
Protocole de Communication Multimédia Multi-réseaux :

Un nouveau Concept

Pascal Berthou¹, Thierry Gayraud^{1,2}, Philippe Owezarski¹,
Michel Diaz¹

¹ LAAS-CNRS

7, Avenue du Colonel Roche
31077, Toulouse cedex 4, France

² Université Paul Sabatier

118, Rte de Narbonne
31062 Toulouse cedex 4, France

e-mail : {berthou, gayraud, owe, diaz}@laas.fr

RÉSUMÉ. Les applications multimédias distribuées nécessitent des Qualités de Service (QoS) différentes suivant le type de média utilisé et les catégories de réseau de communication disponibles, chacun d'eux fournissant des qualités de Service variées. Cet article présente un nouveau protocole MMPOC-MN ayant pour objectif de combiner les deux aspects énoncés précédemment. Ce protocole repose sur les concepts d'ordre et de fiabilité partiels et a été spécialement conçu pour supporter des flux de données multimédias. Il permet de tirer parti des réseaux de communication disponibles en accordant la QoS requise par les flux multimédias à celle fournie par ces réseaux. Après avoir décrit l'architecture générale de MMPOC-MN, cet article s'intéressera au problème de la mise en œuvre et du déploiement facile de ce protocole grâce au concept des « réseaux actifs ». L'implémentation et des mesures de performances de ce protocole seront également présentées dans cet article.

ABSTRACT. As on one side current distributed multimedia applications require various Quality of Services (QoS) depending on the media types they have to transmit, and on the other side, many networks are available, each providing different types of QoS. This paper presents a new protocol called MMPOC-MN that aims to combine the two previous aspects. This protocol based on partial order and reliability has been especially designed to handle multimedia data streams, and proposes to take advantage of the underlying available networks according to the QoS media streams required and the one provided by networks. The architecture of such a protocol is described in this paper. On the other side, this paper also addresses the problem of easy protocol prototyping and deployment, and because of that it takes under consideration the new active networking mechanisms. This paper then presents the MMPOC-MN implementation and some performance measurements.

MOTS-CLÉS : Multi-réseaux, MMPOC-MN, Transport, Java, Réseaux actifs

KEY WORDS: Multi-network, MMPOC-MN, Transport, Java, Actives networks

1. Introduction

Le sujet abordé dans cet article concerne l'architecture d'un nouveau protocole de communication pour des applications multimédias distribuées. Aujourd'hui il existe plusieurs types de réseau de communication (ATM, mobile, RNIS, satellite, etc.), chacun d'entre eux fournissant des services présentant des Qualités différentes. De plus, les applications actuelles utilisent des flux de données ayant des exigences différentes en matière de Qualités de Service et peuvent bénéficier de services fournis par tout ou partie des réseaux disponibles. Par exemple, une application de Vidéo-à-la-Demande (VoD) peut profiter d'un réseau RNIS pour la partie signalisation de ses communications et d'un réseau satellite pour la partie audio et vidéo. Notre approche a pour objectif de fixer le choix à effectuer en faisant correspondre, autant que faire se peut par rapport aux paramètres de QoS considérés, la QoS fournie par les divers réseaux accessibles et la QoS requise par les applications multimédias distribuées concernées. Ces applications doivent alors transmettre leurs flux multimédias en accord avec la QoS requise. Les protocoles Internet bien connus TCP et UDP ne sont pas adaptés aux données multimédias (telles que l'audio et la vidéo par exemple), puisqu'ils fournissent des services respectivement soit totalement ordonnés et fiables, soit ni fiables ni ordonnés, alors que les données multimédias nécessitent des services partiellement fiables et ordonnés. Par contre, le protocole MMPOC (MultiMedia Partial Order Connection) repose, lui, sur des mécanismes à ordre et fiabilité partiels pour améliorer les performances des communications multimédias. Cette amélioration a été précédemment démontrée dans [DIA95].

De plus, MMPOC (conçu initialement pour des communications multimédias sur un réseau ATM) repose sur un principe de séparation des médias permettant de tirer parti des classes de service proposées par ATM (VBR, CBR, etc.). Dans le cas où plusieurs réseaux de communication sont accessibles, MMPOC peut être étendu pour tenir compte des possibilités offertes par chacun de ces réseaux par rapport aux besoins en QoS de chacun des médias à transmettre. Le nouveau protocole ainsi obtenu, adapté au concept de communication "multi-réseaux" est appelé MMPOC-MN (Multi-Networking).

En ce qui concerne le développement de MMPOC-MN, et à cause des difficultés rencontrées pour le déploiement d'un nouveau protocole, de nouvelles techniques ont été étudiées, notamment celles basées sur le concept de « réseau actif ». L'objectif de déploiement visé peut être atteint avec des mécanismes de type « Code Mobile », et avec comme support soit HotJava [JAV98] – navigateur Web proposé par Sun – soit une plate-forme type ANTS [WET98] – du MIT – comme dans le projet RNRT Amarrage¹. Une implémentation réalisée en Java et ce principe des « Codes Mobiles » permettent de faciliter la mise en œuvre et le déploiement des nouveaux protocoles.

Cet article est structuré de la manière suivante : La première partie présente la problématique liée aux communications multimédias et à leur QoS, ainsi que la façon dont on peut tirer parti des différents services de communication offerts par

¹ Amarrage : Architecture Multimédia & Administration Réparties sur un Réseau Actif à Grande Echelle (Projet RNRT, Labellisé en 1999)

différents réseaux accessibles (partie 2). La partie suivante (partie 3) présente l'architecture et le protocole pour un service de bout en bout dans le cas de communications multimédias "multi-réseaux": MMPOC-MN. Deux points seront particulièrement étudiés : (1) le protocole de transport MMPOC, solution pour des communications multimédias et (2) comment MMPOC-MN tire parti de l'accès à des réseaux de communication multiples. La partie 4 présente alors comment l'approche « réseau actif » peut apporter de nouvelles facilités d'implémentation, soit avec HotJava, soit avec ANTS. La partie 5 décrit l'implémentation de MMPOC et de MMPOC-MN, présente tous les mécanismes et principes, puis donne les mesures de performances pertinentes pour ces protocoles. Ces mesures ont été réalisées à l'aide d'une application de type VoD, bénéficiant d'un réseau ATM pour l'audio et d'un service « best effort » sur un réseau type IP pour la vidéo. Quelques travaux futurs sont présentés dans la partie 6 surtout sur la généralisation de MMPOC-MN, et pour finir, la partie 7 conclut cet article.

2. La problématique du Multi-réseaux

Les applications Multimédias requièrent différents types de QoS en fonction des types de données utilisés. Cette QoS comporte divers paramètres tels que la fiabilité, le débit, le délai de bout en bout... Par exemple, supposons qu'une application de visioconférence ait 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. Il a de plus 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. 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, GSM/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 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 1 montre que le flux audio utilise une liaison RNIS, la vidéo un lien satellite ATM tandis que la connexion de signalisation utilise Internet : MMPOC-MN propose 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 MMPOC-MN est la re-synchronisation des 3 flux, envoyés sur des réseaux différents tant physiquement que logiquement. On parlera dans ce cas de « multi-réseaux parallèle ».

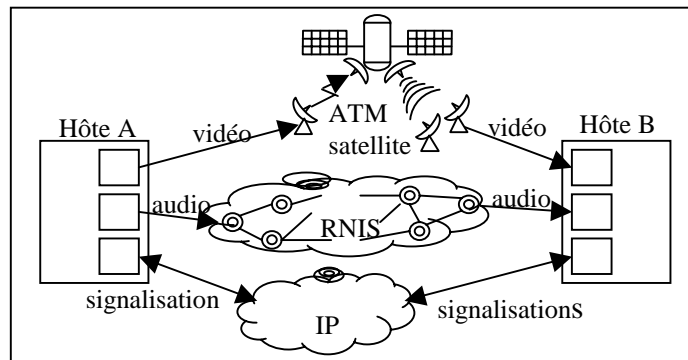


Figure 1. *Multi-réseaux parallèle*

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 intranet MAN (Metropolitan Area Network) terrestre (Figure 2), les implémentations et/ou configurations de TCP

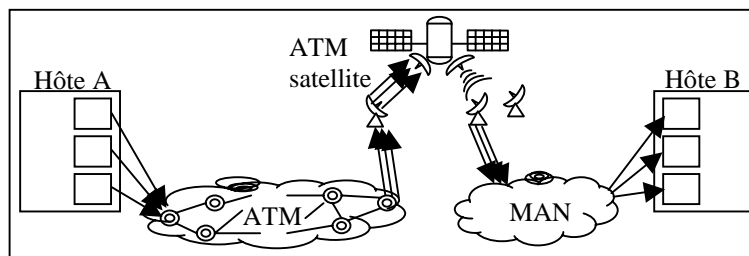


Figure 2. *Multi-réseaux série*

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 ».

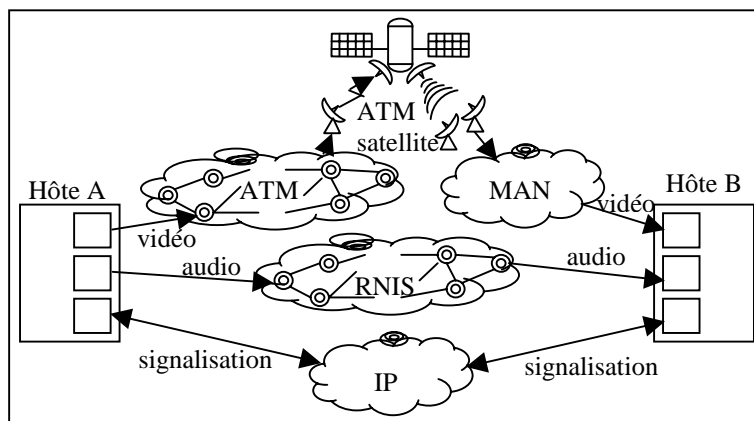


Figure 3. *Multi-réseaux parallèle et série*

Bien sûr, l'application de VoD peut aussi traverser des compositions série et parallèle de différents réseaux comme sur la Figure 3. 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.

Vues les difficultés de ces deux dernières approches, cet article se focalisera sur le multi-réseaux parallèle, la partie série devant encore être approfondie dans des travaux futurs.

3. Description de l'architecture et du protocole MMPOC-MN

3.1 *Un protocole de transport 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 4 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 des pertes à l'intérieur d'une séquence, un nombre maximum des pertes consécutives... 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 fiabilité partielle exigée, définie dans le service de transport à ordre partiel et ayant pour résultat des indications de perte et de délivrances au plus tôt, se déduit des besoins de l'application [OWE98].

En fait, deux approches existent pour contrôler la fiabilité partielle : média par média ou par groupe de médias. 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.

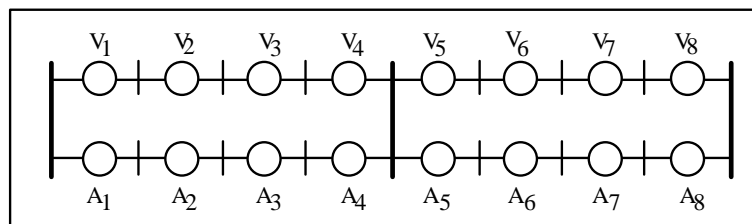


Figure 4. Exemple d'ordre partiel

Considérons la Figure 4 qui représente un réseau de Pétri de composition série/parallèle d'une connexion multimédia (son et vidéo d'une application de visioconférence par exemple). Supposons que le nombre maximum de pertes autorisées sur chaque flux pendant une période de synchronisation (V1-V4, A1-A4) est de un. Supposons également que les données V1, V2, A1, A2, et A3 ont été reçues par l'entité de réception et délivrées à l'application.

Puis supposons que l'entité distante reçoive V4 ; le respect fort de l'ordre partiel sur la connexion vidéo indique que V4 ne peut pas être délivrée à l'application : c'est

parce qu'il est logiquement localisé après V3 et que V3 n'a pas encore été reçu. Maintenant, V4 pourrait être délivrée si V3 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 V4 dès que possible, V3 sera déclarée comme perdue, V4 étant alors délivrée. C'est le principe des pertes et des délivrances au plus tôt.

Maintenant supposons que V5 soit reçue. Concernant l'ordre partiel, cet objet ne peut pas être délivré, parce qu'il est logiquement après A4 (en raison de la synchronisation inter-flux après A4 et V4).

Avec une gestion média par média de l'ordre partiel, V5 doit être stockée jusqu'à ce qu'A4 soit reçue ou déclarée perdue.

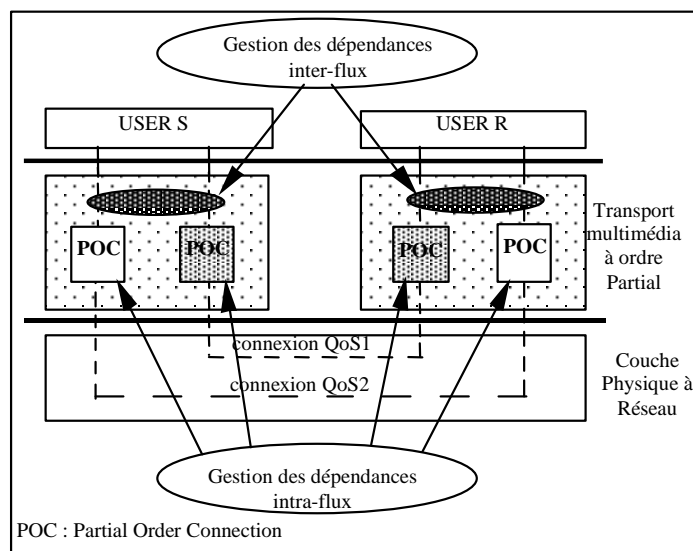


Figure 5. Architecture d'un transport à ordre partiel

Cependant, 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 5, qui montre également la gestion de multi-connexions [OWE98].

3.2 *Multi-réseaux*

3.2.1 *Précédents travaux*

Aujourd'hui peu d'architectures offrent de manière intégrée la possibilité d'utiliser des réseaux différents. Néanmoins, ce concept apparaît dans des applications spécifiques, comme par exemple certains environnements de formation linguistique. Des liaisons analogiques sont utilisées pour l'audio et la vidéo, alors que des liaisons numériques servent à l'échange des données et au contrôle de session. Toutefois, ces environnements sont limités à des environnements locaux, et sur de courtes distances. Ce principe est aussi présent dans le domaine des communications mobiles. En effet, certains travaux de recherche portent d'une part sur le basculement automatique de réseau, dans les cas où celui utilisé serait temporairement inaccessible [STE98], et d'autre part sur la sélection manuelle par l'utilisateur de l'interface réseau sur laquelle il veut émettre ses paquets [INO97][LAN98]. La qualité de service étant généralement restreinte à sa plus simple expression (niveau du signal reçu, coût, etc.), et l'usage restreint à une seule interface réseau à la fois, ces systèmes s'avèrent être trop peu flexibles pour être facilement utilisables.

Plus généralement, à cause des nombreuses API réseaux existantes, le développement d'une application qui tire parti des caractéristiques spécifiques des réseaux sous-jacents, conduit à rendre l'application complètement dépendante du système et par conséquent difficilement portable.

Une application utilisant l'Internet et un réseau RNIS doit aujourd'hui être implémentée à l'aide des protocoles de transport TCP et UDP. Les primitives de gestion de la qualité de service de cette API étant très réduites, et la sélection du réseau étant implicite par l'utilisation d'adresses IP prédéfinies, le portage d'une telle application sur un autre réseau, et utilisant de façon optimale les caractéristiques de ce réseau, est rendu difficile. En effet, la partie concernant le système de communication doit être radicalement modifiée. L'utilisation d'un protocole de transport capable de sélectionner dynamiquement le réseau à utiliser, ne poserait pas ce genre de problèmes. En effet, aucune référence ne serait faite directement à un réseau particulier, et par conséquent ce protocole de transport pourrait être vu comme une API de communication générique.

3.2.2 *Solution*

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 basée sur le protocole MMPOC est proposée. (Figure 6). 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.

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 1).

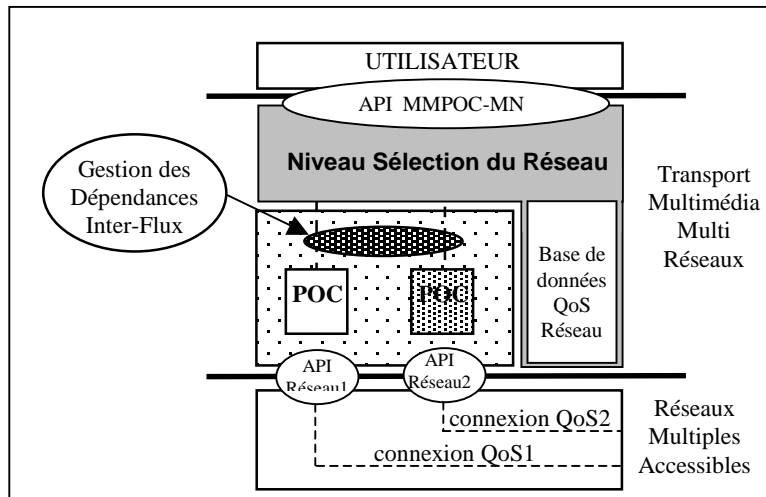


Figure 6. Architecture MMPOC-MN

	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 1. QoS offerte par certains services réseaux

Il est clair que certains paramètres n'ont pas de sens pour certains services réseaux : Par exemple la bande passante pour un service « best effort » comme celui 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 1, la base de connaissances ne comporte 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

dépendantes directement des drivers réseaux et fournies par le système, et les dynamiques qui doivent être mesurées (comme de 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 MMPOC-MN 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. Déploiement à l'aide de réseaux actifs

4.1 *Problématique*

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 parallèles 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.

4.2 *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 un 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. Trois réponses complémentaires avec des niveaux de granularité différents sont apparues : l'approche des routeurs programmables (1) [ALE98] qui offre un mécanisme de téléchargement manuel de programmes (dans les routeurs), mais qui maintient le format classique des paquets. La séparation du téléchargement de programmes et du routage des paquets est intéressante en termes de sécurité dans la mesure où cela reste à la charge de l'administrateur du réseau et non des utilisateurs. 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 au niveau qui permet la mobilité et l'exécution de programmes au travers d'un environnement hétérogène.

4.3 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.

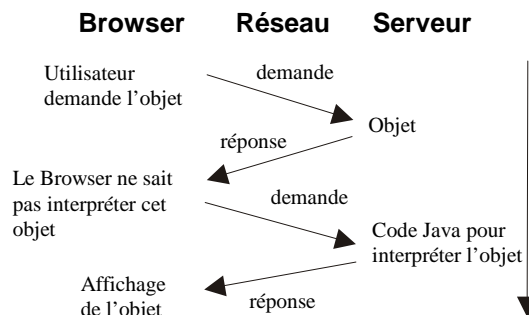


Figure 7. *Transfert du code de l'application*

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 7). 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).

5. Implémentation et résultats

5.1 Implémentation en Java

L'implémentation du protocole MMPOC-MN à 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 monomédia, 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.

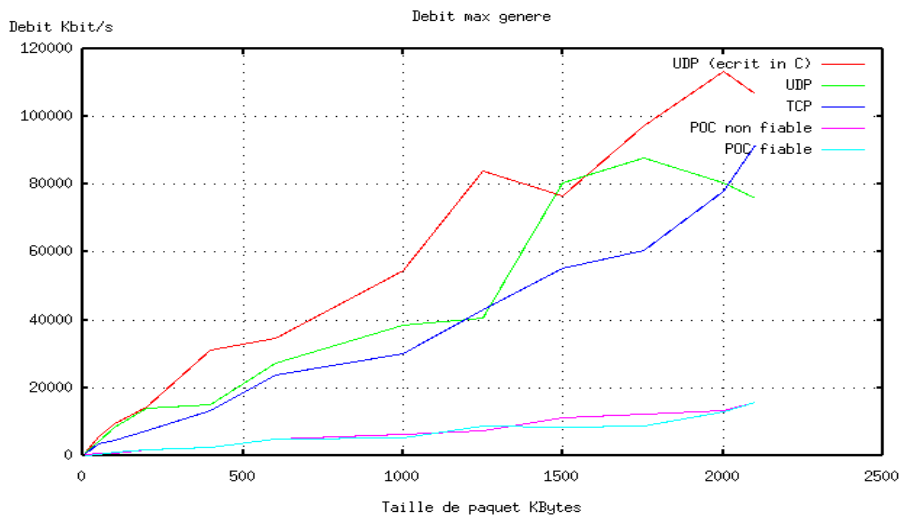


Figure 8. Mesures de performances du protocole

La figure 8 présente une comparaison entre, les performances du protocole MMPOC 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 l'impact du système de communication uniquement, 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 faites utilisant le service UDP du langage C. D'autre part, le débit maximum pour le protocole MMPOC, croit 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 sans ordre et avec fiabilité nulle, et celle d'un protocole MMPOC 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 légère.

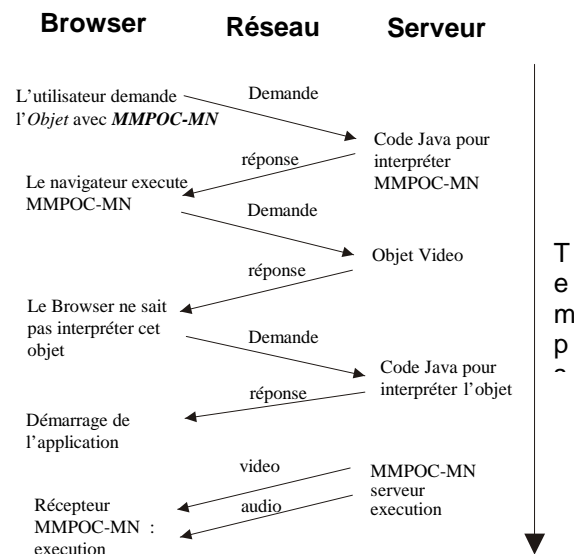


Figure 9. *MMPOC-MN Architecture de déploiement*

Une application de VoD à 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 joignant le protocole MMPOC-MN et l'application de VoD.
- Une partie statique comportant le serveur de fichiers utilisant 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 9 représente les différents échanges d'informations.

5.2 Résultats

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'avions pas accès aux réseaux satellitaires ou mobiles, mais seulement au réseau Internet classique, ou au réseau ATM national RENATER 2. Toutefois, dans le but de confirmer les résultats obtenus par simulation, les mesures qui sont présentées ci-dessous ont été faites sur l'implémentation réelle sur un réseau ATM. L'application de référence est la VoD qui génère deux flux audio et vidéo.

5.2.1 Mesures de performances du protocole MMPOC

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 en terme de qualité de service, 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 10. 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.

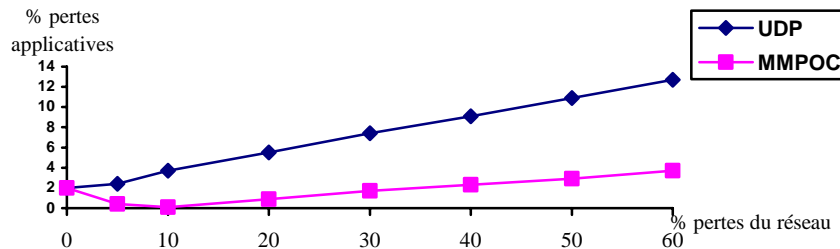


Figure 10. *QoS VoD sur POC vs UDP*

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.

5.2.2 *Evaluation de performances de MMPOC-MN*

Comme dit précédemment, l'évaluation de performances « multi-réseaux » a été faite 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 VoD 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 Internet, il apparaît qu'une dégradation du service réseau a un impact sur la qualité de service des deux flux, ce qui rend la présentation difficile à suivre. Dans le cas où plusieurs réseaux sont utilisés, des améliorations ont pu être notées lors des simulations. Transmettre les données audio sur un réseau RNIS, et la vidéo sur un réseau Internet, 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), l'audio 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 (ATM, satellite GEO), des pertes sur le lien ATM sont moins pénalisantes que des pertes sur le lien satellite. Il reste donc important de bien adapter la taille des tampons de l'application de VoD aux deux RTT observés sur les deux réseaux, pour pouvoir garantir la synchronisation audio/vidéo entre les deux flux. Néanmoins, même si ce problème n'est pas vital pour une application de VoD, qui peut toujours avoir de très gros tampons, elle est extrêmement stratégique pour des applications plus interactives, comme une application de visioconférence. Ainsi, il a été montré qu'il est quasiment impossible d'utiliser un lien satellite géostationnaire pour une application de visioconférence fortement interactive. Eventuellement, si les contraintes de synchronisation audio / vidéo peuvent être relâchées, et si on accepte que la vidéo soit constamment en retard par rapport à la partie son, alors un lien satellite peut être utilisé pour la vidéo, si la partie son est transportée sur un autre type de réseau à faible RTT. Néanmoins, la qualité obtenue sera critiquable.

6. Prospectives

6.1 ANTS: une solution pour le « multi-réseaux série »

Dans cet article nous ne présentons que les résultats obtenus concernant le « multi-réseaux parallèle ». Toutefois, l'extension vers ce que nous appelons le « multi-réseaux série » permettra de garantir la qualité de service sur une connexion qui traverse plusieurs réseaux ayant des QoS différentes. Cette partie semble plus délicate à traiter. Idéalement, cela voudrait dire, proposer un protocole de bout en bout qui serait efficace pour tous les types de réseaux traversés. Ceci semble, aujourd'hui, impossible, comme l'ont prouvé les difficultés rencontrées par le groupe TCPSAT de l'IETF.

Il paraît essentiel de fournir au protocole MMPOC-MN, des informations très précises sur les réseaux traversés. Les réseaux actifs à capsules [WET98] peuvent fournir ce genre d'informations grâce à leur capacité à injecter du code dans les nœuds du réseau, et à modifier leur comportement. Ce concept peut, par exemple, être utilisé pour déployer une « application active » qui permet de mesurer le délai entre les nœuds du réseau traversé. Une application qui envoie un message « PING » au nœud suivant, et qui attend la réponse, peut très bien satisfaire à ces besoins. Les travaux en cours utilisent la plate-forme active ANTS, ainsi qu'une de ses extensions, pour tester des mécanismes ayant les mêmes objectifs.

Ainsi une connexion traversant plusieurs réseaux, peut-être vue comme une série de sous – connexions, chacune ayant une QoS différente, et chacune devant utiliser un protocole adapté.

7. Conclusion

Cet article 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 [AME94] [OWE98].

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. Des évaluations, ainsi que des mesures, ont été faites sur les protocoles MMPOC et MMPOC-MN, montrant les bénéfices apportés par l'utilisation de tels protocoles.

Néanmoins, il apparaît que la métrique permettant l'adaptation du service réseau aux besoins applicatifs est très difficile à définir. Celle choisie et utilisée pour la visioconférence montre des limitations, et doit être raffinée afin d'être plus générique.

Finalement, comme dit en section 6, il est aussi prévu d'améliorer les mécanismes de MMPOC-MN, en utilisant de nouveaux mécanismes, grâce par exemple la plate-forme ANTS. Une évaluation de ces implémentations futures devrait être réalisée prochainement dans le cadre de la plate-forme active française Amarrage.

8. Références

- [ALE 98] D.S. ALEXANDER, W.A. ARBAUGH, M.W. HICKS, P.K., ANGELOS D.K., J.T. MOORE, C.A. GUNTER, S.M. NETTLES, « The SwitchWare Active Network Architecture » *IEEE Network Special Issue on Active and Controllable Networks*, Vol 12 n°3, pp.29-36, 1998
- [AME 94] 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
- [DIA 95] 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
- [FOU 96] M. FOURNIER, C. CHASSOT, A. LOZES, M. DIAZ, "Multimedia partial order transport architecture : Design and implementation. Protocols for High Speed Network", (*PfHSN96*), 1996
- [HAR 96] 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>)
- [JAV 98] JavaSoft, Sun Microsystems, Inc, "HotJava™ Browser : A White Paper". 1998 (<http://sunsite.nus.sg/hotjava/1.0alpha3/doc/overview/hotjava/index.html>)

- [JAV 99] JavaSoft, Sun Microsystems, Inc, "Jini™ Technology Architectural Overview". January 1999 (<http://www.sun.com/jini/whitepapers/architecture.html>)
- [INO 97] J. INOUE, J. WALPOLE, J. BINKLEY, "Physical Media Independence : system Support for Dinamically Available Network Interfaces". *OGI CSE Technical Report 97-001*, Janvier 1997. (<ftp://www.cse.ogi.edu/pub/tech-reports/1997/97-001.ps.gz>)
- [KRU 98] B. KRUPCZAK, K.L. CALVERT, M. AMMAR, "Implementing Communication Protocols in Java ", *IEEE Communications Magazine*, Octobre 1998
- [LAN 98] B. LANDFELDT A. SENEVIRATNE, "User Service Assistant(USA): An End-to-End Reactive QoS Architecture " *Sixth International Workshop on Quality of Service*, 1998.
- [OWE 97] 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, 16-18Juin, 1997
- [OWE 98] 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
- [SCH 99] B. SCHWARTZ A.W. JACKSON, W.T. STRAYER, W. ZHOU, D. ROCKWELL ET C. PARTRIDGE "Smart Packets for Active Networks", *OpenArch*, Mars 1999 (<ftp://ftp.bbn.com/pub/AIR/smart.ps>)
- [STE 98] M. STEMM ET R. H. KATZ., "Vertical Handoffs in Wireless Overlay Networks", *ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet*, 1998. (<http://http.cs.berkeley.edu/~stemm/publications/monet97.ps.gz>)
- [TEN 97] TENNENHOUSE ET AL., "A survey Active network Research ", *IEEE Communications Magazine*, vol. 35, No. 1, pp 80-86. Janvier 1997. (<http://www.tns.lcs.mit.edu/publications/ieeecomms97.html>)
- [VIL 98] T.VILLEMUR , V.BAUDIN , S.OWEZARSKI , M.DIAZ, "An integrated platform for cooperative teleteaching", *Lecture Notes in Computer Science 1483, Eds. T.Plagemann, V.Gobel, 1998, Springer*, ISBN 3-540-64955-7, pp.59-70
- [WET 98] WETHERALL, GUTTAG, TENNENHOUSE, "ANTS : A Tool kit for Building and Dimamically Deploying Network Protocols ", *IEEE OPENARCH'98*, San Francisco, Avril 1998. (<http://www.tns.lcs.mit.edu/publications/openarch98.html>)