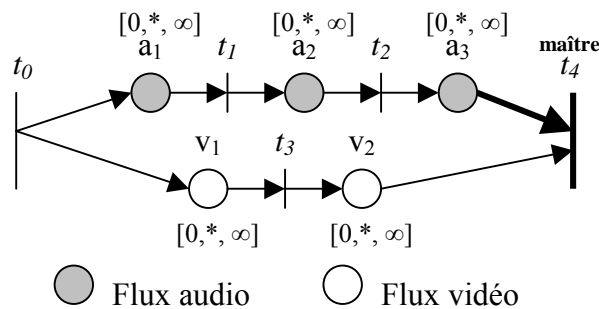


Figure III-2 - Exemple de spécification d'ordre et de fiabilité à l'aide d'un RdPFT

Cette façon d'exprimer les pertes est basée sur l'aspect graphique des réseaux de Pétri. Malgré l'apport incontestable du dessin dans la compréhension et la représentation d'une spécification, ce modèle ne peut pas s'intégrer directement dans les langages de programmation courants et les spécifications doivent par conséquent être réalisées dans un environnement de développement spécialisé. Cette dernière particularité est une chose pénalisante pour les environnements de développement déjà très lourds.

D'autre part, même si une spécification qualitative du type de pertes admises est possible, il est impossible de quantifier explicitement des pertes. Par exemple, il n'est pas possible, à l'aide de ce formalisme, de spécifier que l'on accepte la perte d'un échantillon sonore quelconque parmi les 3 émis.

2.4 Exemples

Pour conclure ce paragraphe présentant les différents formalismes de représentation d'une spécification d'ordre et fiabilité, deux exemples de spécification, telles qu'elles doivent être écrites dans les implémentations de [FOU97] et [ROJ00], sont donnés.

L'application de référence souhaite transférer un flux audio et deux flux vidéo. Le même ordre partiel sera utilisé pour les deux formalismes. Toutefois, à cause d'un pouvoir d'expression différent, la spécification de la fiabilité diffère : Dans la spécification MM-POC les flux vidéo tolèrent 34% et 40% de pertes, et le flux audio 5%, alors que pour la spécification RdPFT, le flux audio est maître (par conséquent ne supporte pas de pertes) et les flux vidéo sont, quant à eux, esclaves (et tolèrent donc 100% de pertes).

2.4.1 MM-POC

```

/* Codage de l'ordre partiel intra flux */
int QoSMM[] = {
    6,          /* nombre de places */
    /* étiquette du flux, nombre de prédécesseurs, prédécesseur 1, ..., prédécesseur n */
    VIDEO1, 3, 0, -1, -3,      /* (donnée 1) */
    VIDEO2, 3, 0, -1, -3,      /* (donnée 2) */
    SON,     3, 0, -1, -3,      /* (donnée 3) */
    VIDEO1, 1, 1,              /* (donnée 4) */
    VIDEO2, 1, 2,              /* (donnée 5) */
    VIDEO1, 1, 4,              /* (donnée 6) */
} ;

/* Codage des fiabilités partielles intra flux */
int data_QoSMM_video1 [] = {
    40,          /* pourcentage de pertes */
    3            /* nombre maximum de pertes consécutives */
} ;

int data_QoSMM_video2 [] = {
    34,          /* pourcentage de pertes */
    2            /* nombre maximum de pertes consécutives */
} ;

int data_QoSMM_son [] = {
    5,          /* pourcentage de pertes */
    1            /* nombre maximum de pertes consécutives */
} ;

```

Figure III-3 - Exemple de spécification MM-POC

Le paramètre QoSMM de la Figure III-3 représente l'ordre partiel. Chaque ligne représente une donnée suivie du nombre de ses prédécesseurs directs ainsi que de leur numéro. Par exemple, comme cela est visible sur le Réseau de Pétri suivant dont les places ont été numérotées, la donnée 5 du flux vidéo 2, est précédée d'une seule donnée numérotée par 2. Dans le cas particulier des données débutant la période représentative du flux (*partie encadrée du RdP*), les numéros de leurs prédécesseurs sont calculés.

Avantages

Spécification intégrée au langage de programmation

Inconvénients

Spécification quantitative des pertes uniquement (cf. 2.3.2)
Complexité du langage de spécification

2.4.2 RDPTS

Ci-dessous (Figure III-4), le même exemple traité à l'aide d'un RdPFT. Les contraintes temporelles n'ont pas été fixées. Pour ce faire, on laisse une valeur infinie aux intervalles temporels, autorisant ainsi constamment le tir des transitions. Seule la période représentative (partie encadrée du schéma) suffit à spécifier la QoS requise par l'application. Le reste a été représenté pour expliciter la numérotation des données référencées par le précédent exemple.

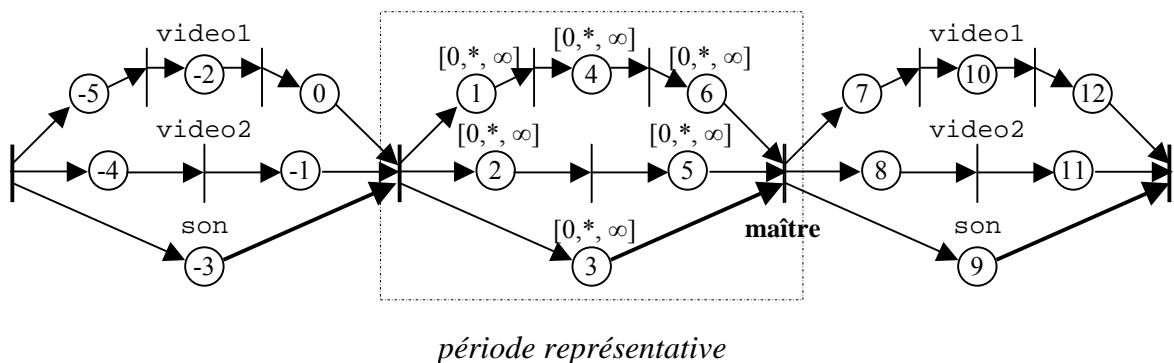


Figure III-4 - Exemple de spécification RdPFT

Avantages

Spécification naturelle des paradigmes de synchronisation multimédia (cf. 2.2.2 & 2.3.3)

Inconvénients

Spécification qualitative des pertes uniquement (cf. 2.3.3)
Spécification séparée du langage de programmation

3 Modèle de flux

3.1 Introduction

Les problèmes posés par la gestion de la fiabilité et des ordres partiels dans les protocoles de communication résident dans la complexité des spécifications de Qualité de Service (ordre et fiabilité), ainsi que dans la spécification et la validation des protocoles. Ceci est un frein à leur déploiement et l'utilisation de méthodes formelles devient alors indispensable. Le travail présenté ici propose un *langage de spécification et une modélisation de flux de données*, permettant de spécifier et représenter les flux échangés par une application multimédia répartie, ainsi qu'une *classe d'automates* définissant le comportement des protocoles associés à ces flux.

La reconnaissance d'un flux, côté récepteur, est effectuée par un automate qui analyse le numéro de séquence des données reçues et qui indique au protocole l'opération à appliquer sur ces données. La donnée peut être délivrée ou « bufferisée » pour garantir une délivrance dans l'ordre demandé, ou bien des retransmissions peuvent être demandées pour garantir la fiabilité. Afin de permettre une implémentation des modules de reconnaissance efficace et au comportement fiable, un modèle d'automate est développé en parallèle de la syntaxe. A chaque terme de la syntaxe est associé un automate chargé de la vérification de la conformité du flux par rapport à la QoS demandée, fonction que nous appellerons par la suite « reconnaissance » du flux.

L'objectif de ce modèle est de fournir un cadre regroupant un langage de spécification de flux, un modèle de flux, et la définition d'un type d'automate accepteur de ces flux. L'intérêt de cette démarche est double. Premièrement l'utilisation d'un langage de spécification garantit au programmeur la validité de sa spécification. Deuxièmement, de la définition des automates, les algorithmes de reconnaissance peuvent être déduits, garantissant ainsi leur fonctionnement. Ce modèle permet à partir de la spécification du flux de synthétiser l'automate associé.

L'apport principal de ce travail est donc un cadre formel définissant, structurant et reliant entre eux le langage de spécification de flux, la modélisation des flux par des langages d'ordres partiels, et les automates accepteurs permettant de traiter et reconnaître ces langages.

3.2 Concepts formels

Afin de faciliter la compréhension du modèle, ce paragraphe se propose de définir clairement les termes utilisés dans la suite de la description du modèle. Les différentes sémantiques rencontrées sont ensuite présentées.

Le modèle fournit deux sémantiques, celle des flux et celle des automates. Pour chacune de ces deux sémantiques, l'aspect compositionnel sera mis en évidence. Bien que présentés séparément, nous montrerons la « complémentarité » qui existe entre les deux sémantiques.

3.2.1 Présentation du modèle

Le modèle propose un langage qui permet de spécifier des flux multimédias, en spécifiant d'une part leur structure, c'est à dire l'ordonnancement des données qui les composent, mais

aussi la fiabilité de chacun des flux monomédias. Les domaines sémantiques du modèle sont décrits un peu plus loin. [Figure III-5]

Un terme du langage, en d'autres mots une *spécification*, représente une façon formelle d'exprimer la structure et la qualité que l'on souhaite associer au flux.

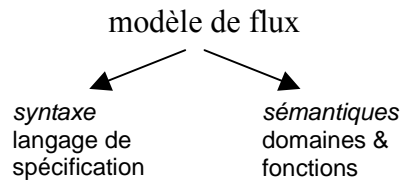


Figure III-5 - Structure du modèle

Le langage de spécification est défini par une syntaxe abstraite. Il comprend un ensemble de *types* et d'*opérateurs* pour la modélisation du flux [Figure III-6]. Tous les types de la syntaxe peuvent être assemblés à l'aide de ces opérateurs, et ce afin de définir une nouvelle spécification de qualité de service. Cette grammaire est appelée *grammaire de flux*.

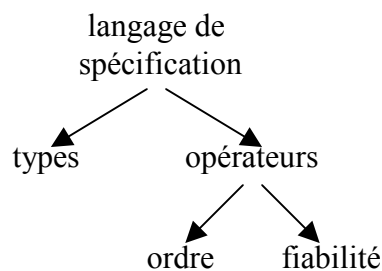


Figure III-6 - Grammaire de flux

Différentes sémantiques du modèle associent à chaque terme de la grammaire un élément d'un domaine sémantique. À chacune des spécifications produites par la syntaxe de flux sont associés un ensemble de flux et un automate.

- 1) L'ensemble de flux est composé par la structure du flux qui sera effectivement envoyé par l'application émettrice, et par l'ensemble des flux qui seront acceptables (délivrables) pour l'application réceptrice.
- 2) L'automate est défini de façon à reconnaître l'ensemble des flux acceptables.

Les domaines sémantiques du modèle [Figure III-7] sont les suivants :

- Les flux émis,
- L'ensemble des flux acceptables,
- Les automates.

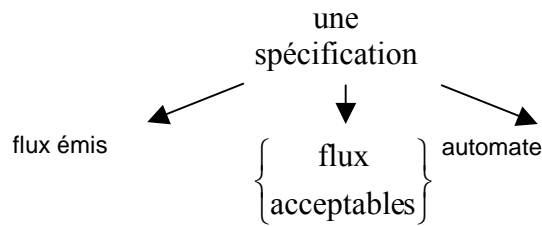


Figure III-7 - Les sémantiques du modèle

Les sémantiques de ce langage sont dites compositionnelles. Des opérateurs ont été définis dans la syntaxe, comme dans les domaines, de sorte que la composition de deux éléments du même domaine sémantique appartienne encore au domaine et que l'image de la composée soit la composée des images [Figure III-8]. Cette propriété est valable pour le domaine des flux ainsi que pour celui des automates.

La complexité est l'un des principaux écueils lors de la définition d'un modèle. Les preuves deviennent alors difficiles et l'apport du modèle devient discutable. Dans le but d'offrir un modèle complet et validable, la notion de composition permet la simplification du langage et des preuves. Il réduit considérablement leur complexité, puisque les flux comme les automates sont définis par composition d'éléments plus simples, et les preuves peuvent être faites par induction sur la structure de ces objets à partir de preuves simples à établir.

Syntaxe	Domaines
$t_1 \text{ op } t_2$	$F(t_1 \text{ op } t_2) = F(t_1) \text{ op}_F F(t_2)$

Figure III-8 - Compositionnalité des sémantiques

3.2.2 Sémantique de flux

Les données échangés par le protocole peuvent être représentés par un ensemble d'éléments étiquetés et numérotés. Une spécification de flux définit l'ensemble partiellement ordonné de ces éléments. Spécifier un flux, à l'aide de la grammaire, revient à définir l'ensemble des ordres partiels délivrables qui respectent la spécification, ainsi que l'ordre d'émission des données. Nous appellerons cela un *langage d'ordres partiels*.

Prenons l'exemple d'un flux, dont la période représentative est composée de six données de deux types différents (*a* et *b*). L'ordre partiel spécifié sous forme graphique pour ce flux est défini dans la partie gauche de la Figure III-9. Une séquence choisie arbitrairement parmi les séquence de données respectant l'ordre partiel, déterminera l'ordre d'envoi des données par le protocole.

Une séquence d'émission logique pour l'exemple peut être $\{(a,1),(b,2),(a,3),(a,4),(a,5),(a,6)\}$.

Un ensemble regroupe toutes les séquences acceptables par l'application réceptrice, c'est à dire celles qui respectent l'ordre partiel. Cet ensemble est, pour le protocole, l'ensemble des séquences délivrables.

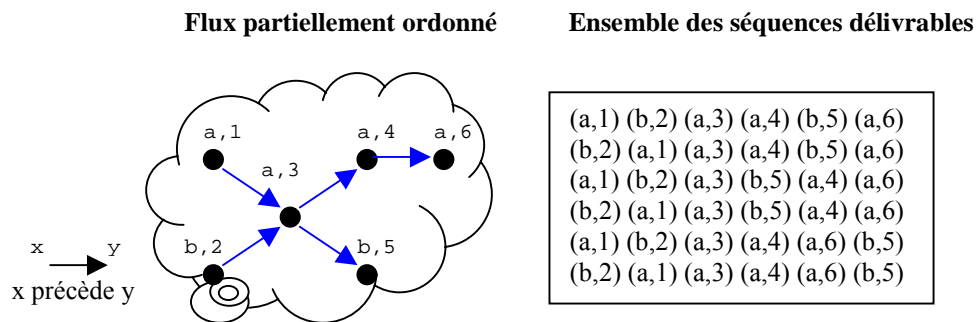


Figure III-9 - Exemple de spécification de flux partiellement ordonné

La gestion de la fiabilité partielle (Figure III-10) étend la cardinalité de l'ensemble des séquences délivrables, puisque si l'on autorise la perte de certains éléments du langage, tous les sous-ensembles (*sous – mots*) des séquences acceptables devront aussi être acceptables.

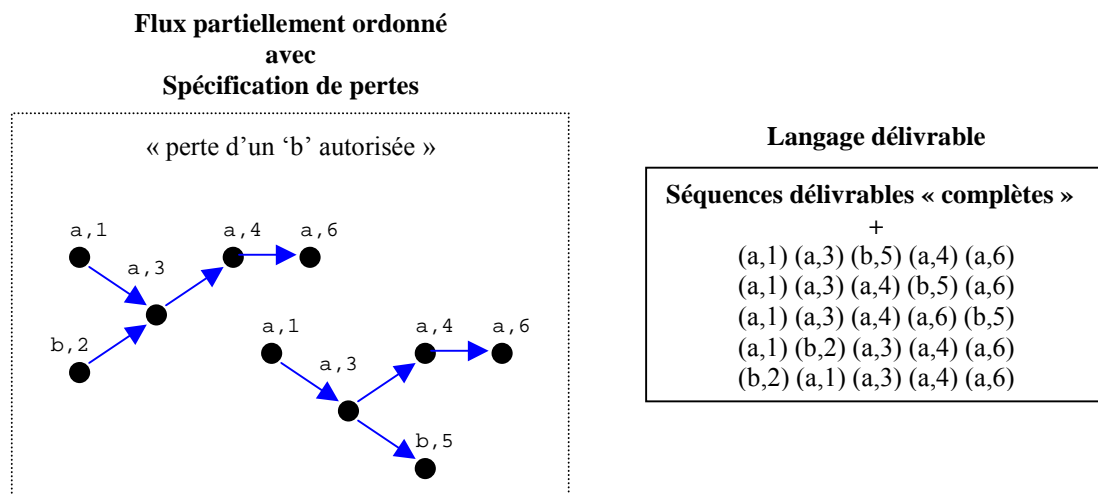


Figure III-10 - Exemple de spécification de l'ordre et de la fiabilité d'un flux

Avec la notion de fiabilité partielle deux langages sont définis, le *langage émis* $EL(t)$ et le *langage délivrable* $DL(t)$. Pour un terme d'une spécification (t), le premier correspond intuitivement à l'ensemble des séquences d'émissions possibles, c'est à dire sans perte, et le second correspond à l'ensemble des sous-mots du *langage émis* qui respectent les contraintes de fiabilités, c'est à dire à l'ensemble des séquences admissibles au regard de la spécification.

Le langage de spécification est composé de motifs finis, de flux élémentaires et de flux synchronisés : Un *flux synchronisé* est défini comme étant la composition en parallèle d'un ensemble fini de *flux élémentaires* assemblés par des *expressions de synchronisation*. Chaque flux élémentaire est structuré par l'itération d'un *motif fini*, c'est à dire composé d'occurrences consécutives d'un ensemble fini partiellement ordonné d'éléments étiquetés.

Cet ensemble particulier est appelé *motif*. Une expression de synchronisation contraint l'ordre de délivrance entre les motifs composant deux flux élémentaires.

Des pertes acceptables peuvent être spécifiées sur chaque flux élémentaire, et cela de deux façons complémentaires :

- Soit par un ensemble de données dites « obligatoires », qui doit être un sous mot du motif constituant le flux. Ce motif est appelé *motif minimal*, il appartiendra à tous les mots du langage délivrable et sera, par conséquent, délivré à chaque itération du motif.
- Soit en spécifiant un nombre maximal de pertes admissibles calculées lors de la réception d'un ensemble de motifs, cette gestion étant dite des *pertes par fenêtres*.

Les éléments du langage sont présentés en détail dans les paragraphes suivants.

3.2.3 Sémantique d'automate

La sémantique d'automate associe à un terme t un automate noté $Aut(t)$, automate dont l'alphabet des entrées est l'ensemble des étiquettes des flux associés au terme t .

Les automates utilisés possèdent, non seulement, des états étendus (comprenant des variables et des compteurs), mais aussi des propriétés de communication et de composition.

Nous définissons deux classes d'automates correspondant à différents types de flux :

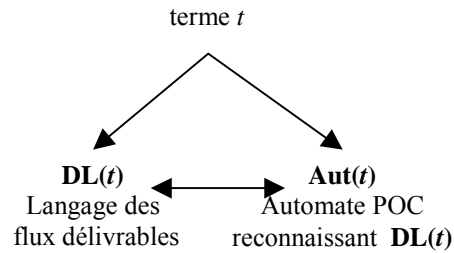
- La première permet de reconnaître des flux élémentaires (simples). Elle définit les *automates élémentaires EA*. Ces automates retirent de leur canal de communication avec l'environnement, des données étiquetées, et décident si la donnée est acceptable à la vue de la spécification. Cette opération sera appelée par la suite « acceptation ». Cette décision est prise en fonction de leur propre état « local », et de l'étiquette de la donnée.
- La seconde permet d'accepter les flux multimédias synchronisés. Les *automates synchronisés SA*, plus complexes, sont composés d'automates élémentaires reconnaissant les flux élémentaires à synchroniser. Chaque composant lit sur son canal de communication les données présentes puis les accepte, non seulement en fonction de son état local et de l'étiquette de la donnée, mais aussi en fonction des états locaux de tous les flux élémentaires composant le flux synchronisé.

Chaque *EA* lit sur son canal de communication les données présentes puis les accepte, non seulement en fonction de son état local et de l'étiquette de la donnée, mais aussi en fonction des états locaux de tous les flux élémentaires composant le flux synchronisé. La principale différence entre les automates élémentaires et les automates synchronisés réside dans une capacité étendue de test global pour les seconds qui permettent de mettre en œuvre les fonctions de synchronisation entre les flux élémentaires.

3.2.4 Dualité des sémantiques

A chaque terme de la syntaxe sont donc associés un langage d'ordres partiels et un automate, et la cohérence de ces deux modèles est représentée par la Figure III-11. La relation existant entre les deux sémantiques est définie par la propriété suivante : l'automate associé à un terme a la propriété de reconnaître exactement le langage des flux délivrables associés à ce terme. Le langage délivrable $DL(t)$ d'un terme t est le langage $L(Aut(t))$ accepté par l'automate $Aut(t)$, plus précisément $L(Aut(t))$ est l'ensemble des extensions linéaires (par la suite nous emploierons le terme linéarisation) de $DL(t)$, ce qui se traduit par : $\forall t \in T, Lin(DL(t)) = L(Aut(t))$.

En particulier, un flux composé est reconnaissable par une composition des automates élémentaires reconnaissant chacun des flux. Cette propriété rend possible la synthèse des automates de reconnaissance et par conséquent une forme de synthèse automatique de protocole.


Figure III-11 - Dualité des sémantiques

3.2.5 Les ordres partiels

Si Σ est un ensemble d'étiquettes, décrites plus loin, alors un *ensemble partiellement ordonné* étiqueté par Σ , appelé Σ -poset, est un triplet (D, \leq, λ) tel que D est un ensemble, \leq une relation d'ordre sur D , et $\lambda : D \rightarrow \Sigma$ la fonction d'étiquetage associant une étiquette de Σ à chacun des éléments de D . L'ensemble des Σ -poset est noté $P(\Sigma)$. Un *mot partiel*, aussi appelé *poset* $[D, \leq, \lambda]$ est la classe d'équivalence de tous les $(D', \leq', \lambda') \in P(\Sigma)$ tel qu'il existe un isomorphisme d'ordre $\varphi : (D, \leq) \rightarrow (D', \leq')$ et que $\lambda = \lambda' \circ \varphi$. L'ensemble des mots partiels est noté $PW(\Sigma)$.

Une relation de *sous-mots* \triangleright sur les Σ -poset s'exprime ainsi : (D', \leq', λ') est un sous-mot de (D, \leq, λ) , noté $(D, \leq, \lambda) \triangleright (D', \leq', \lambda')$ ssi D' est un sous-ensemble de D , \leq' la restriction de \leq aux éléments de D' , et λ' la restriction de λ à D' . Cette relation est aussi définie sur les mots partiels. On note $Sub(\sigma)$ l'ensemble des sous-mots de

$$\sigma : Sub((D, \leq, \lambda)) = \{ (D', \leq', \lambda') : D' \subseteq D, \leq' = \leq / D' \times D', \lambda' = \lambda / D' \}.$$

Si $\sigma = (D, \leq, \lambda)$ est un ensemble étiqueté partiellement ordonné, alors $Ext(\sigma)$ est l'ensemble des *extensions d'ordre* de σ : $Ext(\sigma) = \{ (D, \leq', \lambda) \in P(\Sigma), \leq \subseteq \leq' \}$. L'ensemble des extensions linéaires $Lin(\sigma)$ est le sous-ensemble de $Ext(\sigma)$ composé de toutes les extensions totalement ordonnées de σ .

Les compositions séquentielles et parallèles de deux Σ -poset, $\sigma_i = (D_i, \leq_i, \lambda_i)$ $i=1,2$ sont définies ssi $D_1 \cap D_2 = \emptyset$ par $\sigma_1 \cdot \sigma_2 = (D_1 \cup D_2, \leq_1 \cup \leq_2 \cup (D_1 \times D_2), \lambda_1 \cup \lambda_2)$

$$\text{et } \sigma_1 \parallel \sigma_2 = (D_1 \cup D_2, \leq_1 \cup \leq_2, \lambda_1 \cup \lambda_2).$$

L'ensemble des langages des Σ -poset, i.e. l'ensemble des sous-ensembles de $P(\Sigma)$, est noté $PL(\Sigma)$. Dans ce qui suit, les étiquettes des éléments sont un couple formé par une étiquette appartenant à l'ensemble A des étiquettes représentant le type des PDU et d'un ou plusieurs entiers. Cet ensemble A est utilisé afin de représenter les caractéristiques de chaque donnée à l'intérieur d'un flux, comme cela peut-être le cas, par exemple, dans la définition d'un flux MPEG avec les trames I, P et B. Les entiers servent à identifier de manière unique une donnée étiquetée à l'intérieur d'un motif. Ce couple est étendu pour les autres termes de la spécification, où un entier est ajouté pour identifier le numéro de motif auquel correspond la donnée d'un flux, et un autre entier identifiant le numéro de canal est utilisé afin de repérer le flux d'appartenance d'une donnée faisant partie d'un flux composé.

Le langage de flux et les automates d'acceptation sont présentés ensemble dans la section suivante. Pour chaque terme de la syntaxe, l'automate d'acceptation correspondant est décrit en suivant. Les définitions et preuves des opérateurs ne sont pas détaillées dans ce chapitre, mais sont disponibles dans le rapport [FAN00A]. La gestion des flux monomédia partiellement

ordonnés et partiellement fiables est tout d'abord abordée, puis les expressions de synchronisation et la gestion des flux multimédia clôturent la présentation du modèle.

3.3 Flux monomédia

Les éléments du langage de flux relatifs à la modélisation des flux monomédias sont d'abord présentés. Un flux monomédia permet de représenter un flot de données émanant d'une entité communicante, ou d'une application. Ce flot de données doit être structuré de manière cyclique, c'est à dire qu'il doit être composé d'un motif se répétant indéfiniment. N'incluant pas de contraintes temporelles, les échanges de données peuvent être erratiques, mais doivent respecter la structure du flux. Ainsi, les flux synchrones comme asynchrones peuvent être modélisés. Un échange « anarchique », sans structure interne, ne peut, par contre, pas être modélisé sous la forme d'un flux, mais sous la forme d'une connexion particulière ne respectant pas le modèle. Par conséquent, il ne pourra ni être synchronisé, ni posséder des contraintes de fiabilité.

3.3.1 Motifs numérotés

Les *motifs finis* p représentent la structure itérative d'un flux : ce sont des ordres partiels finis. Ils sont définis par la syntaxe suivante :

$$p := a \mid p \parallel p \mid p \cdot p \mid p^n \mid \parallel_n p$$

où $a \in A$, l'alphabet formé par les étiquettes, et n est un entier. Les motifs finis ne sont pas employés tels quels dans la syntaxe, mais sont présentés car utilisés comme base de l'implémentation. Les *motifs numérotés* m seront utilisés à la place, associant un entier à chaque élément du motif, afin d'identifier de manière unique chaque donnée d'un flux. Un algorithme de numérotation est utilisé dans l'implémentation afin de convertir les motifs finis en motifs numérotés, tout en masquant à l'utilisateur la numérotation qui n'ajoute rien à l'expression d'une spécification. Les *motifs numérotés* peuvent ainsi être considérés comme la première étape de spécification.

$$m := (a, i) \mid m \parallel m \mid m \cdot m$$

où $a \in A$, et $i \in \mathbb{Z}$. Une expression de motif numéroté est bien formée si tout entier i n'apparaît qu'une fois (la numérotation des éléments est unique), et que tout entier dans m' précède tout entier dans m'' , pour toute sous expression $m' \cdot m''$ de m .

L'ensemble des motifs numérotés bien formés est noté T_{np} .

A chaque motif numéroté bien formé est associé un ordre partiel, étiqueté par $A \times \mathbb{N}$, et constituant le langage émis, $EL(m)$. L'ordre partiel associé à un singleton (a, i) est $EL(a, i) = (\{i\}, \{(i, i)\}, \{(i, (a, i))\})$. Les opérateurs de composition séquentielle et parallèle sur les ordres partiels étiquetés sont utilisés pour définir inductivement $EL(m \cdot m') = EL(m) \cdot EL(m')$ et $EL(m \parallel m') = EL(m) \parallel EL(m')$. Le langage délivrable $DL(m)$ est l'ensemble des sous mots du langage émis : $DL(m) = Sub(EL(m))$.

3.3.2 Motifs a-fiabiles

Les *motifs a-fiabiles* représentent la première façon d'exprimer la fiabilité partielle dans la syntaxe, en introduisant la notion de séquence de données non perdables à l'intérieur d'un motif. La notion de partie minimale d'un motif permet d'exprimer, par exemple, le rôle essentiel d'un entête dans le codage d'une image. Sans ce dernier, le reste des données de l'image n'a aucun sens. Des mécanismes de retransmission sont mis en œuvre afin de garantir la présentation à l'utilisateur de cette partie du motif, appelé *motif minimal*. La syntaxe des motifs a-fiabiles est la suivante :

$$\boxed{\rho := m \mid \rho / \triangleright m}$$

où m est un motif numéroté ($m \in T_{np}$).

L'expression $\rho / \triangleright m$ définit m comme étant un sous-motif minimal, imperdable, du motif ρ . L'ensemble des motifs a-fiabiles est noté T_{lp} . Un motif a-fiable est bien formé, si le motif numéroté de base est bien formé, et si le motif minimal est un sous mot du motif de base.

Le langage émis $EL(\rho)$ est le même que celui du motif numéroté de base, $EL(\rho/\triangleright m) = EL(\rho)$. Le langage délivrable $DL(\rho/\triangleright m)$ quand à lui est l'ensemble des mots de $DL(\rho)$ ayant $EL(m)$ comme sous mot : $DL(\rho / \triangleright m) = \{u \in DL(\rho), u \triangleright EL(m)\}$.

3.3.3 Automates élémentaires de motifs

Les définitions des automates de reconnaissance associés aux termes du langage sont proposées dans [FAN00A].

Les automates décrits sont utilisés comme spécification pour l'implantation des modules de reconnaissance et par conséquent sont brièvement décrits par la suite. A chaque terme du langage correspond un automate capable d'effectuer la reconnaissance et l'acceptation du langage délivrable. Les automates seront présentés en parallèle de la syntaxe.

Les automates élémentaires ont une structure traditionnelle $\Gamma = (Q, \Sigma, T, q_0, F)$ (états, alphabet d'entrées, transitions, état initial, états finaux), avec la particularité d'avoir des variables entières dans la définition des états et de lire des entrées de type entier. Par ailleurs le produit synchronisé de deux automates est défini et reconnaît l'intersection des langages de ces automates s'ils ont même alphabet, et leur shuffle (ensemble d'entrelacements) si leurs alphabets sont disjoints .

Un automate de motif a-fiable $Aut(\rho)$ est un n-uplet $Aut(\rho) = (S, AxI, T, state_0, F)$. L'ensemble AxI des entrées est l'ensemble des étiquettes ou en-têtes du motif de base (chaque élément du motif numéroté étant caractérisé par une étiquette ($a \in A$) et un numéro unique ($i \in I \subset \mathbf{N}$)). S est l'ensemble des états, chaque état associant à un un élément du motif un marquage indiquant s'il est attendu (x), s'il a été précédemment reçu et délivré (r), ou bien s'il n'a pas été reçu mais appartient aux prédécesseurs d'un élément reçu et que sa perte à été autorisée lors de la réception de celui-ci (l), c.a.d. $S = \{I \rightarrow \{x, l, r\}\}$. Les transitions de T sont des paires (garde, opération), une garde de transition prend en compte l'étiquette en entrée et l'état courant, et si la réception de l'élément en entrée est compatible avec cet état, un nouvel état est défini par l'opération.

Pour chaque motif numéroté m deux automates sont définis : un automate reconnaissant le motif (m) de manière fiable (noté $Aut_r(m)$), et un autre automate, appelé non fiable $Aut_u(m)$,

reconnaissant tous les sous-motifs du motif m . L'automate permettant la reconnaissance d'un motif a-fiable $\rho / \triangleright m$, est alors défini inductivement comme étant la résultante du produit synchronisé de l'automate reconnaissant le motif a-fiable ρ , avec l'automate fiable reconnaissant le motif minimal m , formellement $Aut(\rho / \triangleright m) = Aut(\rho) \parallel Aut_r(m)$. Cette propriété permettra une conception modulaire des algorithmes de reconnaissance des motifs.

Pour un motif de base sans motif minimal, l'automate est $Aut(m) = Aut_u(m)$. La propriété recherchée et obtenue par ces définitions est

$$L(Aut(\rho)) = Lin(DL(\rho)).$$

Le langage reconnu par $Aut(\rho)$ est le (les linéarisations du) langage délivrable associé au motif a-fiable ρ .

3.3.4 Flux élémentaires

Un flux élémentaire correspond à la source de données d'une application monomédia ; il est caractérisé par des séquences répétitives d'information, et peut supporter une qualité plus ou moins grande. La structure interne du flux est définie comme étant la répétition (ou l'itération) d'un motif a-fiable, définissant d'une part l'ordonnancement des données, mais aussi la notion de pertes autorisées. Un nouvel élément est ajouté à la syntaxe, permettant de raffiner la spécification des pertes, en définissant un nombre maximal de pertes admissibles pour une fenêtre de motifs. La spécification de *pertes par fenêtre* s'effectue pour un ensemble d'étiquettes, et sur un nombre consécutif de motifs.

Un exemple de spécification de ce type, dans le cas d'un flux MPEG, pourrait être :

La perte de plus de 2 trames 'B' tout les 4 motifs n'est pas acceptée dans le langage délivrable.

La syntaxe d'un flux a donc la forme suivante :

$$f := \rho^* \mid f / [B, n, \lambda]$$

Où ρ^* est l'itération d'un motif a-fiable. ($\rho \in T_{lp}$), et $[B, n, \lambda] \in 2^A \times \mathbb{N} \times \mathbb{N}$ étant une expression de pertes. B indique l'alphabet sur lequel s'applique la spécification ; c 'est un sous-ensemble de l'alphabet du motif. n définit la taille de la fenêtre (en nombre de motifs). λ est le nombre d'éléments perdables dans cette fenêtre. L'ensemble des flux élémentaires est noté T_{es} .

Les langages $EL(f)$ et $DL(f)$ sont des ensembles de flux libellés sur $A \times \mathbb{N} \times \mathbb{N}$. Les éléments sont étiquetés par des triplets (a, i, h) : où (a, i) est une étiquette dans le motif et h représente le numéro de motif auquel l'élément appartient.

Formellement la sémantique de l'itération de motif peut être définie à partir de l'itération numérotée que l'on note abusivement L^* , d'un langage L , définie par

$$L^* = \bigcup_n L \times [n]. \text{ Où } L \times [n] = \{(u_1 \times 1)(u_2 \times 2) \dots (u_n \times n), u_i \in L, i = 1..n\}.$$

Chaque élément $(u_i \times i)$ d'une séquence $(u_1 \times 1)(u_2 \times 2) \dots (u_n \times n), u_i \in L, i = 1..n$ étant défini, pour

$$u_i = (D, \leq, l \times v), \text{ par : } (u_i \times i) = (D, \leq, l \times v \times \mu), \text{ où quelque soit } \forall d \in D, \mu(d) = i.$$

Du fait de la numérotation par motif et de la conservation des propriétés de *poset* par la composition en séquence de *poset* disjoints, ces ensembles sont des *poset* étiquetés par $A \times \mathbb{Z} \times \mathbb{Z}$. Les langages d'émission $EL(\rho^*)$ et délivrables $DL(\rho^*)$ peuvent être définis comme les itérations des langages de ρ :

$$EL(\rho^*) = \bigcup_{n \in \mathbb{N}} EL(\rho) \times [n] = [EL(\rho)]^* \text{ et } DL(\rho^*) = \bigcup_{n \in \mathbb{N}} DL(\rho) \times [n] = [DL(\rho)]^* .$$

Concernant la sémantique des expressions de pertes, si $f / [B, n, \lambda]$ est un flux, alors l'expression $[B, n, \lambda]$ contraint la perte des éléments libellés par une lettre de B d'être au plus λ pour chaque fenêtre de n motifs consécutifs émis. Si $|m|_B$ est le nombre d'éléments du motif de base m libellés par une lettre de B , alors pour chaque motif consécutif, au moins $(n \times |m|_B - \lambda)$ éléments libellés par B devront être délivrés. Le langage émis n'est pas modifié par une expression de perte : $EL(\rho^* / [B, n, \lambda]) = EL(\rho^*)$. Nous renvoyons le lecteur à [BER01] et [FAN01] pour la définition formelle de $DL(\rho^* / \{[B, n, \lambda]\})$.

Par composition, un flux peut-être écrit : $f = \rho^* / \{[B_i, n_i, \lambda_i]_{i=1,k}\}$, et le langage délivrable de ce flux est constitué de tous les éléments du langage délivrable de ρ^* qui satisfont chacune des expressions de pertes :

$$DL(\rho^* / \{[B_i, n_i, \lambda_i]\}) = \bigcap_{i \in [k]} DL(\rho^* / [B_i, n_i, \lambda_i]) .$$

3.3.5 Automates élémentaires de flux

Un flux élémentaire étant constitué par l'itération d'un motif fini, et ses éléments étiquetés par des triplets (a, i, h) : où (a, i) est une entrée de l'*automate de motif numéroté* et h indique le numéro du motif auquel appartient la donnée reçue l'*automate de motif a-fiable* doit être modifié, et les états des *automates de flux* comportent un entier indiquant le numéro de motif courant analysé par l'automate. Soit $Aut(\rho) = (S, A \times I, T, state_0, F)$ un automate de motif, on définit l'itération de $Aut(\rho)$ par un n-uplet $Aut(\rho)^* = (S \times \mathbb{N}, A \times I(\rho) \times \mathbb{N}, T^*, (state_0 \times 1), F \times \mathbb{N})$. Les transitions des automates de motifs sont elles aussi redéfinies : on compare le numéro de motif d'une entrée (a, i, h) avec le numéro de motif courant, et deux cas peuvent mener à l'acceptation d'une donnée :

La donnée appartient au motif courant : la garde de la transition est identique à celle de l'*automate de motif*.

La donnée appartient au motif suivant : si l'automate est dans un état final, le numéro de motif courant est incrémenté, et la donnée acceptée, sinon la donnée doit être stockée ou rejetée par le protocole.

La propriété recherchée et obtenue par ces définitions est

$$L(Aut(\rho^*)) = Lin(DL(\rho^*)) .$$

Le langage reconnu par $Aut(\rho)^*$ est le (les linéarisations du) langage délivrable associé au flux élémentaire ρ^* .

Pour la reconnaissance d'un flux élémentaire avec spécification de pertes $\rho^* / [B, n, \lambda]$, les automates précédents sont étendus par une mémoire de taille n qui comptabilise le nombre de pertes acceptées pour l'ensemble des n derniers motifs. La propriété résultante est $L(Aut(\rho^* / [B, n, \lambda])) = Lin(DL(\rho^* / [B, n, \lambda]))$.

3.4 Les flux multimédias

3.4.1 Flux synchronisés

Considérons maintenant plusieurs flux monomédias synchronisés pour introduire la notion de flux multimédias, appelés aussi *flux synchronisés*.

Un *flux synchronisé* regroupe un ensemble fini de *flux élémentaires* composés par l'opérateur \parallel , chacun d'eux associé à un *port de communication* (appelé aussi *canal*), et les synchronise à l'aide d'expressions. A chaque nom de canal correspond un flux élémentaire unique, et l'opérateur d'association d'un flux élémentaire f à un canal c est notée $c :: f$.

La syntaxe est la suivante :

$$\boxed{\psi := c :: f \parallel \psi \parallel \psi[\varepsilon]}$$

Où c est un canal ($c \in C$), et f un flux élémentaire ($f \in T_{es}$). L'opérateur d'association d'un flux élémentaire à un *port de communication* est dénoté $c :: f$. $\xi \in SE$ qui représente les expressions de synchronisations qui sont définies dans le paragraphe suivant. Les termes des ensembles de flux synchronisés sont notés T_{st} .

Les données étant caractérisées par un nom de canal, et une étiquette dans le flux élémentaire associé, les langages de flux synchronisés sont étiquetés par des quadruplets (c, a, u, h) où c est le nom du canal et ou (a, u, h) est une étiquette du flux élémentaire f . Les langages $EL(c :: f)$ et $DL(c :: f)$ sont dérivés de $EL(f)$ et $DL(f)$ par l'ajout du nom de canal aux étiquettes des éléments. Les langages émis et délivrables du terme $(\psi_1 \parallel \psi_2)$ sont les composées parallèles des langages des termes ψ_1 et ψ_2 . Ils sont plus précisément définis comme les extensions d'ordre de ces composées : $EL(\psi_1 \parallel \psi_2) = Ext(EL(\psi_1) \parallel EL(\psi_2))$ et $DL(\psi_1 \parallel \psi_2) = Ext(DL(\psi_1) \parallel DL(\psi_2))$.

3.4.2 Expression de synchronisation

Ce paragraphe présente la partie de la syntaxe permettant la spécification des contraintes d'ordre inter flux. Une expression de synchronisation simple est une paire $((c_1, k), (c_2, h)) \in (C \times \angle) \times (C \times \angle)$, avec $c_1 \neq c_2$. Cette expression contraint le motif k du flux associé au canal c_1 à être délivré uniquement avant le motif h du flux associé au canal c_2 .

Afin de synchroniser périodiquement deux flux des *expressions paramétriques* peuvent être utilisées. Supposons η_1 et η_2 deux fonctions linéaires, les paires $((c_1, \eta_1), (c_2, \eta_2))$ contraignent chaque motif numéroté $\eta_1(i)$ du canal c_1 d'être délivrés avant les motifs numérotés par $\eta_2(i)$ du canal c_2 . Cette opération correspond à une synchronisation interdisant le flux du canal c_2 à être délivré en avance sur le flux du canal c_1 . Cette synchronisation est asymétrique, où la délivrance des données d'un flux est cadencée sur celle de l'autre. La syntaxe des expressions de synchronisation est alors la suivante :

$$\boxed{\xi := ((c_1, k), (c_2, h)) \mid ((c_1, \eta_1), (c_2, \eta_2))}$$

Les langages émis et délivrables du terme $\psi[\xi]$ sont les extensions d'ordre des langages de ψ qui respectent les contraintes de l'expression ξ .

Une expression de synchronisation symétrique permettant une synchronisation de type *ET* est définie par composition comme suit :

$$((c_1, k_1) \lt \gt (c_2, k_2)) = \{ ((c_1, k_1 \times i), (c_2, k_2 \times i + 1)), ((c_2, k_2 \times i), (c_1, k_1 \times i + 1)), i \in \mathbb{Z} \}$$

La Figure III-12 présente un exemple d'expression de synchronisation.

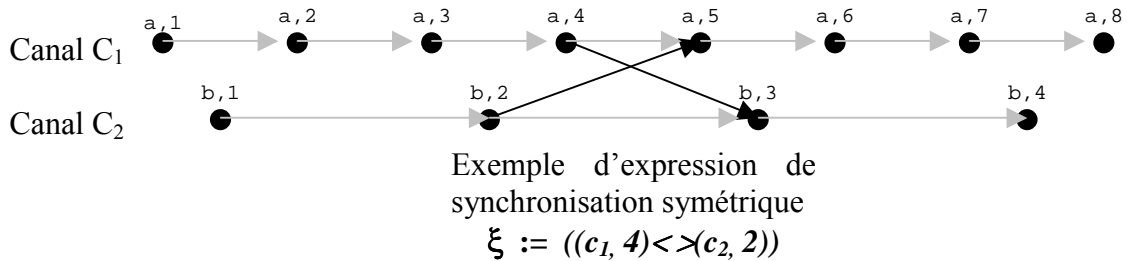


Figure III-12 - Modèle de synchronisation

3.4.3 Les automates synchronisés

Complétant la syntaxe de flux, la synchronisation des automates élémentaires de flux pour la reconnaissance des flux multimédias est proposée. La contribution est de deux ordres :

- La définition des automates synchronisés,
- La définition de nouvelles gardes pour la synchronisation des automates.

Un **automate synchronisé** est un tuple $\Delta = (C, \{\Gamma_c, c \in C\})$ est composé d'un ensemble de canaux C , chaque canal étant associé à un automate particulier Γ_c . Les données sont reçues sur les canaux et leur traitement est délégué à l'automate associé au canal concerné.

Les automates associés aux canaux sont dérivés des automates de flux élémentaires, par l'introduction de *gardes globales* dans les transitions qui permettent leur synchronisation. Le composant d'un automate synchronisé associé à un canal c est un n-uplet $\Gamma_c = (Q_c, \Sigma_c, T_c, q_{oc}, F_c)$ défini à partir d'un automate de flux élémentaire $Elem(\Gamma_c) = (Q_c, \Sigma_c, Loc(T_c), q_{oc}, F_c)$, et seules les gardes des transitions T_c sont modifiées par rapport à $Loc(T_c)$. En effet une transition de T_c est un triplet composé d'une garde locale et d'une opération provenant de $Loc(T_c)$, auxquelles on a ajouté une garde globale :

$T_c \subseteq G_c^l \times G_c^g \times O_c$ où G_c^g est l'ensemble des gardes globales qui testent l'état des automates avec lesquels l'automate Γ_c est synchronisé. Les gardes de l'automate élémentaire de flux sont $Loc(T_c) = T_c / G_c^l \times O_c$.

L'automate $Aut(c :: f)$ ne comporte qu'un $Aut(c :: f) = (\{c\}, c.Aut(f))$, ou $c.Aut(f)$ est juste dérivé de $Aut(f)$, par l'ajout du champ c dans l'alphabet des entrées. En particulier $Aut(c :: f)$ ne comporte aucune garde globale.

La mise en parallèle de deux tels automates n'est définie que si leurs ensembles de canaux sont disjoints, par la simple union des ensembles de canaux et d'automates qui les composent :

si $\Delta_i = (C_i, \{\Gamma_c, c \in C_i\})$, $i=1,2$, alors $\Delta_1 \parallel \Delta_2 = (C_1 \cup C_2, \{\Gamma_c, c \in C_1 \cup C_2\})$

Le langage reconnu par l'automate obtenu est le mélange des langages des automates composants et la propriété recherchée en découle :

$$L(\text{Aut}(\psi_1 \parallel \psi_2)) = L(\text{Aut}(\psi_1) \parallel \text{Aut}(\psi_2)) = \text{Lin}(DL(\psi_1) \parallel DL(\psi_2)) = \text{Lin}(DL(\psi_1 \parallel \psi_2)).$$

La **synchronisation** des automates est définie comme suit :

L'automate $\Delta[\xi]$ oblige la synchronisation entre les deux composants élémentaires de l'automate Δ dont les canaux de lecture apparaissent dans l'expression ξ . Chaque automate élémentaire traite les données présentes dans son canal de manière indépendante, sauf lorsqu'il reçoit une donnée possédant un numéro de motif qui, d'après l'expression de synchronisation, doit être synchronisé avec le motif d'un autre composant. Si $\Delta = (C, \{\Gamma_c, c \in C\})$, alors $\Delta[\xi] = (C, \{\Gamma_c[\xi], c \in C\})$ où $\Gamma_c[\xi]$ est différent de Γ_c uniquement si c apparaît dans l'expression ξ . Si c'est le cas, alors des gardes globales sont ajoutées à la définition de certaines transitions de Γ_c . Une expression de synchronisation n'étant pas symétrique, deux cas sont possibles : Soit le flux doit attendre un autre flux pour atteindre ce numéro de motif et présenter la donnée, soit le flux est lui même attendu.

Nous définissons, sous forme de gardes globales, l'algorithme de synchronisation suivant, lors de la réception d'une donnée étiquetée par (c, a, u, h) et considérant une expression de synchronisation ξ :

- Cas 1 : Si le canal de réception c de la donnée ne correspond pas aux canaux concernés par l'expression de synchronisation, alors aucune garde n'est ajoutée à l'automate.
- Cas 2 : Si c appartient à la partie droite de ξ alors une garde est ajoutée pour garantir que la condition de dépendance soit vérifiée.
- Cas 3 : Si c appartient à la partie gauche de ξ alors une garde est ajoutée pour vérifier que la donnée ne soit plus valable du fait d'un changement de motif anticipé sur l'autre canal.

Pour les expressions paramétriques, la définition de l'automate synchronisé $\Gamma_c[\xi]$ est une simple adaptation de l'algorithme précédent, du fait de la propriété suivante :

Pour un flux $\psi = (\|_{i \in [n]} c_i :: f_i)[\xi_1][\xi_2] \dots [\xi_k]$, $\text{Aut}(\psi) = (\|_{i \in [n]} \text{Aut}(c_i :: f_i))[\xi_1][\xi_2] \dots [\xi_k]$.

De ce fait, l'ajout d'une nouvelle expression de synchronisation à l'automate ne nécessite que l'ajout de nouvelles gardes, tel que défini précédemment.

3.5 Conclusion

Nous avons défini un langage de spécification permettant d'exprimer les contraintes de qualité de service des flux multimédias du point de vue de l'ordre et de fiabilité. Nous y avons associé deux sémantiques, l'une modélisant les flux par des langages d'ordres partiel, l'autre par des automates reconnaissant les flux satisfaisant les contraintes de QoS.

L'application des méthodes formelles dans le domaine des protocoles multimédias donne lieu à plusieurs avancées :

- Elle simplifie les spécifications de QoS dans les protocoles de transports de nouvelle génération tels que les transports à ordre et fiabilité partiels. L'expression des caractéristiques de QoS est faite de manière compositionnelle, à l'aide d'opérateurs de composition spécifiques.
- Elle définit une structure sur les protocoles directement liée à la structure des flux, permettant d'en synthétiser automatiquement une partie et garantissant son bon

fonctionnement. La définition des automates est elle aussi compositionnelle, les automates de reconnaissance des flux peuvent être générés automatiquement à partir de la spécification de QoS. La complexité de l'implémentation des protocoles associés en est ainsi réduite et simplifiée.

Une implémentation du langage, sous la forme d'une API, sera présentée dans la suite du document. Les automates seront générés durant les appels successifs des fonctions de cette API.

La hiérarchie des éléments constituant la grammaire de flux est présentée dans la Figure III-13.

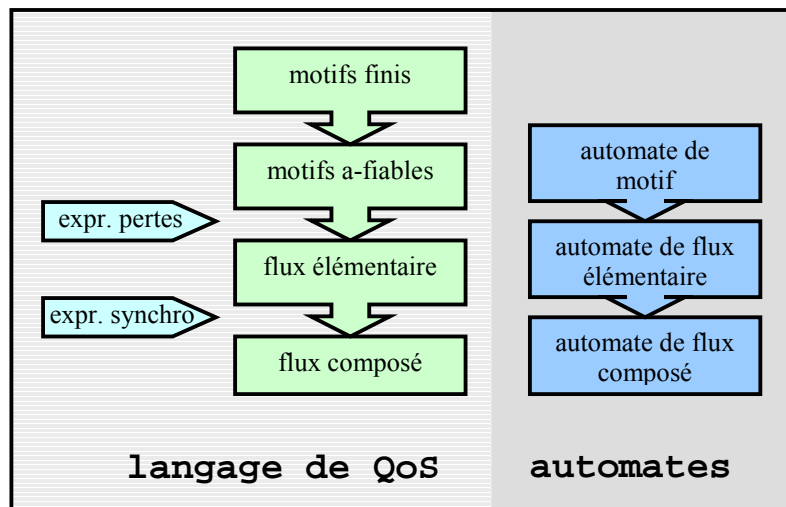


Figure III-13 - Eléments de la grammaire de flux

4 Exemples de spécification

Afin d'illustrer l'utilisation du modèle de flux, ce paragraphe se propose de donner des exemples de définitions de qualité de service pour des applications monomédias et multimédias classiques.

MPEG-1 [MPE93] étant l'un des formats de codage des données de type vidéo les plus répandus, ayant détenu le monopole jusqu'à la naissance du format *realvideo*, la définition d'un flux MPEG est d'abord présentée. Une approche simple, introduisant la notion de fiabilité partielle pour la présentation d'un flux traditionnellement considéré comme fiable, est premièrement abordée. Puis l'exemple proposé par [ROJ00] en conclusion de sa thèse, qui combine les approches ALF (*Application Level Framing*) et MPEG, pour permettre de tirer profit au mieux des possibilités offertes par la gestion de l'ordre et de la fiabilité partielle, est utilisé comme comparaison du pouvoir d'expression et de la compacité du modèle par rapport à d'autres modèles.

Une application multimédia clôturera ce paragraphe dédié aux exemples, en présentant la façon de modéliser ces contraintes de synchronisation inter flux.

4.1 Modélisation d'un flux MPEG

4.1.1 Structure de flux MPEG

La création des normes internationales pour la compression vidéo, telles que MPEG, à accéléré le développement de nouvelles applications distribuées. Le nom de cette norme, adoptée depuis maintenant près de huit ans, est directement issu de celui du groupe d'experts qui à eu en charge de l'élaborer (Moving Pictures Experts Group) sous l'égide de l'ISO (International Standard Organisation).

Cette norme MPEG 1, qui concerne aussi bien la vidéo que l'audio, a été adoptée à la fin de l'année 1992 sous la référence ISO 11172. C'est une norme de restitution et non de production, destinée uniquement à la diffusion éditoriale.

Cette brève introduction n'a pas pour objectif de détailler l'ensemble des techniques de codage définis dans la norme MPEG, mais simplement de présenter les principes d'ordonnement des données et les tolérances, en termes de qualité de service, d'un tel flux.

Ordonnement des images

Pour tirer parti des redondances, toutes les images ne sont pas traitées et compressées de la même façon. C'est ainsi qu'une séquence vidéo à la norme MPEG 1 est composée de trois types d'images (Figure III-14) :

- les images “Intra” (I)
- les images “Prédictives” (P)
- les images “Bidirectionnelles” (B).

a) Les images “Intra”

Les images “Intra” sont codées intégralement, sans aucune référence aux images voisines de la séquence vidéo. C'est la redondance spatiale qui est exploitée et éliminée à l'aide d'une opération mathématique appelée *Transformation en Cosinus Discret* ou DCT (*Discret Cosinus Transform*).

Les images “Intra” constituent les images de référence à partir desquelles est réalisé le décodage. C'est pourquoi chaque changement de plan dans une séquence vidéo commence obligatoirement par une image de type “I”. À l'intérieur de celle-ci, la norme MPEG 1 indique qu'une image sur douze doit être une image “Intra”.

b) Les images “Prédictives”

Les images “Prédictives” exploitent à la fois la redondance spatiale et la redondance temporelle des images d'une séquence vidéo. Elles sont codées à partir de l'image “I” ou “P” précédente à l'aide de vecteurs de mouvement. Les images sont découpées en blocs de 16 x 16 pixels. Les vecteurs de mouvement sont ensuite calculés en fonction du déplacement de chacun de ces blocs de pixels d'une image à la suivante puis codés en DPCM (*Differential Pulse Code Modulation*), c'est-à-dire que seule la différence entre les vecteurs de l'image n et les vecteurs de l'image n + 1 est prise en compte. Ces valeurs sont ensuite soumises au codage à longueur variable (VLC). La norme MPEG 1 fixe à deux le nombre d'images séparant deux images prédites ou une image prédite et une image “Intra”. Elle prévoit aussi trois images prédites en deux images “Intra”.

c) Les images “Bidirectionnelles”

Les images “Bidirectionnelles” sont les plus compressées. Elles sont codées à l'aide de vecteurs de mouvement avant et arrière, par prédiction bidirectionnelle, c'est-à-dire à partir des images voisines passées et futures. Au décodage, elles sont entièrement reconstruites par interpolation, c'est-à-dire que chaque bloc de pixels constituant ce type d'images prend la valeur moyenne du bloc de pixels correspondant de l'image “I” ou “P” précédente et à venir dans la séquence d'affichage, à la sortie du décodeur.

Il faut noter que le codage et le décodage d'une image “B” ne sont possibles que si les images “I” et “P” lui servant de référence sont disponibles. C'est pour cette raison que la séquence utilisée pour le codage et le décodage est modifiée par rapport à celle requise pour l'affichage: l'ordre des images est réorganisé à l'aide de mémoires de trames.

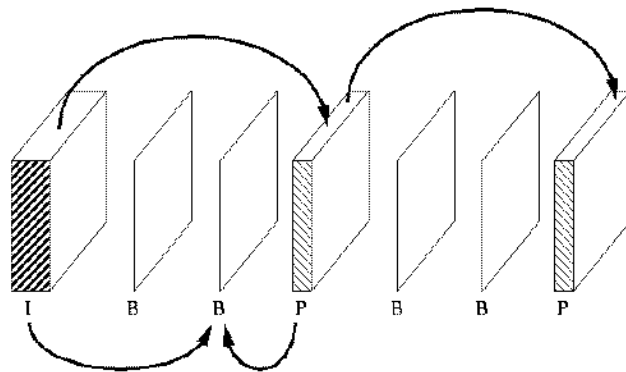


Figure III-14 - Groupe d'image MPEG (GOP)

Décompression: un exemple

La séquence présentée est la séquence la plus souvent utilisée pour le codage d'images pour les USA et le Japon (Fréquence de 60 Hz) : IB1B2P1B3B4P2BBPBBI

On constate que l'intervalle entre le codage de deux images indépendantes de leur passé et avenir (Image I) est de 12 images. La synchronisation est donc effectuée toutes les 0,4 s.

La proportion d'images codées P et B a été déterminée par l'expérience.

<i>Décodeur</i>	<i>Processeur</i>
Décode (I)	
Décode (P1)	Affiche (I)
Décode (B1)	
Décode (B2)	Affiche (B1)
Décode (P2)	Affiche (B2)
Décode (B3)	Affiche (P1)

Figure III-15 - Exemple de décompression

La figure ci-dessus (Figure III-15) montre comment le décodeur opère pour décompresser les images à partir de la séquence reçue.

La principale chose à noter est la recherche de la séquence P1 et son décodage avant le traitement des séquences B1 et B2 qui temporellement lui sont antérieures. Cette recherche est nécessaire car le calcul de l'image de type B doit s'effectuer à partir d'une image de type I ou P qui lui est postérieure.

Tolérance d'un flux MPEG

A la différence des applications informatiques classiques qui imposent des services de communication totalement fiables, les applications fondées sur le transfert de vidéo sont capables de tolérer certaines altérations introduites par le réseau, telles que les pertes, les déséquilibrés et les délais. Bien entendu, ces erreurs doivent rester limitées et contrôlées.

Dans ce cadre, le but de cette section est de présenter la nature et les limites des erreurs de transmissions qu'un flux vidéo MPEG peut tolérer.

a) Tolérances aux pertes

La perte de données peut être tolérée par une application de présentation de flux MPEG grâce aux informations de synchronisation faisant partie des entêtes de chaque donnée. Toutefois, même si les applications MPEG sont capables de tolérer des pertes, leur type doit-être contrôlé afin de maintenir une bonne qualité de présentation.

Les images composant un flux vidéo MPEG n'étant pas toutes codées de la même façon, le type de la perte est un facteur qui détermine l'ampleur de l'impact et la qualité de la présentation. En effet si la donnée endommagée ou perdue n'est pas une image de référence (en l'occurrence une image de type B), seule l'image concernée sera dégradée ou non présentée. Dans le cas d'une image de référence, une image de type I ou P, la dégradation de l'image s'étendra sur les images en amont et en aval de la perte. Par conséquent, si des pertes peuvent être tolérées pour les paquets constituant une image B, elles doivent être contrôlées pour les images de type P, voire refusées pour les images de référence I, afin de conserver une présentation acceptable. L'impact de la perte d'une donnée constituant un flux MPEG en fonction de son type est détaillée dans [ROJ00].

b) Déséquilibrés tolérés

Si les déséquilibrés de données induits par les réseaux de communications sont une réalité, la norme de video MPEG n'offre pas les moyens nécessaires à l'application de les tolérer. Les données doivent être présentées selon un ordre total, qui est celui défini dans la norme et fonction du format utilisé (IBBPBBPBBPBB dans le cas de notre exemple précédent). Un traitement déséquilibré de ces données provoquerait un résultat visuel incohérent.

D'une part des travaux montrent que certaines données, en particulier des parties d'image [HEY91] [ROJ98B], peuvent tolérer un certain traitement déséquilibré. L'idée sous-jacente proposée par les auteurs, est de segmenter chacune des images afin d'en autoriser le traitement de manière parallèle. Ainsi, une image peut-être sectionnée en plusieurs parties, appelées tranches, qui peuvent être affichées de manière indépendante par l'application et de ce fait dans l'ordre de leur réception. Les déséquilibrés tolérés se limiteront aux tranches d'une même image (nommé déséquilibré *intra-image*), et non pas aux tranches d'images consécutives, ce qui aurait pour conséquence un impact visuel important.

D'autre part, si les données doivent être présentées selon un ordre défini, les trames de référence peuvent être utilisées, même si elles ont été délivrées au delà de leur temps de présentation. Ces trames I ou P ne pourront pas être présentées, mais en revanche pourront permettre de décoder correctement les données qui en dépendent (les trames B suivantes).

Ainsi, les trames de référence I et P peuvent être acceptées pendant toute la durée d'un groupe d'images (GOP).

4.1.2 Définition d'un flux MPEG à fiabilité partielle

Cet exemple propose de définir à l'aide de la syntaxe de flux, la qualité de service requise par une application présentant un flux MPEG diffusé depuis un serveur vidéo distant. Afin de réduire les délais engendrés par l'attente de retransmissions de données perdues, nous proposons de définir le flux MPEG comme ne nécessitant pas une fiabilité totale, c'est pourquoi le protocole POC sera utilisé à la place de TCP (traditionnellement choisi pour ce type d'application).

Pour ce premier exemple, dans un but simplificateur, seule la notion de fiabilité partielle est prise en compte. Afin de conserver une qualité globale de la présentation, la perte des images de référence est contrôlée. Les images « Intra » (I), contraignant le décodage de l'ensemble du groupe d'images auquel elles appartiennent, doivent être délivrées au décodeur dans tous les cas. Le rôle moins important des images de référence (P) permet d'en tolérer la perte, mais seulement dans une certaine limite. En effet, la perte de deux images (P) consécutives génère autant de trouble dans le décodage que la perte d'une image I. Nous choisirons donc d'interdire la perte de deux images (P) consécutives appartenant au même groupe d'images. La perte d'une image de type B n'entraînant qu'un défaut de présentation de cette dernière, leur perte sera tolérée.

La démarche de spécification est présentée pas à pas : d'abord le motif constituant la trame du flux est élaboré, puis l'expression de la fiabilité est déterminée.

Définition de la structure du flux

Conformément à la spécification du flux MPEG résumée dans le paragraphe précédant, nous définissons la structure par un *motif fini*. Dans une optique simplificatrice chaque image est contenue dans une unité de données du protocole de transport (TPDU). Une vue plus complexe sera présentée dans l'exemple suivant. Toutefois la taille des images d'une vidéo MPEG de petite taille n'étant pas très importante, cette hypothèse est tout à fait justifiée. Ne prenant pas en compte les possibilités de traitement parallèle des images, l'ordonnancement des données est naturellement celui décrit dans la norme, c'est à dire la séquence présentée dans la Figure III-14. La séquence des différentes images constituant un groupe d'image (GOP) MPEG est représentée alors par le motif nommé *gop* suivant :

$$gop = I . B . B . P . B . B . P$$

I, P, B étant les étiquettes qui identifient chacune des TPDU émises par le protocole POC de l'entité émettrice.

Une vidéo MPEG étant constituée d'une succession de groupe d'images (GOP), le flux pourrait être simplement modélisé par l'itération infinie (opérateur *) du motif *gop* : $f = gop^*$. Toutefois, ce serait ne pas tenir compte de la nature partiellement fiable que nous souhaitons donner à ce flux.

Définition de la fiabilité du flux

Afin de spécifier la qualité et la quantité de pertes que l'utilisateur sera prêt à accepter dans le flux, la traduction des contraintes exprimées en langage naturel doit être faite vers le langage défini dans la syntaxe de flux.

Deux niveaux de spécification des pertes sont disponibles :

Le premier correspond à la définition qualitative des pertes, en spécifiant un ou plusieurs motifs minimaux.

Le second permet la spécification quantitative des pertes en indiquant le seuil de dégradation acceptable en fonction du nombre de données qu'il est acceptable de perdre.

Seule les images de type I nécessitant d'être impérativement délivrées, un motif minimal constitué de la seule image I du GOP est alors défini :

$$min = I$$

Toute donnée dépendant directement de la donnée portant l'étiquette I, c'est à dire le GOP entier, ne peut-être délivrée sans qu'elle n'ait été délivrée auparavant.

Le motif associant motif fini, et motif minimal sera spécifié par :

$$p = gop / \triangleright min$$

La traduction des contraintes quantitatives se fait de manière aussi intuitive :

- Tolérer la perte des images de type B correspond à accepter la perte des quatre éléments libellés par B du motif. La première expression de perte s'écrit : $\xi_1 = [\{ B \}, 1, 4]$
- Accepter la perte d'une image P sur deux est équivalent à autoriser la perte d'une seule entité libellée par P par GOP : $\xi_2 = [\{ P \}, 1, 1]$

Le flux est alors défini par l'itération du motif et par ses expressions de pertes. Il est ensuite associé à une connexion réseau en vue d'être transmis :

$$f_1 = p^* / [\{ B \}, 1, 4] / [\{ P \}, 1, 1]$$
$$video = c_1 :: f_1$$

La Figure III-16 récapitule la spécification du flux.

$gop = I . B . B . P . B . B . P$ $min = I$ $p = gop / \triangleright min$ $f_1 = p^* / [\{ B \}, 1, 4] / [\{ P \}, 1, 1]$ $video = c_1 :: f_1$

Figure III-16 - Exemple de spécification d'un flux MPEG

4.1.3 Définition d'un flux MPEG à fiabilité partielle et ordre partiel

Dans le cadre de diffusion de vidéo haute définition, l'exemple précédent peut-être optimisé afin de prendre en compte les nouvelles caractéristiques d'un tel flux. Une vidéo haute définition est principalement différenciée par deux paramètres, qui sont une définition des images plus importante, et une fréquence d'image supérieure.

La définition des images étant supérieure, l'hypothèse qu'une image puisse être émise à l'intérieur d'une seule TPDU n'est plus réaliste, en particulier pour les images de référence I, qui sont des images au sens propre du terme ayant une taille équivalente à une image de la même définition sous le format JPEG. Les images devront donc être fractionnées en plusieurs TPDU, et libellées pour une reconstruction correcte après réception. Cette approche est proposée par [HEY91] sous le nom « Application Level Framing » (ALF).

[ROJ98B] étend ce concept par la gestion d'un ordre partiel.

- D'une part, aucun ordre n'est requis parmi les tranches d'une même image afin de permettre un affichage plus rapide, sans avoir à attendre la réception de l'intégralité de l'image.
- D'autre part, les parties des images de référence sont acceptées toute la durée du GOP afin d'en permettre la délivrance pour le décodage des données qui en dépendent, même lorsque celles-ci sont en retard et ne peuvent être présentées.

Spécification à l'aide de Réseaux de Pétri

Le formalisme utilisé par [ROJ98B] pour définir l'ordre partiel de ce flux est celui des *Réseaux de Pétri à Flux Temporels* (RdPFT) proposés par [DIA93,SEN96]. Néanmoins, les auteurs ignorent volontairement la spécification du temps car le service « best effort » du réseau Internet utilisé ne permet aucune garantie temporelle. La définition de fiabilité, quant à elle, est spécifiée grâce à la sémantique des transitions. Le réseau de Pétri proposé par les auteurs est présenté dans la Figure III-17.

Définition de la fiabilité :

La sémantique des transitions utilisées dans le modèle permet la définition de la fiabilité partielle. Parmi les différents types de transition définis dans les RdPFT, deux d'entre eux sont employés dans la spécification du flux :

- les transitions classiques (ST_i) définissant une fiabilité totale,
- les transitions maîtres (MT_i) définissant une fiabilité partielle.

Lorsqu'une transition est accessible depuis plusieurs places, si l'un des arcs est défini comme étant maître, son tir entraînera automatiquement le tir de la transition, sans attendre que l'ensemble des places ait été marqué. Ainsi les places en amont de la transition, qui représentent les données délivrables à un instant donné, ne seront plus accessibles.

Les places qui étaient marquées avant le tir de la transition auront permis la délivrance de la donnée associée. Celles qui ne l'étaient pas correspondent à des données manquantes qui ne sont plus délivrables. Elles doivent, par conséquent, être déclarées perdues. Les transitions maîtres sont représentées par la bande grise de la Figure III-17, et sont nommées (MT_i). Ainsi, toutes les données de la bande grise, nommées TT (*enTèTe*) ne peuvent être perdues, à l'inverse des autres, nommées TG (*Tranche d'imaGe*).

Comme on peut le constater, la perte des tranches non reçues des images de type B est déclenchée par la réception de l'entête (TT) de l'image B suivante.

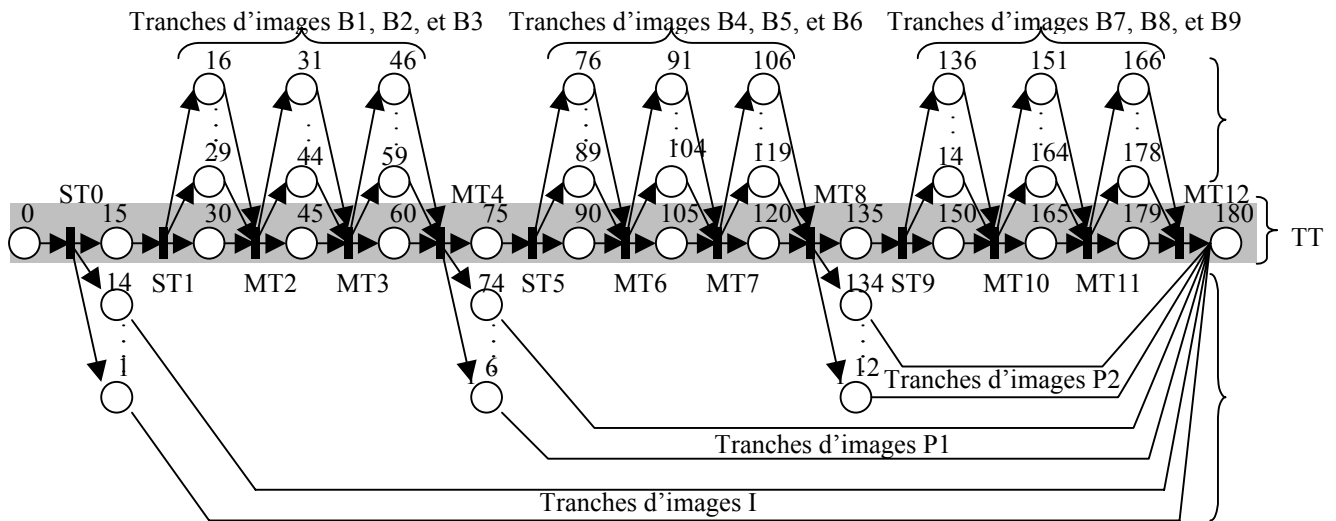


Figure III-17 - Spécification TSPN du service POC pour une séquence vidéo MPEG

Spécification à l'aide de la grammaire de flux :

Nous avons souhaité présenter cet exemple relativement complexe pour montrer le pouvoir d'expression et la simplicité de spécification qui découlent de la grammaire de flux que nous avons créée.

Plusieurs étiquettes sont définies afin d'identifier chacun des types de données :

- Les entêtes de début et fin de GOP : D, et F marquent le début et la fin d'un GOP (0,180).
- Les entêtes d'images : I₀, B₀, P₀ représentent les entêtes de chacune des images.
- Les tranches d'images : I, P, et B.

Pour simplifier l'expression du motif final, nous avons choisi les trois motifs intermédiaires suivant :

- *Img_B* qui correspond à la structure d'une image de type B, composé d'un entête (B₀) et des 14 tranches de l'images. Aucun ordre n'est spécifié entre ces éléments.
- *Trames_P*, et *Trames_I* correspondent aux tranches des images P et I. Aucun ordre n'est spécifié pour les tranches d'images.

Le motif fini (*GOP*) sans l'expression des pertes acceptables, correspondant à un groupe d'images, se définit alors comme étant la composition des différents éléments précités. Il est notable que l'expression est terminée par la composition en parallèle des tranches d'images P et I. C'est ainsi que peut être exprimé le fait que les tranches d'images de référence I et P peuvent être reçues jusqu'à la fin du GOP dans l'unique but de servir de base au décodage des images qui pourraient en dépendre.

Nous pouvons exprimer la fiabilité d'une manière toute aussi naturelle :

Puisque, d'après le modèle de [ROJ98B], seules les entêtes sont nécessaires, un motif minimal composé de toutes ces entêtes (*Min*) est défini. Les entêtes devant être délivrées en séquence, le motif est linéaire.

En ne définissant qu'un motif minimal, les données qui n'appartiennent pas au motif minimal (i.e. toutes les tranches d'images) sont considérées comme perdables. Ainsi, la fiabilité définie dans le modèle RdP par l'utilisation de transitions maîtres peut être exprimée seulement par l'emploi un motif minimal.

Le flux (*MPEG*) est alors représenté par l'itération du motif avec pertes. La Figure III-18 illustre la spécification du service POC pour un flux MPEG-1 à haute définition.

$\begin{aligned} \text{Img_B} &= \parallel_{14} B \parallel B_0 \\ \text{Trames_P} &= \parallel_{14} P \\ \text{Trames_I} &= \parallel_{14} I \\ \\ \text{GOP} &= D \cdot (I_0 \cdot \text{Img_B}^3 \cdot (P_0 \cdot \text{Img_B}^3 \cdot (P_0 \cdot \text{Img_B}^3) \parallel \text{Trames_P}) \parallel \text{Trames_P}) \parallel \text{Trames_I} \cdot F \\ \\ \text{Min} &= D \cdot I_0 \cdot B_0^3 \cdot P_0 \cdot B_0^3 \cdot P_0 \cdot B_0^3 \cdot F \\ \\ \text{SEQ} &= \text{GOP} / \triangleright \text{Min} \\ \\ \text{MPEG} &= \text{SEQ}^* \end{aligned}$

Figure III-18 - Spécification du service POC pour une séquence vidéo MPEG

Comparaison des deux modèles :

Pour conclure ce long exemple, une rapide comparaison entre le modèle RdPFT et le notre est effectuée :

- Tout d'abord, il faut noter que contrairement aux RdPFT, notre modèle de flux n'intègre pas de notion de temps.
- Le pouvoir d'expression de notre grammaire est supérieur, en tout cas pour la spécification de flux, puisqu'elle permet de spécifier des pertes quantitatives et non pas seulement qualitatives.
- Pour notre modèle, il n'est pas nécessaire de faire une validation de la spécification, car le respect de la grammaire de flux dans une spécification garantit une modélisation valide du flux.
- La syntaxe de notre modèle est compacte et peut s'intégrer très facilement dans un langage de programmation. Ce point sera d'ailleurs présenté dans le chapitre traitant de l'implémentation.
- Les automates de reconnaissance des flux peuvent être automatiquement générés.

4.2 Visioconférence MPEG

Pour conclure cette série d'exemples, et présenter les mécanismes de synchronisation, prenons la spécification des contraintes de qualité de service d'une application de visioconférence. Cette application sera composée d'un flux MPEG, identique à celui décrit dans le premier exemple (4.1.2), et d'un flux audio stéréo PCM. Un des principaux critères de qualité d'une telle application étant la qualité de la synchronisation entre le flux audio et le flux vidéo, nous cherchons à garantir la « lip synchronisation ». Même avec une qualité d'image et de son correcte, il est très difficile de suivre une conférence dont le son est très décalé.

Pour illustrer la méthode de spécification de flux synchronisés, un flux audio stéréo PCM va d'abord être défini, puis ses contraintes de synchronisation avec le flux vidéo vont être exprimées.

Le type le plus courant d'enregistrement audio numérique est appelé modulation par impulsions codées (pulse code modulation, PCM). C'est la technique utilisée en téléphonie, par les disques compacts et la plupart des fichiers WAV. Dans un système d'enregistrement PCM, un microphone convertit les variations de pression de l'air (ondes sonores) en variations de voltage. Ensuite, un convertisseur analogique-numérique échantillonne le voltage à intervalles de temps réguliers. Le flux numérique d'informations ainsi obtenu est fragmenté en paquets de petite taille afin d'être émis dans une connexion réseau. Pour un nombre de 20 paquets par seconde un flux audio mono peut s'exprimer ainsi :

$$\begin{aligned} PCM_seq_droit &= d^t \\ PCM_seq_gauche &= g^t \end{aligned}$$

où d représente l'étiquette associée aux données audio du canal droit, et PCM_seq_droit représente une séquence audio de $1/5^{\text{ème}}$ de seconde (200 ms). Le canal gauche, PCM_seq_gauche est défini de la même manière.

Le flux stéréo est composé de deux séquences audio mono qui doivent être fortement synchronisées. Une séquence stéréo peut alors être définie par la composition parallèle des deux séquences audio (canal droit et gauche). Ceci aura pour conséquence d'introduire un point de synchronisation à chaque fin de motif, c'est à dire toutes les 200 ms.

$$PCM_seq_stéréo = PCM_seq_droit \parallel PCM_seq_gauche$$

Notre flux stéréo, dont les deux canaux seront synchronisés avec une granularité de 200 ms, sera alors :

$$audio_stéréo = PCM_seq_stereo^*$$

En introduisant la notion de fiabilité partielle, où une tolérance de 25 % de perte est souhaitée, l'expression du flux stéréo associé à une connexion réseau sera :

$$flux_stéréo = c_2 :: (audio_stéréo / [\{ d \}, 1, 2] / [\{ g \}, 1, 2])$$

où c_2 représente l'identifiant de la connexion utilisée pour la transmission des données audio (typiquement un numéro de port UDP). L'expression de perte autorise la perte de deux éléments par motif pour chaque canal.

Une expression de perte globale tel que $[\{ d, g \}, 1, 4]$ ne peut pas remplacer l'utilisation des deux expressions de pertes, car dans ce cas, les pertes ne pourraient avoir lieu que sur un seul flux, ce qui dégraderait fortement sa présentation.

Si la synchronisation entre les deux flux audio doit être forte, la synchronisation entre le flux vidéo et le flux audio stéréo peut être plus lâche. Relaxer les contraintes de synchronisation permet de diminuer les pertes de temps induites par l'attente de données en retard ou perdues [OWE96].

Dans le cas de la synchronisation audio / vidéo, une resynchronisation de l'audio avec la vidéo toutes les 400 milli-secondes est largement suffisante pour garantir une présentation correcte.

La durée d'un échantillon sonore étant de 200 ms et celle d'un GOP pour le format MPEG de 400 ms, deux échantillons sonores doivent être joués pendant la durée de présentation d'un GOP. En plaçant un point de synchronisation ξ à la fin de chaque GOP, et toutes les deux séquences audio, on obtient la synchronisation requise :

$$\xi = [(c_1, 1) < > (c_2, 2)]$$

où c_1 et c_2 correspondent aux canaux d'allocation des différents flux précédemment définis.

Le flux final se définit alors par la composition des deux médias, et de leur contrainte de synchronisation :

$$audio_video = (audio \parallel video) [\xi]$$

5 Conclusion

Ce chapitre a présenté un état de l'art concernant l'expression de la qualité de service pour les protocoles à ordre et fiabilité partiels. L'accent a été mis plus particulièrement sur les formalismes utilisés lors de précédentes implantations de ce type de protocole [CHA95] [FOU97] [ROJ00]. Les points forts et les faiblesses des différents formalismes pour la définition de la QoS ont été mis en évidence au travers d'exemples simples.

La contribution apportée par le modèle que nous avons créé est de deux ordres.

Tout d'abord un langage pour la modélisation de flux multimédia associant ordre et fiabilité est proposé. Même si la spécification d'ordre partiel par des langages série / parallèle [GRA81] n'est pas une nouveauté, ce formalisme est *le premier* à associer d'une manière duale définition de l'ordre et spécification des pertes. Notre modèle est plus expressif que ses prédécesseurs quant à la spécification des pertes. En effet, à une définition *qualitative* des pertes, comme cela est fait à l'aide des RdPFT, est associée à une définition *quantitative* des pertes comme cela est proposé dans MM-POC. Le langage de spécification de ce nouveau modèle possède aussi la propriété d'être très compact. La spécification d'un flux, aussi complexe qu'il soit, tient sur quelques lignes. Le fait d'utiliser un langage, plutôt qu'une méthode graphique, rend la méthode de spécification compatible avec les langages utilisés pour la programmation des applications multimédias distribuées. La distance entre un langage de programmation et le langage de spécification n'est pas très grande. Nous avons élaboré une interface de programmation orientée objet déduite du langage de spécification. Elle sera décrite dans le chapitre concernant l'implémentation.

D'autre part, un modèle d'automates permettant la reconnaissance de ces flux est présenté. Les algorithmes, ainsi que les structures de données y sont détaillés. De ce fait, ces automates peuvent être générés automatiquement à partir d'une spécification. Leur intégration est alors facilitée pour la construction de nouveaux protocoles de communication à ordre et fiabilité partiels.

Le langage de spécification joue un rôle d'interface entre l'application et le protocole de communication. Les automates, quand à eux, constituent le cœur de ce protocole.

La synchronisation multimédia est un élément clé de l'architecture de communication MMPOC-MN. Nous avons vu, dans le chapitre précédant, qu'il était possible de profiter de l'hétérogénéité des réseaux, en tout cas dans une configuration de réseaux parallèles.

Le cas de multi-réseaux série pose de nouveaux problèmes, et nécessite une évolution de l'architecture. Le chapitre suivant expose l'architecture de communication pour le multi-réseaux série que nous proposons.

Chapitre IV

Multi-réseaux Série

1 Introduction

D'une part, l'émergence de nouvelles technologies de communication, de nouveaux opérateurs, de nouveaux besoins de communication, font que de plus en plus de réseaux naissent, chacun offrant une qualité de service spécifique à sa vocation (réseaux mobiles, réseaux hauts débits, ...).

D'autre part, pour assurer le besoin de « connectivité permanente » et offrir un moyen à chaque utilisateur d'accéder depuis n'importe où à ses données, les réseaux s'interconnectent de plus en plus.

Il en résulte que l'homogénéité de technologies et de services qui existait dans les réseaux d'ancienne génération n'est plus un état de fait. Il n'est pas rare que, pour mettre en relation deux utilisateurs, plusieurs types de réseaux soient traversés, chacun offrant une qualité de service particulière. Les plus grandes disparités se trouvent souvent à la rencontre des réseaux d'accès et des réseaux d'interconnexion.

Le protocole de transport TCP [RFC793], conçu depuis près de 20 ans, a été, et continue à être, largement utilisé par un très grand nombre d'applications Internet et Intranet. Toutefois, dans certains environnements, le « bon » fonctionnement de TCP est limité par les caractéristiques des liens qu'il emprunte : mauvaise utilisation de la bande passante sur des liens haut débit, reprise d'erreur que l'on peut qualifier pour le moins de maladroite sur des liens sans fil, etc. Que faire ?

Il peut être envisagé de chercher à améliorer le service de liaison de données afin d'offrir un environnement de meilleure qualité aux protocoles de niveau supérieur, mais ceci n'est pas toujours faisable à moindre coût : certaines caractéristiques sont parfois difficiles – voire impossibles - à compenser, tel le délai de propagation d'une liaison satellite.

L'alternative consiste alors à travailler au niveau transport, en modifiant cette couche pour l'adapter aux caractéristiques de la couche liaison empruntée, comme cela est fait pour l'adaptation de TCP aux liaisons satellites ou aux liaisons mobiles. Nous avons décrit dans le chapitre 2 ces modifications à apporter aux protocoles de transport.

Toutefois, que faire lorsque seule une partie de la liaison pose problème ? Il n'est souvent pas possible d'utiliser la version du protocole modifié sur toute la liaison, car les modifications apportées perturbent la plupart du temps le comportement du protocole dans son environnement d'origine (i.e. les réseaux terrestres).

Prenons le cas d'une connexion TCP empruntant un réseau terrestre puis un réseau d'accès par satellite, l'utilisation d'un contrôle de flux spécialisé pour les réseaux par satellite risquerait de provoquer des congestions sur le réseau terrestre à cause d'une trop grande taille de fenêtre. D'autre part, du fait de l'agressivité supérieure des mécanismes de contrôle de congestion implantés au sein des souches TCP satellite, le comportement de telles connexions face à une congestion n'est plus équitable avec celui des connexions TCP classiques. Elles seraient favorisées lors des phases de slow-start, ce qui est contraire au principe d'équité des flux TCP.

Ces problèmes posés par la traversée de réseau à qualité de service très distincte constituent la *problématique du multi-réseaux série*.

Ce chapitre a pour objectif de décrire notre solution qui améliore le comportement des protocoles de transport dans ces configurations particulières, tout en préservant les principes fondamentaux de l'Internet.

Un état de l'art est d'abord présenté. L'architecture du protocole est ensuite décrite. Diverses solutions de déploiement seront discutées. Enfin, une série de mesures viendront conclure ce chapitre.

2 Exemples de solutions multi-réseaux série

Contrairement à ce que l'on a pu observer pour le multi-réseaux parallèle, ce que nous appelons multi-réseaux série a soulevé de nombreuses interrogations dans la communauté scientifique, et de nombreux travaux ont été développés, sans toutefois afficher une très grande coordination dans les activités de recherche et de développement. De ce fait, de nombreuses solutions spécifiques sont apparues dans le domaine des caches et des proxys. Depuis peu a été créé un groupe IETF étudiant ce domaine. Des propositions de rupture de connexions des protocoles de transport, et en particulier de TCP, ont été recommandées par ce groupe allant dans le même sens que nos propositions de protocole multi-réseaux série. Des extensions de ce type d'architectures ont été proposées afin de conserver la sémantique de bout en bout des protocoles de transport, comme cela sera présenté au travers de *snoop-protocol*. Ces deux solutions composent les deux principales classes de solutions, le *splitting* et le *spoofing* autour desquels de nombreuses variantes et implémentations sont faites.

Dans un autre registre, des solutions permettant de modifier l'architecture de l'Internet afin d'éviter les configurations multi-réseaux série sont présentées. Nous citerons les cas des mécanismes de caches ainsi que celui des Réseaux de Transport de Contenu (CDN Content Delivery Network) qui sont une autre réponse à cette problématique multi-réseaux.

2.1 Le *splitting*

La première référence à citer est issue du groupe de travail « *Performance Implications of Link Characteristics* » (PILC) de l'IETF. Il s'agit du draft « *Performance Enhancing Proxies* » (PEP) datant de juin 1999 [BOR99] et qui préconise l'utilisation de proxys. Ces derniers ont pour but d'optimiser les performances des protocoles Internet dans des environnements multi-réseaux. Le principe exposé est d'adapter le fonctionnement du protocole considéré uniquement sur la liaison posant problème. Plus précisément, il consiste à scinder la connexion TCP en plusieurs tronçons, en intercalant un proxy à chaque raccord, de manière à pouvoir utiliser une version modifiée du protocole ou un autre protocole, sur le tronçon problématique. Ainsi, certaines fonctionnalités, comme le contrôle d'erreur ou le maintien de la connexion, peuvent être gérées au cas par cas entre certains nœuds du réseau. Le draft propose des exemples de proxys, que nous reprenons ici brièvement.

Un proxy, par définition est un serveur recevant des requêtes qui ne lui sont pas destinées et qui les transmet aux autres serveurs. Dans le cas particulier adressé par PEP, il s'agit plus particulièrement d'un équipement de réseau interceptant les demandes d'ouverture et de fermeture de connexions TCP, afin d'établir une multi-connexion TCP. Tous les paquets sortant d'une connexion TCP sont redirigés vers l'entrée de l'autre connexion. Pour ce faire, cet équipement possède les couches 3 et 4 du modèle Internet. La Figure IV-1 présente l'architecture du proxy ainsi que le mécanisme de **TCP-Splitting**.

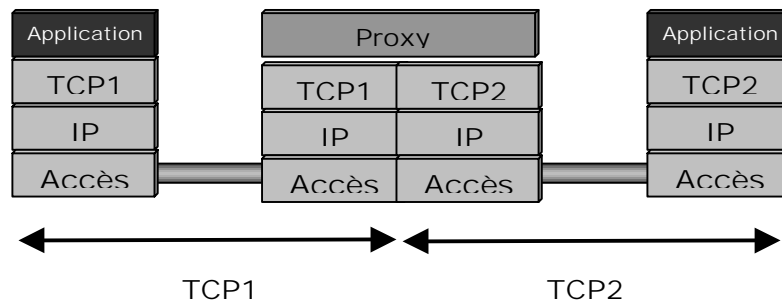


Figure IV-1 - TCP splitting

Cette technique, nommée rupture de connexion, est généralement utilisée pour modifier une non-adéquation entre deux systèmes d'extrémités, lorsqu'il n'est pas possible de modifier efficacement et de manière sûre ces derniers.

L'une des principales motivations pour l'utilisation de proxys réside dans l'auto adaptation des mécanismes de contrôle de flux et de congestion de TCP induits par son utilisation dans un environnement de réseaux homogènes. Les acquittements locaux à chaque connexion permettent, par exemple, d'accélérer l'ouverture de fenêtre de transmission de l'émetteur et contribuent à améliorer le débit. C'est un système très facile à mettre en œuvre, offrant une solution d'adaptation des protocoles à moindre coût de développement. Les proxys se trouvent généralement en bordure de réseau, à l'intersection entre les réseaux que l'on veut gérer séparément.

Un des effets de bord de cette architecture est présenté dans le draft [BOR99] étudiant les mécanismes de retransmissions locales. Grâce à l'utilisation du proxy le contrôle d'erreur est effectué pour chacune des connexions. Ceci a pour conséquence une détection et une récupération d'erreur beaucoup plus réactive. Toutes les données d'une connexion découpée sont interceptées et réémises vers l'autre connexion. Elles sont donc mémorisées au niveau du buffer d'émission de la connexion sortante.

Le proxy joue alors le rôle de point de retransmission des données perdues sur la liaison proxy-récepteur, ce afin de limiter les délais de retransmission.

En sus du contrôle de congestion et du contrôle d'erreurs, d'autres fonctionnalités peuvent être prises en charge : il s'agit, par exemple de la compression de données permettant une meilleure utilisation de la bande passante pour les liens mobiles.

Afin de supporter les environnements de communication mobiles, une implémentation compatible avec ce modèle a été réalisée par le protocole I-TCP [BAR94]. Une souche optimisée pour les communications mobiles est utilisée pour la liaison proxy - équipement mobile.

L'un des principaux freins au déploiement massif d'une telle architecture est la perte de la sémantique de bout en bout de la connexion transport. En effet, même si l'utilisation de deux connexions totalement fiables et ordonnées garantit une intégrité totale des données en fonctionnement normal de la communication, rien ne garantit le comportement en cas de congestion du proxy. La sémantique de TCP garantit que chaque donnée acquittée a été reçue et délivrée au récepteur. Elle peut alors être détruite côté émetteur. Si une congestion a lieu sur le proxy, au niveau de la deuxième connexion, il se peut que des données aient été acquittées sur la première connexion et n'aient jamais été transmises sur la seconde. De ce fait, une couche supérieure garantissant la sémantique de bout en bout est nécessaire au dessus de telles architectures. Toutefois, ceci ne peut se faire sans modifications de

l'application. Ainsi, le déploiement n'est possible qu'à condition de modifier les programmes existants.

2.2 Le spoofing

Une extension a été proposée par [BAL95] avec le *snoop-protocol* qui, pour la même architecture, conserve la sémantique de bout-en-bout en conservant une seule connexion TCP. L'idée est de garder la connexion TCP et d'intercepter les données et les acquittements au niveau d'une passerelle. Des acquittements sont générés depuis la passerelle afin de permettre un contrôle de flux adéquat, et pour conserver l'intégrité du flux, au cas où une donnée acquittée serait perdue ensuite, un mécanisme de cache et de retransmission est mis en place. La Figure IV-2 présente la technique d'acquittement, et la compare par rapport à TCP. Si une donnée est perdue avant la passerelle, un acquittement négatif est émis par la passerelle permettant une retransmission au plus tôt. Au contraire, si une donnée acquittée par la passerelle est perdue sur la deuxième partie de la connexion elle sera renvoyée par cette dernière. La connexion TCP, qui est conservée, préserve alors la sémantique de bout en bout. La Figure IV-3 détaille cette configuration appelée **TCP-Spoofing**.

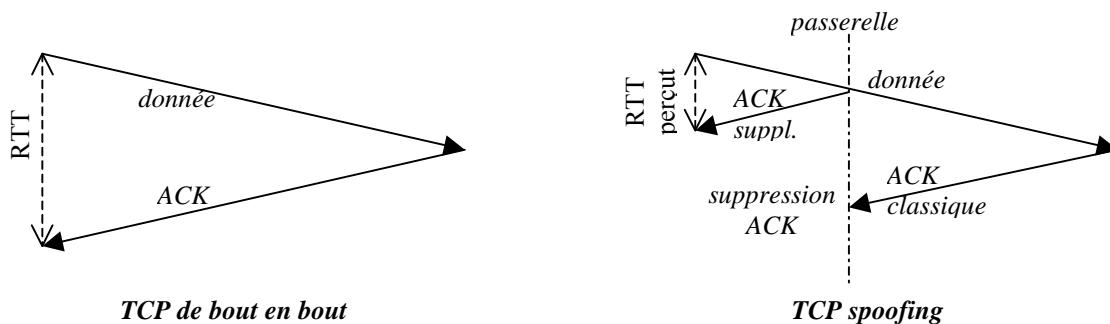


Figure IV-2 - Mécanisme d'acquittement de TCP-spoofing

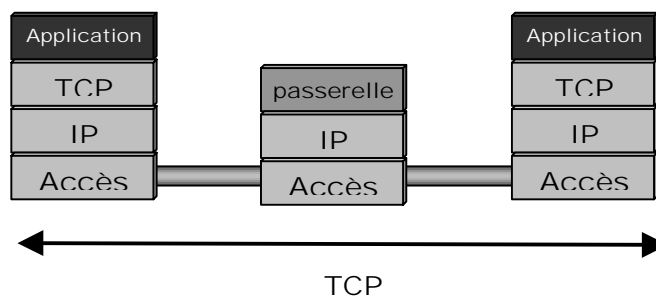


Figure IV-3 - TCP Spoofing

Le principal frein à cette architecture est le coût de développement d'une telle passerelle, qui est beaucoup plus lourd qu'une simple solution de *splitting*. En effet, des mécanismes d'acquittements et de caches, ainsi que leur régulation, doivent être mis en place.

D'autres mécanismes peuvent avantageusement être rajoutés à une telle plate-forme, comme par exemple des systèmes de lissage et de filtrage des flots d'acquittements TCP, qui permettraient une meilleure évolution du contrôle de flux.

2.3 WAP 2.0

Le protocole WAP (*Wireless Application Protocol*) offre l'accès à l'Internet à partir des terminaux mobiles. Ce protocole désigne l'ensemble des spécifications techniques issues du WAP Forum, une alliance créée en 1997 regroupant les principaux acteurs du domaine des communications sans fil. Les objectifs de cette alliance sont de définir des protocoles de communication qui permettent de développer des applications et des services opérant sur des réseaux de communications sans fil. Ce protocole tient compte des fortes contraintes liées à la nature des terminaux : mémoire de faible capacité, écrans de petites tailles, faible taux de transfert, etc.

Les apports de la version WAP 2.0 [WAP] récemment finalisée, et notamment son adaptation au haut débit, viennent d'être publiés par le WAP Forum. Cette nouvelle version se destine désormais aussi aux transmissions de type Internet. Les évolutions du protocole concernent essentiellement la réception de flux, l'introduction de nouveaux langages dédiés à Internet de types XML, et surtout l'introduction des protocoles Internet dans l'environnement WAP. Cette compatibilité a été motivée par l'émergence des réseaux mobiles hauts débits (2.5G et 3G) qui offrent directement un support IP aux mobiles, et de ce fait rend totalement transparent l'accès au réseau Internet. La pile de protocoles précédente (WAP 1.0) qui n'intégrait pas de support IP a toutefois été conservée pour la compatibilité. Cette pile est représentée à gauche sur la Figure IV-4.

L'intégration de la pile TCP/IP dans la pile WAP a été rendue possible par la définition de trois nouveaux protocoles ainsi que l'extension d'une architecture déjà chère au WAP : les proxys.

La Figure IV-4 détaille cette architecture dans le cadre de l'accès à un serveur web depuis un portable. La *passerelle WAP* chargée précédemment de la conversion du contenu et des protocoles Internet en un format compatible avec celui du WAP, qui jouait le rôle de point d'accès au réseau Internet pour l'opérateur du réseau mobile, est étendue par une pile TCP/IP. Il est remarquable de noter que cette passerelle possède deux piles TCP (une classique et une optimisée) pour les liaisons sans fil (TCP*). Ceci explique pourquoi cette passerelle porte maintenant le nom de « proxy ». Elle est en effet utilisée pour rompre les connexions, afin de les adapter au type de réseau traversé. Cette architecture est une implantation opérationnelle des techniques de découpage de connexions.

Les nouveaux protocoles ajoutés à la pile sont :

WP-TCP (Wireless Profiled TCP) (TCP*) qui est une souche de TCP optimisée pour les communications sans fil. Les mécanismes apportés pour l'amélioration des performances de TCP résultent des recommandations prônées par le groupe PILC de l'IETF précédemment cité sur l'implémentation de TCP pour les réseaux « longs et fins ».

TLS (Transport Layer Security) fournissant des mécanismes de sécurisation des connexions (cryptage, certificats, etc ...), et surtout un mécanisme de gestion de session qui a pour rôle de garantir la sémantique de bout en bout pour la couche supérieure (WP-HTTP).

WP-HTTP (Wireless Profiled HTTP) qui est une implantation du protocole HTTP pour les environnements sans fil compatible avec la version 1.1 de HTTP. Cette version supporte la compression des messages de réponses et l'établissement de tunnels sécurisés.

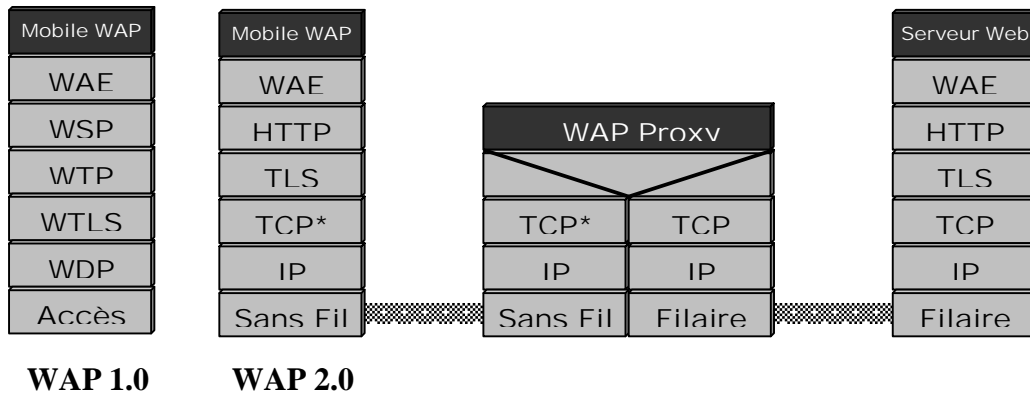


Figure IV-4 - Proxy WAP possédant une pile TCP optimisée

2.4 Mécanismes de caches

Hormis les principes d'optimisation des protocoles de transport, comme cela a été annoncé dans l'introduction, il existe des fonctions bien connues de l'Internet qui proposent des solutions au problème de ce que nous appelons multi-réseaux. C'est en particulier le cas pour les mécanismes de caches, qui contribuent à l'amélioration des performances de l'Internet. Le *Web caching* [MIG01] consiste d'une part à télécharger des pages web pendant la nuit (quand la bande passante est maximale) en fonction de prévisions statistiques, et d'autre part à stocker des pages qui sont fréquemment demandées par les utilisateurs. Il a pour principal intérêt de contourner une configuration multi-réseaux en déplaçant l'information aussi près que possible de l'utilisateur. Les serveurs de caches se trouvent sur le même réseau que l'utilisateur (i.e. le réseau du fournisseur de service). Ce principe de cache remplit parfaitement son rôle tant que les pages web ne sont pas souvent réactualisées.

Néanmoins, l'idée sous-jacente aux approches de caches consiste à changer l'architecture du Web. Ces caches participent à la réduction du trafic hors du réseau et évitent la traversée des points d'interconnexion de réseaux (*peering point*) qui sont généralement des goulots d'étranglement du réseau global. En d'autres termes, cette approche intéressante résout les problèmes multi-réseaux en supprimant les connexions multi-réseaux !

Bien sûr, cette approche ne fonctionne que pour un trafic asynchrone et ne peut pas prendre en compte les trafics temps réels. Toutefois, les réseaux d'acheminement de contenu (*Content Delivery Network*) poursuivent un objectif similaire, en déplaçant (en répliquant plus exactement) les serveurs web des extrémités vers le cœur du réseau. Ceci permet d'éviter les goulots d'étranglement liés aux capacités limitées des réseaux d'accès. Le concept de CDN est né avec des sociétés comme *Akamai* ou *Digital Island*. Elles ont construit par-dessus Internet un réseau de serveurs de cache près des points de présence des fournisseurs d'accès. On rapproche le contenu de l'utilisateur pour accélérer sa livraison. Cette technique permet la livraison des données asynchrone comme synchrone. Le succès d'*Akamai* s'est fait sur la diffusion vidéo sur l'Internet.

Ces types de techniques ont clairement un gros impact sur la réduction de trafic dans le réseau et sont en train de changer l'architecture classique de l'Internet.

3 MNP : Protocole multi-réseaux série

3.1 Justification de l'architecture

Si l'Internet est aujourd'hui déployé sur des liens très variés, c'est toujours de manière transparente pour les couches protocolaires supérieures. Les communications de bout en bout utilisent généralement les protocoles TCP et UDP quels que soient les liens sous-jacents utilisés. Mais, comme cela a été présenté précédemment, cette approche est très restrictive en terme de performances. Néanmoins, le principal avantage d'utiliser TCP pour la programmation d'applications distribuées réside dans la possibilité offerte au programmeur de travailler sans la connaissance des réseaux qui seront utilisés.

Pour améliorer les performances des protocoles de transport, tout en conservant cet avantage, il faut fractionner la connexion de bout en bout en plusieurs connexions bout à bout. L'objectif est l'adaptation du protocole de transport utilisé sur chacune des portions aux caractéristiques du réseau. Cette idée novatrice, proposée dans PEP [BOR01], était jusqu'à aujourd'hui réservée aux utilisateurs de TCP.

Basée sur cette idée, l'architecture que nous proposons dans cette section se nomme **MNP**, pour « **MultiNetwork Protocol** ». Ce protocole peut être vu comme un protocole de communication dédié aux transmissions multimédias qui intègre des mécanismes permettant l'optimisation des communications dans les configurations multi-réseaux parallèle et série. Tout au long d'une connexion MNP, la couche transport est rompue lorsque cela est nécessaire (i.e. lors des changements de réseaux). Sur chacune des parties, le protocole MMPOC-MN précédemment décrit est utilisé comme protocole de transport. La sémantique de bout en bout de la couche transport est préservée par les fonctionnalités de MNP, comme cela sera détaillé plus loin.

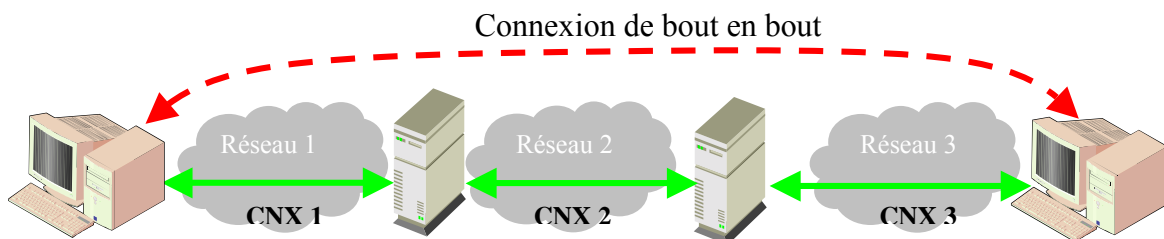


Figure IV-5 - Configuration multi-réseaux série

Considérons la configuration réseau de la Figure IV-5, où pour relier deux utilisateurs une connexion de niveau transport doit traverser trois domaines dont les qualités de services sont très différentes. Typiquement, ce genre de configuration est engendré par l'interconnexion de réseaux de natures très différentes.

Utiliser une connexion de bout en bout dans une telle configuration n'est pas judicieux, car si une donnée est perdue sur la dernière partie de la liaison, maillon pouvant être le plus faible de la connexion quand il s'agit d'un réseau d'accès, la retransmission de la donnée devra être faite depuis l'émetteur, c'est à dire retraverser tout le réseau.

- Comme solution alternative à l'utilisation d'une connexion de bout en bout, plusieurs mécanismes, connus sous le nom de PEP, permettent d'offrir à l'utilisateur final une optimisation des performances du réseau.

Parmi les deux façons de réaliser les fonctionnalités de ces équipements, qui sont le *splitting* et le *spoofing*, nous avons opté pour la première. En effet, les solutions basées sur la conservation de la sémantique de bout en bout du protocole, comme le *spoofing*, n'offrent pas tous les avantages des cascades de connexions, et en particulier leur généricité et leur relative simplicité de mise en œuvre.

3.2 Architecture

Selon la technique du *splitting* adaptée à notre architecture, l'ouverture d'une connexion MNP sur un chemin de données consiste en l'établissement d'une cascade de connexion MMPOC-MN sur chacun des domaines traversés par ce chemin. Ainsi, chaque connexion MMPOC-MN sera établie sur un domaine dont la QoS est, par définition, homogène.

Ces considérations conduisent à la définition de l'architecture du protocole MNP présenté à la Figure IV-6. L'élément central de cette architecture est le proxy MNP, qui a en charge de relayer des données provenant des connexions MMPOC-MN et de garantir la cohérence des connexions entre elles.

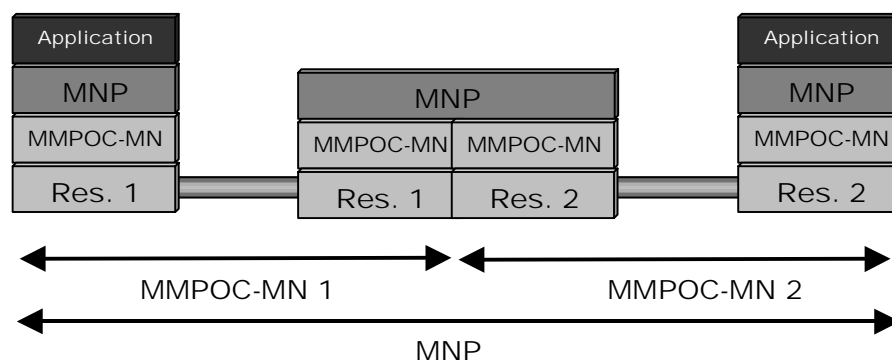


Figure IV-6 - Architecture du protocole MNP

L'architecture proposée pour le protocole MNP peut être positionnée par rapport à une architecture en couche classique ainsi : elle possède des fonctionnalités de la couche transport offertes par les connexions MMPOC-MN, ainsi que des fonctionnalités de la couche applicative. La sémantique de bout en bout du modèle TCP/IP est rompue et de nouvelles fonctionnalités sont offertes aux bordures des domaines (les domaines « réseau » ayant la même signification que dans DiffServ), de manière à relayer les paquets d'un domaine à l'autre jusqu'au récepteur final. Le proxy MNP est implanté comme équipement spécial qui inclut un routeur IP classique, augmenté par une couche Transport étendue.

Cette architecture a deux conséquences directes :

- Une optimisation globale de la connexion MNP par l'optimisation locale de chaque connexion MMPOC-MN.
- Une réduction des délais de recouvrement d'erreur du fait des retransmissions locales effectuées sur les entités émettrices des connexions MMPOC-MN.

Les principaux problèmes devant être résolus par le protocole multi-réseaux série sont alors de deux ordres :

- Optimiser les performances du protocole sur les liaisons homogènes (problématique MMPOC-MN)
- Rétablir la sémantique de bout en bout (problématique MNP).

Tous ces points sont abordés plus en détail au fil de la description de chacune des couches. MNP étant la couche la plus proche de l'utilisateur, ses propriétés et son fonctionnement sont tout d'abord décrits. Les problèmes de recouvrement de la sémantique y sont adressés. Les configurations des connexions MMPOC-MN sont ensuite détaillées, répondant aux interrogations portant sur l'optimisation des connexions.

3.3 Couche MNP

L'objectif de la couche MNP est double. Le premier rôle assuré par cette couche est le rétablissement de la sémantique de bout en bout perdue par l'utilisation des mécanismes de proxys. Au dessus de cette couche, l'utilisation de passerelle est totalement transparent à l'utilisateur qui n'observe que l'amélioration de la liaison. L'autre rôle de cette couche est la détermination des critères de qualité de service à fixer à chacune des connexions MMPOC-MN en fonction de ceux fournis par l'utilisateur.

3.3.1 La sémantique de bout en bout

L'introduction d'une rupture de connexion induit une rupture de la sémantique de bout en bout du protocole de transport, inconvénient toutefois compensé par les nettes améliorations du protocole.

Considérons d'abord les conséquences de cette rupture. Que ce passerait-il, dans la configuration donnée à la Figure IV-5 si une donnée est perdue sur la dernière connexion (*cnx3*) ? Le message ayant déjà été acquitté par les nœuds précédents, la suppression du message dans les buffers de l'émetteur a pu déjà être effectuée. Pour que la retransmission puisse avoir lieu de manière transparente pour l'émetteur, le dernier nœud traversé par le message doit en avoir gardé une copie. Il est donc naturel qu'un mécanisme de proxy couplé à celui de cache soit déployé afin de garantir la sémantique de bout en bout de la connexion découpée.

Mais ceci n'est pas suffisant pour garantir la sémantique de bout en bout. Considérons maintenant le cas où la connexion (*cnx3*) soit fermée à cause de problèmes réseaux survenus sur la dernière partie de la liaison. Deux solutions s'imposent :

Soit les connexions en amont doivent être fermées afin d'avertir l'émetteur que la connexion n'est plus valable.

Soit un mécanisme doit être employé pour essayer de rétablir la connexion défailante et retransmettre les données qui étaient en attente dans le buffer d'émission.

Ces mécanismes sont assurés par la couche MNP. La sémantique de bout en bout, au moins en ce qui concerne la fiabilité de bout en bout, est conservée dans les deux cas :

- Le recouvrement d'erreur sur les portions de connexion est effectué par les connexions MMPOC-MN. Si une donnée est perdue sur l'une de ces connexions, l'entité réceptrice du protocole MMPOC-MN détectera la perte, et s'il y a rupture de qualité de service, (i.e. si l'ordre ou la fiabilité du flux spécifié ne sont pas respectés) la retransmission sera demandée.

- La fermeture de l'ensemble des connexions en cas de défaillance de l'une d'elles est quant à elle assurée par la couche MNP. Lorsqu'une fermeture imprévue de connexion est détectée, par exemple sur le proxy, l'entité MNP responsable de la connexion défaillante initie la fermeture unilatérale de la connexion (amont ou aval) encore active. Cette fermeture aura un effet de propagation en cascade, puisque les entités MNP adjacentes détecteront à leur tour la fermeture de connexion. A la fin de ce processus les utilisateurs sont prévenus de la fermeture de la connexion MNP.

D'autres conséquences induites par l'utilisation de proxy seront présentées à la fin de ce chapitre, mais ne sont pas couvertes par le protocole.

3.3.2 Technique de relais

Le transfert des données reçues sur la connexion entrante vers la connexion sortante est assuré par la couche MNP. Ce relais d'une connexion vers l'autre soulève un problème qu'il est intéressant de présenter.

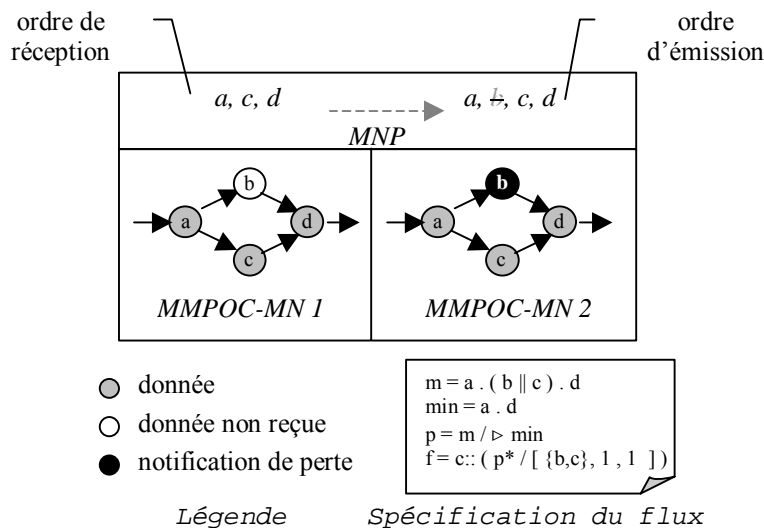


Figure IV-7 – Relayage des données par la couche MNP

La couche MNP a pour rôle d'extraire les données délivrées par la première connexion (*MMPOC-MN 1*) et de les réinjecter dans la suivante (*MMPOC-MN 2*). Il faut noter que ces données ne sont délivrées à la couche MNP que si l'ordre et la fiabilité définis dans la spécification du flux sont respectés.

Prenons l'exemple de la Figure IV-7 qui présente le comportement d'un proxy MNP pour un flux partiellement ordonné et partiellement fiable. La spécification de QoS (indiquée sur la figure) est la même pour les deux connexions.

Supposons que sur la première connexion, au cours de la transmission du motif *abcd* composant le flux, la donnée étiquetée par *b* ait été perdue. Puisque la perte d'un élément *b* ou *c* est autorisée par la spécification de QoS, le motif délivré sera *a, c, d*. Si ce lot de données était émis directement sur la seconde connexion, il en résulterait un problème d'étiquetage, puisque l'élément *b* a disparu. Le comportement de la souche émettrice du protocole MMPOC-MN serait perturbé car les mécanismes d'émission sont basés sur l'émission d'un flux fiable, où la perte d'un élément n'est pas envisagé.

Heureusement, grâce au principe de notification des pertes du protocole MMPOC-MN, la réception de la donnée d provoquera la délivrance d'un message d'information à la couche MNP indiquant que la donnée étiquetée par b est considérée comme perdue. Ainsi, un message particulier (une donnée vide) peut être émis sur la connexion sortante, à la place de la donnée manquante, signalant la perte de cette donnée. Le mécanisme d'étiquetage du protocole ne sera alors pas perturbé.

Ce principe de notification des pertes permet de conserver la même numérotation des données sur les deux connexions ce qui simplifie la gestion de la QoS.

3.3.3 Critères de QoS

Le découpage de la connexion transport pose un nouveau problème quant à la fourniture de cette qualité de service. Puisque la connexion MNP est découpée en sous connexions MMPOC-MN, quelle qualité de service doit être fournie par chacune des sous-connexions ?

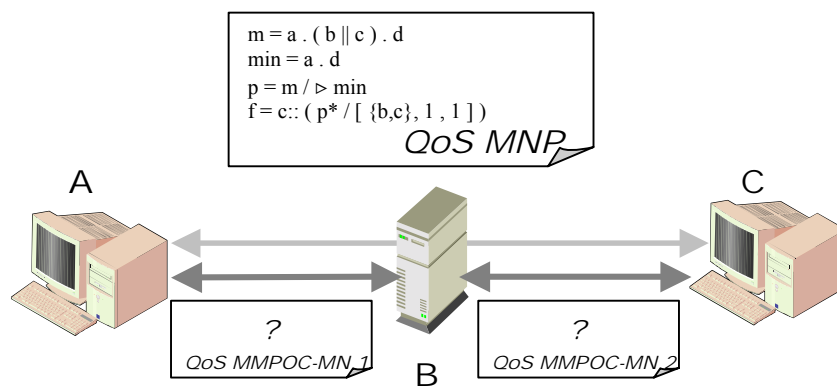


Figure IV-8 – Détermination de la QoS des sous-connexions

L'exemple Figure IV-8 présente la QoS définie pour le transport du flux référencé dans l'exemple précédent. La problématique réside dans la détermination des QoS à fournir pour l'établissement des connexions $A \leftrightarrow B$ et $B \leftrightarrow C$.

Plusieurs stratégies se démarquent, et l'efficacité du service de communication dépendra fortement de la stratégie appliquée. Mais avant d'aborder la comparaison de ces stratégies, deux remarques sur les connexions MMPOC-MN concernées doivent être faites :

- La fiabilité requise pour une connexion sortante ne peut être supérieure à la fiabilité requise pour la connexion entrante.
Supposons, par exemple, que la perte de b et de c soit autorisée sur la première connexion et pas sur la seconde. Si b et c sont perdus sur la première partie, il est impossible de garantir la QoS de la seconde car la quantité d'éléments transmis n'est pas suffisante.
- Le langage émis de chaque spécification doit appartenir au langage délivrable de la spécification de la dernière connexion.
Ceci implique, entre autre, que la période et l'alphabet des motifs de toutes les QoS établies sur une connexion MNP doivent être égaux. Cette règle permet de garantir que toute donnée reçue sur une connexion pourra être re-émise sur la connexion suivante et surtout que la spécification globale du flux sera égale à la QoS spécifiée pour le dernier flux.

Ces deux remarques sont essentielles, puisqu'il ne serait pas possible de réaliser ce service autrement. Ceci dit, analysons maintenant les possibilités de combinaison de QoS.

La Table IV-1 présente un ensemble de spécifications qui ont été déduites de la spécification du flux de l'exemple, en relaxant ou en augmentant, les contraintes sur la QoS. Toutes respectent les deux hypothèses énoncées plus haut, sauf les deux spécifications qui définissent un motif non ordonné (*ON_FP*, *ON_FT*) et données à titre d'exemple. Il existe d'autres spécifications dont l'ordre et la fiabilité sont compris dans les bornes définies par ces spécifications, mais dont le comportement généré sera semblable à l'un des cas cités. Par la suite, seules les abréviations seront utilisées.

Type de QoS		Spécification du flux
Ordre Nul, Fiabilité Partielle		$m = a \parallel b \parallel c \parallel d$ $ON_FP = c:: (m^* / [\{b,c\}, 1, 1])$
Ordre Nul, Fiabilité Totale		$m = a \parallel b \parallel c \parallel d$ $ON_FT = c:: (m^*)$
Ordre Total, Fiabilité Partielle		$m = a . b . c . d$ $OT_FP = c:: (m^* / [\{b,c\}, 1, 1])$
Ordre Total, Fiabilité Totale		$m = a . b . c . d$ $OT_FT = c:: (m^*)$
Ordre Partiel, Fiabilité Partielle		$m = a . (b \parallel c) . d$ $OP_FP = c:: (m^* / [\{b,c\}, 1, 1])$
Ordre Partiel, Fiabilité Totale		$m = a . (b \parallel c) . d$ $OP_FT = c:: (m^*)$

Table IV-1 - QoS dérivées

Analysons maintenant l'effet de la composition en série de deux connexions qui garantissent ces spécifications. Il faut noter que, comme nous l'avons dit plus tôt, pour garantir l'ordre et la fiabilité spécifiés dans l'exemple [Figure IV-8], la seconde connexion doit forcément garantir cette spécification (*OP_FP*). Le paramètre variable est donc uniquement la QoS de la première connexion. La QoS de bout en bout est donc définie par (*OP_FP*).

QoS MMPOC_MN 1	Comportement MNP
<i>ON_FP</i> <i>ON_FT</i>	Ces deux spécifications ne respectant pas les hypothèses sur les langages émis, un mécanisme de réordonnement devrait être ajouté à MNP pour émettre un flux adéquat.
<i>OT_FP</i>	La gestion d'un ordre total implique un risque d'attente lors du réordonnement des données durant la livraison au proxy qui n'est pas nécessaire si l'on tient compte des spécificités du flux.
<i>OT_FT</i>	La gestion d'une fiabilité totale implique un risque d'attente superflue lors de la livraison des données au proxy, puisque des retransmissions non nécessaires pourraient être attendues. Cette attente se reporte évidemment sur la seconde connexion.
<i>OP_FP</i>	C'est la solution la plus adaptée puisqu'elle n'implique pas les contraintes des solutions précédentes.
<i>OP_FT</i>	Idem <i>OT_FT</i>

Table IV-2 - Analyse des compositions de QoS

L'analyse des compositions [Table IV-2] révèle que la solution la plus efficace réside dans l'utilisation, pour toutes les connexions MMPOC-MN, de la même spécification de flux. Ceci peut aussi se justifier d'un point de vue plus pragmatique. Si la QoS offerte par le réseau supportant la première connexion est inférieure à celle offerte par le réseau utilisé pour la seconde, il vaut mieux garantir le minimum acceptable, c'est à dire *OP_FP*, afin de réduire le délai de livraison. Inversement, si la QoS offerte par le premier domaine est bonne, le service requis par *OP_FP* pourra être dépassé, sans que cela ne nécessite la garantie d'un ordre total ou d'une fiabilité totale.

3.3.4 Contrôle de congestion

Un mécanisme associé de contrôle de congestion est nécessaire afin de prévenir le risque de débordement des buffers internes du proxy MNP. Cette situation peut avoir lieu lors du passage d'une liaison de bonne qualité à une liaison de mauvaise qualité. Si le débit de la seconde connexion est limité par une faible bande passante disponible ou par des reprises de pertes nombreuses, la quantité de données stockées dans le nœud intermédiaire croîtra régulièrement jusqu'à la saturation des buffers. Afin d'éviter cette situation, l'augmentation de la taille des buffers doit être surveillée.

Le scénario d'une telle congestion est le suivant : Le buffer d'émission de la connexion sortante se remplit jusqu'à saturation puisque le débit sortant est inférieur au débit entrant. L'émission de données par la couche MNP sur la connexion sortante n'étant plus possible, les données ne sont plus extraites du buffer de réception de la connexion entrante. De ce fait, si la situation stagne, la saturation est aussi inéluctable pour la connexion entrante.

Comme un contrôle de congestion par fenêtre glissante est mis en place dans le protocole MMPOC-MN [Chapitre II], une réduction de la fenêtre d'émission va être générée par la réduction préventive du nombre d'acquittements renvoyés par l'entité réceptrice proche du débordement mémoire. L'entité émettrice passe en mode de contrôle de congestion en réduisant son débit d'émission. A terme, le débit des connexions se réduisant, les buffers du proxy se videront progressivement. La congestion pourrait, d'autre part, être réduite par l'ajout d'un mécanisme de suppression sélective des paquets qui ne sont pas strictement nécessaires au regard de la QoS garantie par le proxy MNP.

3.4 Couche MMPOC-MN

3.4.1 Limitations de l'étude

Une connexion MMPOC-MN est établie sur chaque portion de réseau homogène pour offrir un service à ordre et fiabilité partiels garantis. Dédié aux communications multimédias le protocole MMPOC-MN permet de tirer parti des configurations multi-réseaux parallèles. Toutefois, le spectre de l'étude que nous avons menée se limite à l'utilisation d'un service monomédia à ordre et fiabilité partiels, la combinaison des architectures multi-réseaux parallèles et série n'étant pas abordée. Il sera donc sous-entendu qu'un seul flux sera transporté par connexion MMPOC-MN sur chaque domaine de réseaux. Plusieurs flux émanant de médias différents peuvent toutefois être multiplexés par le protocole sur la même connexion. En reprenant les termes de la syntaxe de flux précédemment présentée, les connexions MMPOC-MN sont restreintes à un seul canal, mais peuvent véhiculer plusieurs flux.

3.4.2 Dimensionnement des buffers MMPOC-MN

Afin de réduire la taille des buffers intermédiaires, il est possible de profiter des mécanismes d'ordre et de fiabilité partiels pour maintenir seulement un service de retransmission minimal dans les nœuds du réseau. Puisqu'une demande de retransmission n'est faite par le récepteur que dans le cas où la qualité de service demandée n'est pas satisfaite (pertes trop nombreuses), un nœud intermédiaire n'ayant stocké que les TPDU's nécessaires pour assurer la qualité de service minimale demandée sera suffisant (OP_FP). Cette technique a pour principal avantage de limiter la taille des buffers, et donc de réduire les besoins en mémoire des nœuds. Ceci confirme l'intérêt de cette stratégie de la gestion de la QoS sur le proxy MNP. La réduction de la taille des buffers est proportionnelle au taux de fiabilité minimum demandé par l'application [GAY00].

3.4.3 Réduction du délai de bout en bout

Ce paragraphe aborde deux effets induits par l'utilisation de multi-connexions qui visent à réduire les délais de bout en bout, principal critère de qualité pour l'application réceptrice.

Adaptation de la connexion aux caractéristiques du domaine :

Les connexions MMPOC-MN établies sur des portions homogènes de réseaux sont, par définition, optimisées pour le réseau sous-jacent, puisque c'est l'un des rôles du protocole MMPOC-MN [Chapitre II, 3.2]. Les mécanismes dont le comportement est amélioré sont nombreux et ils contribuent tous à une optimisation substantielle du protocole.

Nous citerons parmi eux :

- Les contrôles de flux et de congestion qui sont plus efficaces, ce qui a pour conséquence une meilleure utilisation de la bande passante disponible,
- Les délais de retransmission qui sont ajustés au mieux, ce qui tend à améliorer le contrôle d'erreur.

Réduction du délai de détection et de recouvrement d'erreurs :

Le second effet induit par l'utilisation d'une série de connexions est la réduction du délai nécessaire au contrôle d'erreur. Les phases de détection d'erreur et de recouvrement d'erreur le composent. Classiquement effectués aux extrémités du réseau par la couche transport, ces mécanismes permettent de garantir la délivrance des données selon l'ordre et la fiabilité spécifiée.

Analysons le temps nécessaire au contrôle d'erreur lorsque celui-ci est effectué aux extrémités du réseau. Lorsqu'une donnée est perdue, dans le meilleur des cas, la détection de sa perte par l'entité réceptrice du transport est immédiate. Sa notification à l'entité émettrice par un acquittement (négatif) prend le temps de traversée du réseau. A la suite de cette demande, une retransmission est effectuée qui prend à nouveau le même temps, soit un $\frac{1}{2}$ RTT⁴. Il s'en suit, que lorsqu'une donnée est perdue, le temps de recouvrement de la perte est de 2 fois $\frac{1}{2}$ RTT, soit au minimum un RTT.

Maintenant, étudions ce délai dans le cas de figure de deux connexions MMPOC-MN en série. Le contrôle d'erreur est effectué sur l'extrémité de chacune des connexions, et de ce fait est cette fois-ci fait à l'intérieur du réseau. Prenons l'hypothèse simplificatrice que le proxy se trouve exactement à mi-chemin (en terme de délai) entre les deux utilisateurs. Le délai

⁴ Round Trip Time

nécessaire à la détection et à la notification d'une erreur survenue sur la première partie de la connexion est alors réduit de moitié, puisque l'acquittement ne parcourt plus que la moitié du chemin (proxy - émetteur). La retransmission, quant à elle, prendra le même temps car la donnée devra de toute façon traverser le réseau. Le délai de recouvrement d'une perte est alors égal à $\frac{3}{4}$ de RTT, ce qui constitue une amélioration du délai de 25%. Si la perte survient sur la deuxième connexion, le gain est identique puisque cette fois-ci, c'est le délai de recouvrement de la perte qui est réduit dans les mêmes proportions.

La Figure IV-9 résume ces mécanismes en prenant comme exemple la détection d'erreur sur un flux totalement fiable et ordonné.



Séquence vidéo

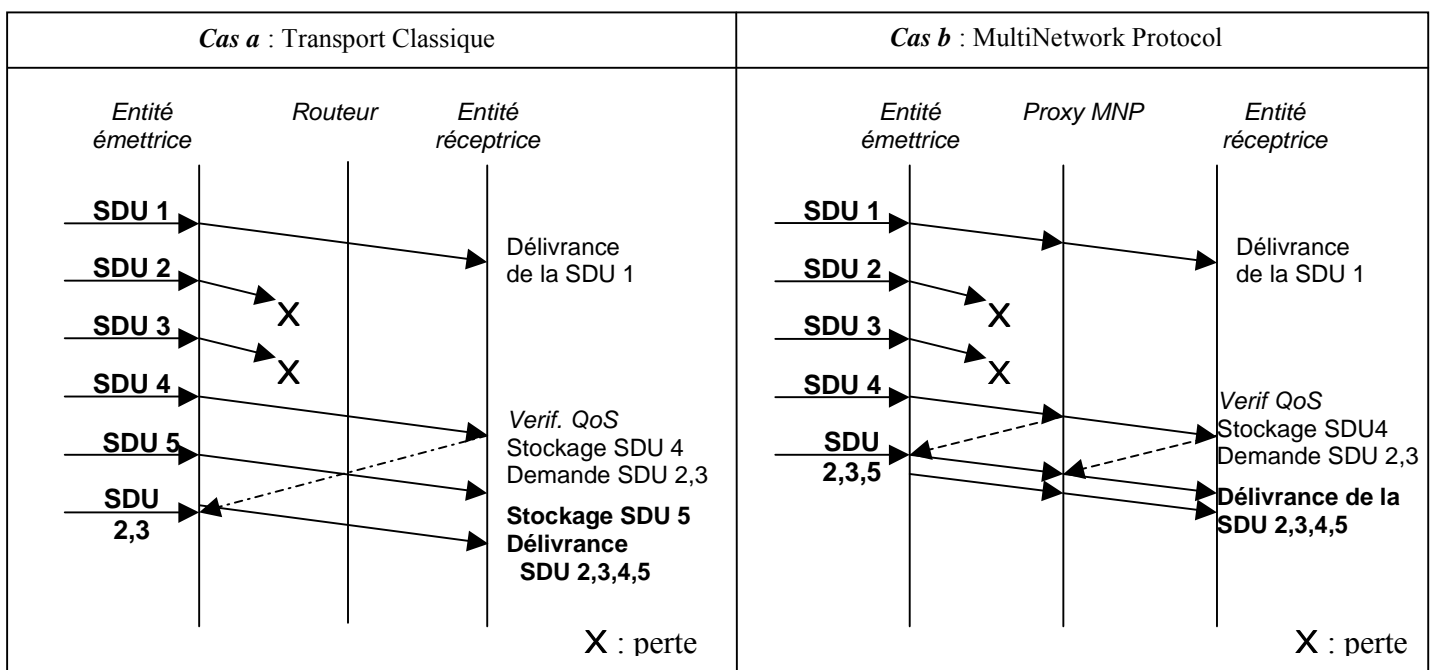


Figure IV-9 - Réduction du délai de recouvrement de pertes

La QoS est améliorée dans tous les cas, grâce aux recouvrements locaux des pertes, du fait de la réduction du délai de bout en bout. Ces résultats sont confirmés dans [LIU01] par la modélisation de connexions TCP interrompues par un proxy.

4 Déploiement de protocole

Comme pour le protocole MMPOC-MN, le déploiement reste parmi l'une de nos préoccupations les plus grandes. Une première solution basée sur l'installation manuelle des proxy MNP aux endroits sensibles du réseau est d'abord présentée. Mais comme nous l'avons souligné dans le chapitre concernant le protocole multi-réseaux parallèle (Chapitre II), le déploiement d'un nouveau protocole n'est pas une chose facile. Comme il serait très difficile de convaincre les administrateurs réseau d'installer massivement des proxy MNP, une

solution de déploiement automatique a été envisagée. Ces deux solutions complémentaires sont présentées indépendamment dans cette section.

4.1 Déploiement manuel

L'installation de proxy par un administrateur réseau est une chose courante de nos jours. On trouve sur la majorité des réseaux aujourd'hui, des proxys HTTP, des proxys NAT, des Firewall, tous étant configurés manuellement lors de leur mise en place. Ces outils se trouvent sous la forme d'équipements spécifiques, sous forme de boîtiers intégrés, ou bien sous forme logicielle à installer sur un routeur logiciel.

Développé en Java au dessus d'un routeur Linux, notre proxy MNP possède quelques particularités qu'il est bon de présenter. L'installation est simplement faite par l'ajout d'une couche logicielle au routeur.

Le langage Java ne permettant pas toutes les possibilités de programmation réseau du C, quelques astuces ont été utilisées. En effet, de nombreuses solutions de proxys sont transparentes à la couche transport (NAT, Firewall) en interceptant directement les paquets IP, sans remonter jusqu'à la couche transport. L'API réseau Java étant de haut niveau, les primitives de niveau le plus bas concernent uniquement la gestion des sockets TCP et UDP. Il n'est alors pas possible d'intercepter les connexions transport qui ne sont pas directement adressées au proxy. Ceci rend impossible l'interception des segments d'ouverture et de fermeture de connexion nécessaire au découpage transparent des connexions MMPOC-MN.

L'alternative consiste, comme cela est fait pour certains proxys HTTP, d'indiquer l'adresse du (ou des) proxy(s) à la couche MNP. Cette dernière se chargera d'établir les connexions nécessaires entre les utilisateurs et les proxys car toutes les adresses IP sont connues.

Il n'est pourtant pas concevable que pour initier une connexion MNP l'utilisateur soit au courant des adresses des proxys MNP à utiliser. Pour pallier à cette situation nous avons mis en œuvre une solution basée sur l'application *traceroute*, solution détaillée dans l'exemple suivant [Figure IV-10]. Cet exemple présente notre plate-forme locale de test. Elle est constituée de 3 machines Linux RedHat 7.0 isolées du réseau, où *mnp-racine* est un routeur.

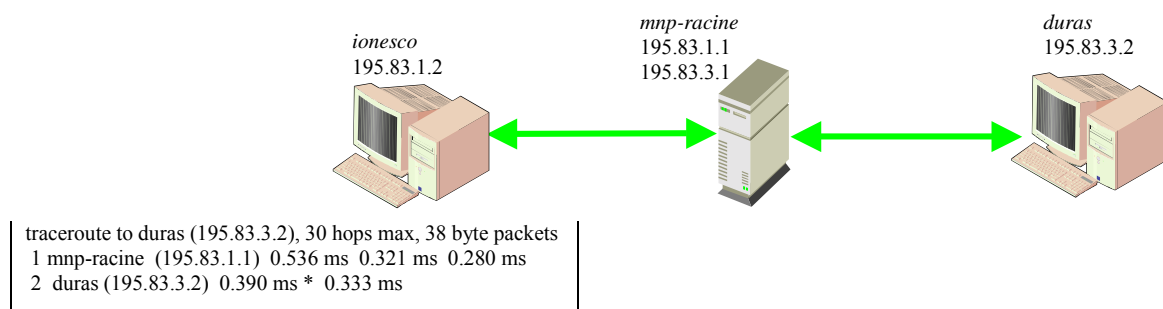


Figure IV-10 - Mécanisme de détection des proxys MNP

Pour initier une connexion MNP, l'utilisateur spécifie l'adresse IP du correspondant à la couche MNP, qui, à l'aide d'un *traceroute* va à son tour déterminer l'ensemble des adresses des routeurs traversés sur le chemin. Si l'un deux porte un nom débutant par la chaîne de caractères « *mnp-*», la couche MNP suppose que cette machine est un proxy et tente une demande de connexion sur le port dédié. Le résultat du *traceroute*, contenant la liste de l'ensemble des proxys potentiels est ensuite transmis à la couche MNP du proxy pour établir les autres connexions.

Tous les proxys situés sur le chemin sont ainsi très facilement découvrables. Cette technique est très facile à mettre en œuvre, rapide, et sans conséquence sur l'architecture du réseau. Un administrateur souhaitant offrir ce service n'a qu'à installer la couche logicielle correspondante et donner un nom correct à la machine. Toutes les connexions MNP entrantes et sortantes pourront alors utiliser le proxy MNP de manière transparente.

4.2 Déploiement automatique

4.2.1 Choix de la plate-forme

Comme précédemment argumenté pour le déploiement du protocole MMPOC-MN, même si un protocole apporte de réels gains de performances, son succès dépend très fortement de son déploiement dans le réseau pour atteindre un « effet de masse critique ». Il est très difficile de convaincre les utilisateurs et les administrateurs d'utiliser un nouveau protocole. Par conséquent, nous avons étudié le problème du déploiement dynamique du protocole MNP.

Le cas d'étude est passablement différent de celui déjà rencontré concernant le déploiement d'un protocole dans les entités d'extrémités (MMPOC-MN), car une contrainte supplémentaire est ajoutée :

- Déployer le protocole dans un des nœuds du réseau.

A ce jour il n'existe pas d'équipements de réseau permettant le déploiement de services de ce type dans le réseau. Le réseau n'est pas conçu pour laisser la possibilité à un utilisateur, ou bien à un fournisseur de service, de déployer dynamiquement un nouveau service sur les équipements traditionnels de routage.

Les réseaux actifs offrent une réponse au manque de flexibilité des réseaux classiques, et parmi les trois niveaux de granularité proposés par les différentes approches, une nous a paru plus prometteuse : celle des *réseaux à capsules*. L'approche *téléchargement de service* par l'administrateur réseau nous semblait trop contraignante et trop proche de l'installation manuelle d'une machine. L'approche *code mobile* est particulièrement bien adaptée au déploiement de programmes chez les utilisateurs, mais pas au cœur du réseau.

L'approche *capsules*, qui permet d'ajouter à l'entête des paquets traditionnels, des programmes miniatures qui peuvent être exécutés dans tous les nœuds du réseau traversé, offre le niveau d'action dans le réseau que nous souhaitons.

La plate-forme la plus développée au moment où nous avons fait ce choix était ANTS [WET98]. Elle est de plus basée sur le langage Java, ce qui la rend compatible avec les travaux précédemment effectués sur HotJava. L'architecture proposée intègre alors un moyen de déployer les protocoles MNP/MMPOC-MN au travers et à l'intérieur du réseau. Le déploiement du protocole MNP chez les utilisateurs est rendu possible par l'implémentation Java et le support de HotJava. Les protocoles et les groupes de codes (au sens ANTS) fournissent un moyen d'ajouter du traitement défini par l'utilisateur dans le réseau, ce qui a pour conséquence de permettre le déploiement dynamique des passerelles multi-réseaux série. Pour ce faire, un *nœud actif* doit être configuré pour permettre le téléchargement du code MNP et son exécution. Aujourd'hui, seuls les routeurs logiciels peuvent héberger un routeur actif qui peut-être constitué, par exemple, par le code Java de la plate-forme ANTS. Notre architecture nécessite donc la construction d'un réseau virtuel de nœuds actifs (appelé communément réseau actif), au dessus d'un réseau IP. Ainsi, quand une connexion est

ouverte, si un ou plusieurs nœuds actifs sont traversés sur le chemin, le code des passerelles peut être téléchargé et activé sur ces nœuds.

4.2.2 Principes de déploiement

Dans le cadre de la plate-forme active ANTS, l'utilisation du service de distribution de code pour déployer un protocole de transport impose certaines contraintes.

Un protocole ANTS est un programme Java référencé par les capsules qui est exécuté sur chacun des nœuds actifs traversés après son téléchargement. Il a la particularité de devoir être exécuté entièrement (c'est à dire de se terminer) avant que la capsule l'ayant référencé puisse être transmise au nœud suivant. Cette sémantique d'exécution impose de sévères restrictions quant au déploiement de protocoles. En effet, il n'est alors pas possible de laisser, sur un nœud actif, une socket MMPOC-MN ouverte, comme cela est suggéré par l'architecture.

D'autre part, les seuls états rémanents que puisse conserver le protocole, sont ceux laissés dans un cache spécifique à chaque nœud actif. La capacité de ce cache est assez restrictive, et ne permet à l'heure actuelle de ne stocker que des objets simples (entiers, paquets, ...). Il est impossible d'y laisser un programme actif après le départ du nœud de la capsule; ceci rend particulièrement délicat la gestion de « timers » classiquement mise en œuvre dans les protocoles de transport.

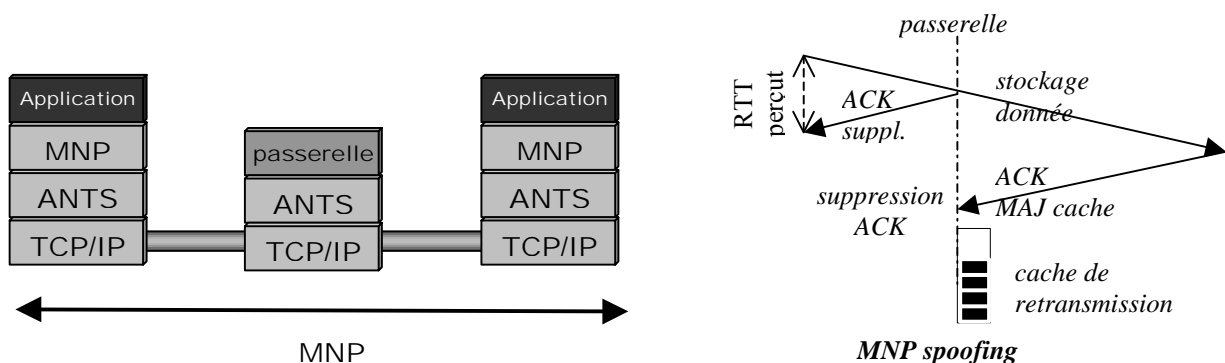


Figure IV-11 - Principe du spoofing appliqué à MNP

Il semble difficile de développer un protocole de transport mobile dans ces conditions, et dans l'état actuel le déploiement d'une solution de *splitting* par un proxy MNP est impossible. Toutefois, une solution alternative, utilisant les principes des PEP de type *spoofing* a été testée et offre des résultats encourageants [Figure IV-11].

Dans cette solution, les paquets de données MNP contiennent le code nécessaire pour être acquittés par les nœuds intermédiaires, stockés dans le cache du nœud et relayés vers le nœud suivant. Au retour, les acquittements, possèdent le code pour mettre à jour le cache du nœud précédent en supprimant les références des données acquittées. Sur chacune des passerelles est par conséquent stocké :

- Un numéro de séquence permettant la détection d'erreur,
- Un seuil de fiabilité et le nombre de pertes acceptées à la vue de la fiabilité partielle,
- Un cache de données.

Ainsi les mécanismes d'acquittements et de retransmissions locales peuvent être fournis.

Par rapport à la solution de *splitting* les restrictions sont nombreuses quant à la gestion de la qualité de service sur les nœuds intermédiaires. La gestion d'un ordre partiel est impossible sur un nœud actif car cela demande trop de ressources en termes de calculs et de mémoire. Seule la comparaison des numéros de séquence des paquets permet de détecter une rupture de séquence dans un ordre total et déclencher une demande de retransmission.

Une gestion limitée de la fiabilité partielle est rendue possible par le comptage du nombre de pertes acceptées et la comparaison par rapport à un taux de pertes acceptables. Mais cette gestion de la fiabilité est loin de celle proposée par le protocole MMPOC-MN, seule une gestion par pourcentage peut être réalisée.

Pour toutes ces raisons, nous continuons à penser que le principe de *splitting* est mieux adapté et envisageons de nouvelles solutions pour tendre vers ce but. L'évolution des plates-formes de réseaux actifs va dans ce sens.

4.2.3 Décision de localisation du proxy

Le déploiement dynamique du proxy MNP soulève un problème assez complexe à résoudre : La décision de déploiement d'un ou plusieurs proxys, et leur lieu d'implantation.

En d'autres termes, la question peut être posée différemment, en se demandant « quelles sont les portions du chemin qui pourraient poser problème à une connexion de bout en bout ? ». Si une réponse est trouvée à cette interrogation, alors des proxys MNP peuvent être déployés aux extrémités, de (ou des), portion(s) pouvant engendrer une baisse des performances d'une connexion transport.

Pour ce faire, il est nécessaire d'offrir une vision globale de la qualité de service offerte par la succession des liens constituant l'ensemble du chemin reliant l'émetteur au récepteur. Les seules techniques disponibles aujourd'hui reposent sur la combinaison des applications *traceroute*, pour connaître le chemin, et *ping* pour déterminer les temps de réponse et une estimation du taux de perte entre chaque nœud. Les mesures seront prises uniquement entre chaque nœud actif du réseau puisqu'ils sont les seuls à pouvoir supporter un proxy MNP.

Le délai et la fiabilité d'une liaison étant les principaux responsables du mauvais fonctionnement des protocoles de transport, une méthode d'estimation de la QoS pour l'évaluation de ces critères est mise en œuvre. L'objectif est, sur un chemin comportant un certain nombre de liaisons, de repérer où peut avoir lieu une rupture de la QoS réseau. (i.e. repérer à partir de quels nœuds changent radicalement les paramètres de délai et de fiabilité). Ainsi, un long délai, imputable par exemple à une liaison satellite, ou un taux d'erreur important, induit par une liaison mobile, peuvent être aisément repérés sur le chemin.

La Figure IV-12 présente un exemple de l'estimation qui peut être réalisée à l'aide de cette méthode. Les nœuds actifs renvoyés par la commande *traceroute* (dans sa version ANTS) sur le chemin entre A et D sont B et C. Une série de *ping* est successivement envoyée à chacun de ces nœuds, et le résultat analysé. En comparant les temps d'aller et retour fournis par le *ping*, le délai de la liaison B-C est estimé à 253 ms. Ce délai peut correspondre à une liaison satellite, et afin de préserver la QoS fournie par la connexion MNP, un proxy peut être déployé sur B et C.

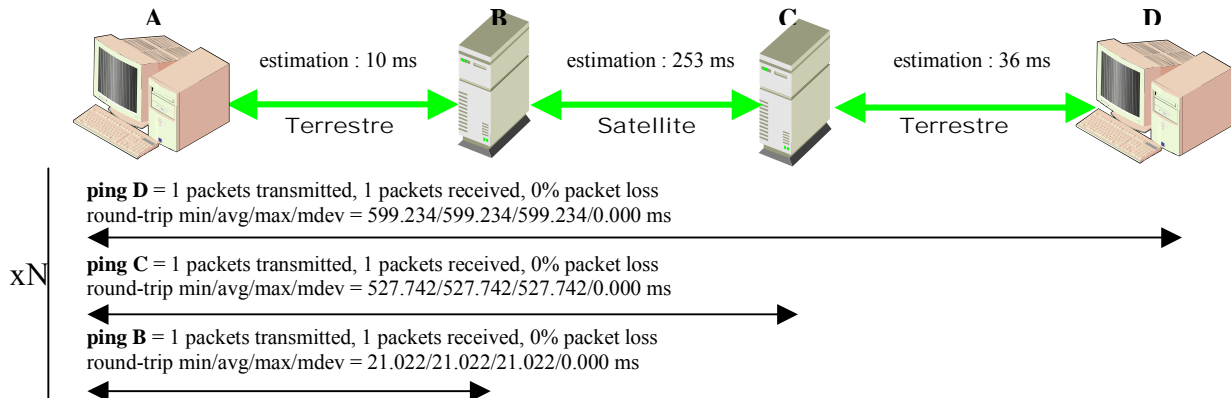


Figure IV-12 - Estimation de la QoS inter-nœud du réseau

Cette méthode, bien que manquant de fiabilité, possède l'avantage d'être très simple, et donne une approximation qui s'améliore au fil des séries de mesures. Seuls les paramètres de délai et de fiabilité peuvent être mesurés. D'autres méthodes plus complexes pourraient être envisagées, en particulier en utilisant les principes de capsules mis en œuvre dans un réseau actif qui permettraient des mesures bien plus variées. Toutefois, le temps de réponse d'un nœud actif étant encore très variable dans les implantations actuelles d'ANTS, ces méthodes semblent encore peu fiables.

4.2.4 Conclusion

Le déploiement d'un mécanisme de contrôle d'erreur local a été rendu faisable grâce aux fonctionnalités offertes par ANTS. Il est donc possible, à l'aide d'un tel réseau actif, de déployer un PEP (au sens IETF) de manière dynamique à la demande de l'application. Toutefois, l'approche ANTS ne semble pas si prometteuse que nous le pensions à l'origine en terme de flexibilité. Il est difficile de déployer des services complexes, nécessitant de nombreuses ressources, autant en termes de mémoire que de calculs. D'autre part, les aspects de sécurité sont pour l'instant très peu abordés, et il ne semble encore pas concevable de déployer une telle plate-forme dans un avenir proche.

Par contre, l'expérience acquise par l'utilisation d'ANTS nous a permis de cibler les besoins des protocoles de transport pour leur amélioration au travers des réseaux actifs. Des solutions de déploiement dynamique de services plus complexes sont aujourd'hui à l'étude dans le cadre de projets auxquels nous participons [AMA99][GCA99], et semblent susciter un intérêt grandissant dans la communauté scientifique et industrielle.

5 Problèmes liés à l'utilisation de proxys

Nombreux sont les partisans des solutions d'optimisation des protocoles de transport par les PEPs, mais encore plus nombreux sont les adversaires de telles solutions. Le déploiement généralisé de proxy dans le réseau possède des inconvénients qu'il faut présenter et évaluer.

Ces techniques de proxys, en violant les principes de bout en bout de la couche transport, entraînent des implications pernicieuses sur le fonctionnement global de l'Internet. Le paragraphe suivant liste ces problèmes.

Maintient d'état dans le réseau :

Si une panne survient dans le réseau, la capacité d'une connexion à survivre à la panne dépend principalement du nombre d'états maintenus dans le réseau pour la connexion et si ces états peuvent être automatiquement réparés. Si aucun état spécifique ne réside dans le réseau, ou qu'un tel état est auto-réparable, comme c'est le cas dans des connexions de bout en bout, alors l'incident réseau ne coupera la connexion que seulement si aucun chemin alternatif dans le réseau n'a pu être trouvé. D'ailleurs, dans ce cas, les deux entités d'extrémités détecteront l'incident. Au contraire, si la connexion dépend d'états sauvegardés dans le réseau (i.e. un PEP), alors dans le cas d'un problème réseau (i.e. une défaillance du PEP), l'état est perdu, forçant la connexion à se terminer, même si un autre chemin de secours aurait pu être utilisé.

Il est toutefois défendable, que lorsque de nettes améliorations peuvent être obtenues grâce à l'utilisation d'un proxy, en particulier dans le maintien de la connexion par des mécanismes spécifiques supportés par le PEP, un utilisateur averti choisira cette solution, d'autant plus que la fiabilité des réseaux croît d'années en années.

Sécurité :

L'un des grands arguments militant contre l'utilisation des proxys est l'impossibilité d'utiliser des mécanismes de sécurité comme IPSEC sur une multi-connexion. En effet, IPSEC est basé sur le partage d'un secret (une clé de codage) entre les entités d'extrémités et le cryptage des données encapsulées sur toute la connexion. Si un Proxy est utilisé, pour pouvoir accéder aux paquets décryptés il doit partager le secret afin de pouvoir décoder et relayer les données. Pour garantir la sécurité ceci est souvent indésirable.

Notons toutefois que ceci n'est pas incompatible avec l'architecture MNP, puisque la sécurisation des données concerne particulièrement les applications de commerce électronique. Elles échangent des flux asynchrones et sortent ainsi du cadre d'un protocole multimédia. D'autre part, l'utilisation de proxy peut, bien sûr, être désactivée pour certains cas spécifiques, où par exemple un organisme souhaiterait crypter ses communications.

Routage et mobilité :

Dans les environnements de communication mobile il se peut que lors d'un changement de position d'un utilisateur, un chemin plus court puisse être trouvé entre les deux entités extrémités ne passant pas forcément par le proxy. Le maintien de la communication au travers du proxy contraint alors à utiliser un chemin non optimal.

Les proxys, très souvent utilisés dans ce cadre de réseaux sans fil, sont situés, soit sur les points d'entrées du réseau d'un opérateur mobile, soit en amont d'un réseau local sans fil. Dans les deux cas, ces points d'entrées sont si peu nombreux, que l'optimisation du chemin serait difficile. D'autre part, la faible mobilité des utilisateurs, tout au moins à grande échelle, ne justifie pas de telles hypothèses.

Routage asymétrique :

Certains environnements, comme certains liens satellites, impliquent un routage asymétrique des flux, c'est à dire que le canal retour (pour les acquittements en particulier) n'est pas le même que le canal aller (pour les données). L'utilisation d'un proxy contraint les deux flux à se rejoindre en un point, d'où la possibilité d'obtenir un routage non optimal.

Généralement, les liaisons impliquant un routage asymétrique posent d'autres problèmes de QoS, comme un long délai pour les liens satellites, et l'utilisation du proxy compense largement les baisses de performances induites par un routage non optimal.

Conclusion :

L'architecture de communication MNP n'engendre pas tous les problèmes sus cités, néanmoins l'utilisation des proxy MNP doit être, à notre avis, prônée uniquement dans des cas spécifiques, tels que la traversée de réseaux d'accès mobiles. En effet, les architectures actuelles de réseau évoluent vers des cœurs de réseau très simples pour atteindre de grandes performances, et déplacent l'intelligence vers les bordures de réseau. Généralement installés par les opérateurs, le déploiement des proxys MNP peut vraisemblablement être envisagé comme contrôlé.

6 Mesures

Pour illustrer les bénéfices apportés par l'architecture MNP, des simulations ont été réalisées à l'aide du simulateur OPNET. C'est une fois de plus, comme pour les simulations du protocole MMPOC-MN, l'étendue de la gamme des réseaux simulables par OPNET qui nous a orienté vers des évaluations de performances par simulation plutôt que par expérimentations. Il est très difficile, à l'aide des outils actuels, et nous pensons en particulier aux émulateurs tels que *dummysnet* [RIZ97] de recréer des conditions d'expérimentations proches des qualités de services offertes par les nouvelles générations de réseaux. De ce fait, l'étude et le développement de protocoles de communication est difficile sans une phase préalable de simulation. Il est toutefois regrettable que pour atteindre un niveau de précision suffisant dans la simulation du comportement de protocole et en particulier de protocoles de transport, que la modélisation du protocole s'apparente plus à une implantation réelle qu'à la définition d'une maquette. De plus, les nombreuses heures de développement du protocole nécessaires à la réalisation du modèle sont souvent perdues, car il n'est pas encore possible de dériver le code de l'implantation du code du modèle et qu'une réécriture est systématique, dans la majorité des environnements.

Plutôt que de présenter les performances du protocole de manière globale et dans un cas très favorable, le parti a été pris de présenter séparément les résultats obtenus pour les différents critères qui permettent l'optimisation du protocole. Dans tous les cas, la solution MNP est comparée à une solution de bout en bout.

L'optimisation induite par l'adaptation des multi-connexions aux multi-domaines réseaux est d'abord évaluée. Les gains induits par la gestion d'un ordre et d'une fiabilité partiels sont ensuite mesurés sur un autre exemple.

Pour illustrer les bénéfices de l'utilisation du protocole multi-réseaux série, la configuration suivante a été choisie [Figure IV-13] : Une connexion traversant deux domaines offrant des QoS distinctes est scindée par un proxy.

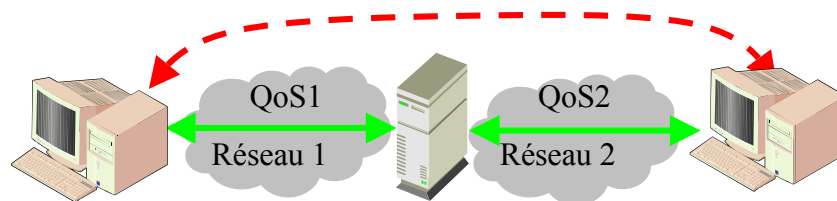


Figure IV-13 – Configuration multi-réseaux série

6.1 Influence de l'adaptation du transport au réseau

L'adaptation du protocole de transport au réseau sous jacent a montré être une idée intéressante en terme d'amélioration des performances. L'évaluation du protocole MMPOC-MN quantifie cette optimisation, en comparant les performances de protocoles classiques et de protocoles adaptés aux réseaux Hertziens.

Comme l'utilisation de protocoles optimisés n'est pas toujours possible sur une connexion entière, le protocole MNP propose l'utilisation de ce type de protocoles uniquement sur les domaines de réseaux où cela est nécessaire.

Afin de mesurer uniquement le gain apporté par ce type de gestion, les mécanismes d'ordre et de fiabilité partiels n'ont pas été utilisés pour cet ensemble de simulations. Ainsi, le délai de bout en bout fourni par une série de connexions TCP est évalué pour la configuration présentée sur la Figure IV-14.

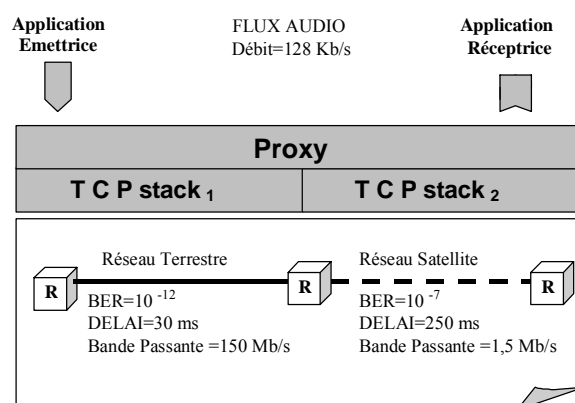


Figure IV-14 - Conditions de simulation

Deux domaines sont traversés par la connexion multi-réseaux. Les réseaux sous jacents sont :

- Un réseau terrestre offrant une très bonne QoS, avec un taux de perte et un délai faibles.
- Un réseau par satellite dont le long délai et le taux d'erreur assez élevé sont les principaux défauts.

Le délai de bout en bout est mesuré pour deux configurations de TCP *splitting*, dont la première est composée de deux souches TCP classiques (*new Reno*) et la seconde d'une souche classique et d'une optimisée pour les réseaux par satellite. Ces deux types de configurations sont à comparer avec une solution TCP classique. La Figure IV-15 présente ces résultats.

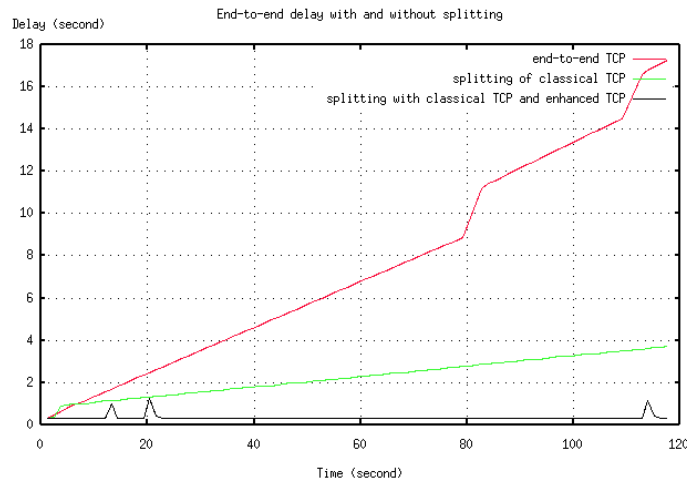


Figure IV-15 - Comparaison du délai de TCP dans une configuration multi-réseaux

On peut tout d'abord remarquer sur la Figure IV-15 que l'utilisation d'une connexion classique de TCP n'est pas satisfaisante dans la mesure où le retard pris par la connexion est d'environ 20 secondes après 2 minutes de simulation. Cela signifie qu'une donnée émise par l'utilisateur sera reçue et présentée au récepteur 20 secondes plus tard. On peut ainsi estimer que le temps de téléchargement d'un fichier audio dans cette configuration durera 16 % de temps en plus que sa lecture (20 secondes sur une durée de 2 minutes). Lorsqu'une solution classique de *splitting* est employée, ce retard diminue à moins de 4 secondes après 2 minutes de simulation. Cette configuration devient acceptable pour un flux temps réel dans la mesure où l'application de présentation peut accepter, de temps en temps, d'effectuer un rattrapage du décalage en déclenchant la perte de quelques données pour atteindre à nouveau un faible délai. Cette resynchronisation pourrait avoir lieu par exemple toutes les 30 secondes en engendrant une coupure d'une ½ seconde.

Finalement, lorsqu'une solution combinant l'approche de *splitting* et l'adaptation de TCP au réseau satellite (modification de la taille de fenêtre, SACK, etc ...) est mise en œuvre, le délai de la connexion multi-réseaux chute à près de 250 ms (le délai physique du réseau), ce qui est une valeur tout à fait acceptable pour un flux temps réel.

6.2 Influence de l'ordre et de la fiabilité partielle

Analysons maintenant l'impact de la fiabilité partielle sur une multi-connexions. Afin de ne mesurer que le gain lié à cette gestion, les problèmes d'adaptation précédemment étudiés sont éliminés, en changeant la configuration de simulation. Cette fois ci, un réseau local sans fil remplace la problématique liaison satellite. La même configuration de protocole peut convenir alors aux deux environnements.

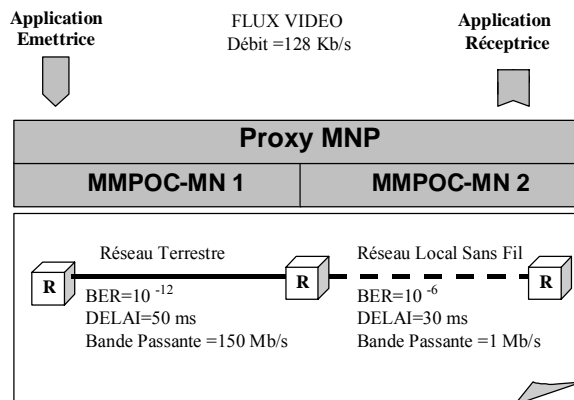


Figure IV-16 - Conditions de simulation

La Figure IV-16 décrit les conditions de simulation. Le premier réseau est un réseau longue distance à fibre optique ($BER=10^{-12}$ et $Delai=50$ ms) et le second est un réseau local sans fil ($BER=10^{-6}$ et $Delai=30$ ms).

La simulation est ainsi paramétrée, et le délai de bout en bout ainsi que la taille des buffers de l'émetteur ont été mesurés.

Les mesures confirment à nouveau que le délai de bout en bout est réduit lorsque la connexion transport est splittée.

Les résultats sont visibles sur la Figure IV-17. Les courbes correspondant aux cas suivants sont indiquées sur le même graphique : TCP de bout en bout, MMPOC-MN (de bout en bout) et MNP. Deux configurations des cas MMPOC-MN et MNP sont étudiés : lorsqu'une fiabilité totale est requise pour le flux, et lorsqu'une fiabilité partielle de 90 % est demandée.

Les résultats montrent que TCP n'est pas si inefficace que ça : 60% des paquets TCP enregistrent un délai inférieur à 150 ms, mais la croissance lente de la fin de la courbe montre que, à contrario, certains paquets peuvent subir un retard très long.

De tels retards sont très gênant pour l'ensemble de la QoS de la connexion, et impliquent l'utilisation de très grands tampons de mémoire. La taille de ces tampons est présentée sur la Figure IV-17.

MMPOC-MN configuré avec 100% de fiabilité est assez similaire à TCP et ceci est tout à fait logique puisque ces deux protocoles sont sensés assurer le même service. Les différences s'expliquent par des choix d'implémentation différents entre les deux protocoles.

Ce qui est très intéressant est que la fiabilité partielle augmente considérablement les performances. Le retard est bien moins important du fait qu'il est possible de supprimer les paquets qui tardent trop et de ce fait les performances de la connexion ne sont pas touchées. En découpant la connexion de bout en bout, les résultats sont encore meilleurs du fait de la proximité des retransmissions et de la gestion de la fiabilité partielle. En conséquence, les retards sont beaucoup moins nombreux. Ceci se démontre par une courbe qui finit rapidement, prouvant que les paquets n'enregistrent pas de très longs retards.

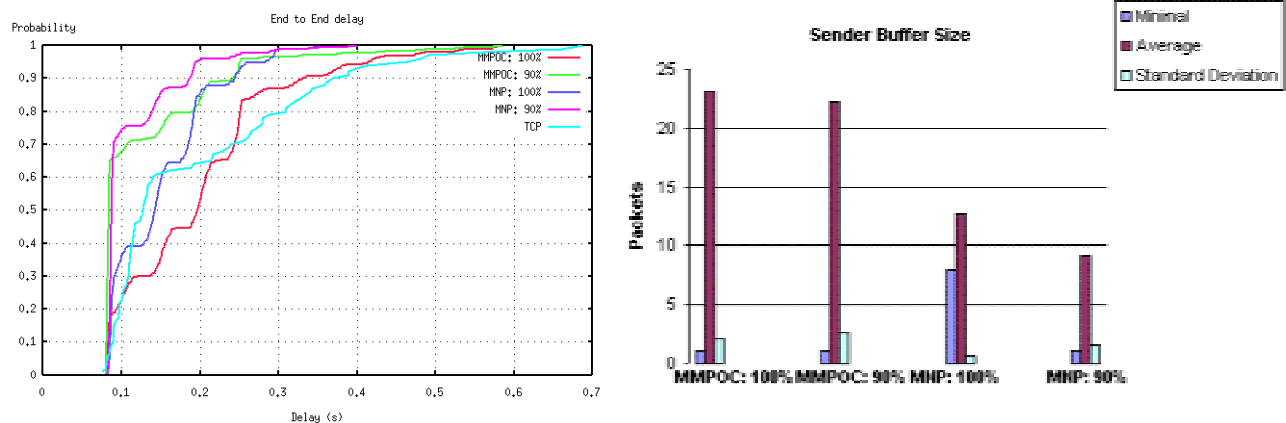


Figure IV-17 – Influence conjuguée de la fiabilité et du découpage de connexion

7 Conclusion

L'émergence de nouvelles technologies de communication est en train de changer les hypothèses sur lesquelles reposaient le bon fonctionnement des réseaux actuels. Dans ce chapitre, nous avons mis en évidence que le réseau ne pouvait plus être considéré comme homogène de bout en bout et que les mécanismes mis en œuvre dans les protocoles de transport s'en trouvaient perturbés.

Les configurations évoquées deviennent incontournables du fait d'un besoin omniprésent de connectivité. Les réseaux mobiles de troisième génération deviendront, dans un futur proche et pour les zones urbaines, une technique d'accès incontournable, de la même façon que la téléphonie mobile a supplanté la téléphonie fixe. D'autre part, les réseaux par satellite, qui offrent une solution à faible coût pour l'équipement des zones rurales, devraient voir leur utilisation se développer pour répondre aux nouveaux besoins issus du télétravail.

Pour faire face à cette hétérogénéité, nous proposons une nouvelle architecture de protocoles, basée sur l'utilisation de séries de connexions adaptées aux domaines réseaux séparant deux utilisateurs. Une solution de déploiement dynamique de ce protocole est abordée, permettant l'utilisation de tels protocoles lorsqu'il ne sont pas proposés par un fournisseur de service, mais aussi une plus grande liberté quant à leur utilisation. Les principes de ruptures de connexion peuvent être utilisés ou non, en fonction des désirs de l'utilisateur qui peut souhaiter conserver une solution classique pour répondre, par exemple, à des exigences de sécurisation du transfert de données.

Cette dernière caractéristique étend la gamme des réponses qui peut être donnée aux détracteurs de l'utilisation de multi-connexions transport, dans le sens où le fournisseur de service et l'utilisateur peuvent conjointement décider d'appliquer ou non cette méthode : l'un en décidant d'offrir le service et l'autre en le sélectionnant.

Chapitre V Implémentation

1 Introduction

Cette architecture a donnée lieu à deux souches de protocoles interopérables, toutes deux réalisées en Java : MMPOC-MN et MNP. L'implémentation de ces protocoles sera détaillée dans le dernier rapport de la tâche multi-réseaux du projet GCAP [GCAP99]. Plutôt que présenter l'aspect technique de ces implémentations nous avons choisi de présenter les interfaces de programmation ainsi que les mesures de performances de ces protocoles. L'accent sera mis principalement sur MMPOC-MN puisque MNP est encore en cours de développement. En effet les premières versions du protocole MNP ont donné de faibles résultats en terme de performances (imputables à une mauvaise gestion d'ANTS). Ce dernier point sera discuté plus en avant.

Les interfaces de programmation du protocole (API) sont tout d'abord présentées. La première permet la spécification de QoS par l'application, la seconde est celle du protocole MMPOC-MN à proprement parler.

Les mesures de performances réalisées sur la souche MMPOC-MN sont ensuite détaillées. L'influence de l'environnement Java et de la gestion de l'ordre partiel y est analysée.

2 Interfaces de programmation

2.1 API du langage de flux

Comme cela a été annoncé dans le chapitre présentant le modèle de flux, afin d'offrir une interface de programmation (API) permettant la spécification de flux à ordre et fiabilité partiels, le modèle a été traduit dans un langage de programmation orienté objet.

Les paramètres de qualité de service peuvent être ainsi spécifiés à l'aide du langage de flux, et ce au travers d'une API. La spécification est faite dans le code source de l'application multimédia distribuée pour permettre la transcription des besoins applicatifs au protocole de MMPOC-MN. La qualité de service ainsi définie est ensuite garantie par le protocole pendant la durée de la communication.

2.1.1 Concepts

L'intérêt de spécifier la qualité de service dans le code source du programme, plutôt qu'à l'aide d'un logiciel spécifique, est multiple.

L'impact psychologique doit être tout d'abord souligné. La qualité de service ainsi décrite au sein de l'application, en utilisant les mêmes outils et les mêmes techniques de développement que le programme, est « démystifiée ». La qualité de service devient une composante banale d'un système réparti, et n'est plus vue par le programmeur comme une entité externe qui n'est ni un algorithme, ni une structure de donnée classique.

D'autre part, l'utilisation d'un langage de programmation comme moyen d'expression de la QoS possède des attraits techniques. Plusieurs alternatives s'offraient quant à la réalisation des analyseurs du langage de flux, nécessaires à la vérification des spécifications de QoS et à leur traduction en structures informatiques.

La première aurait été de créer des analyseurs lexicaux et syntaxiques pour interpréter les spécifications, comme cela peut être fait à l'aide des outils Lex et Yacc. Lors de l'exécution de l'application, ces analyseurs auraient été appelés et auraient permis la détection

d'éventuelles incohérences dans la spécification. Cette solution, bien que classique, recèle plusieurs inconvénients :

- La détection d'incohérences dans la spécification ne peut se faire que lors de l'exécution du programme, ce qui peut allonger la durée de la phase de tests.
- Lors de chaque lancement de l'application, la vérification consomme des ressources en temps de calcul puisque l'analyse est faite à ce moment.

La seconde alternative, que nous avons développée, consiste à bénéficier des phases d'analyse lexicale et syntaxique ayant lieu lors de la compilation du programme. L'analyseur utilisé étant celui du compilateur, la syntaxe et la grammaire du langage de flux doit être compatible avec celle du langage de programmation.

Pour ce faire, Java, langage de programmation Orienté Objet, a été choisi pour son pouvoir d'expression et la rigueur des vérifications de type (*typage*). A partir du modèle, un diagramme de classe UML a été déduit en respectant une simple méthode de traduction :

- A chaque *type* du modèle est associé une *classe* dans le diagramme.
- A chaque *opérateur* du modèle est associé une *méthode* de classe.

L'API est ainsi composée de six classes : `FinitePattern` (*motifs finis*), `NumberedPattern` (*motifs numérotés*), `LossyPattern` (*motifs avec pertes*), `ElementaryStream` (*flux élémentaires*), `Stream` (*flux multimédia*), `Synchro` (*expressions de synchronisations*).

Chaque classe est composée d'un ensemble de méthodes. Par exemple la classe `FinitePattern`, est composée, entre autres, des méthodes `par()` et `seq()` qui correspondent aux opérateurs `.` et `|`. Le diagramme de classe est présenté sur la page suivante. La partie encadrée correspondant à l'API, les autres classes servent à la définition des reconnaisseurs.

La correction syntaxique de la spécification est assurée par le compilateur qui décèle les incorrections orthographiques ainsi que les erreurs de parenthésage. La correction grammaticale est garantie par la vérification de type effectuée sur les paramètres lors des appels de méthode. Il serait impossible, par exemple, de composer en série un motif fini avec un flux élémentaire, car la méthode `seq()` a été construite de façon à interdire ce genre de déclarations. L'analyse de la validité de la spécification est rendu ainsi très simple, et n'est faite qu'à la compilation du programme, c'est à dire pendant la phase de codage du cycle de développement de logiciel.

2.1.2 Exemple

L'exemple de la visioconférence présenté dans le chapitre III, section 4.2 est à nouveau présenté pour détailler l'utilisation de l'interface de spécification de la QoS. La Figure V-1 rappelle la spécification proposée pour le flux audio. La spécification du flux vidéo n'est pas détaillée dans un but de clarté, le flux `video` est donc considéré comme déjà défini.

$$\begin{aligned} PCM_seq_droit &= d^4 \\ PCM_seq_gauche &= g^4 \\ PCM_seq_stéréo &= PCM_seq_droit \parallel PCM_seq_gauche \\ audio_stéréo &= PCM_seq_stéréo^* \\ flux_stéréo &= c_2 :: (audio_stéréo / [\{ d \}, 1, 2] / [\{ g \}, 1, 2]) \\ \xi &= [(c_1, 1) <> (c_2, 2)] \\ audio_video &= (audio \parallel video) [\xi] \end{aligned}$$

Figure V-1 - Spécification d'une visioconférence

La Figure V-2 présente la traduction de la spécification de la visioconférence (Figure V-1) au travers de l'API. Deux nouvelles classes apparaissent dans ce schéma.

`NumberedPattern` remplace les motifs finis pour construire les flux élémentaires à partir de motifs numérotés.

`StreamAutomaton` apparaît à la fin de la spécification. Elle correspond à la déclaration des automates permettant la reconnaissance des flux. En paramètre est passé l'objet `Stream` qui correspond à la spécification de QoS.

```
// Spécification des motifs de base des canaux audio
FinitePattern PCM_seq_droit = new FinitePattern("dddd");
FinitePattern PCM_seq_gauche = new FinitePattern("gggg");
// spécification du motif de base du flux audio stereo
FinitePattern PCM_seq_stereo = PCM_seq_droit.par(PCM_seq_gauche);

// définition du flux élémentaire audio stereo à partir du motif numéroté
ElementaryStream audio_stereo = new ElementaryStream(new
    NumberedPattern(PCM_seq_stereo, new DepthAlgorithm()));
audio_stereo.addLossSpecification("d", 1, 2);
audio_stereo.addLossSpecification("g", 1, 2);

// définition des flux audio et video
Stream audio = new Stream(audio_stereo);
Stream video = new Stream(video_elem);

// définition du flux audio / video
Stream audio_video = audio.par(video);
// spécification de la synchronisation inter-flux
audio_video.synchronize(new
    HardSync(audio_stereo.getLocation(), 2, video_elem.getLocation(), 1));

// Construction de l'automate de flux élémentaire
A = new StreamAutomaton(audio_video);
```

Figure V-2 - Spécification d'une visioconférence à l'aide de l'API

2.2 API du protocole

```
public class MMPocMN extends Thread
{
    public MMPocMN ();
    public MMPocMN (MMPocMNWarning Callback);
    public int FlowNumber();
    public int setInputBlockingOption(int i,Thread Reader);
    public int setOutputBlockingOption(int i,Thread Writer);
    public int connect(String Host, int Port, MMPOCMN_QoS Demande);
    public int listen(int port, MMPOCMN_QoS Recu);
    public int accept(MMPOCMN_QoS Accepte);
    public int close ();
    public int closeOneway ();
    public int read (int Flux, byte[] tab, int length);
    public int available(int Flux);
    public int write (int Flux, byte[] tab, int length);
}
```

Figure V-4 - API du transport à ordre et fiabilité partiel

L'API détaillée dans la Figure V-4 est écrite en Java. Cette interface de programmation permet l'accès au transport à ordre partiel. Elle est composée d'un ensemble de primitives de services qui assurent l'ouverture et la fermeture des connexions ainsi que le transfert des données. Cette interface complète permet le développement d'applications au dessus du transport à ordre partiel. Elle peut être décomposée en trois groupes.

2.2.1 Primitives d'ouverture et de fermeture de connexion

L'ensemble de ces primitives est calqué sur les primitives qui existent déjà dans d'autres protocoles orientés connexion. Elles assurent l'ouverture et la fermeture de la connexion multimédia par le biais de la connexion de contrôle.

L'ouverture de connexion est dissymétrique, avec, d'un côté un serveur en attente de connexion grâce à la primitive *listen*, et de l'autre un client se connectant à l'aide de la primitive *connect*. Un paramètre particulièrement remarquable dans ces fonctions est nommé *MMPOCMN_QoS* : il représente la qualité de service demandée. La définition de la qualité de service, pour la connexion, s'effectue de manière concertée. L'émetteur propose une qualité de service (paramètre *Demande* de la primitive *connect*), celle-ci est confirmée ensuite par le récepteur (grâce au paramètre *Accepte* de la primitive *accept*). La QoS acceptée peut être différente de celle émise par le client. En cas de refus, le client peut fermer la connexion et en ouvrir une suivante avec d'autres paramètres à nouveau négociables.

La fermeture peut être négociée (*close*) ou impérative (*closeOneway*).

2.2.2 Primitives d'échange de données

Les primitives classiques *read* et *write* permettent l'échange de données monomédias. Le premier paramètre (*Flux*) de ces deux primitives représente le flux monomédia sur lequel doit être fait l'émission ou la réception des données. Les deux autres paramètres correspondent aux données, en spécifiant le contenu et la taille de ces dernières. La primitive *Available* renvoie la taille des données en attente dans le buffer de réception d'un *Flux* donné.

2.2.3 Primitives de gestion de la connexion

Les deux primitives restantes :

setInputBlockingOption, *setOutputBlockingOption* permettent une gestion plus simple des problèmes de débordement mémoire et de pertes liés au contrôle de flux. Grâce à ces options, les buffers qui ont une taille limitée peuvent bloquer les processus d'émission en cas de remplissage et le processus de réception en cas de famine. Un contrôle de débit à l'émission très simple peut être ainsi aisément implanté. Ces fonctions restent cependant orientées vers des applications simples. Si elles ne sont pas activées (cas par défaut), une gestion classique des débordements mémoire est mise en œuvre par les mécanismes d'*exception* de Java.

3 Performances de l'implémentation Java

L'implémentation du protocole MMPOC-MN a été réalisée en Java, à l'aide du service UDP. La portabilité du langage nécessaire au déploiement par réseaux actifs est une des principales raisons de ce choix. Une étude préalable portant sur les performances du langage a été réalisée afin de valider ce choix. Les résultats ainsi que les prévisions d'accroissement de la puissance des stations de travail, ont laissé penser que la position d'un protocole dans l'espace utilisateur n'est plus une hérésie. Cette constatation a ensuite été confirmée par [KRU98].

Implémenter un protocole en Java présente quelques limitations, surtout en ce qui concerne le traitement des PDUs. Les mécanismes d'encapsulation sont traditionnellement réalisés par des accès directs à la mémoire nécessitant l'utilisation de pointeurs. Pour éviter le parcours séquentiel répétitif des entêtes de paquet, un mécanisme d'accès à la mémoire a été implémenté, sur la base des travaux de [KRU98] par l'intermédiaire d'un tableau d'octets. Toutefois, l'architecture du protocole MMPOC-MN reste fidèle aux mécanismes présentés dans [FOU96]. L'indépendance des flux monomédias, permet d'implémenter des couples d'émetteur / récepteur à l'aide de Threads. Les mécanismes de retransmission peuvent ainsi être locaux du côté émetteur, et supervisés par une entité gérant les flux multimédias du côté récepteur. L'API socket UDP est utilisée comme support de transmission de toutes les PDUs (Data, Ack), puisqu'elle fournit le service minimal sur lequel l'ordre et la fiabilité partiels sont fondés.

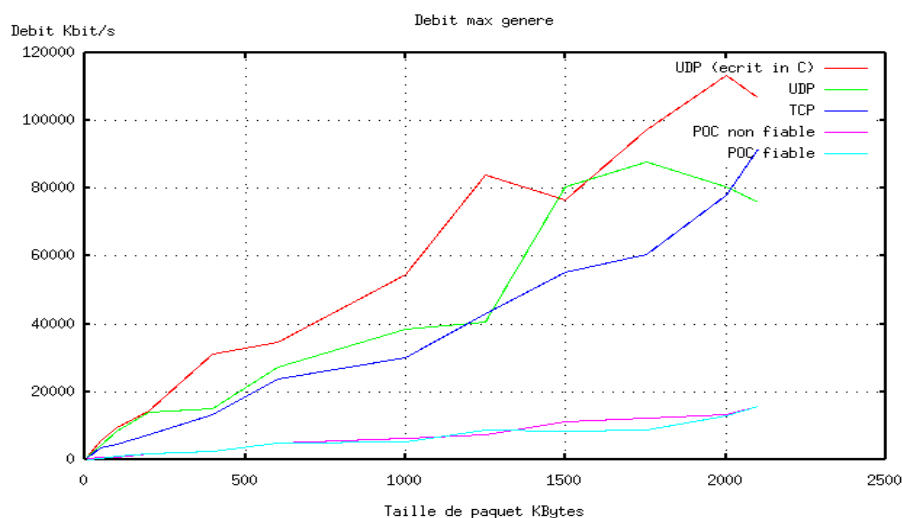


Figure V-5 - Mesures de performances du protocole MMPOC-MN

La Figure V-5 présente une comparaison entre les performances du protocole MMPOC-MN implémenté en Java et celles des services classiques fournis par Java et C. Le débit maximum généré par une application de bourrage est mesuré en fonction de la taille des paquets émis. Bien sûr, le débit nominal généré n'est pas intéressant en soi, puisque complètement dépendant de la machine sur laquelle les mesures ont été faites. Cependant le rapport de performance entre les différents services nous intéresse fortement. Afin de mesurer uniquement l'impact du système de communication, les données sont émises sur l'interface de communication locale (loopback) et par conséquent aucune donnée ne circule sur le réseau. L'environnement de test est un Pentium III 500Mhz, avec un système Linux, et une version 1.2.2 du JDK.

Les courbes présentées, montrent tout d'abord un rapport de performance global de 1 pour 6 en faveur des mesures réalisées en utilisant le service UDP du langage C. Ce rapport tend à diminuer au fil des nouvelles versions du langage. D'autre part, le débit maximum pour le protocole MMPOC-MN croît linéairement en fonction de la taille des paquets émis. Dans ce cas, tant que la taille des paquets émis se trouve en deçà de la taille maximum des PDUs, il n'y a pas de segmentation de données, ceci s'explique si le temps de traitement des PDUs est relativement indépendant de leur taille. Nous pouvons donc en conclure que la gestion des copies de données dans notre implémentation est relativement efficace. Finalement, la différence de performance entre l'utilisation d'un protocole MMPOC-MN sans ordre et avec fiabilité nulle, et celle d'un protocole MMPOC-MN totalement fiable et ordonnée, est quasiment nulle dans le cas où le réseau sous-jacent (loopback pour ces mesures) est sans perte. Ceci prouve que l'impact, en terme de temps de traitement de la gestion de l'ordre et de la fiabilité reste très léger.

4 Bilan

L'API de spécification des flux s'est avérée être d'une grande facilité d'utilisation. De nombreuses spécifications, plus ou moins complexes, ont été réalisées sans réelle difficulté. Aucune incohérence de spécification n'a été introduite par mégarde. Même si l'utilisation d'une API est moins intuitive, au premier abord, qu'un formalisme à base de réseaux de Pétri, le temps nécessaire à réaliser une spécification est relativement court. Les séquences répétitives peuvent être aisément dupliquées par la technique du « *copier/coller* ». Ceci rend d'autant plus facile le développement d'applications qui possèdent des similarités.

Le développement de l'API et des automates de reconnaissance a été rapide grâce au support formel du modèle. La génération de code a pu être mise en œuvre à partir du diagramme de classe UML. Les algorithmes des automates ont été dérivés de leur spécification sans trop de difficultés. Néanmoins, quelques incohérences du modèle découvertes lors de l'implémentation, ont pu être corrigées a posteriori.

Le bilan des performances de l'implantation en Java est en demi teinte.

D'un côté, les performances du protocole MMPOC-MN sont cohérentes avec ce que nous attendions. Des débits de plusieurs mégabits sont atteints et ceci malgré la faible efficacité de l'environnement HotJava. De telles performances sont amplement suffisantes pour une application multimédia classique.

De l'autre côté, si les performances des proxys MNP, comparables avec l'implantation MMPOC-MN, sont acceptables pour un déploiement « *classique* », les performances de la version « *active* » du protocole MNP (sur ANTS) sont très décevantes. Des débits maximaux de seulement quelques centaines de kbit/s ont pu être atteints et ceci de manière très irrégulière. Les paquets subissent de très grandes variations de leur temps de traitement sur les proxys et des pertes fréquentes sont à déplorer. Ceci est d'autant plus flagrant que le nombre

de flux à traiter augmente. Ces faibles performances sont imputables à une programmation peu optimisée d'ANTS. Pour chaque flux d'une application multimédia une machine virtuelle Java (JVM) est chargée en mémoire. Dans l'état actuel de l'implantation, nous ne pouvons pas espérer traiter plus de 4 flux par proxys. Toutefois ces résultats sont les tous premiers et de plus amples expérimentations devraient être réalisées. De nouvelles pistes nous permettent d'envisager de n'utiliser qu'une seule JVM par application, ce qui réduirait de manière conséquente les perturbations du système.

Conclusion générale

Dans ce mémoire de thèse, nous avons présenté les travaux effectués autour de la définition d'une architecture de communication conçue pour le transfert de données des applications multimédias distribuées dans un contexte de réseaux hétérogènes. La problématique que nous nous étions fixée était de concevoir un système de communication qui unifie les caractéristiques des différents systèmes de communication pour offrir la qualité de service la plus adaptée aux applications multimédias tout en préservant la synchronisation inhérente à ce type d'application.

Rappel des contributions

Du développement incessant de nouvelles technologies de communications a émergé une multitude de réseaux de communication, tous dédiés à la transmission d'un type de donnée ou à un environnement particulier. Certains offrent des faibles débits, mais une qualité de service adaptée à la transmission de flux audio. D'autres sont spécialisés dans le transport de données et permettent des transferts à hauts débits. Les solutions polyvalentes sont rares et ne sont généralement pas accessibles à moindre coût. Le déploiement d'applications multimédias distribuées est alors très dépendant des caractéristiques des réseaux utilisés et les résultats, en termes de qualité d'utilisation, sont très variables.

C'est dans ce contexte, baptisé multi-réseaux, que sont situées les contributions de nos travaux. Une architecture de communication adaptée à ce contexte a été proposée.

Architecture de communication multi-réseaux

La possibilité de posséder plusieurs interfaces d'accès à ces réseaux sur le même ordinateur ouvre de nouvelles opportunités pour les applications multimédias. L'approche que nous avons développée prône un accès multiple aux réseaux de communication pour offrir une « palette » de QoS réseau aux applications distribuées. Le protocole MMPOC-MN permet de sélectionner la meilleure interface de communication en fonction des besoins spécifiques des flux multimédias. Une application multimédia étant, par définition, potentiellement composée de plusieurs flux, ce protocole peut diriger ces flux, en fonction du service requis et sur des réseaux différents. Afin de préserver la synchronisation essentielle à la qualité d'une application multimédia, des mécanismes d'ordres partiels ont été intégrés au protocole. De plus, la qualité de présentation des flux est garantie par la gestion optimisée d'une tolérance aux pertes réseaux. Cette architecture a montré que d'une part, il était possible d'optimiser l'utilisation des réseaux sous-jacents grâce à la mise en œuvre de protocoles adaptés et que, d'autre part, dans certains cas, des garanties de qualité de service pouvaient être offertes aux applications dans des environnements où cela était jusqu'alors impensable. Il en résulte un bien meilleur comportement des applications distribuées et un plus grand confort d'utilisation nécessaire à leur utilisation à grande échelle.

Toutefois, cette problématique n'est pas la seule soulevée par le contexte multi-réseaux. L'interconnexion de domaines hétérogènes (au sens Diffserv) est un nouveau frein au développement d'architectures de communication de « nouvelle génération » et le maintien de la qualité de service inter domaines est un des challenges actuels de l'Internet. Les contributions à ce problème sont nombreuses et leurs implications sont présentes à tous les niveaux des couches protocolaires. Nous avons proposé une solution qui permet de préserver la qualité de service des protocoles de transport lors de la traversée de plusieurs domaines très

hétérogènes. Basée sur le principe de la rupture des connexions transport par des proxys situés en bordure de domaine, notre architecture tire avantageusement profit de la caractéristique multi-protocolaire de MMPOC-MN. Les performances des mécanismes de contrôle de congestion et contrôle d'erreurs sont adaptées aux réseaux traversés. Il en résulte une bien meilleure réactivité ainsi qu'une optimisation de l'utilisation des liaisons sous jacentes. De plus, la réduction conséquente des délais de recouvrement des pertes diminue considérablement les risques de rupture de la qualité de service inhérents à l'attente, par l'application de données perdues. Finalement, notre architecture préserve la sémantique de bout en bout et évite ainsi à l'application une délicate gestion des pannes.

Modèle de flux

Il était jusque là difficile d'exprimer les caractéristiques des flux émanant des applications multimédias. D'un côté, les Réseaux de Pétri étaient utilisés pour représenter des contraintes d'ordonnancement et, d'un autre côté, étaient spécifiées les tolérances aux pertes de chacun des flux. Cette vision bipartite diminuait le pouvoir d'expression des spécifications et rendait particulièrement complexe la gestion des protocoles correspondants. Nous avons développé un modèle de flux associant un langage de spécification des flux multimédias à des mécanismes de traitement de ces flux par le protocole. Le langage permet la description de la qualité de service requise par un flux multimédia en termes de fiabilité et de synchronisation. Les traitements sont exprimés sous la forme d'automates à intégrer dans le protocole. Ces automates permettent de garantir les contraintes spécifiées par le langage. Ce modèle s'est avéré plus expressif et plus compact que celui des Réseaux de Pétri. Basé sur un principe de composition des spécifications ce modèle est en complète adéquation avec le développement d'applications complexes par agrégation d'outils. La gestion, par le protocole, de ces flux ainsi modélisés est simplifiée grâce à l'apport fourni par le modèle d'automates.

Déploiement dynamique de protocoles de communication

Cette thèse a fortement contribué à développer des mécanismes de déploiement dynamique de protocoles de transports. Que ce soit pour le protocole multi-réseaux parallèle ou série, ces principes ont été étudiés et mis en œuvre. Le navigateur Web HotJava, en partie prévu à cet effet, a tout d'abord été utilisé pour la « version » parallèle du protocole. Une nouvelle façon de concevoir les protocoles de transport a ensuite été élaborée et appliquée au développement du protocole multi-réseaux série pour assurer son déploiement sur un réseau actif. Le déploiement dynamique de protocole de transport est une approche novatrice qui ouvre la voie à une ère nouvelle de protocoles adaptés à la fois aux applications et aux réseaux supports des architectures distribuées.

Perspectives

Une évolution du transport prenant en compte la renégociation de Qualité de Service doit également être proposée. Actuellement la QoS est fixée à l'ouverture de la connexion et reste la même jusqu'à la fermeture. La possibilité de modifier la QoS dynamiquement, c'est à dire durant un fonctionnement normal sans altérer la continuité du flux, doit être offerte à l'utilisateur. Deux raisons principales peuvent motiver un changement de QoS : (a) le changement de Qualité de Service offerte par le (ou les) réseaux utilisés, voire même la perte de connectivité sur une des interfaces ; (b) la volonté de l'utilisateur de modifier les paramètres d'une connexion dont il n'est pas entièrement satisfait. Pour cela nous envisageons, en réponse à ces deux cas, deux évolutions possibles de l'architecture : (a) offrir

la possibilité de changer d'interfaces réseau en cours de communication ; (b) intégrer dans le modèle de flux le changement de QoS. Un opérateur de *séquence* pourrait être ajouté aux *flux élémentaires* et aux *flux synchronisés* pour modéliser ces changements. Du point de vue de l'architecture, une telle renégociation de QoS impliquerait de modifier « à chaud » le code déployé sur les proxys MNP pour garantir la nouvelle spécification.

Le déploiement de protocoles de communication sur les réseaux actifs a été abordé tout au long de cette thèse. Nous avons précisé que, pour ce besoin particulier, les plates-formes actuelles n'étaient pas entièrement satisfaisantes. Nous nous orientons maintenant vers des mécanismes plus proches de ceux de téléchargement de code que de paquets actifs. La définition de nouvelles plates-formes de réseaux programmables est au centre de nos préoccupations actuelles. La plate-forme développée dans cet esprit par 6WIND au sein du projet GCAP [LAR01] doit être testée très prochainement.

Enfin, nous avons délibérément choisi de ne pas aborder la combinaison des architectures de réseaux parallèle et série, car nous n'avons, à ce moment là, pas de scénarios à proposer. Du fait de l'avancement des travaux, cet aspect de l'architecture pourra désormais être étudié dès que la plate-forme d'expérimentation multi-réseaux sera mise en place. De plus, le comportement du protocole MNP dans un environnement réseau congestionné pourra être évalué. En effet l'impact du contrôle de congestion de MNP sur les flux TCP n'a pas encore été mesuré et de nouvelles techniques pourraient être développées.

Bibliographie de l'auteur

Revue

- [BER01C] P.BERTHOU, T.GAYRAUD, P.OWEZARSKI, M.DIAZ, « Multimedia Multi-networking: a new concept », *Annales des télécommunications*, 24p, à paraître Février-Mars 2002.

Conférences avec actes

- [BER99A] P.BERTHOU, T.GAYRAUD, P.OWEZARSKI, « Partial Ordered and Reliable Transport Multimedia Protocol for Satellite Communications », 5th European Conference on Satellite Communications, Toulouse, November 3, 4 & 5, 1999.
- [BER00A] P.BERTHOU, T.GAYRAUD, P.OWEZARSKI, M.DIAZ, « Protocole de Communication Multimédia Multi-réseaux : un nouveau concept », 8è Colloque Francophone sur l'Ingénierie des Protocoles, Toulouse, October 2000.
- [GAY00] T.GAYRAUD, P.BERTHOU, P.OWEZARSKI, M.DIAZ, « M3POC : A Multimedia Multicast Transport Protocol for Cooperative Applications », ICME2000, New York City, July 30, August 2, 2000.

Rapports de contrats

- [BER00B] P.BERTHOU, T.GAYRAUD, M.ERRECART, K.L.THAI, P.SPATHEIS, « Etude préalable des fonctions à déporter et des contraintes sur l'architecture », Rapport LAAS No00488, Projet AMARRAGE RNRT'99 N°41, Novembre 2000, 49p.
- [BER00C] P.BERTHOU, T.GAYRAUD, « Spécification du protocole multimédia MMPOC », Rapport LAAS No00489, Projet AMARRAGE RNRT'99 N°41, Novembre 2000, 25p.
- [BER01A] P.BERTHOU, M.DIAZ, E.EXPOSITO, T.GAYRAUD, S.OWEZARSKI, P.SENAC, L.WOLF, « Specification of a multicast multimedia protocol », Rapport LAAS No01186, Project IST-1999-10 504 GCAP, Janvier 2001, 35p.
- [BER01D] P.BERTHOU, T.GAYRAUD, P.OWEZARSKI, L.PLATEAUX, M.DIAZ, D.GENTY, « Specification of the MC MM multinet network protocol », Rapport LAAS No01351, Project IST-1999-10 504 GCAP, Juin 2001, 43p.

Rapports LAAS

- [BER00D] P.BERTHOU, E.EXPOSITO, J.FANCHON, « A model for partially ordered and partially reliable connections », Rapport LAAS N°00456, Novembre 2000, 24p.
- [BER01B] P.BERTHOU, J.FANCHON, « A model and implementation of partially ordered and partially reliable connections », Rapport LAAS N°01128, Mars 2001, 17p.

Bibliographie

- [ALE97] D.S. ALEXANDER, B. BRADEN, C.A. GUNTER, and al. « ANEP : Active Network Encapsulation Protocol », <http://www.cis.upenn.edu/~switchware/ANEP/>, July 1997.
- [ALE98A] D.S. ALEXANDER, W.A. ARBAUGH, P. HICKS, and al., « The SwitchWare Active Network Architecture ». IEEE Network Special Issue on Active and Controllable Networks, 12(3):29-36, July 1998.
- [ALE98B] D.S. ALEXANDER, W.A. ARBAUGH, A.D. KEROMYTIS and al., « A Secure Active Network Environment Architecture : Realization in SwitchWare », IEEE Network, 12(3):37-45, May 1998.
- [ALE98C] D.S. Alexander, « ALIEN : A Generalized Computing Model of Active Networks. », PhD thesis, University of Pennsylvania, December 1998.
- [ALE99] D.S. ALEXANDER, W.M. ARBAUGH, A.D. KEROMYTIS and al., « Security in Active Networks. In Secure Internet Programming: Issues in Distributed and Mobile Object Systems. », Springer Verlag, LNCS State of the Art Series, 1999.
- [ALL98] M. ALLMAN, D. GLOVER, J. GRINER, « Ongoing TCP Research Related to Satellites », May 1998, Internet-Draft draft-ietf-tcpsat-res-issues-03.txt
- [ALM98] K.C. ALMEROOTH, Y. ZHANG, « Using Satellite Links as Delivery Paths in Multicast Backbone (MBone) », Third International Workshop on Satellite-based Information Services (WOSBIS 98), Dallas, Texas, Oct 1998.
- [AMA99] « Architecture Multimédia & Administration Réparties sur un Réseau Actif à Grande Echelle », Projet RNRT-1999-41, Amarrage, (<http://www.amarrage.org>)
- [AME93] P. AMER, C. CHASSOT, M. DIAZ, « Partial order quality of service to support multimedia connections. Reliable channels », 2nd International Symposium on High Performance Distributed computing, Spokane, Etats Unis, juillet 1993.
- [AME94] P.D. AMER, C. CHASSOT, T. CONNOLLY, P. CONRAD, M. DIAZ, « Partial order transport service for multimedia and other applications », *IEEE/ACM transactions on Networking*, Vol. 2, No. 5, 1994.
- [ARD98] S. ARDON, R. DE SILVA, R. LANDFELDT, AND A. SENEVIRATNE. « *Total management of transmissions for the end user: a framework for user control of application behaviour.* » In Proceedings of HIPPARCH'98 Workshop, pages 189-205, London, June 1998.
- [BAL95] H. BALAKRISHNAN, S. SESHAN, E. AMIR, and al., « Improving TCP/IP Performance over Wireless Networks Networks, Mobicom », November 1995.
- [BAL99] H. BALAKRISHNAN, V. PADMANABHAN, R. KATZ, « The effect of asymmetry on TCP performance », 3^{ième} ACM/IEEE mobicom, sept. 1999.
- [BAR94] A. BARKE, B.R. BADRINATH, « I-TCP: Indirect TCP for Mobile Hosts », Technical Report DCS-TR-314, Rutgers University, October 1994.

- [BES94] L. BESSE, L. DAIRAINÉ, L. FEDAOUI, W. TAWBI, et K. THAI, « Towards an Architecture for Distributed Multimedia Application Support », Proc. International Conference Multimedia Computing Systems, Boston, Mai 1994.
- [BHA98] S. BHATTACHARJEE, K. L. CALVERT, E. W. ZEGURA, « Active Networking and End-to-End Argument », IEEE Network, May/June 1998.
- [BLA98] S. BLAKE, D. BLACK, M. CARLSON, and al., « An Architecture for Differentiated Services », RFC 2475, December 1998.
- [BOR99] J. BORDER, M. KOJO, J. GRINER, G. MONTENEGRO, « Performance Enhancing Proxies », Internet draft : draft-ietf-pilc-pep-00.txt.
- [BOR01] J. BORDER, M. KOJO, J. GRINER, G. MONTENEGRO, « Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations », IETF draft of the TCPSAT Working Group PILC : draft-ietf-pilc-pep-07.txt.
- [CAL98] K.L. CALVERT, S. BHATTACHARJEE, E. ZEGURA and al., « Directions in Active Networks. », IEEE Communications Magazine , October 1998.
- [CAM94] A. CAMPBELL, G. COULSON, et D. HUTCHINSON, « A Quality of Service Architecture », ACM Computer Communication Review », Avril 1994.
- [CHA95] C. Chassot, « Architecture de transport multimédia à connexion d'ordre partiel », Thèse de doctorat de l'institut National Polytechnique de Toulouse (INPT), Décembre 1995.
- [COM98] H.COMON,Y.JURSKI, « *Multiple counters automata, safety analysis and Presburger arithmetic.* », CAV'98. LNCS vol. 1427,(1998) 268-279.
- [DAL00] S. D'ALU, G. CHELLIUS, I. CHRISMENT, O. FESTOR, E. FLEURY « Intégration du support IPv6 dans l'environnement de supervision de réseaux ANAIS », CFIP'2000, octobre, 2000 Toulouse, France.
- [DAS] S. DA SILVA, Y. YEMINI, « Netscript A language and environment for programmable networks », <http://www.cs.columbia.edu/dcc/netscript>.
- [DIA01] M. Diaz, R. Canonico, L. Costa, « GCAP : A New Multimedia Multicast Architecture for QoS », Proc. 6th International Conference on Protocols for Multimedia Systems (PROMS 2001), Springer LNCS, October 17 - 19, Enschede, The Netherlands, 2001.
- [DIA93] M. DIAZ, P. SENAC, « Time stream Petri nets : a model for multimedia streams synchronization », 1st International Conference on Multimedia Modeling (MMM'93), Singapore, novembre 1993, pp. 257-273.
- [DIA94] M. DIAZ, A. LOZES, C. CHASSOT, and al., « Partial order connections : a new concept for high speed and multimedia services and protocols », Annals of telecommunications, vol. 49, n°5-6, 1994.
- [DIA95] M. DIAZ, K. DRIRA, A. LOZES, C. CHASSOT, «Definition and representation of the quality of service for multimedia systems», *6th International conference on high speed Networking, HPN'95*, Palma de Mallorca, Spain, September 11-15, 1995.
- [DRO96] M. DROSTE, P.GASTIN. « *Asynchronous Cellular automata for Pomsets without autoconcurrency.* », CONCUR 96, Lecture Notes in Computer Science.
- [DRO00] M. DROSTE, P.GASTIN, D. Kuske, « *Asynchronous Cellular automata for Pomsets* », Theoretical Computer Science Vol. 247, 1-2, pp. 1-38, September 2000.

- [FAN99A] J. FANCHON, « *Trace Channel Nets.* » ICATPN 99. Lecture Notes in Computer Science, Vol. 1639. Springer-Verlag, 1999.
- [FAN00A] J.F. FANCHON, P. BERTHOU, E. EXPOSITO, « *Model for Partially Ordered and Partially Reliable Connections.* » LAAS research report 00456 , november 2000
- [FES00] O. FESTOR, I. CHRISMENT, E. FLEURY, « Les réseaux programmables », Rapport de Recherche INRIA No. 3913, Mars 2000.
- [FOU96] M. FOURNIER, C. CHASSOT, A. LOZES, M. DIAZ, «Multimedia partial order transport architecture : Design and implementation. Protocols for High Speed Network», (*PfHSN96*), 1996.
- [FOU97] C. FOURNIER, «Réalisation et évaluation de performances d'un protocole de transport multimédia à ordre partiel », Thèse de doctorat de l'Université Paul Sabatier de Toulouse, Mars 1997.
- [GAY00] T.GAYRAUD, P.BERTHOU, P. OWEZARSKI, et al., «M3POC : A Multimedia Multicast Transport Protocol for Cooperative Applications », *Proc. ICME2000, New York City, August 2000.*
- [GCAP99] « Global Communication Architecture and Protocols for new QoS services over IPv6 networks », Project IST-1999-10 504 GCAP, (<http://www.laas.fr/GCAP>)
- [GRA81] J. GRABOWSKI, «*On partial Languages.*», *Fundamenta Informaticae* IV.2. (1981).
- [HAR96] J. HARTMAN, UDI MANBER, L. PETERSON, T. PROEBSTING, «Liquid Software: A New Paradigm for Networked Systems», *TR96-11, University of Arizona Department of Computer Science*, 1996. (<http://www.cs.rice.edu/~adve/CS615/liquidsw.tr96-11.ps>)
- [HEN97] T.R. HENDERSON, R.H. KATZ, « Satellite Transport Protocol (STP): An SSCOP-based Transport Protocol for Datagram Satellite Networks », WOSBIS'97, Budapest, Hungary, October 1997.
- [HEY91] A.T. HEYBEY, « Video Coding and the Application Level Framing Protocol Architecture », Massachusetts Institute of Technology, 1991.
- [HIC98] M. HICKS, P. KAKKAR, J.T. MOORE, and al., « PLAN : A Packet Language for Active Networks. », In Proceedings of the Third ACM SIGPLAN International Conference on Functional Programming Languages, 1998.
- [HOR00] E. Horlait, M. Bouyer, G. Harmel et al. « BMRP: Bandwidth Management and Reservation Protocol », Internet Draft, draft-horlait-clep-01.txt, mars 2000.
- [JAV98] JavaSoft, Sun Microsystems, Inc, « HotJava™ Browser : A White Paper ». 1998 (<http://sunsite.nus.sg/hotjava/1.0alpha3/doc/overview/hotjava/index.html>)
- [JAV99] JavaSoft, Sun Microsystems, Inc, « Jini™ Technology Architectural Overview ». January 1999 (<http://www.sun.com/jini/whitepapers/architecture.html>)
- [JIN01] G. Jin, G. Yang, B. Crowley, D. Agarwal, « Network Characterization Service (NCS) », Proceedings of the 10th IEEE Symposium on High Performance Distributed Computing HPDC-10, August 2001, LBNL-47892.
- [KRU98] B. KRUPCZAK, K.L. CALVERT, M. AMMAR, « Implementing Communication Protocols in Java », IEEE Communications Magazine, Octobre 1998

- [KUS98] D. KUSKE, « *Asynchronous Cellular and Asynchronous Automata for Pomsets.* » Lecture Notes in Computer Science, Vol 1466, CONCUR 98, 517-523, Springer-Verlag 1998.
- [LAR01] D. LARRABEITI, M. CALDERÓN, J.E. KRISTENSEN, F. PHAM-KHA, « Design of the Active Protocol entities for the IPv6 support of multinet ABN », Project IST-1999-10 504 GCAP, December 2000.
- [LAN98] B. LANDFELDT A. SENEVIRATNE, «User Service Assistant(USA): An End-to-End Reactive QoS Architecture » *Sixth International Workshop on Quality of Service*, 1998.
- [LAN98] B. LANDFELDT, A. SENEVIRATNE, C. DIOT, «User Services Assistant: an End-to-End Reactive QoS Architecture» , IWQOS98 Napa, California May 1998.
- [MCC00] S. McCREARY, K. CLAFFY, «Trends in wide area IP traffic patterns», ITC seminar on IP traffic measurement, modelling and management, Monterey, CA, September 18th - 20th, 2000.
- [MIG01] J.C. MIGNOT, « Cache web : un état de l'art des techniques et prototypes. », *TSI*, 20(6):719–748, 2001.
- [LEC01] V. LECUIRE, F. LEPAGE, K. KAMMOUN, « Enhancing Quality of MPEG Video through Partially Reliable Transport Service in Interactive Application », 4th IFIP/IEEE International Conference on Management of Multimedia Networks and Services, MNS 2001, Chicago, IL, USA October 29 - November 1, 2001, Proceedings LNCS 2216, p. 96 ff.
- [LIU01] M. LIU, N. EHSAN, « Modeling TCP Performance with Proxies », *Rapport Interne, Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor*.
- [MPE93] Comité Consultatif International Télégraphique et Téléphonique (CCITT) « CCITT Recommendation MPEG-1, Coded Recommendation of Picture, Audio and Multimédia / Hypermedia Information », *ISO/IEC 111172*, Genève, Suisse, 1993.
- [NAH95] K. NAHRSTEDT ET J.M. SMITH, « The QoS Broker », *IEEE Multimedia*, Vol. 12(1), Spring, 1995.
- [OWE96] P. OWEZARSKI, « Conception et formalisation d'une application de visionconférence coopérative. Application et extension pour la téléformation », Thèse de doctorat de l'Université Paul Sabatier de Toulouse, Décembre 1996.
- [OWE97] P. OWEZARSKI, D. WARTELLE, P. PERROT, G. SEGARRA, S. GUILLOUET, K. DRIRA, A. MEFTAH, M. DIAZ, «Assessment methodology of new technologies for collaborative automotive design», *2nd International distributed conference on network interoperability*, Madeire, Portugal, 16-18 Juin, 1997
- [OWE98] P. OWEZARSKI, M. DIAZ, C. CHASSOT, «A Time Efficient Architecture for Multimedia Applications», *IEEE Journal on Selected Areas in Communication, Special Issue on Protocols and Architectures for Applications of the 21st Century*, 1998.
- [PAR97] C. PARTRIDGE, T.J. SHEPARD, « Performance of TCP/IP over satellite. », October 1997.
- [PER96] C. PERKINS, «IP Mobility Support», RFC 2002, October 1996.

- [RFC793] POSTEL J, « Transmission Control Protocol », *RFC793*, Defense Advanced Research Projects Agency, 1981.
- [RFC1958] B. CARPENTER. « Architectural principles of the internet. », RFC 1958, IETF, June 1996.
- [RFC2330] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, « *Framework for IP Performance Metrics.* », IETF, RFC 2330 (Informational), May 1998.
- [RFC2488] M. ALLMAN, D. GLOVER, L. SANCHEZ. « Enhancing TCP Over Satellite Channels. », IETF, RFC 2488, Janvier 1999.
- [RFC2678] J. Mahdavi, V. Paxson, « *IPPM Metrics for Measuring Connectivity.* », RFC 2678, IETF, Sept. 1999
- [RFC2679] G. Almes, S. Kalidindi, and M. Zekauskas. « *A One-Way Delay Metric for IPPM.* », RFC 2679, available from <http://www.ietf.org/rfc>, September 1999.
- [RFC2680] G. Almes, S. Kalidindi, and M. Zakeuskas, « *A one-way packet loss metric for IPPM* », RFC 2680, IETF, Sept. 1999.
- [RFC2681] G. Almes, S. Kalidindi, and M. Zakeuskas, « *A Round-trip Delay Metric for IPPM* », RFC 2681, IETF, Sept. 1999.
- [RFC3155] S. Dawkins, G. Montenegro, M. Kojo et al. , « End-to-end Performance Implications of Links with Errors », RFC 2681, IETF, August 2001.
- [RIZ97] L. RIZZO, « Dummynet: a simple approach to the evaluation of network protocols », *ACM Computer Communication Review*, v. 27, n.1, Jan. 1997. (http://info.iet.unipi.it/~luigi/ip_dummynet)
- [ROJ98A] L. ROJAS-CARDENAS, E. CHAPUT, L. DAIRAINÉ, and al., « MPEG Video Transport over Partial Order Connections », International conference on Multimedia and Telecommunication Management, Hong Kong, December 1998.
- [ROJ98B] L. ROJAS-CARDENAS, E. CHAPUT, L. DAIRAINÉ, P. SENAC, M. DIAZ, « Video Transport Over Partial Order Connections », Fourth International Workshop on High Performance Protocol Architectures (HIPPARCH'98), Londres, juin 1998.
- [ROJ00] L. ROJAS-CARDENAS, « *Architecture de Transport à Ordre et Fiabilité Partiels pour les Applications Multimédias Adaptatives à Temps Contraint.* », Thèse de Doctorat, Ecole Nationale Supérieure d'Ingénieurs de Constructions Aéronautiques, Novembre 2000.
- [SEN94] P. SENAC, M. DIAZ, P. DE SAQUI-SANNES, et al., « Toward a formal specification of multimedia synchronization », *Annals of telecommunication*, May / June 1994.
- [SEN96] P. SENAC, M. DIAZ, P. DE SAQUI-SANNES, et al., « Modelling Logical and Temporal Synchronisation in Hypermedia Systems », *IEEE Journal on Selected Area in Communication*, special issues on multimedia synchronization, vol. 14, n°1, janvier 1996, pp. 84-103.
- [SCH99] B. SCHWARTZ A.W. JACKSON, W.T. STRAYER, and al., « Smart Packets for Active Networks », *OpenArch*, Mars 1999 (<ftp://ftp.bbn.com/pub/AIR/smart.ps>)
- [SMI97] J. SMITH, D. FARBER, C.A. GUNTER, and al., « SwitchWare : Towards a 21st Century Network Infrastructure. », <http://www.cis.upenn.edu/switchware/papers/sware.ps>, 1997.

- [TEN96] D.L. TENNENHOUSE, D.J. WETHERALL, « Towards an Active Network Architecture » *Computer Communication Review*, April 1996.
- [TEN97] D.L.TENNENHOUSE, D. SINCOSKIE, D.J. WETHERALL, « A survey of Active Network Research », *IEEE Communications Magazine*, vol. 35, No. 1, pp 80-86. Janvier 1997.
- [VEG96] A. VEGA GARCIA, « Mécanismes de contrôle pour la transmission de l'audio sur l'Internet », Thèse de doctorat de l'université de nice-sophia antipolis, octobre 1996.
- [VIL98] T.VILLEMUR , V.BAUDIN , S.OWEZARSKI , M.DIAZ, «An integrated platform for cooperative teleteaching», *Lecture Notes in Computer Science 1483*, Eds. T.Plagemann, V.Goebel, 1998, Springer, ISBN 3-540-64955-7, pp.59-70
- [WAP] Wireless Application Protocol, White paper, Août 2001, http://www.wapforum.com/what/WAPWhite_Paper1.pdf
- [WET] D.J. WETHERALL « Developing Network Protocols With the ANTS Toolkit », MIT
- [WET96] D.J. WETHERALL, D.L. TENNENHOUSE « The Active IP Option. », In Proceedings of ACM SIGOPS European Workshop '96, Connemara, Ireland, September 1996.
- [WET98] D.J. WETHERALL, J. GUTTAG, D. L. TENNENHOUSE, « ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols », *IEEE OPENARCH'98*, San Francisco, CA, April 1998.
- [WET99] D.J. WETHERALL, J. GUTTAG, D. TENNENHOUSE, « ANTS: Network Services Without the Red Tape », *IEEE Computer*, April 1999.
- [WOL00] T. WOLF, D. DECASPER, C. TSCHUDIN FWOLF., « SAPF Tags for High Performance Active Networks », <http://www.docs.uu.se/~tschudin/pub/cft-2000-tan.pdf>, 2000.