# Mémoire d'Habilitation à Diriger des Recherches

Présenté par

Thierry TURLETTI
Projet Planète
INRIA Sophia Antipolis

à

L'École Doctorale STIC
de L'Université de NICE - Sophia Antipolis

# *Étude et Conception de Mécanismes pour Applications Multimédias sur Réseaux IP Filaires et Sans Fil*

Soutenue à l'INRIA le vendredi 13 janvier 2006 devant la commission composée de :

| | |
|---|---|
| M. Jean-Paul Rigault | Président |
| M. Jon Crowcroft | Rapporteur |
| M. Andrzej Duda | Rapporteur |
| Mme Christine Guillemot | Rapporteur |
| M. Ernst Biersack | Examinateur |
| M. Ken Chen | Examinateur |

*À mes parents, Isabelle, Théo et Robin*

# TABLE DES MATIÈRES

# 1. AVANT-PROPOS

Comme préambule, je dresse dans l'ordre chronologique la liste de mes principales contributions scientifiques et présente à la suite l'organisation de ce mémoire.

## *1.1 Parcours scientifique*

Lors de ma de thèse, je me suis intéressé aux problèmes liés au codage et à la transmission de la vidéo sur Internet. J'ai contribué à montrer d'une part que la technologie actuelle permettait de réaliser des codecs vidéo à faible coût en logiciel, et d'autre part qu'il était possible de transmettre de la vidéo de qualité acceptable sur Internet. Il est bon de rappeler que pour la communauté scientifique et industrielle des années 90, la compression vidéo était essentiellement une affaire de "hardware" et la transmission vidéo ne se concevait que sur des réseaux offrant des garanties de qualité de service (alors qu'Internet n'offre pas de telles garanties). Ainsi, j'ai mis en œuvre l'un des premiers logiciels de vidéoconférence sur Internet, IVS [14, 66, 67], basé sur le standard H.261. J'ai été amené à concevoir un schéma de découpage en paquets du flux vidéo H.261, ainsi que des algorithmes d'adaptation du codeur vidéo en fonction de la capacité disponible dans le réseau, en particulier des algorithmes de contrôle d'erreurs et de contrôle de débit vidéo [9, 10, 13, 7, 71]. L'algorithme de découpage de flot vidéo H.261 en paquets a été standardisé à l'IETF (organisme de standardisation d'Internet) avec le RFC 2032 [70] et a aussi été adopté par l'UIT (Union Internationale des Télécommunications). Tous ces travaux ont été menés dans le cadre des projets Européens MICE, MICE2 et MERCI.

A la suite de ma thèse, j'ai effectué un séjour postdoctoral d'un an dans le groupe "Telemedia, Networking and Systems" (TNS) dirigé par le Professeur David Tennenhouse au LCS (Laboratory for Computer Science) du MIT. Ce groupe de recherche[1] était réputé pour ses développements d'applications multimédias haut débit. Mes travaux ont porté sur le projet SpectrumWare, un des tout premiers projets concernant la radio logicielle. J'ai étudié les problèmes et limitations techniques liés à la mise en œuvre logicielle d'applications Radio-Fréquence (RF) et réseaux mobiles. En particulier, j'ai travaillé sur l'adaptation d'algorithmes de traitement du signal (habituellement implantés dans des circuits hardware spécialisés) pour implanter la partie Interface Radio d'une station de base GSM en logiciel [68, 65, 73].

De retour dans le projet RODEO en octobre 1996, j'ai poursuivi mes travaux concernant la transmission multimedia en multipoint sur Internet vers des récepteurs

[1] Voir aussi l'URL <http ://www.tns.lcs.mit.edu/>.

hétérogènes [8]. J'ai mis au point un codage hiérarchique audio, robuste à la perte de paquets et à faible complexité, ainsi qu'un des premiers algorithmes de contrôle de congestion TCP-*Friendly* [72]. Dans le cadre du projet Européen HIPPARCH, j'ai implanté des mécanismes de transmission FEC adaptatif et les ai intégrés au compilateur de protocoles *ALFred* [17]. Enfin, sur le thème des radio logicielles, j'ai proposé une nouveau mécanisme, indépendant de la machine utilisée, pour évaluer la complexité des algorithmes afin de prévoir la puissance CPU nécessaire à leur implantation logicielle [74, 69].

J'ai été admis comme chargé de recherche à l'INRIA en 1998 avec un programme de recherche qui portait sur deux thématiques : les applications radio logicielles et l'optimisation de la transmission multimédia vers un grand nombre de récepteurs hétérogènes.

En ce qui concerne la première thématique, je me suis intéressé plus particulièrement à la mise en oeuvre de ces applications ainsi qu'aux nouvelles fonctionnalités qu'elles apportent. Les applications radio logicielles sont particulièrement complexes à mettre en oeuvre car elles font appel à de multiples domaines scientifiques. Dans le cadre du projet européen ITEA DESS, j'ai étudié comment simplifier l'implantation de la partie contrôle d'applications radio logicielles via l'utilisation du langage formel Esterel. Nous avons ainsi développé une extension Esterel à l'environnement de développement PSPECTRA du MIT [28, 30, 29, 24] qui permet de vérifier de manière automatique certaines propriétés du code généré. D'autre part, une implantation logicielle de ce type d'applications à forte composante en traitement du signal permet d'adapter les algorithmes de transmission à tous les niveaux de la pile protocolaire, y compris au sein de la couche physique. Je me suis intéressé en particulier à la possibilité de choisir le mode de transmission d'une station WLAN IEEE 802.11 non seulement en fonction de caractéristiques du canal de transmission mais aussi en fonction de la tolérance aux erreurs des paquets à transmettre [48, 49, 47]. Ce mécanisme fait l'objet du chapitre 5.

Mes travaux sur la seconde thématique ont porté sur les problèmes suivants. Je me suis intéressé à l'élaboration d'un protocole de communication pour des environnements virtuels à grande échelle [39, 40, 18]. Ce protocole, détaillé dans le chapitre 2 de ce manuscrit, est utilisé pour l'application monde virtuel sur Internet V-Eye [55] développée dans notre projet. Dans le cadre du projet RNRT VISI, j'ai travaillé sur l'élaboration de mécanismes de contrôle de congestion TCP-courtois pour des flots vidéo hiérarchiques [60, 78, 77, 79]. Ce sujet fait l'objet du chapitre 3. Au sein du projet RNRT VIP, j'ai étudié la problématique du contrôle de congestion pour de la vidéoconférence sur des réseaux hybrides filaires et sans fil [5]. Enfin, dans le cadre des projets RNRT VTHD et VTHD++, j'ai travaillé sur le support de mécanismes de différenciation de services pour de la transmission multimédia sur des réseaux Piconet Bluetooth [34] ainsi que sur des réseaux locaux sans fil IEEE 802.11e [58, 45, 4, 52, 56]. Un de ces mécanismes est décrit dans le chapitre 4 du manuscrit. D'autres travaux ont aussi porté sur le WIFI : l'élaboration de nouveaux algorithmes de sélection de mode 802.11 [33, 26] plus efficaces, des mécanismes pour rendre le partage de ressources d'un canal 802.11 plus équitable [15]

et l'amélioration des performances du mécanisme d'accès DCF lorsque le réseau est saturé [2, 3].

## 1.2 Organisation du mémoire

Dans ce manuscrit, j'ai choisi de présenter quatre échantillons de mes travaux que j'ai eu le plaisir d'effectuer avec des étudiants depuis ma thèse. Pour chacun de ces mécanismes : 1/ le problème à résoudre, son importance ainsi que les verrous scientifiques à lever sont introduits ; 2/ un bref état de l'art du domaine suit ; 3/ les principes de la solution élaborée sont présentés en identifiant les contributions (l'article qui détaille la solution étant reproduit dans l'Annexe Technique) ; 4/ enfin, la pertinence de la solution et sa mise en oeuvre dans le réseau sont discutées.

Le mémoire est organisé de la manière suivante. Le chapitre 2 traite du problème de passage à l'échelle des protocoles de communication pour des applications d'environnements virtuels à grande échelle. Le chapitre 3 étudie la problématique de la transmission multimédia sur Internet vers un ensemble hétérogène de récepteurs. Le chapitre 4 porte sur le support de mécanismes de différenciation de services dans les réseaux WIFI. Le chapitre 5 présente un exemple de mécanisme intercouches pour améliorer les performances de la transmission multimédia sur les réseaux sans fil. Enfin, le chapitre 6 conclut le mémoire en présentant quelques axes de recherche que j'ai l'intention de développer dans le futur.

# 2. PROTOCOLE DE COMMUNICATION POUR EVGE

## *2.1 Problématique*

Ce chapitre porte sur l'élaboration d'un protocole de communication pour des applications d'environnements virtuels à grande échelle (EVGE). Afin de mettre en évidence les différents problèmes que les EVGE engendrent, étudions tout d'abord quelles sont les principales caractéristiques de ces applications ainsi que leurs besoins.

### *Grand nombre de participants*

Dans un EVGE, chaque utilisateur est à la fois récepteur et source de différents types de données. Comme chaque participant n'interagit à tout instant qu'avec un nombre limité de voisins, il n'est intéressé que par le trafic associé à ces derniers. Ce sont donc des applications de type "*many few-to-few*" dans lesquelles la complexité croît avec le nombre de participants. Si l'on considère l'ensemble des informations émises par les participants de l'EVGE, lorsque le nombre de participants augmente, le pourcentage d'informations que désire recevoir un participant diminue. Il est clair que la transmission de tous les paquets de données émis par chaque participant vers l'ensemble des participants est inefficace. À l'intérieur du réseau, elle peut entraîner une saturation des tampons mémoires au niveau des routeurs et une intensification de la congestion sur les liens de transmission. Au niveau des machines hôtes, les effets peuvent se traduire par un débordement des files d'attente et un gaspillage des ressources CPU.

### *Hétérogénéité des participants*

Les EVGE sont susceptibles de comporter un très grand nombre de participants et ces derniers peuvent avoir des caractéristiques très hétérogènes. Cette hétérogénéité se traduit par la diversité des liens qui relient les participants à Internet, et par la différence de puissance de calcul disponible sur les machines-hôtes. Elle peut également s'exprimer en terme de préférences de l'utilisateur. Toutes ces différences entre les utilisateurs doivent donc être prises en compte par les EVGE afin que les ressources dont ils disposent soient utilisées de la meilleure manière. Il est nécessaire de trouver un bon compromis entre l'état rajouté dans les routeurs, la bande passante utilisée et la satisfaction de l'utilisateur final. Chaque participant d'un EVGE est ainsi caractérisé par deux paramètres qui lui sont propres : ses centres d'intérêts, et sa capacité à recevoir et traiter en temps réel les informations qu'il reçoit.

#### *Besoin d'interactivité*

Un EVGE est constitué d'un ensemble d'*entités* régies par des règles d'interaction. Par exemple, deux participants face à face dans le monde virtuel doivent pouvoir observer en temps-réel chacune de leurs actions. Cependant, la technologie actuelle ne permet pas de traiter l'information et de la communiquer de manière instantanée. Le délai de transmission physique auquel s'ajoute le temps de traitement au niveau des routeurs doit être pris en compte dans le développement des applications interactives distribuées. Un délai maximum de 150 ms est généralement reconnu pour ce type d'applications [53]. En conséquence, des mécanismes de contrôle de transmission adaptés aux différents besoins d'interactivité doivent être mis en œuvre afin de masquer les effets du réseau sur l'application.

## *2.2 État de l'art : mécanismes de filtrage d'information*

Dans la littérature scientifique, deux classes d'approches de filtrage d'information ont été proposées, une approche au niveau de la couche réseau et une approche au niveau de la couche transport. La première approche consiste à rajouter de nouveaux mécanismes au sein des routeurs afin de limiter la propagation des paquets vers un sous-ensemble de destinataires. Dans le deuxième cas, l'architecture de communication est basée sur l'utilisation de plusieurs groupes multipoint. Des mécanismes de répartition dynamique des participants dans les différents groupes doivent donc être mis en œuvre.

### *2.2.1 Filtrage au niveau de la couche réseau*

Le filtrage au niveau de la couche réseau consiste à rajouter de nouveaux mécanismes au sein des routeurs pour ne propager les paquets que vers le sous-ensemble de destinataires intéressés par leur contenu [64]. Pour cela, chaque paquet doit comporter des *meta-informations* précisant la nature de l'information qu'il contient. Dans l'architecture AIM (*Addressable Internet Multicast*)[37], ces meta-informations sont comparées à des labels assignés à chaque routeur de l'arbre multipoint sous-jacent. Grâce à ces labels, chaque source ou récepteur du groupe multipoint possède une adresse au sein de l'arbre multipoint associé. Les paquets émis par une source contiennent le label de la source ainsi que les labels des destinataires, permettant ainsi un sous-routage implicite au sein du même arbre. Les labels de flots, quant à eux, permettent de construire des sous-arbres en associant un label avec un ensemble de sources (générant un flot de données). Chaque routeur doit alors maintenir une liste de labels désignant les flots en cours de réception ainsi que la liste des interfaces sur lesquelles faire suivre les paquets pour chacun de ces flots.

L'approche de filtrage au niveau réseau permet en outre d'éviter les temps de latence liés à l'établissement de l'arbre de transmission multipoint et les coûts liés au maintien d'un grand nombre d'arbres multipoint. En effet, à partir d'un seul arbre multipoint existant, des techniques de division en sous-groupes peuvent facilement

être utilisées par les applications afin de limiter leur trafic vers un sous-ensemble de destinataires. Cependant, le problème de l'organisation des groupes au niveau de l'application (ou comment regrouper les sources et les récepteurs entre eux) et le déploiement de ce mécanisme à grande-échelle restent des problèmes ouverts. En effet, cette approche entraîne des modifications à l'intérieur de chaque routeur du réseau et, par conséquent, est beaucoup plus délicate à être déployée.

### *2.2.2 Filtrage au niveau de la couche transport*

Les premiers travaux entrepris autour de ce thème aboutirent à l'apparition de deux normes : SIMNET [38] puis DIS (*Distributed Interactive Simulation*) [63], toutes deux très orientées vers la simulation militaire. Leurs architectures distribuées ont été à l'origine d'une des premières thèses dans le domaine [42] qui porte sur le problème du passage à l'échelle des LSDS (*Large-Scale Distributed Simulations*). Les solutions proposées consistent à découper logiquement l'environnement virtuel en différentes classes d'entités selon des critères d'ordre spatial, temporel et fonctionnel, et d'associer chacune de ces classes à un groupe multipoint différent. Le verrou scientifique consiste alors à déterminer une méthode effectuant ces associations qui soit à la fois suffisamment générique, efficace et surtout passe à l'échelle.

La première tentative fut mise en œuvre dans NPSNET [43] qui utilise un découpage statique du monde virtuel en cellules. À chacune des cellules est associée un groupe multipoint. Les différentes entités sont réparties dans les différents groupes multipoints par l'intermédiaire d'un gestionnaire de centres d'intérêts. Toutefois, le découpage est réalisé de façon statique et la répartition des entités dans les différents groupes est supposée changer peu fréquemment, ce qui n'apporte au problème qu'une solution partielle. D'autres environnements virtuels, comme VREng (*Virtual Reality Engine*) [21], sont également basés sur des principes similaires à ceux implantés dans NPSNET.

Dans DIVE (*Distributed Interactive Virtual Environment*) [16] et MASSIVE (*Model, Architecture, and System for Spacial Interaction in Virtual Environments*) [22], plusieurs groupes multipoints sont également utilisés. Cependant, les environnements virtuels sont décrits par une hiérarchie d'objets plutôt que par un découpage en cellules. Dans ces deux implantations, plusieurs types de médias sont supportés et le concept d'auras [6] est utilisé, généralisant ainsi la notion de centre d'intérêt d'une entité virtuelle.

Diamond Park est un environnement virtuel basé sur une architecture multi-serveur appelée SPLINE (*Scalable Platform for Large Interactive Networks Environments*) [80]. Le monde virtuel est partagé en plusieurs zones appelées *locales* à l'intérieur desquelles les communications se font en multipoint. Lorsqu'un participant change de locale, il obtient par l'intermédiaire des serveurs, une matrice lui permettant de changer de système de coordonnées.

## 2.3 Contributions : le protocole de communication SCORE

Cette section présente une vue d'ensemble du protocole de communication SCORE [39, 40] dont l'objectif est de rendre scalable le déploiement des EVGE sur Internet tout en prenant en compte l'hétérogénéité des utilisateurs. Nous invitons le lecteur à se référer à l'annexe A pour une description plus détaillée du protocole et de ses performances.

Comme nous l'avons vu dans la section 2.1, définir une architecture de communication pour EVGE est particulièrement complexe tant le nombre de paramètres à considérer et les besoins pour ce type d'applications sont importants. Nous avons opté pour un filtrage des données au niveau de la couche transport, en supposant toutefois que chaque utilisateur est capable de recevoir et d'émettre du trafic en multipoint avec le modèle ASM (*Any Source Multicast*). Ceci limite le problème de déploiement du protocole, car nous ne faisons pas l'hypothèse que les routeurs ont la possibilité d'implanter des mécanismes de filtrage élaborés.

SCORE est une architecture de communication multi-agent basée sur une stratégie de découpage spatial de l'EVGE en cellules. Les agents sont des machines ou des processus placés à différents endroits du réseau (par exemple, sur le LAN d'un campus ou hébergé chez des fournisseurs de services). Les agents ne reçoivent pas les données que les participants se transmettent, d'où l'utilisation du terme "agent" plutôt que "serveur". Ils ne sont pas impliqués dans le calcul d'états, la synchronisation entre participants, ou la centralisation du trafic émis puis sa redistribution vers les participants concernés. Leur rôle est de définir de manière dynamique des zones à l'intérieur de l'EVGE en tenant compte de la répartition des participants. Ils doivent aussi calculer des tailles de cellules appropriées en fonction des diverses densités de participants dans ces zones ; une zone étant par définition une sous-partie de l'EVGE à l'intérieur de laquelle les cellules ont toutes la même taille. Les agents évaluent également périodiquement la satisfaction de chaque participant en tenant compte de leur capacité respective, de la taille de leur zone d'intérêt, ainsi que de la densité de participants dans la zone dans laquelle ils se trouvent.

Dans SCORE, les groupes multipoints jouent le rôle d'outils de filtrage de trafic pour les participants. À chaque cellule de l'EVGE est associé une adresse de groupe multipoint distincte. Le calcul d'état, la synchronisation des données et les abonnements/désabonnements aux groupes multipoints sont laissés à la charge des participants. Comme ce sont les participants qui prennent la décision finale d'abonnement et désabonnement aux groupes multipoints, les agents doivent leur fournir la liste des groupes multipoints auxquels ils doivent s'abonner afin qu'ils puissent communiquer avec leurs voisins.

En fonction de sa position dans l'EVGE et de son intérêt pour le contenu des données en transit sur chaque groupe, chaque participant peut ensuite s'abonner et se désabonner dynamiquement aux groupes multipoints. Enfin, pour préserver l'interactivité de l'application, les participants doivent anticiper l'abonnement aux groupes multipoints afin de masquer la latence liée à l'abonnement multipoint.

## *2.4 Discussion*

La présence des agents et des messages de contrôle et de signalisation échangés dans le protocole de communication SCORE introduit un coût supplémentaire. Nous avons analysé ce coût avec l'amélioration des performances qu'apporte SCORE pour différents degrés de mobilité et d'hétérogénéité des participants. A noter qu'une limitation du nombre de groupes multipoints disponibles pour l'application EVGE peut avoir des conséquences néfastes sur les performances de l'application. En effet, une telle limitation entraîne implicitement une réduction du nombre de cellules dans l'EVGE. Par conséquent, lorsque la densité de participants par cellule excède un certain seuil dans l'EVGE, les agents ne sont plus capables d'améliorer la satisfaction de leurs participants. On peut toutefois imaginer des solutions pour préserver la qualité perçue par les utilisateurs de l'EVGE. Par exemple, on peut élaborer un EVGE "extensible" dont la taille s'adapte en fonction du nombre d'utilisateurs, de telle sorte à ce que la densité moyenne de participants dans l'EVGE reste toujours inférieure à un certain seuil. On peut y associer des protocoles tels que ceux définis dans *MAAA* [32], afin de permettre une allocation dynamique de groupes multipoints, lorsque la densité de participants excède un seuil maximal.

Le protocole SCORE a été implanté dans l'application monde virtuel V-Eye [55] développée dans le projet Planète. Ce logiciel combine un monde en 3 dimensions, avec des participants qui ont la possibilité d'envoyer des messages textuels ou des flots multimédias (par le biais des outils de vidéoconférence vic et rat), ainsi qu'une salle de cinéma pour projeter des flots vidéo en Haute Définition. Ce développement a servi de plate-forme pour expérimenter des transmissions multimédias avec un grand nombre de participants sur des réseaux hétérogènes avec, par exemple, une dorsale très rapide comme le réseau VTHD, des réseaux locaux filaires et WIFI.

A la suite de ces travaux, nous avons étudié le cas où le multipoint traditionnel ASM n'était pas disponible mais qu'à la place, les participants pouvaient utiliser le multipoint SSM (*Source Specific Multicast*). SSM pose moins de problèmes de déploiement que ASM et, comme son nom l'indique, permet de spécifier le sous-ensemble de sources d'un groupe multipoint auquel on désire s'abonner. Alors que le modèle SSM peut sembler, à première vue, plus approprié aux applications "de 1 vers N" comme la diffusion en ligne, nous avons montré qu'il était possible de tirer profit d'une telle couche de communication pour résoudre les nouveaux problèmes qui se posent à l'application dans le monde SSM. Avec un surcoût de traffic de signalisation (en particulier, pour annoncer aux participants les adresses IP des sources voisines), la version SSM du protocole SCORE obtient même de meilleures performances car elle permet de sélectionner de manière plus fine les flots reçus par les participants [54].

# 3. CONTRÔLE DE TRANSMISSION POUR FLOTS MULTIMÉDIAS HIÉRARCHIQUES SUR INTERNET

## *3.1 Problématique*

La transmission multimédia temps-réel en multipoint sur Internet est confrontée à plusieurs problèmes. Tout d'abord, Internet ne supporte qu'un service de type *best-effort*, sans fiabilité ni aucune garantie de délai. Les applications, en l'occurrence les codeurs-décodeurs (codecs) audio et vidéo, doivent donc utiliser des mécanismes pour s'adapter de manière dynamique aux conditions du réseau et être robuste à la perte éventuelle de paquets.

Pour lutter contre la perte de paquets, des mécanismes de correction d'erreurs FEC (*Forward Error Correction*) peuvent être utilisés, mais ces derniers introduisent inévitablement un surcoût en terme de bande passante qui doit être pris en compte par l'algorithme de contrôle de transmission. Lorsque le service IP multipoint est supporté dans le réseau, les applications de type vidéoconférence ont la possibilité d'utiliser le protocole de transport UDP au dessus de IP multipoint. Cependant, il leur faut implanter un mécanisme de contrôle de congestion afin qu'elles puissent adapter leur débit d'émission en fonction des ressources disponibles dans le réseau. Ce mécanisme doit faire en sorte que ces applications ne représentent pas une menace pour les autres applications qui partagent les différentes liaisons, principalement TCP. C'est pour cette raison que de nombreux algorithmes de contrôle de congestion "TCP-Courtois" (*TCP-Friendly*) ont vu le jour dans la communauté scientifique réseau.

D'autre part, pour donner entière satisfaction aux utilisateurs, il est important que les applications puissent s'adapter à l'hétérogénéité de ces derniers. En effet, les utilisateurs sont reliés entre eux par des liaisons qui ont des caractéristiques (bande passante disponible, latence, jigue, taux de pertes) dynamiques et non homogènes. De plus, les utilisateurs n'ont pas forcément la même puissance de calcul disponible, le terminal pouvant très bien être une station de travail ou un assistant personnel (*PDA*). Le problème de l'hétérogénéité des utilisateurs peut-être résolu en partie en utilisant des flots multimédias hiérarchiques. Cependant, pour des raisons d'efficacité de codage, le nombre de couches doit rester faible (pour des raisons d'efficacité de codage, il excède rarement 3 ou 4 couches), il est donc nécessaire de regrouper les utilisateurs qui ont des caractéristiques communes pour pouvoir s'adapter au mieux à leur profil. Bien sûr, si l'on veut pouvoir s'adapter de manière optimale aux caractéristiques des différents récepteurs, il est indispensable de pouvoir estimer leurs capacités de manière dynamique, ce qui pose le problème bien

connu d'implosion de la voie de retour lorsque les utilisateurs sont nombreux.

## 3.2 État de l'art : mécanismes de contrôle de congestion

Le codage hiérarchique (ou codage en couches) a souvent été proposé comme solution pour contrôler la transmission multimédia vers un ensemble hétérogène d'utilisateurs. Principalement, deux types d'approches ont été proposées, selon que le contrôle se fait par le récepteur (*receiver-driven*) [50, 75], ou par la source et les récepteurs à la fois (approche hybride) [62, 76, 25]. Le premier type d'approche, utilisé par RLM [50] et RLC [75], consiste à émettre les différentes couches avec des débits statiques en multipoint sur des adresses de groupe distinctes. La décision d'abonnement ou de désabonnement aux couches est prise par les récepteurs en fonction de leur capacité disponible. Cependant, ces mécanismes souffrent de comportements pathologiques tels que des périodes de congestion transitoires, provoquant des pertes de paquets périodiques [35]. PLM [36] est un protocole de contrôle de congestion qui évite ce dysfonctionnement en utilisant un mécanisme basé sur l'envoi simultané de deux paquets pour inférer la bande passante du réseau. Cependant, il fait l'hypothèse que l'ensemble des routeurs utilisent une politique d'ordonnancement de type *Fair Queuing* alors que la grande majorité des routeurs d'Internet utilisent *FIFO* aujourd'hui.

TFMCC [82] est un mécanisme de contrôle de congestion TCP-courtois pour des applications multipoints. Il utilise un mécanisme d'estimation du temps aller-retour (*RTT*) source-récepteur pour calculer le débit *TCP-Friendly* que l'application est susceptible de recevoir ainsi qu'un mécanisme de suppression de rapports de réception pour éviter le phénomène d'implosion. Cependant, il ne considère qu'un flot unique de données (donc un débit unique) et n'est donc pas en mesure de régler le problème d'hétérogénéité des récepteurs.

FLID-DL [12] est un mécanisme de contrôle de congestion pour des sources multipoints à débits multiples qui élimine le besoin des sondes périodiques utilisées par RLC. Cependant, le trafic important de messages de contrôle IGMP et PIM-SM que génère chaque récepteur rend ce schéma prohibitif. WEBRC [41] a été récemment proposé pour résoudre les principaux inconvénients de FLID-DL en utilisant une approche originale d'émission des données à la manière d'ondes. Cependant WEBRC, tout comme FLID-DL ne sont efficaces que pour des applications de type transmission fiable[1], et ne peuvent pas être utilisés pour transmettre des flots multimédias temps-réel comme H.263+ ou MPEG-4.

SAMM [76] est l'un des premiers algorithmes de contrôle de congestion pour flots multi-couches dans lequel la source adapte le débit des différentes couches en fonction de la bande passante disponible. Il utilise une fonction de suppression partielle des rapports de réception dans les nœuds du réseau pour éviter le problème d'implosion de messages. Toutefois, ce mécanisme n'est pas en mesure de

[1] Ces protocoles peuvent éventuellement être utilisés pour du streaming si la latence n'est pas une limitation.

refléter précisément les besoins des différents récepteurs et l'algorithme n'est pas *TCP-Friendly*.

MLDA [62] est un algorithme de contrôle de congestion TCP-courtois et, tout comme SAMM, il adapte le débit des différentes couches émises en fonction des rapports de réception. Pour éviter le problème d'implosion de rapports, ces derniers sont envoyés avec des délais générés de manière exponentiellement distribué et un mécanisme de suppression partielle des rapports est utilisé. Cependant, lorsque les récepteurs sont très hétérogènes, le nombre de rapports de réception émis peut devenir très élevé et mener quand même au phénomène d'implosion. D'autre part, MLDA ne permet pas de quantifier le nombre de récepteurs par débit demandé et ni SAMM ni MLDA ne proposent de solution à la perte de paquets sur les différentes couches transmises.

Un mécanisme de contrôle de congestion qui classifie et regroupe de manière centralisée les rapports de réception RTCP [61] est proposé dans [23]. Ces rapports comportent la bande passante disponible ainsi que le taux de perte de paquets observé. Néanmoins, comme le mécanisme de regroupement est centralisé, des récepteurs de bande passante équivalente mais avec des taux de perte différents peuvent se retrouver dans la même classe de récepteurs. D'autre part, pour éviter le phénomène d'implosion des rapports, la fréquence d'émission des rapports RTCP est fonction du nombre de récepteurs. Ainsi, lorsque le nombre de récepteurs augmente, la pertinence des rapports diminue de manière significative.

## *3.3 Contributions : le mécanisme de contrôle de transmission SARC*

Le mécanisme SARC [79] (*Source-channel Adaptive Rate Control*) que nous avons élaboré a pour objectif d'adapter à la fois le nombre de couches, le débit source par couche et le débit canal (FEC) par couche en fonction des caractéristiques des différents récepteurs, et ceci, tout en restant TCP-compatible avec les autres applications. Nous ne décrivons ici que les principes de ce protocole et invitons le lecteur à se référer à l'Annexe B pour plus de détails.

Au problème de la génération de rapports de réception pertinents qui passe à l'échelle, nous avons proposé un mécanisme à base d'agents d'agrégation placés dans des endroits stratégiques au sein du réseau, voir Figure 3.1. Les récepteurs sont regroupés en une hiérarchie de régions locales ; chaque région contient un agent d'agrégation qui reçoit des rapports de réception, classe les récepteurs en groupes de caractéristiques homogènes et renvoie un rapport de réception agrégé à l'agent d'agrégation de niveau supérieur.

Le mécanisme d'agrégation des rapports permet de regrouper les différents récepteurs en classes de caractéristiques homogènes. Il est d'autant plus efficace que le réseau présente de fortes corrélations spatiales, comme c'est le cas dans ce type de transmission sur le MBone [83]. Pour bénéficier de la redondance temporelle des rapports de réception, les anciens rapports de réception sont agrégés (avec un poids

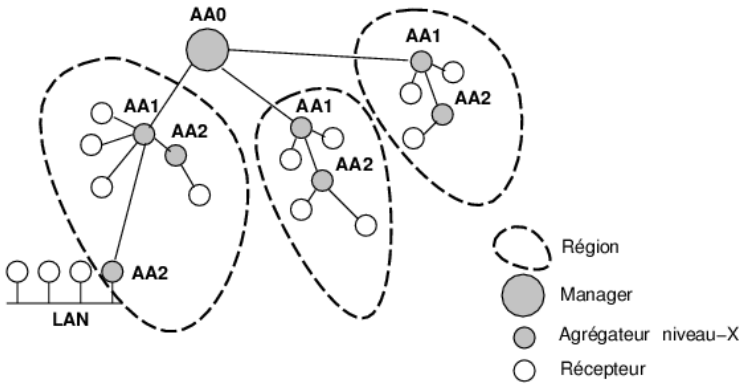



*Fig. 3.1:* Organisation hiérarchique des agents d'agrégation dans le réseau.

qui dépend de leur ancienneté) avec les nouveaux rapports.

Au début de la session, la source envoie aux récepteurs la gamme de débits (c'est-à-dire l'intervalle de débit $[R_{min}, R_{max}]$) estimée à partir des caractéristiques débit-distorsion de la source. La valeur $R_{min}$ correspond au débit en dessous duquel la qualité du signal décodé n'est plus acceptable et la valeur $R_{max}$ au débit au dessus duquel il n'y a plus d'amélioration sensible de la qualité du signal vidéo. L'intervalle $[R_{min}, R_{max}]$ est discrétisé en sous-intervalles pour éviter d'avoir des couches de qualités équivalentes et donc inutiles.

A partir de ce moment, l'algorithme de contrôle de congestion multi-couches peut démarrer. On divise alors le temps en *rondes* qui correspondent aux intervalles de réception des nouveaux rapports de réception. Chaque ronde comporte les quatre étapes suivantes.

- Au début de chaque ronde, la source annonce le nombre de couches et leurs débits respectifs en utilisant les rapports d'émission SR (*Sender report*) de RTCP. Chaque couche est émise sur un groupe IP multipoint distinct.
- Chaque récepteur estime son taux de perte et son débit TCP-compatible avec la source. Ce débit estimé sera utilisé pour choisir les couches auxquelles s'abonner ou se désabonner. Chaque récepteur envoie un rapport de réception RR (*Receiver Report*) qui comporte le débit TCP-courtois estimé et son taux de perte en direction de la source. Ce rapport n'est pas émis en multipoint comme dans le protocole RTP, mais en point-à-point vers un agent d'agrégation.
- Des agents d'agrégation, placés à des endroits stratégiques dans l'arbre de distribution multipoint entre la source et les récepteurs, effectuent une classification des récepteurs selon une mesure de similarité préalablement définie, et produisent des rapports de réception agrégés. Plus précisément, ces rapports représentent des groupes (ou *clusters*) de récepteurs qui ont des caractéristiques de bande passante disponible et taux de perte observées analogues. Ces

agents sont aussi responsable d'estimer le temps d'aller-retour avec la source qui est utilisé par les récepteurs pour estimer le débit *TCP-Friendly*.
– La source reçoit un rapport de réception final qui lui permet de se représenter le nombre de groupes, le nombre de récepteurs par groupe ainsi que les caractéristiques de chaque groupe. Elle peut alors calculer le nombre de couches optimal, leurs débits ainsi que les niveaux de protection à utiliser (en fonction des débits demandés et des taux de perte observés) pour maximiser la qualité globale de réception.

À l'émetteur, les caractéristiques débit-distorsion de la source ainsi que le rapport agrégé final de réception sont utilisés pour maximiser la qualité globale de réception de la session. Le débit source et le débit FEC (c'est-à-dire le rapport $k/n$ du niveau de protection) sont optimisés conjointement. Le nombre de couches, leur débit ainsi que leur niveau de protection sont alors choisis de manière à maximiser le rapport signal-à-bruit (PSNR) global de réception.

## *3.4 Discussion*

Le protocole SARC a été implanté dans le simulateur réseau NS et expérimenté avec des codecs H.263+ et MPEG4. Ses performances ont été analysées en détail dans l'article [79] qui fait l'objet de l'Annexe B. Les simulations effectuées montrent que cet algorithme de contrôle de congestion parvient à suivre de manière précise les variations de caractéristiques des différents récepteurs tout en restant *TCP-Friendly*. Toutefois, comme la couche de base est supposée à débit constant, il est bon de préciser que les récepteurs doivent se désabonner complètement de la session vidéo si leurs caractéristiques ne leur permettent pas de recevoir le flot de base.

À notre connaissance, SARC est le seul protocole qui permet de choisir de manière dynamique en cours de session à la fois le nombre de couches transmises, leur débit et leur niveau de protection tout en conservant la propriété *TCP-friendly* avec les autres applications d'Internet.

Évidemment, ce mécanisme introduit des coûts supplémentaires et, en particulier, la nécessité d'agents d'agrégation placés à des endroits stratégiques dans le réseau (idéalement, à l'intérieur de certains routeurs). On peut envisager d'implanter ce type de fonctionnalités avec des *routeurs actifs*, en utilisant par exemple l'approche Chameleon [11]. Cependant, avec les problèmes d'ossification du réseau, il est peu probable que ce type de services actifs se répande un jour sur Internet. Une alternative pourrait bien être l'utilisation du mécanisme SARC dans un réseau multipoint applicatif de type *overlay* ou pair-à-pair. L'évaluation de cette solution alternative fait partie de mes perspectives de recherches qui fait l'objet de la section 6.1.

Le mécanisme SARC a été conçu à l'origine pour de la transmission multimédia en multipoint sur des réseaux filaire à commutation de paquets, et sans réservation de ressources. Une autre extension de ces travaux est de considérer non plus un

réseau filaire mais de multiples canaux sans fil, voir la section 6.3.

# 4. SUPPORT DE DIFFÉRENCIATION DE SERVICES POUR RÉSEAUX WIFI

## *4.1 Problématique*

Internet devient de plus en plus hétérogène. En quelques années seulement, les réseaux locaux sans fil (*wireless local area networks* ou WLAN) ont connu un déploiement extraordinaire, non seulement au sein des entreprises mais aussi chez les particuliers. Les réseaux locaux sans fil de type IEEE 802.11 sont les plus répandus aujourd'hui. Or, il est bien connu que les caractéristiques d'un canal de transmission sans fil sont très différentes de celles des liaisons filaires plus classiques d'Internet. Par exemple, la bande passante disponible est plus limitée : dans le cas de la norme 802.11b, le débit maximal au niveau physique est de 11 Mb/s et il est de 54 Mb/s pour les normes 802.11a et 802.11g. Le taux d'erreurs de transmission (*bit error rate* ou BER) est beaucoup plus élevé : il est de l'ordre de $10^{-5}$, à comparer à un BER inférieur à $10^{-9}$ pour une transmission sur fibre optique. De plus, les caractéristiques du canal de transmission sont très variables et dépendent de la distance entre la source et l'émetteur, des réflexions et de l'évanouissement éventuel du signal, de la vitesse du mobile, etc.

Avec un accroissement exponentiel de la puissance des stations de travail et une bande passante disponible de plus en plus élevée, les applications multimédias se sont rapidement développées sur Internet. Ces applications ont des caractéristiques sensiblement différentes de celles des protocoles FTP ou HTTP. En particulier, elles sont beaucoup plus sensibles à la latence et à la gigue sur le réseau. Elles sont aussi généralement plus gourmandes en bande passante mais la plupart d'entre elles peuvent tolérer un certain taux de perte de paquets ou de bit erronés. Le déploiement d'applications multimédias sur un réseau sans fil pose problème dès lors que les données multimédias sont en concurrence avec d'autres types de flots à transmettre. En effet, le principal mécanisme d'accès au médium (*medium access control* ou MAC) utilisé par les interfaces IEEE 802.11 est un mode d'accès avec contention au medium (distributed coordination function ou DCF) de type CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*).

Le principe de DCF est le suivant. Avant d'émettre, on vérifie que le canal de transmission est libre et on attend alors un temps fixe DIFS (*DCF InterFrame Space*), auquel est ajouté un délai variable et aléatoire calculé par l'émetteur dans une plage donnée appelée fenêtre de contention (CW, *Contention Window*). Ce délai aléatoire permet de réduire la probabilité de collisions des paquets sur la canal. Pendant le temps d'attente, l'émetteur continue d'écouter le canal de transmission.

Si le canal devient occupé, alors l'émetteur doit attendre la fin de la transmission en cours pour que le canal devienne libre de nouveau, et ensuite il attend un DIFS et le temps restant du délai aléatoire. Dès que le temps d'attente est terminé, si le canal est toujours libre, alors l'émetteur peut transmettre ses données. Les données transmises sont suivies d'un temps fixe appelé SIFS (*Short InterFrame Space*) et d'un accusé de réception en provenance du destinataire. La fenêtre de contention reprend la valeur minimale $CW_{min}$ et le cycle peut reprendre. En cas de collision, la fenêtre de contention est doublée (mais bornée par la valeur $CW_{max}$) afin de réduire la probabilité de collision. Le mode d'accès DCF fournit donc un service de type *best-effort*, sans aucune possibilité d'attribuer plus de priorité aux flots multimédias pour assurer une certaine qualité de service.

Un second mécanisme d'accès optionnel est décrit dans la norme MAC IEEE 802.11 [81]. Il s'agit du mode PCF (*Point Coordination Function*), un schéma d'élection centralisé contrôlé par un coordinateur situé au niveau du point d'accès (AP). En présence de ce mode, il y a alternance entre une période sans contention appelée CFP (*Contention Free Period*) et une période avec contention appelée CP (*Contention Period*). Pendant la période de non-contention, les stations ne peuvent pas émettre à moins que l'AP ne leur demande. Afin de s'assurer que les stations ne l'interrompent pas, le coordinateur utilise un délai d'accès au médium PIFS (*PCF InterFrame Space*) inférieur à celui des stations (DIFS). Les stations sont prévenues du mode de non-contention (pour éviter des interprétations de collision) par l'envoi de la trame beacon au départ de la session qui contient également la durée d'occupation du canal dans ce mode (le canal peut aussi être libéré par un signal de fin de période de non-contention appelé *CF-end*). Bien que ce mécanisme ait été conçu pour des applications multimédias qui nécessitent une certaine qualité de service, il n'est en pratique jamais utilisé. En fait, ce mécanisme a plusieurs inconvénients :

- il n'est pas possible d'effectuer de transmission directe entre deux stations (toutes les communications doivent transiter via le point d'accès,
- le temps alloué à une station est difficile à contrôler car une fois autorisée à émettre, la durée de transmission est très variable, la taille du paquet étant comprise entre 0 et 2346 octets,
- la transmission de la trame de balise (appelée plus couramment *beacon*) peut être retardée car la station ne vérifie pas que la fin de transmission du paquet intervient avant le début prévu du prochain beacon
- enfin, comme le mode PCF est optionnel, les différents constructeurs n'ont pas jugé utile son implantation, il est donc inutilisé en pratique.

Depuis l'élaboration du standard IEEE 802.11, de nombreuses études ont été menées afin de proposer un mécanisme d'accès au médium avec différenciation de services pour les réseaux locaux sans fil. Le groupe de travail 802.11e a été formé à l'IEEE dans le but de proposer un nouveau standard MAC avec support de qualité de services. Dans ce nouveau standard, un mécanisme d'accès au canal sans contention est défini pour des applications à débit constant (*constant bit rate* ou CBR). Nous avons proposé un nouveau mécanisme d'accès au canal appelé FHCF qui améliore les performances du protocole HCF en présence de flots non rigoureusement CBR.

Ainsi, ce nouveau protocole d'accès peut être utilisé par la majeure partie des applications multimédias, qui le plus souvent ne sont pas strictement CBR mais à débit variable (*variable bit rate* ou VBR).

La suite de ce chapitre est organisée de la manière suivante. Nous présentons tout d'abord un bref état de l'art des mécanismes de différenciation de services qui sont proposés dans le standard IEEE 802.11e. Puis nous présentons les principes du protocole FHCF. Enfin, nous discutons de l'utilité d'un tel mécanisme et de son utilisation éventuelle dans les réseaux locaux sans fil IEEE 802.11e à venir.

## 4.2 État de l'art : Le standard IEEE 802.11e

De nombreuses études ont été menées pour proposer un service différencié dans les réseaux IEEE 802.11. Nous avons dressé un état de l'art de ces travaux dans l'article [51] et d'une manière plus exhaustive dans un chapitre de livre [52]. Dans cette section, nous ne donnons qu'un aperçu du standard IEEE 802.11e en cours de standardisation et en particulier du mécanisme d'accès HCCA qui sert de base à notre étude.

Le standard IEEE 802.11e en cours d'élaboration introduit une nouvelle fonction de coordination appelée HCF (*Hybrid Coordination Function*). Celle-ci supporte deux nouveaux mécanismes d'accès avec différenciation de services : un mécanisme d'accès avec contention au médium appelé EDCA (*Enhanced Distributed Channel Access*) et un mécanisme d'accès sans contention appelé HCCA (*HCF Controlled Channel Access*). HCF introduit la notion de TXOP (*Transmission Opportunity*), un temps de transmission variable et alloué pour chaque station. Une station peut obtenir un TXOP en utilisant soit EDCA soit HCCA. Le TXOP lui garantit le droit d'utiliser le canal à un instant donné et pour une durée maximale prédéfinie. La valeur maximale du TXOP est communiquée à l'ensemble des stations dans la trame de beacon en utilisant le mode EDCA. Bien que HCF remplace la fonction de coordination DCF pour les stations qui utilisent 802.11e (ces stations sont appelées QSTA), l'interopérabilité est assurée avec les stations plus anciennes qui utilisent DCF.

### 4.2.1 Le mécanisme d'accès EDCA

EDCA est une extension du mécanisme d'accès avec contention DCF pour fournir un service différencié à l'utilisateur. Avec EDCA, chaque station dispose de quatre files d'attente de priorités différentes, l'utilisateur pouvant choisir jusqu'à 8 priorités différentes pour chacun de ses flots (les mêmes priorités que celles définies dans la norme IEEE 802.1D). Pour obtenir cette différenciation de services, on utilise des délais d'attentes IFS (*InterFrame Space*) différents pour chacune des files d'attente. Ce délai est appelé AIFS (*Arbitration IFS*) dans EDCA et remplace le DIFS qui est utilisé dans DCF. Ainsi, la file d'attente qui a le délai AIFS le plus court est celle qui est la plus prioritaire. De plus, des fenêtres de contention ($CW$)

différentes sont allouées pour chacune des files d'attente. La file d'attente avec la fenêtre de contention la plus courte devient ainsi la plus prioritaire. Dans le cas où la fenêtre de contention de plusieurs files d'attente arrivent à zéro au même instant, un ordonnanceur interne empêche la collision virtuelle en garantissant l'accès au médium à la file la plus prioritaire. Au même instant, les autres files d'attente doublent leur fenêtre de contention comme s'il y avait eu une collision sur le médium. Lorsqu'une station gagne le TXOP, elle peut transmettre plusieurs trames dans la mesure où la durée de transmission reste inférieure au TXOP.

Les valeurs utilisées pour les délais d'attente AIFS et pour gérer les fenêtres de contention de tailles variables sont annoncées à toutes les stations par le point d'accès (AP) dans la trame de beacon. Le standard IEEE 802.11e permet d'adapter ces paramètres de manière dynamique selon la charge du réseau mais aucun algorithme d'adaptation n'est proposé et le problème reste donc ouvert.

Les performances de EDCA peuvent rapidement se dégrader en cas de congestion dans le réseau. Les fenêtres de contention augmentent alors et de plus en plus de temps est gaspillé en attente plutôt que pour émettre les données. Les flots de données les plus prioritaires peuvent complètement inhiber l'émission des autres applications moins prioritaires. Pour ces raisons, un contrôle d'admission est prévu par l'AP pour contrôler l'accès aux files d'attente prioritaires. Pour des applications qui nécessitent des garanties strictes de délais, il est préférable d'utiliser le mécanisme d'accès HCCA.

### *4.2.2 Le mécanisme d'accès HCCA*

Le mécanisme d'accès HCCA (*HCF controlled channel access*) combine les avantages des modes DCF et PCF. Il utilise un coordinateur central appelé HC (*Hybrid Coordinator*) qui utilise des règles différentes de celles du mode PCF. Avec HCCA, le TXOP est alloué par l'AP et peut-être actif à la fois dans la période sans contention (CFP) mais aussi dans la période avec contention (CP). En effet, il est possible de découper l'intervalle de temps CP en une nouvelle période sans contention appelée CAP (*Controlled Access Phase*) qui utilise le mécanisme HCCA, et une période avec contention qui utilise EDCA. Les périodes CAP sont utiles pour rendre indépendante la fréquence d'émission des balises (beacons) des contraintes de latence que peuvent avoir les applications multimédias. D'autre part, pour remédier au phénomène de dé-synchronisation des beacons qui se produit avec le mode PCF avec HCCA, une station n'est autorisée à émettre un paquet que dans la mesure où sa transmission ne gène pas l'émission de la prochaine balise.

Afin de garantir un service différencié, le mécanisme HCCA se base sur une négociation de trafic TSPEC (*Traffic SPECification*) entre le point d'accès (AP) et les stations. Avant de transmettre un flot qui nécessite une garantie de service, un circuit virtuel appelé TS (*Traffic stream*) doit s'établir entre l'AP et les différentes stations pour échanger certains paramètres (comme le débit du flot, la taille des paquets, la latence maximale acceptable, etc.) En fonction des paramètres TSPEC, un ordonnanceur localisé dans l'AP calcule une durée de TXOP pour chacune des stations.

Le standard préconise l'utilisation d'un algorithme *round-robin* comme politique d'ordonnancement. Dans le cas où une station génère plusieurs flots avec garanties de services, un ordonnanceur localisé au sein-même de la station va allouer le TXOP qui lui a été attribué entre ses différents flots en tenant compte des priorités de ces derniers. Par ailleurs, le standard permet au point d'accès d'effectuer de manière optionnelle un contrôle d'admission avant d'accepter un nouveau TS.

Plusieurs études montrent que le mécanisme d'accès HCCA donne de bonnes performances pour des applications multimédias qui ont un débit constant [46]. En revanche, les performances se dégradent lorsque le débit de transmission de l'application n'est pas rigoureusement constant, des pertes de paquets peuvent se produire (ou des paquets qui arrivent trop tard pour être utilisés par le récepteur).

## 4.3 Contributions : le mécanisme d'ordonnancement FHCF

Le mécanisme d'ordonnancement FHCF (*Fair Hybrid Coordination Function*) que nous avons élaboré a pour but d'être efficace et équitable aussi bien pour des flux CBR que pour des flux VBR. Nous ne décrivons ici que les principes de ce protocole et invitons le lecteur à se reporter à l'Annexe C pour une description détaillée et une évaluation de ses performances.

FHCF se compose d'un ordonnanceur situé au niveau du point d'accès (QAP) et d'un autre ordonnanceur implanté au sein de chaque nœud. Schématiquement, l'ordonnanceur du QAP a pour rôle d'estimer les tailles de file d'attente de chaque station (QSTA) avant le prochain intervalle de service (SI). À la différence de l'ordonnanceur préconisé dans le mode d'accès HCCA qui alloue les temps d'accès au medium en fonction des taux moyens d'arrivée de données, l'algorithme d'ordonnancement FHCF utilise des estimations sur les longueurs des files d'attente pour affiner les allocations de temps d'accès des différentes stations du réseau. Plus précisément, l'ordonnanceur utilise une fenêtre d'erreurs d'estimation précédentes pour allouer le TXOP alloué à chaque station. Comme les TXOP sont alloués de manière globale à chaque QSTA, un ordonnanceur est nécessaire au sein de chaque QSTA pour redistribuer le TXOP entre les différents flots de traffic (TS) de la station.

Nous avons comparé les performances des différents mécanismes dans le simulateur réseau NS-2 et les résultats montrent que le protocole FHCF permet d'être équitable envers les différentes stations tout en respectant les besoins en bande passante et en délai de chaque flux indépendamment de la charge du réseau.

## 4.4 Discussion

Comme nous l'avons mentionné dans l'introduction, le support de qualité de service dans les réseaux sans fil est indispensable pour les applications multimédias en présence de contention sur le médium avec d'autres flots. En effet, dans les réseaux sans fil, la bande passante disponible est souvent beaucoup moins importante

que dans les réseaux filaires, le taux d'erreurs de transmission y est plus élevé et les caractéristiques du canal varient fortement au cours du temps.

Il est légitime de se demander pourquoi les mécanismes de support de QoS dans les réseaux sans fil ne connaîtraient pas les mêmes déboires que les protocoles de réservation de ressources IntServ ou DiffServ proposés à l'IETF pour Internet. Le risque d'échec semble tout de même moins élevé, en particulier si on se limite à l'utilisation du support de qualité de services au sein d'un réseau local ou d'un *hotspot*. On évite ainsi le problème de "fonctions coûteuses" à implanter au sein des routeurs dans le cœur du réseau et le problème de scalabilité avec un nombre de flots élevé. On se débarrasse aussi des problèmes de gestion de trafic aux frontières des systèmes autonomes (AS). Cela dit, pour maximiser les chances de déploiement de ces nouveaux mécanismes, il est préférable d'éviter de les proposer comme mode optionnel dans les différents standards. Cela pourrait avoir les mêmes conséquences que pour le mécanisme d'accès sans contention *Point Coordination Function (PCF)* qui a été proposé en option dans le standard IEEE 802.11. Ce mécanisme a été jugé trop complexe à mettre en œuvre par les fabricants et n'a pas pu être déployé [51].

Reste aussi les problèmes de compatibilité avec l'existant ; il est clair qu'il vaut mieux prévoir ces nouveaux mécanismes dès la vente des premières cartes sur le marché afin d'éviter les problèmes de compatibilité ou de mauvaises performances en présence d'anciens composants. Toutefois, on peut noter que ce problème est moins grave pour les WLAN que dans le cas d'Internet en raison des coûts beaucoup moins élevés pour remplacer les anciens composants. Mais pour envisager le remplacement des anciennes cartes, une forte incitation est indispensable comme un gain en performance très élevé ou la possibilité d'executer de nouvelles applications.

Dans l'hypothèse où le support de qualité de services connaît un essor important, il faut veiller à la bonne utilisation de ce service par les utilisateurs. Par exemple, comment peut-on s'assurer que les utilisateurs n'abuseront pas de ce service ? Un tel service est efficace dans la mesure où les utilisateurs choisissent de manière pertinente la classe de services en fonction des caractéristiques de l'application. Une solution est de jumeler le protocole de contrôle d'accès avec un mécanisme de contrôle des flots émis par les différentes stations. Il serait alors possible de s'assurer que les services à forte contrainte de latence sont bien utilisés par des applications multimédias. Ce type de vérification complète certaines études récentes visant à détecter de mauvais comportements des utilisateurs dans les réseaux WIFI [57] ou à estimer les ressources du WLAN consommées par chaque utilisateur [19]. On peut noter qu'il existe d'autres méthodes utilisant la tarification lors de la réservation de ressources dans le réseau sans fil, ce qui permet de limiter la demande en incitant les utilisateurs à utiliser de manière appropriée le canal de transmission [44].

# 5. CONTRÔLE DE TRANSMISSION MULTIMÉDIA INTERCOUCHES POUR RÉSEAUX WIFI

## *5.1 Problématique*

Dans le chapitre précédent, nous avons considéré le support d'une différenciation de services au niveau de la couche MAC afin améliorer la transmission de flots multimédias VBR. De cette manière on prend en compte les caractéristiques de l'application pour choisir le service le plus approprié au niveau de la couche d'accès au médium. Dans ce chapitre, on va étendre cette approche d'optimisation intercouches un cran plus loin, en essayant de choisir les meilleurs paramètres de transmission de la couche physique en fonction non plus uniquement des caractéristiques du canal de transmission mais aussi en prenant en compte les besoins des applications. La mise en œuvre de ces interactions est facilité par l'approche radio logicielle qui ouvre de nouveaux horizons. Avec la montée en puissance très rapide des machines, l'implantation logicielle devient une alternative attirante pour les fonctions de transmission d'information. Elle permet en effet une plus grande flexibilité par rapport aux solutions matérielles figées ; il devient possible de modifier n'importe quel paramètre de transmission de manière dynamique. Évidemment, il faut veiller à ce que le coût de tels mécanismes reste en deçà du gain escompté pour leur utilisation. Ici, nous n'envisageons pas une optimisation conjointe de toutes les couches de la pile de communication, ceci est trop ambitieux étant donné la très grande complexité à mettre en œuvre et la diversité des expertises requises. Nous nous penchons plutôt sur le réglage de certains paramètres de la couche physique en fonction d'informations en provenance de couches supérieures concernant les caractéristiques des flots à transmettre. L'approche classique consiste à optimiser chaque couche du protocole de transmission de manière indépendante aux autres couches. En l'occurrence, pour la couche physique, on vise en général à rendre le canal de transmission le plus fiable possible indépendamment des caractéristiques des données que l'on à transmettre. Lorsque le récepteur reçoit un paquet qui contient des bits erronés, il est automatiquement écarté et ne parvient pas à la couche liaison. La couche liaison de l'émetteur, en ne recevant pas d'acquittement va s'apercevoir de la perte du paquet et procéder éventuellement à une retransmission. Si un tel mécanisme est désirable pour des applications classiques comme la transmission de fichiers, il l'est beaucoup moins pour les applications qui ont des besoins en interactivité plus importants et qui peuvent tolérer un certain taux d'erreurs de transmission. En effet, certaines applications multimédias utilisent des codages qui peuvent concilier soit un certain taux d'effacement de paquets soit un certain taux de bits erronés. Dans ce dernier

cas, pour profiter de la robustesse du codage, il est nécessaire de faire remonter les paquets erronés à l'application et ne pas les rejeter dès la couche physique. C'est précisément ce type de solution que nous envisageons dans ce chapitre et dont nous discutons après un bref état de l'art des mécanismes d'optimisation inter-couches.

## 5.2 État de l'art : mécanismes d'interactions intercouches

L'architecture OSI [20] avec ses interfaces standardisées permet de simplifier l'élaboration des protocoles de communication en optimisant chaque couche de manière indépendante des autres. Cependant, cette architecture est remise en cause avec les réseaux sans fil principalement en raison des caractéristiques du canal de transmission qui sont beaucoup moins prévisibles que dans le cas des réseaux filaires traditionnels. En effet, pour réagir de manière efficace à de brusques changements de caractéristiques du réseau, il devient indispensable d'échanger des informations entre plusieurs couches à la fois, voire d'optimiser de manière conjointe différentes couches. Depuis quelques années, différentes études ont été menées dans cette optique, nous n'en citons ici que quelques exemples.

De nouvelles interactions entre la couche MAC et la couche physique des réseaux WIFI sont proposées dans les articles [27] et [84]. L'idée de base est de renvoyer à l'émetteur des estimations sur les caractéristiques du canal de transmission pour choisir le mode de transmission [27] ou adapter la taille du diagramme de constellation de la modulation.

D'autres types d'interactions entre la couche réseau et la couche MAC sont proposées dans l'article [59] pour améliorer les performances du protocole de routage dans les réseaux ad-hoc. La couche MAC fournit à la couche réseau une indication sur la réception des trames de contrôle CTS et ACK afin qu'elle choisisse la route qui minimise la probabilité d'erreurs de transmission.

Des mécanismes d'optimisation entre la couche MAC et la couche application sont décrits dans l'article [31] pour transmettre de la vidéo robuste sur réseaux WIFI. Ces mécanismes incluent une nouvelle stratégie de retransmission de paquets au niveau MAC, un mécanisme de correction d'erreurs dans la couche d'application et l'utilisation d'un codage vidéo *scalable* pour adapter la compression en fonction de la bande passante disponible.

La nouvelle architecture en couches proposée dans le projet Européen *Mobileman* vise à permettre aux protocoles implantés dans les différentes couches de coopérer entre eux en partageant une information d'état du réseau, tout en maintenant une claire séparation entre les différentes couches de la pile de communication. Ces mécanismes sont proposées pour résoudre les problèmes de sécurité, support de qualité de services et gestion de l'énergie.

On peut noter également les efforts pour prendre en compte le *co-design* des différentes couches dans les standards récents comme CDMA2000, BRAN HiperLAN2 et 3GPP. L'IEEE dans le cadre du groupe d'étude MBWA (*Mobile Broadband Wireless Access*) étudie des mécanismes d'optimisation inter-couches pour accroître

le débit de transmission et réduire la latence des communications.

## 5.3 Contributions : le mécanisme MORSA

Nous avons élaboré le mécanisme MORSA (*Media-Oriented Rate Selection Algorithm*) de sélection de mode de transmission pour les réseaux locaux sans fil 802.11 dont l'objectif est de s'adapter à la fois aux conditions du canal et aux caractéristiques du média transmis. Cette section présente les principes de cet algorithme qui est détaillé dans l'Annexe D.

Ce mécanisme utilise des interactions entre la couche MAC, la couche application et la couche physique afin d'améliorer la transmission de flots multimédias qui sont robustes aux erreurs de bits sur le canal. Pour cette classe d'application, la qualité vidéo au niveau du récepteur résulte d'un compromis entre le débit et la fiabilité de la transmission. En informant la couche MAC que l'application peut tolérer un certain pourcentage de bits erronés, on peut modifier l'algorithme de sélection de mode de transmission pour qu'il prenne en compte non seulement les caractéristiques du canal de transmission mais aussi les celles des flots de données. Nous avons élaboré un mécanisme dans lequel la source a la possibilité de spécifier une certaine qualité de services (comme le débit de transmission et sa tolérance aux erreurs de transmission binaire) afin que le récepteur puisse sélectionner le meilleur mode de transmission. Pour cela, la source inclut dans le paquet RTS une indication sur le taux de tolérance de BER supporté par l'application, ce qui permet au récepteur, en fonction du rapport signal-à-bruit (SNR) du canal, de choisir le meilleur mode de transmission (débit, modulation, code correcteur d'erreur). On utilise pour cela l'algorithme de sélection de mode RBAR [27] que l'on a modifié pour prendre en compte la tolérance au BER. Le mode de transmission que l'on vient de calculer est alors envoyé à la source dans le paquet CTS et la source peut ainsi utiliser le mode de transmission optimal pour émettre ses données. Lorsque le paquet de données arrive au récepteur, comme l'entête PLCP inclut une indication de tolérance de BER des données, le récepteur peut décider de laisser remonter les paquets erronés.

L'algorithme que nous avons proposé peut utiliser le mécanisme d'accès EDCA de la norme IEEE 802.11e afin de distinguer les caractéristiques des différentes applications. Comme nous l'avons décrit dans la section 4.2.1, avec EDCA chaque station dispose de quatre files d'attente de priorités différentes. De cette manière, on peut distinguer jusqu'à quatre niveaux de tolérance aux erreurs de transmission.

Nous avons comparé les performances obtenues avec notre mécanisme avec d'autres algorithmes de sélection de débit pour un réseau local sans fil 802.11a. Avec un flot vidéo qui tolère un BER de $10^{-3}$, le nouveau mécanisme permet un gain maximal de 5 Mb/s sur le débit reçu ou d'accroître de 20 mètres la portée de la transmission.

## *5.4 Discussion*

Le but de cette étude est d'évaluer l'intérêt de faire interagir différentes couches de la pile de communication ensemble pour choisir les paramètres de transmission qui maximisent la qualité vidéo au niveau des récepteurs. En l'occurrence, nous avons évalué quel était le gain potentiel pour une application vidéo lorsque l'algorithme de sélection de mode prend non seulement en compte les caractéristiques du canal de transmission mais aussi celles du codage vidéo. L'algorithme que nous avons présenté fait interagir les couches application, MAC et physique pour choisir le meilleur mode de transmission. Il a été appliqué aux réseaux sans fil 802.11 mais la même idée peut très bien être adaptée à d'autres types de réseaux sans fil.

Dans le cas du 802.11, le mécanisme que nous avons élaboré nécessite des modifications aux standards. Pour pouvoir profiter de la robustesse du codage aux erreurs, il est indispensable que les paquets qui contiennent des bits erronés puissent remonter toutes les couches protocolaires. Cela implique des modifications dans le standard. Tout d'abord, la couche physique ne doit pas les rejeter. Le code détecteur d'erreurs (CRC) au niveau de la couche MAC ne doit plus couvrir l'ensemble du paquet, mais uniquement les en-têtes MAC, IP, UDP et RTP. De plus, pour éviter que la couche transport ne rejette les paquets erronés, il faut invalider la détection optionnelle d'erreurs du protocole UDP. Enfin, il est souhaitable d'émettre les en-têtes MAC, IP, UDP et RTP avec le mode de transmission le moins sensible aux erreurs, comme c'est le cas pour l'entête de la couche physique (PLCP). Cela permet de rendre encore plus robuste notre mécanisme contre les erreurs de transmission. Par ailleurs, les paquets de contrôle RTS et CTS doivent être modifiés de manière à inclure le SNR (comme c'est le cas dans l'algorithme RBAR [27]) ainsi que l'indication de tolérance aux erreurs et le mode de transmission optimal.

En ce qui concerne l'implantation du mécanisme, sa complexité est très faible, car l'algorithme peut utiliser des tables pré-calculées pour différentes valeurs de SNR et de tolérance aux erreurs de transmission. La plus forte contrainte est la nécessité de pouvoir distinguer différents types de flots, c'est-à-dire la disponibilité du mécanisme d'accès EDCA.

Bien qu'il semble peu probable que ce mécanisme se déploie un jour dans les réseaux 802.11 en raison d'incompatibilité avec les stations WIFI existantes, cette étude met en évidence un gain appréciable pour les applications robustes aux pertes et donc, l'intérêt d'intégrer ce genre de mécanismes inter-couches dans les nouveaux standards de communication sans fil.

# 6. PERSPECTIVES DE RECHERCHE

Pour conclure le mémoire, je présente ici quelques axes de recherche que je souhaite développer, certains travaux ayant déjà commencé.

## 6.1 Transmission multimedia en multipoint entre pairs

Comme nous l'avons déjà mentionné à plusieurs reprises, dans l'Internet, il devient très difficile d'implanter de nouveaux mécanismes à l'intérieur des routeurs. Même si certaines compagnies comme CISCO parviennent plus ou moins rapidement à proposer de nouvelles fonctionnalités dans les routeurs, leur déploiement pose toujours de réels problèmes ; on peut prendre par exemple les mécanismes de différenciation de services, IPv6, ou les protocoles de transmission multipoint. Une solution qui nous reste est de proposer les nouvelles fonctionnalités non plus au sein des routeurs mais dans les machines elles-mêmes connectées au réseau (terminaux). Fournir les nouveaux services au niveau des pairs peut ainsi devenir une solution à l'ossification du réseau.

Les mécanismes SCORE et SARC que nous avons proposés dans les chapitres 2 et 3 font l'hypothèse que le service multipoint ASM est disponible au niveau de tous les terminaux, ce qui est malheureusement loin d'être la réalité. Cela limite fortement le déploiement de ces protocoles. Dans le cas du protocole SARC, l'utilisation d'un mécanisme multipoint au niveau applicatif (entre pairs), permettrait aussi d'implanter la fonction d'agrégation des rapports de réception à moindre coût. L'idéal serait d'élaborer des protocoles de communication qui puissent tirer profit du multipoint natif là où il est disponible et de permettre à des terminaux qui n'ont pas ce service de pouvoir néanmoins recevoir les différents flots de données que ce soit pour des application monde virtuel à grande échelle ou pour des transmissions multimédias plus classiques comme le *streaming*. La conception de tels mécanismes est nécessaire pour pouvoir gérer l'hétérogénéité entre les terminaux qui ne cesse d'augmenter sur Internet.

## 6.2 Validation des protocoles de transmission sur WLANs

L'élaboration de nouveaux mécanismes d'accès dans le monde sans-fil (comme le support de différenciation de services dans le cadre du standard IEEE 802.11e) nécessite une phase de validation souvent délicate. L'idéal est de pouvoir simuler ces nouveaux mécanismes de manière la plus réaliste possible mais aussi de les

implanter dans des cartes WIFI et de les expérimenter en vrai. Notons qu'il n'est pas toujours possible d'implanter de nouveaux algorithmes dans les cartes du commerce car quelquefois des modifications sont nécessaires à l'intérieur du firmware, ce qui empêche toute expérimentation à moindre coût. De plus, il est très difficile de maîtriser les caractéristiques d'un réseau sans fil en raison des interférences et des phénomènes d'évanouissement du signal qui sont souvent imprévisibles, ce qui rend très complexe la comparison de performance des différents mécanismes entre eux avec les mêmes conditions du canal de transmission.

Pour aider à la mise en place de ces expérimentations, nous envisageons d'élaborer des sondes WIFI intelligentes qui permettront non seulement d'identifier les différentes stations connectées à un instant donné au réseau mais surtout de caractériser les conditions du canal (charge du réseau, niveau d'interférences) mais aussi de manière précise les différents flots émis par les stations (débit obtenu, taux d'erreur, latence, jigue). L'objectif est de pouvoir analyser précisément les performances obtenues lors des expérimentations mais aussi d'utiliser les sondes pour adapter en temps réel les paramètres utilisés dans le point d'accès et les différentes stations comme par exemple la taille des fragments ou les paramètres de différenciation de services.

En ce qui concerne la simulation des nouveaux mécanismes proposés pour les réseaux sans fil, aujourd'hui la plupart d'entre eux sont implantés et simulés dans le simulateur réseau NS[1]. Il s'agit du simulateur le plus répandu dans la communauté scientifique réseau car c'est un outil disponible dans le domaine public qui intègre la plupart des protocoles de transmission. Cependant, les modules WIFI fournis dans ce simulateur sont peu documentés, difficilement lisibles et les modèles implantés pour simuler la couche physique ne sont pas très réalistes. Très souvent, chaque chercheur utilise sa propre version de NS, avec des correctifs dont il est l'auteur ou qu'il a récupérés en dehors de la version officielle de NS. Ceci rend la validation des protocoles et la comparaison des différents algorithmes entre eux difficile. Nous venons de lancer un projet d'ingénieur d'une année sur la réécriture de modules MAC et PHY IEEE 802.11 plus réaliste dans le simulateur NS. On aura besoin de modèles les plus proches possible de la réalité pour simuler les caractéristiques de la couche physique. Pour cela nous envisageons une collaboration avec des experts de la couche physique des réseaux sans fil à Eurecom et au LIP6. L'objectif est d'intégrer à terme les nouveaux modules développés dans la release officielle de NS afin que toute la communauté scientifique réseau puisse disposer d'un simulateur réseau sans-fil public performant.

## *6.3 Transmission multimédia en multipoint sur canaux sans fil multiples*

Récemment, différents projets sont à l'étude pour accroître les performances en bande passante disponible des réseaux sans fil. Pour citer un exemple, le groupe 802.11n de l'IEEE a pour objectif d'augmenter fortement la capacité des liaisons

sans fil en utilisant une technologie MIMO (multiple-input/multiple-output) et une modulation multi-porteuse de type OFDM.

L'utilisation de canaux sans fil multiples peut aussi s'avérer intéressante dans le cas de transmission en multipoint vers un groupe de récepteurs. Comme dans le cas filaire, les récepteurs sans fil peuvent avoir des caractéristiques différentes, et il faut pouvoir gérer cette hétérogénéité (caractéristiques des canaux sans fil, mais aussi des terminaux en particulier la taille de l'écran ainsi que la capacité CPU disponible). Une solution peu efficace serait de s'adapter au récepteur le plus lent. Mais étant donné que dans un environnement sans fil, le récepteur le plus lent peut souvent avoir un débit quasi nul, s'adapter à ce récepteur pénaliserait l'ensemble des récepteurs. Une solution convenable dans un tel environnement est d'utiliser un codage à descriptions multiples (*Multiple Description Coding*, MDC) : les récepteurs ont alors la possibilité de s'abonner aux couches en fonction de leurs ressources disponibles. Outre les problèmes classiques du choix optimal du nombre de couches, du débit et du taux de protection par couche, il faut pouvoir choisir dynamiquement le *mapping* entre les différentes couches vidéo et les meilleurs canaux sans fil disponible du moment, ainsi que la politique d'accès simultanée aux différents canaux sans fil. La couche MAC doit pouvoir prendre en considération les caractéristiques temps réel des flots multimédias, mais aussi les caractéristiques hétérogènes des différents récepteurs. Une collaboration avec le projet TEMICS de l'IRISA spécialisé en codage source vidéo est en cours sur cette thématique.

# BIBLIOGRAPHIE


[1] NS-2 : The Network Simulator. Software and documentation available via URL http ://www.isi.edu/nsnam/ns.

[2] I. Aad, Q. Ni, C. Barakat, and T. Turletti. Enhancing IEEE 802.11 MAC in congested environments. Technical report, Boston, MA, USA, August 2004.

[3] I. Aad, Q. Ni, C. Barakat, and T. Turletti. Enhancing IEEE 802.11 MAC in congested environments. *to appear in Elsevier Computer Communications Journal (Special Issue on ASWN 2004)*, 2005.

[4] P. Ansel, Q. Ni, and T. Turletti. An Efficient Scheduling Scheme for IEEE 802.11e. In *IEEE Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Workshop (WiOpt)*, University of Cambridge, UK, March 2004.

[5] V. Arya and T. Turletti. AED : An Accurate and Explicit Loss Differentiation Mechanism. In *1st Workshop on Mobile and Wireless Networks (MWN)*, May 2003.

[6] Steve Benford and Chris Greenhalgh. Introducing third part objects into the spatial model of interaction. Technical Report CGR-97, 1997.

[7] J. Bolot and T. Turletti. Adaptive Error Control for Packet Video in the Internet. In *ICIP*, Lausanne, CH, September 1996.

[8] J. Bolot and T. Turletti. Experience with Rate Control Mechanisms for Packet Video in the Internet. *ACM SIGCOMM Computer Communication Review*, 28(1) :4–15, January 1998.

[9] J.C. Bolot and T. Turletti. A Rate Control Mechanism for Packet Video in the Internet. In *IEEE INFOCOM*, Toronto, Canada, June 1994.

[10] J.C. Bolot, T. Turletti, and I. Wakeman. Scalable Feedback Control for Multicast Video Distribution in the Internet. *ACM SIGCOMM Computer Communication Review*, 24(4) :58–67, October 1994.

[11] M. Bossardt, R.H. Antink, A. Moser, and B. Plattner. Chameleon : Realizing automatic service composition for extensible active routers. In *Proceedings Fifth Annual International Working Conference on Active Networks*, December 2003.

[12] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shave. FLID-DL : Congestion control for layered multicast. In *Proceedings of the Second International Workshop on Networked Group Communication, NGC'00*, pages 71–81, Stanford, CA, USA, November 2000.

[13] C. Diot and C. Huitema and T. Turletti. Network Conscious Applications. In *HPCS*, Mystic, Connecticut, April 1995.

[14] C. Huitema and T. Turletti. Software Codecs and Workstation Video Conferences. In *INET*, Kobe, Japon, June 1992.

[15] G.R. Cantieni, Q. Ni, C. Barakat, and T. Turletti. Performance Analysis under Finite Load and Improvements for Multirate 802.11. *Special issue of Elsevier Computer Communications journal*, December 2004.

[16] Carlsson and Hagsand. Dive - a multi user virtual reality system. In *Proceedings IEEE VRAIS, Seattle, Washington*, September 1993.

[17] I. Chrisment, D. Kaplan, and C. Diot. An alf communication architecture : Design and automated implementation. *IEEE JSAC special issue on Architectures for the 21st Century*, 16(3), 1998.

[18] W. Dabbous and T. Turletti. *Multicast Multimédia sur Internet, Traité Collection IC2, Série Réseaux et Télécoms*, chapter Le Multipoint pour les Environnements Virtuels à Grande Échelle. HERMES Science Publications, March 2005.

[19] M. Davis. A wireless traffic probe for radio resource management and qos provisioning in ieee 802.11 wlans. In *Proceedings of the third ACM MSWiM*, Venice, Italy, October 2004.

[20] J. Day. The (un)revised osi reference model. *SIGCOMM Comput. Commun. Rev.*, 25(5) :39–55, 1995.

[21] Adrien Felon. Diffusion, cohérence et contraintes temporelles dans les mondes virtuels 3d - système vreng, September 1999.

[22] Chris Greenhalgh. Dynamic, embodied multicast groups in massive-2. Technical Report NOTTCS-TR-96-8 1, 1996.

[23] Q. Guo, Q. Zhang, W. Zhu, and Y.-Q. Zhang. A sender-adaptive and receiver-driven layered multicast scheme for video over internet. In *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'01*, Sydney, Australia, May 2001.

[24] T. Turletti H. Kim and A. Bouali. EPSPECTRA : A Formal Toolkit for Developing DSP Software Applications. *to appear in Theory and Practice of Logic Programming*, 2005.

[25] X. Hénocq, F. Le Léannec, and C. Guillemot. Joint source and channel rate control in multicast layered video transmission. In *Proceedings of SPIE International Conference on Visual Communication and Image Processing, VCIP'2000*, pages 296–307, June 2000.

[26] C. Hoffmann, M.H. Manshaei, and T. Turletti. CLARA : Closed-Loop Adaptive Rate Allocation for IEEE 802.11 Wireless LANs. In *IEEE WirelessCom'05*, Hawai, USA, June 2005.

[27] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 236–251, 2001.

[28] H. Kim and T. Turletti. An Esterel-based development environment for designing software radio applications. INRIA Research Report RR-4256, INRIA, September 2001.

[29] H. Kim and T. Turletti. Implementation of an Esterel-based Toolkit for Designing DSP Software Applications. In *the 5th International Conference on Real-Time Computing Systems and Applications*, Japan, 2002.

[30] H. Kim, T. Turletti, and A. Bouali. EPspectra : A Formal Approach to developing DSP Software Applications. INRIA Research Report RR-4293, INRIA, October 2001.

[31] S. Krishnamachari, M. VanderSchaar, S. Choi, and X Xu. Video streaming over wireless lans : A cross-layer approach. In *Proceedings of Packet Video Workshop, PV'03*, Nantes, France, April 2003.

[32] S. Kumary, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley. The MASC/BGMP Architecture for Inter-domain Multicast Routing. In *Proceedings ACM SIGCOMM'98*, sept 1998.

[33] M. Lacage, M.H. Manshaei, and T. Turletti. A Practical Approach to Rate Adaptation. In *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Venice, Italy, October 2004.

[34] J.B. Lapeyrie and T. Turletti. FPQ : a Fair and Efficient Polling Algorithm with QoS support for Bluetooth Piconet. In *IEEE INFOCOM*, San Francisco, CA, April 2003.

[35] A. Legout and E. W. Biersack. Pathological behaviors for RLM and RLC. In *Proceedings of International Conference on Network and Operating System Support for Digital Audio and Video, NOSSDAV'00*, pages 164–172, Chapel Hill, North Carolina, USA, 2000.

[36] A. Legout and E. W. Biersack. PLM : Fast convergence for cumulative layered multicast transmission schemes. In *Proceedings of ACM SIGMETRICS'00*, pages 13–22, Santa Clara, CA, USA, 2000.

[37] B. N. Levine and J. J. Garcia-Luna-Aceves. Improving internet multicast with routing labels. In *Proceedings ICNP'97*, 1997.

[38] John Locke. An introduction to the internet networking environment and simnet/dis. Technical report, by John Locke (Internet : jxxl@cs.nps.navy.mil) Computer Science Department,, October 1995.

[39] E. Léty and T. Turletti. Issues in Designing a Communication Architecture for Large-Scale Virtual Environments. In *1st International Workshop on Networked Group Communication*, Pisa, Italy, November 1999.

[40] E. Léty, T. Turletti, and F. Baccelli. SCORE : a Scalable Communication Protocol for Large-Scale Virtual Environments. *IEEE/ACM Transactions on Networking*, April 2004.

[41] M Luby and V. Goyal. Wave and equation based rate control building block. *IETF Internet Draft draft-ietf-rmt-bb-webrc-01*, June 2002.

[42] Michael R. Macedonia. *A Network Software Architecture for Large-scale Virtual Environments*. Ph.d. dissertation, June 1995.

[43] Michael R. Macedonia, Michael J. Zyda, David R. Pratt, Paul T. Barham, and S. Zeswitz. Npsnet : A network software architecture for large scale virtual environments. *MIT Presence 3(4)*, 1994.

[44] P. Maillé and B. Tuffin. Multi-bid auctions for bandwidth allocation in communication networks. In *IEEE INFOCOM*, 2004.

[45] M. Malli, Q. Ni, T. Turletti, and C. Barakat. Adaptive Fair Channel Allocation for QoS Enhancement in IEEE 802.11 Wireless LANs. In *IEEE International Conference on Communications (ICC)*, Paris, France, June 2004.

[46] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. Ieee 802.11e wireless lan for quality of service. In *European Wireless*, Florence, Italy, February 2002.

[47] M.H. Manshaei, T. Turletti, and T. Guionnet. An Evaluation of Media-Oriented Rate Selection Algorithm for Multimedia Transmission in MANETs. *EURASIP Journal on Wireless Communications and Networking, Special Issue on Ad Hoc Networks : Cross-Layer Issues*, 1st Quarter 2005.

[48] M.H. Manshaei, T. Turletti, and M.N. Krunz. Media-Oriented Transmission Mode Selection in 802.11 Wireless LAN. Research Report 4958, INRIA, October 2003.

[49] M.H. Manshaei, T. Turletti, and M.N. Krunz. Media-Oriented Transmission Mode Selection in 802.11 Wireless LAN. In *IEEE WCNC*, Atlanta, Georgia, USA, March 2004.

[50] S. McCanne, M. Vetterli, and V. Jacobson. Low-complexity video coding for receiver-driven layered multicast. *IEEE Journal on Selected Areas In Communications*, 15(6) :983–1001, August 1997.

[51] Q. Ni, L. Romdhani, and T. Turletti. A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN. *Wireless Communication and Mobile Computing (WCMC)*, 4(5) :547–566, August 2004.

[52] Q. Ni and T. Turletti. *Wireless LANs and Bluetooth*, chapter QoS Support for IEEE 802.11 WLAN. Nova Science Publishers, 2005.

[53] K. Park and R. Kenyon. Effects on network characteristics on human performance in a collaborative virtual environment. In *IEEE Virtual Reality (VR)*, Houston, Texas, USA, March 1999.

[54] T. Parmentelat, L. Barza, T. Turletti, and W. Dabbous. A Scalable ssm-based Multicast Communication Layer for Multimedia Networked Virtual Environments. Research Report 5389, INRIA, November 2004.

[55] T. Parmentelat, A. Gourdon, T. Turletti, and E. Larreur. A Very Large Virtual Environment for Multimedia Conferencing. Technical Report 0296, INRIA, May 2004.

[56] T. Turletti Q. Ni, Tianji Li and Y. Xiao. Saturation throughput analysis of error-prone 802.11 wireless networks. *to appear in Wireless Communications and Mobile Computing (WCMC)*, 2005.

[57] M. Raya, J.-P. Hubaux, and I. Aad. Domino : A system to detect greedy behavior in ieee 802.11 hotspots. In *Proceedings of ACM Mobisys*, Boston, USA, June 2004.

[58] L. Romdhani, Q. Ni, and T. Turletti. AEDCF : Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks. In *IEEE WCNC*, New Orleans, Louisiana, March 2003.

[59] A. Safwat, H. Hassanein, and H. Mouftah. Optimal cross-layer designs for energy-efficient wireless ad hoc and sensor networks. In *22nd IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Phoenix, Arizona, USA, April 2003.

[60] K. Salamatian and T. Turletti. Classification of Receivers in Large Multicast Groups using Distributed Clustering. In *Packet Video Workshop*, Taejon, Korea, May 2001.

[61] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson. RTP : A Transport Protocol for Real-Time Applications. Rfc-1889, IETF Standards Track RFC, January 1996.

[62] D. Sisalem and A. Wolisz. Mlda : A tcp-friendly congestion control framework for heterogeneous multicast environments. In *Proceedings of International Workshop on Quality of Service, IWQoS'00*, Pittsburgh, PA,USA, June 2000.

[63] IEEE Computer Society, editor. *IEEE Standard for Interactive Distributed Simulation*. Number Std 1278.2. 1995.

[64] Tony Speakman, Dino Farinacci, Steven Lin, Jon Crowcroft, Dan Leshchiner, Jim Gemmell, Michael Luby, Todd Montgomery, and Luigi Rizzo. Pgm reliable transport protocol. Internet-draft, April 2000.

[65] D. Tennenhouse, T. Turletti, and V. Bose. The SpectrumWare Testbed for ATM-based Software Radios. In *ICUPC*, Boston, MA, September 1996.

[66] T. Turletti. H.261 Software Codec for Videoconferencing over the Internet. INRIA Research Report 1834, INRIA, Janvier 1993.

[67] T. Turletti. The INRIA Videoconferencing System (IVS). *ConneXions - The Interoperability Report*, 8(10) :20–24, October 1994.

[68] T. Turletti. A brief overview of the GSM Radio Interface. Technical Memorandum TM-547, MIT, March 1996.

[69] T. Turletti, H. Bentzen, and D.L. Tennenhouse. Towards the Software Realization of a GSM Base Station. *IEEE/JSAC, Special Issue on Software Radios*, 17(4) :603–612, April 1999.

[70] T. Turletti and C. Huitema. RTP Payload Format for H.261 Video Streams. Rfc-2032, IETF Standards Track RFC, October 1996.

[71] T. Turletti and C. Huitema. Videoconferencing in the Internet. *IEEE/ACM Transactions on Networking*, 4(3) :340–351, 1996.

[72] T. Turletti, S.F. Parisis, and J. Bolot. Experiments with a Layered Transmission Scheme over the Internet. INRIA Research Report 3296, INRIA, November 1997.

[73] T. Turletti and D.L. Tennenhouse. Estimating the Computational Requirements of a Software GSM Base Station. In *ICC*, Montreal, Canada, June 1997.

[74] T. Turletti and D.L. Tennenhouse. Complexity of a Software GSM Base Station. *IEEE Communication Magazine*, 37(2) :113–117, February 1999.

[75] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of the Conference on Computer Communications, IEEE Infocom'98*, pages 996–1003, San Francisco, USA, March 1998.

[76] B. J. Vickers, C. Albuquerque, and T. Suda. Source adaptive multi-layered multicast algorithms for real-time video distribution. *IEEE/ACM Transactions on Networking*, 8(6) :720–733, December 2000.

[77] J. Vieron, T. Turletti, X. Henocq, C. Guillemot, and K. Salamatian. TCP-Compatible Rate Control for FGS Layered Multicast Video Transmission based on a Clustering Algorithm. In *ISCAS*, Scottsdale, Arizona, May 2002.

[78] J. Vieron, T. Turletti, K. Salamatian, and C. Guillemot. Source and channel adaptive rate control for multicast layered video transmission based on a clustering algorithm. Technical Report RR-4580, INRIA, October 2002.

[79] J. Vieron, T. Turletti, K. Salamatian, and C. Guillemot. Source and channel adaptive rate control for multicast layered video transmission based on a clustering algorithm. *EURASIP Journal on Applied Signal Processing (JASP)*, February 2004.

[80] R. Waters, D. Anderson, J. Barrus, D. Brogan, M. Casey, S. McKeown, T. Nitta, I. Sterns, and W. Yerazunis. Diamond park and spline : A social virtual reality system with 3d animation, spoken interaction, and runtime modifiability. Technical Report TR-96-02a, November 1996.

[81] IEEE 802.11 WG. International standard [for] information technology - telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11 :wireless lan medium access control (mac) and physical layer (phy) specifications, 1999. Reference number ISO/IEC 8802-11 :1999(E).

[82] J. Widmer and M. Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of Conference of the Special Interest Group on data COMMunnication, ACM SIGCOMM'01*, San Diego, USA, August 2001.

[83] M. Yajnik, J. Kurose, and D. Towsley. Packet loss correlation in the mbone multicast network. In *Proceedings IEEE Global Internet Conference*, London,UK, November 1996.

[84] W. Yuen, H. Lee, and T. Andersen. A simple and effective cross layer networking system for mobile ad hoc networks. In *PIMRC*, 2002.

# ANNEXE

# A. ARTICLE SCORE

Cette annexe contient un article qui a été publié dans la revue *IEEE/ACM Transactions on Networking, Vol. 12, No 12, pp. 247-260* en Avril 2004. Il décrit le protocole SCORE qui fait l'objet du chapitre 2.

in IEEE/ACM Transactions on Networking Journal, Vol. 12 , No. 2, pp. 247-260, April 2004

# SCORE : a Scalable Communication Protocol for Large-Scale Virtual Environments

Emmanuel Léty[1], Thierry Turletti[1], François Baccelli[2]
Emmanuel.Lety@UDcast.com, {Thierry.Turletti,Francois.Baccelli}@inria.fr
[1]INRIA, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis, FRANCE.
[2]INRIA-ENS, 45 rue d'Ulm, 75005 Paris, FRANCE.

***Abstract*— This paper describes and analyses SCORE, a scalable multicast-based communication protocol for Large-Scale Virtual Environments (LSVE) on the Internet. Today, many of these applications have to handle an increasing number of participants and deal with the difficult problem of scalability. We propose an approach at the transport-layer, using multiple multicast groups and multiple agents. This approach involves the dynamic partitioning of the virtual environment into spatial areas and the association of these areas with multicast groups. It uses a method based on the theory of planar point processes to determine an appropriate cell-size, so that the incoming traffic at the receiver side remains with a given probability below a sufficiently low threshold. We evaluate the performance of our scheme and show that it allows to significantly improve the participants' satisfaction while adding very low overhead.**

***Index Terms*— Area Of Interest Manager (AOIM), Cell-based grouping, communication protocol, Large-Scale Virtual Environments (LSVE), multiple multicast groups, scalability.**

## I. INTRODUCTION

This paper describes and analyses SCORE, a scalable multicast-based communication protocol for Large-Scale Virtual Environments (LSVE) on the Internet. Such Virtual Environments (VE) include massively multi-player games, Distributed Interactive Simulations (DIS) [1], and shared virtual worlds. Today, many of these applications have to handle an increasing number of participants and deal with the difficult problem of scalability. Moreover, the real-time requirements of these applications make the scalability problem more difficult to solve. In this paper, we consider only many-to-many applications, where each participant is both source and receiver. We also make the assumption that a single data flow is generated per participant. However, we believe that most of the results and mechanisms presented in this paper can be easily adapted to more complex applications that use several media types or layered encodings [2].

The use of IP multicast solves part of the scalability problem by allowing each source to send data only once to all the participants without having to deal with as many sequential or concurrent unicast sessions as the number of participants. However, with a large number of heterogeneous users, transmitting all the data to all the participants dramatically increases the probability of congestion within the network and particularly at the receiver side. Indeed, processing and filtering all the packets received at the application level could overload local resources, especially if the rendering module is already processor intensive [3]. [4] shows that in a group communication setting, the percentage of useless (or *superfluous*) information received by each participant increases with the number of data flows and the number of users. This is not surprising since within a VE, each participant simultaneously interacts with only a limited set of other participants. The superfluous information represents a cost in terms of network bandwidth, routers buffer occupation and end-host resources, and is mainly responsible for the degradation of performance in LSVE.

We argue that the superfluous received traffic has to be filtered out before it reaches the end-host. The main difficulty in this filtering mechanism comes from the heterogeneity and the dynamicity of the receivers, not only in terms of bandwidth and processing power but also in terms of data of interest, virtual and physical locations. In [5] and [6], network-layer approaches are proposed to introduce "filters" in the router forwarding process, customizing the data delivered to multicast receivers. However, these propositions require modifications in the routers and are unfortunately not yet deployed in the Internet.

The aim of this paper is neither to propose a new IP multicast model nor to come up with a network-layer approach, adding new mechanisms in the routers. Instead, we present a transport-layer filtering mechanism with multiple agents, assuming that all the users are capable of receiving multicast transmissions. Our approach involves the dynamic partitioning of the VE into spatial areas called *cells* and the association of these cells with multicast groups. We describe a method, based on the theory of planar point processes, to determine an appropriate *cell-size* so that the incoming traffic at the receiver side remains with a given probability below a sufficiently low threshold. We then propose mechanisms to dynamically partition the VE into cells of different sizes, depending on the density of participants per cell, the number of available multicast groups, and the link bandwidth and processing resources available per participant.

The rest of the paper is organized as follows. Section II reviews the limitations of the current IP multicast model, presents the cell-based grouping strategy, and examines the tradeoff in selecting the cell-size parameter. Section III

describes a model allowing one to evaluate various mean values of interest. The section then analyses the impact of the cell-size on the traffic received at the receivers and several quantities such as the participant's mean residence time within a multicast group. Section IV describes SCORE, a scalable communication protocol that implements a dynamic cell-based grouping strategy using a limited number of multicast groups. Section V evaluates the performance and the overhead of SCORE using a set of intensive experimentations. Finally, Section VI discusses related works, and Section VII concludes the paper and presents directions for future work.

## II. MOTIVATION

In this section, we examine the different limitations in using multiple multicast groups and the issues involved in selecting the best size of cell.

### A. Multiple multicast groups limitations

There are several limitations on the use of multiple multicast groups. First, we have to consider that today, multicast groups are not inexhaustible resources: the number of available multicast groups in IPv4 is limited to 268 million Class D addresses[1]. Moreover, there is an increasing number of applications that require several multicast addresses, such as layered coding based video-conferencing, or DIS applications. Therefore, the widespread use of multicast increases the probability of address collisions. A few solutions have already been proposed in the literature to solve the multicast address allocation problem. For example, a scalable multicast address assignment based on DNS has been proposed in [7]. Another alternative could be the use of the Multicast Address Set Claim (MASC) protocol which describes a scheme for the hierarchical allocation of Internet Class D addresses [8]. Some alternatives to the current IP multicast model have also been proposed: [9] describes a multicast address space partitioning scheme based on the port number and the unicast host address. In *Simple Multicast*, a multicast group is identified by the pair (address of the group, address of the core of the multicast tree), which gives to each core the full set of Class D addresses space [10]. In *EXPRESS*, a multicast *channel* is identified by both the sender's source address and the multicast group [11]. Finally, with IPv6, the multicast address space will be as large as the unicast address space, so this will solve the multicast address assignment problem. However, all these proposals are still active research areas and are not currently available on the Internet.

Secondly, multicast addresses are expensive resources. The routing and forwarding tables within the network are limited resources with limited size. For each multicast group, all the routers of the associated multicast tree have to keep information on which ports are in the group. Hosts and routers also need to report periodically their IP multicast group memberships to their neighboring multicast routers using IGMP[12]. Moreover, some routing protocols such as DVMRP[13] rely on the periodic flooding of messages throughout the network. All this traffic has a cost, not only in terms of bandwidth but also in terms of join and leave latency, which should be taken into consideration for interactive applications [14]. Indeed, when a participant sends a join request, it can take several hundreds of milliseconds before the first multicast packet arrives. Such costs should be obviously considered in Large-Scale Multicast Applications (LSMA) and argue in favor of a bigger cell-size, and therefore, of a limited number of multicast groups.

[1]IPv4 Class D addresses use 28-bits address space.

### B. The cell-size tradeoff

In this paper, we focus on the *cell-based grouping strategy* which basically consists in partitioning the VE into cells and assigning to each cell a multicast group. During the session, each participant identifies the cell he is currently "virtually" located in, and sends his data to the associated multicast group. To receive the data from the other participants included in the area in which he is interested in (i.e., his *area of interest*), each participant has to join the multicast groups associated with the cells that intersect his area of interest. Similarly, when a participant moves, he needs to leave the multicast groups associated with the cells which do not intersect his area of interest anymore.

The cell-based grouping strategy is particularly suitable on VEs that can easily be partitioned into virtual areas (e.g., virtual Euclidean spaces). However, the main difficulty in this partitioning is to find the appropriated cell-size. Indeed, decreasing the cell-size increases the overhead associated with dynamic group membership whereas increasing the cell-size increases the unwanted information received per participant [15].

Two approaches are possible to estimate the best cell-size in a LSVE: the first approach requires the pre-calculation of a static cell-size parameter, which remains the same during the whole session. The second approach consists in dynamically re-estimating the cell-size during the session, taking into account various parameters. To motivate the choice of one of these two approaches, let us first identify the parameters involved in the cell-size calculation and then, examine the impact of the variability of these parameters on the appropriate cell-size.

- **The number of available multicast groups** is an important parameter to take into account for the cell-size calculation because it gives a lower bound on the cell-size. As the number of multicast groups used is inversely proportional to the size of the cell, a small set of available multicast groups will lead to a bigger cell-size.
- **The receivers capacities** are determined by the link capacities and the processing power available per receiver. Typically, this parameter limits the amount of traffic that the receivers can handle. Assuming each user roughly generates the same amount of traffic, the incoming traffic per receiver grows linearly with the total number of sources contained in the multicast groups to which he has subscribed. In other words, the incoming traffic per receiver is a function of the number of entities located in the cells included or intersected by his area of interest. Nevertheless, some of these participants may be located outside the area of interest but inside a cell that includes

this area of interest. The ratio between the corresponding number of unwanted participants and the total number of sources received represents the percentage of *superfluous* traffic received. So, the cell-size and more particularly the ratio between the cell-size and the size of the area of interest, have a direct impact on the amount of unwanted traffic.

- **The density of participants** represents the ratio between the number of participants and the size of the VE. In the cell-based grouping strategy, the area of interest is approximated by the smallest set of cells covering the area of interest. In the rest of the paper, we refer to the difference between these two areas as the *superfluous area*, see Figure6. So, the density of participants in a VE not only has an impact on the average number of participants located in the area of interest, but also on the superfluous area. Consequently, the participant density has an impact on the average superfluous traffic. A smaller cell-size could allow a better approximation of the area of interest and a significant reduction of superfluous area and its corresponding traffic. Thus, depending on the participant density, the superfluous traffic and its negative impact on the application performance could also be significantly reduced.
- **The participant velocity** can be used in a cell-based grouping VE to estimate the bandwidth overhead generated when participants cross cells, and the ratio between the join and leave latency and the mean time that the participant stays in each cell. In cell-based grouping, each cell is assigned to a multicast group. Therefore, joining and leaving a cell in a VE corresponds to joining or leaving an IP multicast group in reality. Even though there are enough multicast addresses available to assign each cell, there are several concerns while using multiple multicast groups. First, join and leave control messages use some additional bandwidth between the end-users and their nearest multicast routers. Second, when the participants join or leave multicast groups, they create a significant processing overhead among the routers of the associated multicast trees. Finally, there is a huge concern with the join and leave latency, especially for interactive VE in which the real-time requirement of the application is essential to preserve.

## III. Models and simulations

This section introduces models of area of interest and of participants based on random point processes which are inspired by the stochastic geometry approach proposed in [16]. This model allows us to evaluate various mean values of interest and later on to address issues pertaining to mobility.

### A. Static participants

First, we restrict the problem to static participants using the following assumptions :

- The participants are static and located on the plane according to a random homogeneous Poisson point process of intensity $\lambda$ [17];
- The cells form an infinite regular square grid on the plane;
- The area of interest $IArea$ is a square of area $r^2$ centered on a typical participant (referred to as the observer in what follows).

We denote by $s$ the cell-size (i.e., the distance between two adjacent horizontal or vertical cell boundaries), and $CellArea$ the cell area $s^2$. We focus here on the distribution of the number $M$ of cells intersecting the area of interest and on that of $N$, the number of participants located in these cells (excluding the observer).

Let $\lfloor x \rfloor$ denote the integer part of the real number $x$, namely the largest integer smaller than or equal to $x$. Let

$$k = k(r,s) = \lfloor \frac{r}{s} \rfloor \quad (1)$$

$$p = p(r,s) = \frac{r}{s} - \lfloor \frac{r}{s} \rfloor. \quad (2)$$

Note that $0 \leq p < 1$. We prove below that:

1) The law of $M$ is a point mass distribution on the three integers $(k+1)^2$, $(k+1)(k+2)$ and $(k+2)^2$, with parameters

$$P[M = (k+2)^2] = p^2, \quad (3)$$

$$P[M = (k+1)(k+2)] = 2p(1-p), \quad (4)$$

$$P[M = (k+1)^2] = (1-p)^2. \quad (5)$$

2) The generating function of $N$, is given by the following formula:

$$E[z^N] = p^2 e^{-\lambda s^2 (k+2)^2 (1-z)} + 2p(1-p) e^{-\lambda s^2 (k+1)(k+2)(1-z)} + (1-p)^2 e^{-\lambda s^2 (k+1)^2 (1-z)}. \quad (6)$$

Consider the configuration *seen* by the observer. Assume the observer to be located at point $(r/2, r/2)$, so that the area of interest $I$ is the square $[0,r] \times [0,r]$. Seen from this participant, the grid is as randomly shifted. More precisely, from well known properties of renewal processes [18], this typical configuration is that where the grid has one of its intersection points at $(X,Y)$, where $X$ and $Y$ are independent random variables, each with a uniform distribution on the interval $[0,s]$. Under such a configuration, if $X \leq x_0$, where $x_0$ is defined by the relation

$$x_0 = r - s\lfloor \frac{r}{s} \rfloor,$$

then the number of cells which intersect the horizontal sides of $I$ is exactly $k+2$, with $k$ defined as above. If $X > x_0$, this number is $k+1$. The same argument gives the number of cells intersecting the vertical sides of $I$. Using the independence and the uniformity, we obtain that with probability $\left(\frac{x_0}{s}\right)^2 = p^2$, the number of cells intersecting $I$ is $(k+2)^2$. We obtain the other point masses of the law of $M$ via similar arguments.

We now give the proof of the second formula. We have

$$N = \sum_{i=1}^{M} N_i,$$

where the random variables $N_i$ give the numbers of participants in the cells which intersect $I$. Each of these variables is Poisson

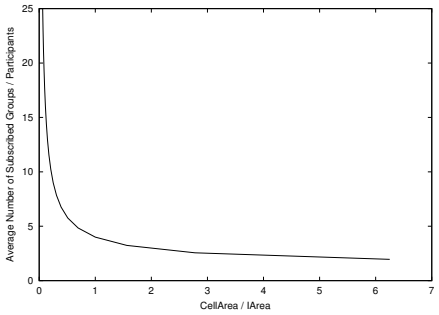



Fig. 1. Average number of subscribed groups / participant

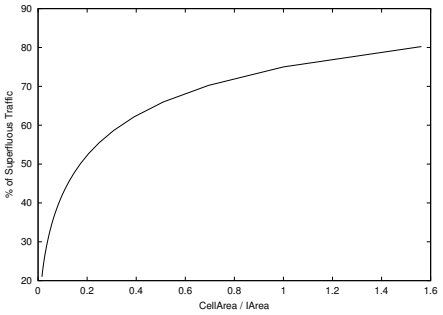



Fig. 2. Percentage of superfluous traffic

with parameter $\lambda s^2$. In addition, the random variables $N_i$ and $M$ are independent. Therefore, we can apply the rule giving the generating function of a random sum of random variables, which states that the generating function of $N$ is $\psi(\phi(z))$ where $\phi$ is the generating function of $N_1$ and $\psi$ that of $M$ [19]. Here, we have

$$\phi(z) = e^{-\lambda s^2(1-z)}$$

and

$$\psi(z) = p^2 z^{(k+2)^2} \; + \; 2p(1-p)z^{(k+1)(k+2)} + \; (1-p)^2 z^{(k+1)^2},$$

and the second formula follows immediately from this. As direct consequences of these formulas, we obtain the following expressions for:

1) The mean value of $M$ and its variance:

$$\begin{aligned} E[M] &= p^2(k+2)^2 + 2p(1-p)(k+1)(k+2) \\ &\quad +(1-p)^2(k+1)^2 \\ \text{var } (M) &= p^2(k+2)^4 + 2p(1-p)(k+1)^2(k+2)^2 \\ &\quad +(1-p)^2(k+1)^4 - (E[M])^2. \end{aligned}$$

2) The mean value of $N$ : $E[N] = \lambda s^2 E[M]$.
3) The variance of $N$ (with the above notation) :

$$\begin{aligned} \text{var } (N) &= E[M]\text{var } N_1 + E[N_1]^2\text{var } M \\ &= E[M]\lambda s^2 + \text{var } M\lambda^2 s^4. \end{aligned} \quad (7)$$

4) The probability that $N$ is less than a threshold $n$, where $n$ is a non–negative integer :

$$\begin{aligned} P[N \leq n] &= p^2 g_n((k+2)^2) \\ &\quad +2p(1-p)g_n((k+1)(k+2)) \\ &\quad +(1-p)^2 g_n((k+2)^2), \end{aligned} \quad (8)$$

where

$$g_n(m) = e^{-\lambda s^2 m} \sum_{i=0}^{n} \frac{(\lambda s^2 m)^i}{i!}. \quad (9)$$

Now, we analyze the impact of $CellArea$ and the participant intensity on the traffic $N$ received by participant, the average number of subscribed multicast groups per participant, and the percentage of superfluous traffic received. In order to be as generic as possible, we focus more particularly on the impact of the ratio between $CellArea$ and $IArea$ (i.e., $\frac{s^2}{r^2}$). Note that we assume here that all the participants generate the same amount of traffic.

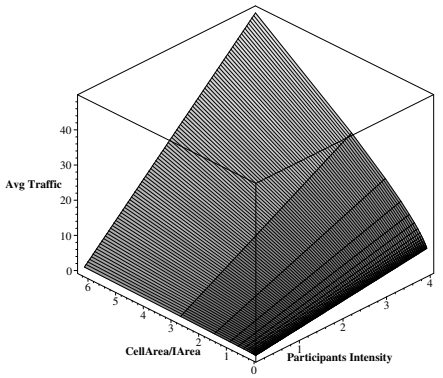



Fig. 3. Average traffic (number of sources) / participant

Figure 1 shows that the average number of subscribed multicast groups per participants $E[M]$ decreases sharply as $CellArea$ approaches $IArea$. However, as $CellArea$ increases further, the average number of subscribed groups decreases slowly to 1.

Figure 2 plots the average percentage of superfluous traffic out of the total received traffic by a participant. Since the participants are located on the plane according to a random homogeneous Poisson point process, this percentage is equal to: $1 - \frac{r^2}{E[M]s^2}$. We observe that when $CellArea$ is larger than $IArea$, more than 70% of the traffic is superfluous. This figure also suggests that when $CellArea$ is smaller than $IArea$, a slight diminution of $CellArea$ decreases significantly the superfluous traffic received. However, it is important to note that 70% of superfluous traffic is acceptable compared to the situation where all the users communicate on a single

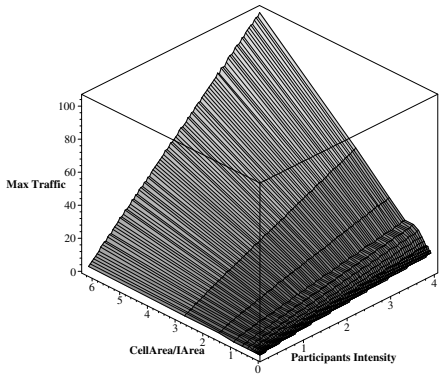



Fig. 4. Max traffic threshold (number of sources) / participant with p = 0.95

multicast group [3]. Indeed, with a single multicast group and a large number of participants, almost all the traffic received would be superfluous.

Figure 3 shows the average traffic received by a participant, depending on the intensity of participants in the VE, and the ratio between $CellArea$ and $IArea$. The participant intensity represents here the average number of participants per $IArea$: $\lambda r^2$. Such a way to express the density of participant in a VE is very useful, as it allows us to modify $CellArea$ without having an impact on the value of the density. The results show that for a given value of participant intensity, it is possible to find the largest ratio between $CellArea$ and $IArea$, so that the average traffic remains under a sufficiently low threshold. The average traffic is given by : $\lambda s^2 E[M]$.
Finally, Figure 4 probably shows the most interesting results. In order to satisfy participants in a VE, it is better to determine an appropriate $CellArea$ so that the incoming traffic remains with a high probability below the maximum traffic that they can handle. This probability reflects the tradeoff between the satisfaction of the users and the required number of multicast groups. Figure 4 shows that for a given intensity of participants, it is possible to find the largest $CellArea$ (i.e., the smallest number of multicast groups), so that the incoming traffic remains below a sufficiently low threshold with a probability of 0.95. Moreover, for a given $CellArea$, we observe that this traffic increases linearly with the intensity of participants.

### *B. Dynamic participants*

This section introduces a model of mobility which is compatible with the assumption that the point process of participants is Poisson at any time, and which allows us to derive various mean values of interest in relation with mobility. This includes quantities such as:

- The handover in and out of a multicast group, defined as the time point process intensity of the boundary crossings of the corresponding cell by moving participants;
- The mean residence time of a typical participant within a multicast group, namely within the corresponding cell.

The assumption is still that participants are initially located according to a Poisson point process of intensity $\lambda$. No participant enters or leaves the game. Nevertheless, each participant moves on the plane according to an independent random motion described as follows: a pair of random variables $(V_i, \theta_i)$, is associated with participant $i$, where $V_i \in I\!R_+$ is the random velocity of the participant and $\theta_i \in [0, 2\pi)$ his random direction. It is assumed that all pairs $(V_i, \theta_i)$ are independent and identically distributed and that the random variables $(V_1, \theta_1)$ are independent, with $V_1$ of density $f$ on $I\!R_+$ and with $\theta_i$ uniform on $[0, 2\pi)$. Thanks to the so called displacement theorem (see [17], p. 61), the point process giving the location of all participants is still a Poisson point process of intensity $\lambda$ at any time $t$, so that the results of the previous section are still valid at any such time.

Let $\sigma$ be a fixed segment of length $u$, which we can assume to be located on the horizontal axis without loss of generality. The set of participants with a motion pair equal to $(v, \theta)$ and which cross $\sigma$ between time 0 and $t$ is that initially located in a parallelogram of area $uvt|\sin\theta|$. The set of participants with a motion pair in the set $[v, v + dv] \times [\theta, \theta + d\theta]$ is Poisson with intensity $\lambda f(v) dv \frac{d\theta}{2\pi}$. Therefore, the mean number of participants crossing $\sigma$ between time 0 and $t$ is

$$\int_0^\infty \int_0^{2\pi} uvt|\sin\theta|\lambda f(v) dv \frac{d\theta}{2\pi} = \frac{2\lambda u E[V] t}{\pi}.$$

Consider a typical cell, namely a square with perimeter $4s$. From what precedes, we get the following expression for the mean value of the handover in and out this cell per unit of time:

$$H = \frac{8\lambda s E[V]}{\pi}. \tag{10}$$

Due to the displacement theorem, we can still use the Poisson law for the number of participants in this cell at any time. Its mean value is $\lambda s^2$. Since the intensity of the entrances into the cell is $\frac{H}{2}$, Little's law gives the following expression for the mean residence time of a participant in a typical multicast group:

$$E[W] = \frac{2\lambda s^2}{H} = \frac{\pi s}{4E[V]}. \tag{11}$$

Figure 5 shows the mean residence time per cell $E[W]$ as a function of the participant mean velocity $E[V]$. We express the velocity in cell-size per second. We observe that the mean residence time decreases exponentially as the mean velocity approaches 1 cell-size per second. This result argues in favor of a limited velocity in LSVE, so that the residence time per cell remains higher by orders of magnitude than the join and leave latency. Indeed, a participant needs to anticipate his join request by subscribing to the multicast groups which map the cells where he can go during the time corresponding to a join latency. Hence, his velocity and the cell-size impact on the number of multicast groups he needs to join by anticipation, and therefore on the IGMP traffic generated.

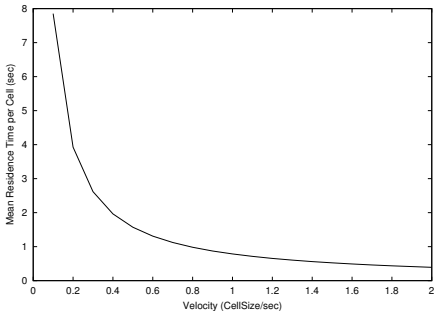



Fig. 5. Mean residence time

### C. Discussion

These models are all based on the assumption that participants are distributed according to planar Poisson point processes. This assumption is primarily made for mathematical convenience. In further studies, models with more clustering such as compound Poisson point processes could also be considered. This was not done here as important properties such as the displacement theorem do not hold for such models.

## IV. Description of SCORE

In Section III, we showed that the cell-size should take different values, depending on the density of participants and on the maximum traffic that participants are able to handle. This result argues in favor of a dynamic partitioning of VEs into cells of different sizes. However, our model can only be applied if we suppose that the distribution of participants in the VE follows a random homogeneous Poisson point process. In a real VE, such a global distribution is not realistic, however if we split the VE into *zones* or parts, we approximate a Poisson distribution of participants inside each zone, with different intensities within each zone. In our scheme, we take into account this model and the corresponding results to split the VE into different zones and to compute an appropriate cell-size in each zone. We implemented this scheme to show its feasibility, then performed several experimentations on our testbed in order to prove its advantages (i.e., the improvements in performance), and to evaluate its cost (i.e., control messages overhead and cost of dynamic join and leave). The goal of this scheme is to make VEs scalable with thousands of heterogeneous users on the Internet. We claim that this solution works with a limited number of available multicast groups. We believe that today, such many-to-many applications, with potentially thousands of users, require minimal management and administration support.

This section is organized as follows. First, we introduce a user satisfaction metric and present the role of the *agents* in our scheme. Then, we describe the information exchange process between participants and agents and finally we present the mapping algorithm and the handover management mechanism.

### A. User satisfaction metric

An ideal situation from the end-user viewpoint can be defined as a situation where the received traffic contains no superfluous data. However, this situation is far from being realistic, considering the cost of multicasting, and therefore, the limitation in the number of available multicast groups (see Section II-A). Moreover, participants have limited network and CPU processing cycles resources. If the participant's area of interest is so large that the traffic he receives cannot be processed in real time, no mechanism could enable him to receive all the data he is interested in. Indeed, in this case, even if the subscribed cells exactly match his area of interest, the received traffic exceeds his capacity. For this purpose, we define the user satisfaction metric $S$ as:

$$S = \frac{U_r}{\min(U_t, C)} \tag{12}$$

where $U_r$ stands for the interesting data rate received and processed; $U_t$ represents the data rate (received or not received), in which the user is interested (in the case of a homogeneous Poisson point process of intensity $\lambda$, this would be proportional to $\lambda r^2$); and $C$ stands for the receiver's capacity, which is the maximum data rate that the receiver can handle (limited by his network connectivity and/or processing power). When a participant receives and processes all the data he is interested in, this satisfaction metric is maximal whatever the superfluous traffic rate. Notice that for a particular user, $S$ is also maximal when $U_r$ is equal to $C$. This is true even though only a part of the data, in which the participant is interested, is received by the application. We justify the choice of this metric by the necessary tradeoff between the superfluous data rate received, the network state, and the overhead associated with dynamic group membership. Note that with this satisfaction metric, the goal of our scheme is not to adapt to the worst receiver in terms of network connectivity and processing power, but to maximize the satisfaction of the receiver with the lowest $S$ value. This approach often referred as *max-min fairness* is described in [20].

### B. Agents responsibility

Let us define *agents* as servers or processes running at different parts of the network (e.g., on a campus LAN, hosted by an ISP or by LSVE developers). Administrators of LSVE are responsible for deploying such agents on the Internet and for positioning them as close as possible to their potential users. Agents are not servers, i.e., they do not aim to process any global state for the VE, so they do not receive data traffic sent between participants. Actually, agents dynamically determine zones with the VE by considering the distribution of participants and they calculate appropriate cell-sizes according to the density of participants in each zone. Agents also have to periodically process the satisfaction of each participant according to his capacity, the size of his area of interest and the density of participants within his current zone. The computation of the participant satisfaction is done in a very simple way, using our Poisson model in the plane within each zone. Once this computation is done, agents can determine new zones (or inversely they can aggregate existing zones), and

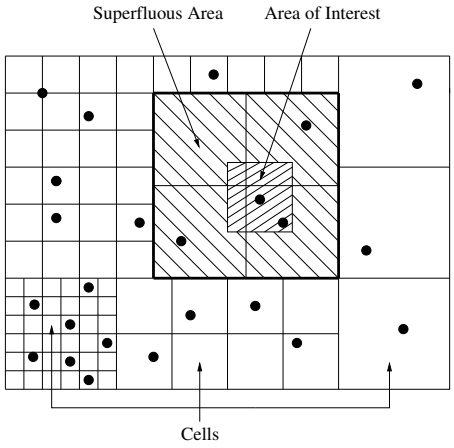



Fig. 6. Partitioning with different cell-sizes

modify the cell-sizes within the zones where the participants with the lowest satisfaction are located. Therefore our approach requires the dynamic partitioning of the VE into cells of different sizes, and the association of these cells with multicast groups. Agents have to dynamically determine appropriate cell-size values in order to maximize users' satisfaction. During the session, four successive operations are required:

- Partition the VE into several *zones*, according to the distribution of users, the users satisfactions, and the VE structure (e.g., rooms, walls, etc.).
- Compute the appropriate cell-size for each zone, according to the parameters listed in Section II-B (see Figure 6).
- Divide each zone into cells, according to his computed cell-size, and assign a multicast group address to each cell of each zone.
- Inform the participants of which multicast groups they need to join in order to interact with participants located around them.

In the rest of the paper, we refer to the first three operations as the *mapping algorithm*. We also designate the results of these operations as the *mapping information*.

### C. Mapping information

In order to communicate mapping information to users, i.e., the association between cells and multicast group addresses, it is necessary to find a way to identify and name these cells within the VE. Moreover, the VE could be a structured environment with walls and rooms of different sizes. Two participants can be very close to each other but as a wall is separating them, there is no possible interaction. This specific information should be taken into account before partitioning a VE into different zones.

First, the VE is statically partitioned into several large parts that we called *start-zones* in the rest of the paper. These start-zones are actually defined according to the intrinsic structure of the VE (e.g., rooms, floor, walls, etc.) and can't ever be combined. Each start-zone is statically partitioned into indivisible *zone-units* which are the smallest unitary zones that compose the start-zone. During the session, start-zones are dynamically divided into zones which all have the same cell size. So, cells are mapping of multicast groups to a number of zone-units. As agents decide to define new zones in order to take into account changes in the distribution of participants, they identify these zones as sets of one or more contiguous zone-units belonging to the same start-zone.
To summarize, a zone is a subset of a start-zone and is composed by $n$ contiguous zone-units $(n \geq 1)$. Within a given zone, all cells have the same size but two distinct zones could have different cell-sizes.

### D. Participants-to-Agent communication

Figure 7 shows the different levels of communication in our scheme :

- Each participant subscribes to one or more multicast groups but sends data packets on a single group.
- Each participant is connected to a single agent, using a UDP unicast connection.
- Agents communicate with each other on a single multicast group: the *Agent Multicast Group* ($AMG$).

A participant has to subscribe to two different kinds of multicast groups:

- *data groups* associated to the cells that intersect his area of interest. Note that a participant only sends data to the multicast group associated to his current cell.
- *control groups* associated to the *start-zones* that intersect his area of interest. For these groups, a participant is only a receiver. Agents use control groups to send mapping information relative to the start zones. These informations are periodically sent for each start-zone (period = $P_{mapping}$), and contain the mapping information for all the zones belonging to the start-zone (i.e., the cell-size for each zone and the associated multicast groups addresses).

For each of these groups, the participant has to make early *join*s taking into account his speed, and the join-latency value[2]. For control groups, the $P_{mapping}$ period is also taken into account in order for the participant to receive the mapping information before his area of interest intersects new cells belonging to new start-zones.

Each participant is connected to his nearest agent using a UDP connection. We do not use a TCP connection for scalability reasons. Each time a participant enters a new zone-unit, he sends a short message to his agent. This message (20 bytes) contains his identity, his position in the zone-unit, his current size of area of interest and his capacity [21]. Therefore each agent is able to track the location of its connected users in the VE. In order to evaluate the density of participants within each zone, agents exchange information on the $AMG$

[2]The join-latency value can be dynamically updated during the session.

8

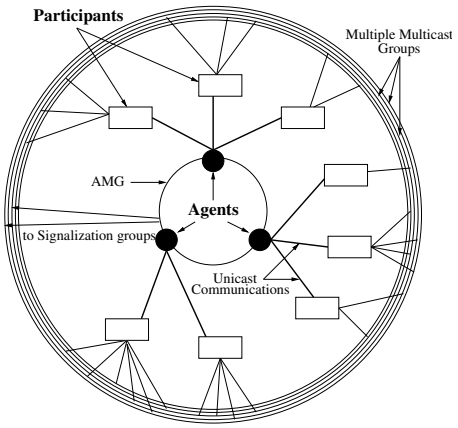



Fig. 7. Different levels of communication

multicast group. However, agents do not need to send the exact virtual position of their associated users. Only the number of users per zone-unit is necessary to allow agents to compute periodically the density of participants per zone-unit.

We have also designed a flow control mechanism between participants and agents along with a dynamic mechanism allowing each agent to know when a participant disconnects from the VE. This creates a soft state in the agent and adds reliability to the UDP transport. Participants have to send low rate *keep-alive* packets so that agents can detect a possible disconnection and have an accurate number of participants in the different zones. According to the number of participants connected, agents compute the minimal sending packet period[3] and send it back to participants. If the participant's timer expires before the participant crosses a new zone-unit, he will send a keep-alive packet including only his identity and his current position.

*Connection to the Virtual Environment*

We assume that before starting a session, participants have already downloaded the VE description and know the agent's multicast group address. When a new participant wants to enter the VE, he first needs to find the "closest" agent before registering and starting a login process. In our scheme, end-users discover agents by sending "Hello" packets on the agent multicast group address (they do not need to request membership to that group). This agent discovery could be done using either an incremental TTL-based mechanism or an RTT-based mechanism, depending on the distance metric we decide to choose. As soon as an agent receives a "Hello" packet from a new participant, it opens a UDP connection with it and starts the login process. Afterwards, an optional authentication process can start.

[3]This information is included within the remapping information sent by agents through the control groups.

*Mapping algorithm*

The same mapping algorithm is used by each agent. Agents use this algorithm to dynamically define zones in the VE, and to dynamically compute an appropriate cell-size within each zone, considering the distribution of participants and their satisfactions. However, the number of cells within a zone is inversely proportional to the cell-size for that zone. Thus, a limited number of multicast groups limits the minimal size of cells (remember that each cell is associated with a unique multicast group). Considering the entire VE, the number of cells remains always the same during the whole session. Nevertheless, the number of multicast groups assigned for each zone dynamically changes according to the evolution of the distribution of participants in the VE.

In order to allow the agents to easily compute the mapping information, we only consider square cells, with an integer number of cells per zone. Throughout the session, agents periodically compute the average density of participants per multicast group, by dividing the number of connected participants with the number of available multicast groups for the application. We refer to this density as the *remapping threshold* of the mapping algorithm. As participants arrive and move in the VE, agents keep track of the density of participants in each zone.

The mapping algorithm consists in three successive operations:

- At first, calculation is done in order to define a cell-size for each zone by only taking into account the distribution of participants in the VE. To perform this calculation, the density of participants per cell in each zone is compared to the remapping threshold.
- Then, the participants with the lowest satisfaction are identified as well as their distribution in the VE. If agents detect a concentration of unsatisfied participants within a part of a zone, this zone is divided into two new zones in order to isolate these participants. If not, the zone remains unchanged. A smallest cell-size is computed in the zones which contain the participants with the lowest satisfactions, so that they can better approximate their area of interest and therefore improve their goodput.
- The final operation is less frequently performed. During this operation, agents can decide to aggregate contiguous zones, if the cell-sizes are the same for these zones and if they belong to the same start-zone. Note that two start-zones can never be merged.

Two possible reasons can lead to the division of a zone into smaller cells:

- It is possible to find a smaller cell-size where the average density of participants per cell still exceeds the remapping threshold.
- The participants with the lowest satisfactions are located in this zone.

In the first case, agents can use the density of participants in the zone to compute a more appropriate cell-size. In the second case, agents first determine the distribution of unsatisfied participants within the zone. In order to detect a concentration

of unsatisfied users in only a part of the zone, agents first compute the minimal satisfactions of participants for each zone-unit of that zone. Then they compare these satisfactions with the average satisfaction of all the zone-units of that zone. Afterwards, they can decide to split or not the zone into two new ones. Conversely, agents can decide to remap a zone using bigger cells. This remapping occurs when the density of participants per cell is smaller than the remapping threshold.

*Handover management*

To inform the participants on which multicast groups they need to join in order to interact with participants located around them, agents have to deal with two different situations. The first situation occurs when a participant is moving in the VE and is about to enter in an area where he does not know the mapping information. The second situation occurs when agents decide that the cell-size of a part of the VE is no longer appropriate; for example if the density of participants in this area suddenly increases. In this case, a new cell-size needs to be computed and the participants who are currently located in this area need to update their group memberships. Moreover, participants need to keep interacting between each other without suffering from this *remapping*. We call this critical operation the *handover management* [22], [23].

Here are the successive operations required to perform a handover:

- When a participant receives the new mapping information, he joins the new groups which map his area of interest. However, the participant keeps sending in the old multicast groups.
- As explained in Section IV-D, agents periodically send the mapping information in each control group. However, when agents decide to change the mapping information for a zone, they temporarily increase their sending rate for the corresponding control group.
- The participant waits for the reception of $n$ mapping information packets before sending to the new multicast groups instead of the old ones. However, if he starts receiving data from the new groups, he immediately switches to the new groups.
- When a participant did not receive any data from the old multicast groups for a given period of time, he leaves these groups.

## V. Evaluation of SCORE

In order to evaluate the performance and the overhead of SCORE, we have implemented the algorithms described in section IV and run a set of intensive experimentations [21] on PC stations: 7 PCs under Linux 2.2 connected on a 10Mb/s Ethernet. To allow experimentations with a very large number of participants, we added an option for the participants to disable the reception of data packets. When this option is used, the participant sends normally his data traffic to the multicast group associated to his current cell but only subscribes to the control groups (not to the data groups). This considerably reduces the CPU load used for the participant and enables us to run a large number of participants on a same machine.

In the following experimentations, we use 1000 participants: 996 of them with the data reception disabled are run on 3 PCs (332 participants per PC) and the 4 remaining participants that receive data traffic are run on 4 others PCs. One agent is enough to handle a set of 1000 participants: it takes only 10% of CPU of a PentiumPro 200MHz machine. To simplify participants' movements, we use a square VE without walls. The ($1200 \times 1200$ $units^2$) VE is partitioned into $3 \times 3$ square start-zones with 144 available multicast groups. Each start-zone includes $2 \times 2$ square zone-units, so the size of a zone can be 1,2,3 or 4 times the size of a zone-unit. For example, in case zones are composed of 1 or 4 zone-units, the number of cells per zone takes its value in $\{1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121\}$. However, the total number of cells in the VE remains always less than or equal to the number of available multicast groups. In order to evaluate the performance of the mapping algorithm, we compare it with a static partitioning strategy dividing the VE into $12 \times 12$ squares cells of the same size. To simulate heterogeneous participants, each participant has a capacity $C$ that is randomly selected at the beginning of the experimentation. For example, if a participant was able to handle a maximum of 20 sources, but 40 participants were located in the cells intersecting its area of interest, then only half of its incoming traffic was received and processed. The presence of variability is introduced in the VE using both the participants velocities and the notion of "hot" and "cold" start-zones: i.e., zones in which the probabilities to contain participants are respectively higher and lower than the average. At the beginning, participants are first randomly placed in the VE with a uniform distribution along x-axis and y-axis. Then the destination start-zone is randomly selected taking into account probabilities to contain participants of each start-zone [21].

Furthermore, to analyze the different experimentations, the following parameters are used:

- *Area of interest (IArea)* expressed according to the cell area in the static case (which is equal to the ratio between the VE area ($1200 \times 1200$ $units^2$) and the number of available multicast groups (i.e., 144)),
- *Remapping period (RP)* standing for the period in seconds between two different remapping decided by agents,
- *Participants velocity (V)* in the VE in units per second (we have compared two cases: $V = 10$ units/s and $V = 100$ units/s given that with a static partitioning the cell area is equal to $100 \times 100$ $units^2$),
- *Distribution of participants' capacity (C)*: capacities are randomly selected with a uniform distribution on either the interval $[20, 40]$ or the interval $[10, 50]$ sources/sec.

In all the experimentations, we use 1000 participants and a set of 144 available multicast groups, so the remapping threshold is equal to $\frac{1000}{144} = 6.94$.

### A. Performance evaluation using the satisfaction metric

In the following set of experiments, we analyze the cumulative distribution of participants' satisfactions based on the model described in Section III. Then, we compare data traffic received per participant with and without SCORE.

10

*1) Comparison of satisfactions static/dynamic:* Figure 8 compares satisfactions obtained with a static partitioning and a dynamic partitioning. Two levels of heterogeneity are shown, with capacities uniformly distributed between either $[20, 40]$ sources/sec or $[10, 50]$ sources/sec. We have done experimentations [21] with ten different values of $IArea$ (between 1 $CellArea$ and 0.01 $CellArea$), but we only present in the paper two of them: $IArea = 0.25CellArea$ for the left figure and $0.04CellArea$ for the right figure. Whatever the level of heterogeneity between participants, the dynamic partitioning curve remains always below the static partitioning curve. For example, in the left figure, we observe that for the dynamic partitioning case, less than 5% of participants for $C \in [20, 40]$, (and less than 20% of participants for $C \in [10, 50]$) have a satisfaction value less than $0.8$; whereas for the static case, between 40% and 50% of participants have a satisfaction value less than $0.8$. In the right figure, minimal satisfactions are respectively $0.9$ ($C \in [20, 40]$) and $0.6$ ($C \in [10, 50]$) for the dynamic case and $0.55$ and $0.3$ for the static case. These results clearly demonstrate the scalability improvements of SCORE with respect to a static partitioning approach. However, when the area of interest is very large, performance decreases whatever the partitioning mode. On the opposite, when the area of interest is very small, participants' satisfactions tend towards 1 whatever the partitioning mode, and the SCORE mechanism becomes useless in this case.

*2) Comparison of satisfactions for different distributions of capacities:* Figure 9 compares mean satisfactions of 10 participants (i.e., 1% of the overall LSVE population), for two different distributions of receiver capacities. In the first distribution (called *non-uniform distribution*), their capacities are uniformly distributed in $[10, 20]$, whereas the 990 remaining participants have higher capacities uniformly distributed in $[30, 50]$. In the second distribution called *uniform distribution*, capacities of the 1000 participants are uniformly distributed in $[10, 20]$. The left figure shows the case where the area of interest is large (equal to $CellArea$). We observe that the two curves are similar and that very few participants obtain a maximal satisfaction. Indeed, when the area of interest is large, the superfluous incoming traffic could become very important. So, whatever their capacities, the participants with the lowest satisfactions are almost all located within the "hot" start-zones. Thus, for both distributions of capacities, the mapping algorithm only allocates more multicast groups within those start-zones.

When the area of interest decreases, (e.g., in the left figure with $IArea = 0.49CellArea$), more and more participants with capacities uniformly distributed in $[30, 50]$ obtain a maximal satisfaction. As soon as the cell sizes have been computed on the different zones according to the density, these participants obtain a maximal satisfaction. So, all the remaining multicast groups can be allocated to zones in which the 10 low-capacity participants are located. Note that less than 40% of satisfactions are less than 0.5 for the non-uniform case, whereas this percentage reaches 80% for a uniform distribution of receivers' capacities. This result shows the aptitude of SCORE to handle heterogeneous participants.

*3) Received data traffic per participant:* Figure 10 compares the mean participant's incoming data rate (in sources/sec) for a static partitioning scheme and a dynamic partitioning scheme using $RP = 1s$. Remember that this traffic is used to compute satisfactions of participants. We can observe that the gap between the 2 curves is almost constant independent of the size of the area of interest. However, relative gaps between curves differs: for $IArea = 0.16CellArea$, the incoming data traffic is $50\%$ less in the dynamic case (20 sources/sec) than in the static case (30 sources/sec); whereas for $IArea = CellArea$, it is only $30\%$ less (50 sources/sec vs. 65 sources/sec). This result shows that mechanisms implemented in SCORE enable participants to better approximate their areas of interest using smaller cell sizes, especially in places where the density of participants is important. Indeed, in such high density places, a small reduction of the superfluous area strongly decreases the superfluous incoming data traffic.

### B. Overhead of SCORE

*1) Impact of SCORE on Multicast Routing Protocols:* It is realistic to assume that multicast-enabled routers can support the needs of multiple multicast groups as required by SCORE. Firstly, even if each participant subscribes to multiple multicast groups, each participant will only send data traffic to a single multicast group. Therefore, with respect to a given participant, a single $(S, G)$ entry will be active in each multicast router. Regarding the other multicast groups where the participant behaves as a passive receiver only, the only impact might be the addition of an outgoing interface in pre-existing entries of each multicast router present in the corresponding multicast tree. Secondly, it is important to realize that, in SCORE, the fact that each participant could be a member of several multicast groups, is limited by the assumption that SCORE deals with a limited number of multicast groups. This implies that routing/forwarding tables could contain several $(S_i, G)$ for a given group G. So depending on the underlying multicast routing protocol, these entries could also be aggregated into a single $(*, G)$ entry [24].

In the following experimentations, we evaluate the overhead of SCORE focusing on the signaling and control traffic. Then

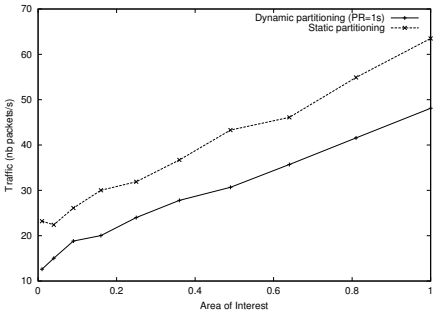



Fig. 10. Received data traffic per participant ($V = 10$ units/s)

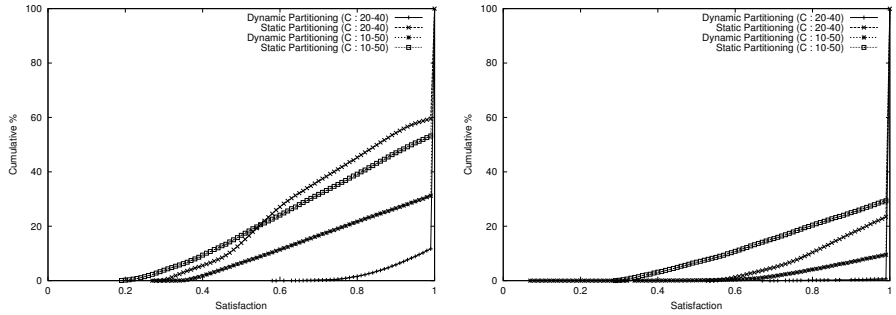



Fig. 8. Comparison of satisfactions with $V = 100$ units/s, $RP = 1s$, $IArea = 0.25$ (left) and $IArea = 0.04$ (right)

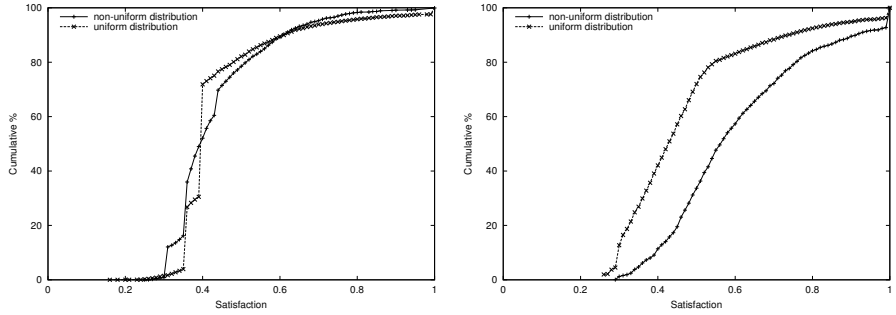



Fig. 9. Comparison of satisfactions with 2 capacity distributions with $IArea = 1$ (left) and $IArea = 0.49$ (right)

we compare the IGMP traffic generated by participant with and without SCORE.

*2) Received signaling traffic per participant:* Figure 11 shows the signaling traffic multicast by agents to participants and the control traffic sent by each participant to his agent according to the size of the area of interest. Note that sizes of signaling and control packets are respectively 8 and 16 bytes (plus 24 bytes of UDP/IP headers). The maximal signaling traffic is obtained for $RP = 1s$ and $IArea = CellArea$ and remains less than 1.5 packet/s. In this worst case, the right figure means that a participant subscribes in average to 1.5 multicast groups and receives a mean traffic rate of 48 bytes/s (i.e., 0.38kb/s). The left figure shows the control traffic and the "keep-alive" traffic sent by two participants to their agents with $V = 10$ units/s. The overhead is very low, less than 0.1 packet/s for the "keep-alive" traffic and about 0.05 packet/s for the control traffic. We used two different participants in order to show that the "keep-alive" traffic decreases when the control traffic increases, and conversely.

*3) Network load caused by the participants:* In Figure 12, we plot the number of subscriptions per second, depending on the area of interest (relative to an average size of cell), the remapping period and the velocity. To obtain the number of IGMPv2 Reports and IGMPv2 Leaves, this number should be multiplied by a factor of two. However, if several participants are located on the same LAN, the number of IGMPv2 packets sent might be reduced as a result of the IGMP-v2 Max Response Time field present in each IGMP-v2 Query packet combined with the duplicate Report suppression mechanism of IGMP-v2.

In Figure 12 left, we observe that in the case of dynamic partitioning the number of reports doubles when the area of interest increases from $0.01 * CellArea$ to $0.49 * CellArea$, and seems to stabilize for larger area. This can be explained by the fact that the area of interest is multiplied by 50 in the left part of Figure 12 left and by 2 in the right part. Thus, the number of cells intersected by the Area of interest grows also much faster in the first part. As each cell is associated with a multicast group, the evolution of the number of IGMP Reports is directly correlated with this behaviour. In this Figure, we also notice that the subscription frequency is 2 times larger where $RP = 6s$ and 4 times larger where $RP = 1s$, compared with a static partitioning strategy.

In Figure 12 right, if we compare the dynamic partitioning strategy with the static strategy where $V = 100$ units/s, we observe that the frequency of IGMP reports in the former case is twice larger than in the latter. However, even if the velocity clearly has a direct impact on the subscription frequency,

12

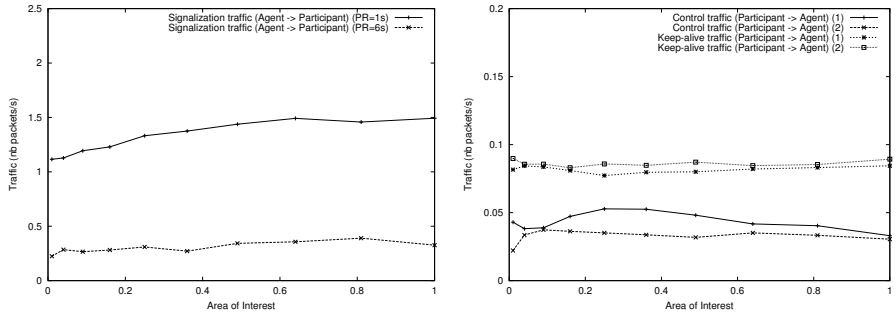



Fig. 11. Signaling traffic received and Control traffic sent (packets/s) with $V = 10$ units/s

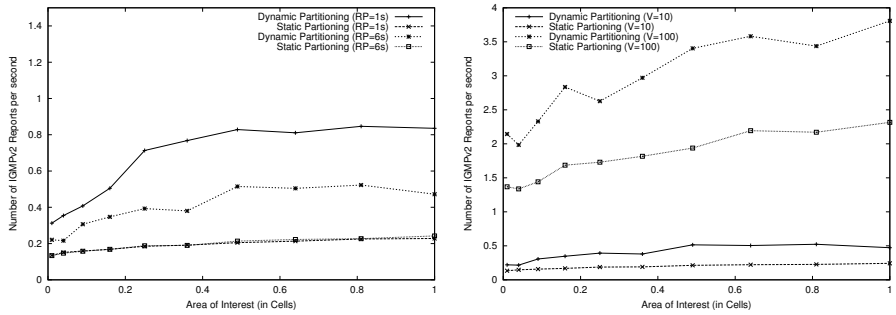



Fig. 12. Comparison of subscription frequency with $V = 10$ units/s (left) and $PR = 6s$ (right)

the same comparison for $V = 10$ units/s shows that the relative difference between the number of IGMP reports for the two partitioning strategies decreases. Indeed, SCORE allows the reduction of cell-size in the areas where the majority of participants are located (as the dynamic partitioning strategy takes into account the density of participants). With $V = 100$ units/s and $RP = 6s$, this statement is mainly true when a remapping of the virtual environment happens. Between two remappings, the distribution of participants changes more drastically compared with the case where $V = 10$ units/s. We have already seen in Figure 8, that this also has an impact on the participant satisfaction.

### C. Multicast groups analysis

In the following experimentations, we analyze the use of multicast groups within the SCORE scheme.

*1) Number of multicast groups subscribed per participant:* Figure 13 shows the evolution of the number of multicast groups subscribed per participant during a session with $RP = 6s$ and $V = 10$ units/s. We have used 10 different values of area of interest in our experimentations [21] but have only plotted in the paper the curves corresponding to $IArea = 0.81CellArea$ and $IArea = 0.36CellArea$. We observe that when $IArea \leq 0.49CellArea$, the number of subscribed multicast groups evolves in the same interval $[1, 4]$ both for the static and dynamic cases (see the right figure). There are several reasons. First, when the area of interest is small, the number of intersected cells is small. Second, since the area of interest is small, agents do not need to compute smaller cell sizes because participants' satisfactions are already maximal. In this case, the number of groups per zone is not increased by a remapping phase and the number of subscribed groups per participant remains low. However, when the area of interest is larger (e.g., $IArea = 0.81CellArea$ in the left figure), more and more cells are intersected by the area of interest. So, the incoming data traffic increases and agents have to remap the "hottest" zones that include unsatisfied participants. This explains the higher number of subscribed groups per participant in the dynamic case.

*2) Distribution of participants within multicast groups:* Figure 14 shows the distribution of participants within multicast groups when $IArea = 0.04$. First, we can observe a peak around $N = 7$ participants. This peak corresponds to the remapping threshold value (i.e., $6.64$, see Section V). This clearly demonstrates that SCORE can adapt to non-uniform and dynamic distributions of participants. On the contrary, the static case leads to a waste of filtering resources: 30% of multicast groups do not contain any participants, and almost

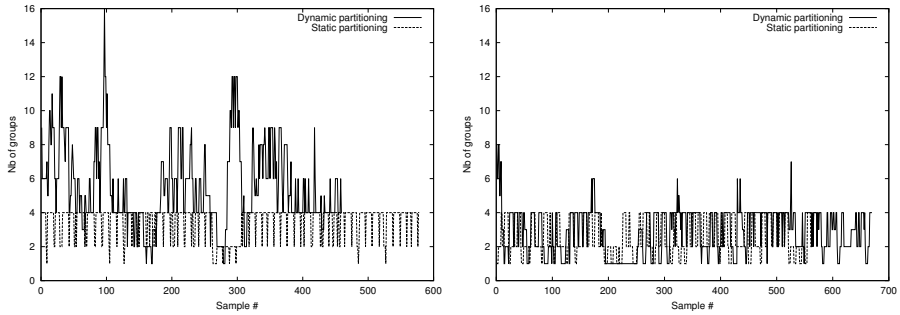



Fig. 13. Subscriptions per participant with $V = 10$ units/s, $RP = 6s$, $IArea = 0.81$ (left) and $IArea = 0.36$ (right)

half of multicast groups contain less than 3 participants. It is interesting to note that the percentage of multicast groups that contain a large number of participants is higher in the static case than in the dynamic case.

## VI. Related Work

There has been a lot of published work on the issue of evaluating grouping strategies for LSVE, but few of them consider network aspects. [25] analyses the performance of a grid-based relevance filtering algorithm that estimates the cell-size value which minimises both the network traffic and the use of scarce multicast resources. However, the paper shows specific simulations done using different granularity of grids for several types of DIS entities, but the generic case is not studied. [26] compares the cost of cell-based and entity-based grouping strategies using both static and dynamic models but the paper does not propose any solution to calculate the cell-size value.

Several architectures such as NPSNET [27], DIVE [28], MASSIVE-2 [29] and SPLINE [30] have already been designed using multiple multicast groups. In NPSNET, the world is partitioned into hexagonal cells which are associated with multicast groups. In the DIVE architecture, the objects in the virtual world are hierarchically composed and associated with a set of hierarchical multicast groups. MASSIVE-2 is a collaborative virtual environment in which the spatial structure is mapped onto a hierarchy of multicast groups. SPLINE [30] is a multi-server architecture that splits the virtual world into several zones (or *locales*) in which multicast transmission is used. [31] also suggests an *octree*-based approach for interest management using multicast groups. The department of Defense has been pursuing its own architecture, called HLA [32], for virtual environment interoperability which has been recently adopted by the IEEE. HLA filtering mechanisms are based on DIS experience with multicast and use the concept of *routing spaces*. A routing space is made of *subscription* regions corresponding to member's expression of interest and *update* regions that express what a member is able to produce; regions are rectangle areas in the routing space. However, none of these different works have presented an architecture to dynamically partition the VE into multicast groups, taking into account the density of participants per cell and the participants' capacities.

## VII. Conclusion and future work

We have described SCORE, a multicast-based communication protocol that enables LSVE applications to run on the Internet today. The intensive experimentations done using the SCORE implementation show that this protocol significantly improves scalability of such applications without adding critical overhead. Moreover, the scheme is flexible enough to benefit from new functionalities like the support for source filtering in IGMPv3[33]. However, we have shown that in some particular cases, a static partitioning scheme is sufficient. This situation occurs when the available number of multicast groups is large enough or when participants have high link bandwidth and processing resources available.

Directions for future work include the extension of the communication protocol to multi-flow sources, the detailed impacts of SCORE on multicast routing protocols, and the experimentation of this communication protocol with a real LSVE application on the Internet. We are currently integrating SCORE into the V-Eye application [34].

## Acknowledgments

The authors would like to thank the anonymous reviewers for helpful suggestions.

## References

[1] J. M. Pullen, M. Myjak, and C. Bouwens, "Limitations of internet protocol suite for distributed simulation in the large multicast environment," *RFC 2502*, February 1999.
[2] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proceedings ACM SIGCOMM*, Stanford, August 1996.
[3] E. Léty, L. Gautier, and C. Diot, "Mimaze, a 3d multi-player game on the internet," in *Proceedings 4th International Conference on VSMM (Virtual Systems and MultiMedia)*, Gifu, Japan, November 1998.
[4] B. N. Levine, J. Crowcroft, C. Diot, J. J. Garcia-Luna-Aceves, and J. F. Kurose, "Consideration of receiver interest in content for ip delivery," in *Proceedings IEEE INFOCOM*, 2000.
[5] B. N. Levine and J. J. Garcia-Luna-Aceves, "Improving internet multicast with routing labels," in *Proceedings ICNP*, Atlanta, GA, 1997.
[6] M. Oliveira, J. Crowcroft, and C. Diot, "Router level filtering for receiver interest delivery," in *Proceedings NGC*, November 2000.

14

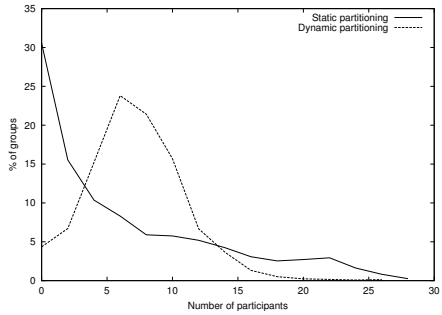



Fig. 14. Distribution of participants within multicast groups with $V = 10$ units/s, $RP = 1s$ and $IArea = 0.04$


[7] M. Sola, M. Ohta, and T. Maeno, "Scalability of internet multicast protocols," in *Proceedings of INET*, 1998.

[8] S. Kumary, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrinz, and M. Handley, "The masc/bgmp architecture for inter-domain multicast routing," in *Proceedings ACM SIGCOMM*, September 1998.

[9] S. Pejhan, A. Eleftheriadis, and D.Anastassiou, "Distributed multicast address management in the global internet," *IEEE Journal on Selected Areas in Communications*, pp. 1445–1456, October 1995.

[10] T. Ballardie, R. Perlman, C-Y Lee, and J. Crowcroft, "Simple scalable internet multicast," *UCL Research Note RN/99/21*, April 1999.

[11] H. W. Holbrook and D. R. Cheriton, "Express multicast making multicast economically viable," in *Proceedings ACM SIGCOMM*, September 1999.

[12] W. Fenner, "Internet group management protocol, version 2," *RFC-2236*, November 1997.

[13] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol," *RFC-1075*, November 1988.

[14] L. Rizzo, "Fast group management in igmp," in *Proceedings 4th Hipparch Workshop*, June 1998.

[15] Daniel J. Van Hook, Steven J. Rak, , and James O. Calvin, "Approaches to relevance filtering," in *Proceedings 11th DIS Workshop*, September 1994.

[16] F. Baccelli and S. Zuyev, "Poisson-voronoi spanning trees with applications to the optimization of communication networks," *in Proceedings of Operations Research*, 1999.

[17] J. Kingman, *Poisson processes*, Oxford Studies in Probability, No3. Clarendon Press, 1993.

[18] F. Baccelli and P. Brémaud, *Elements of Queueing Theory*, Springer Verlag, 1994.

[19] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 2, Chapman and Hall, Limited, 1971.

[20] D. Bertsekas and R. Gallager, *Data Networks*, chapter 6, pp. 524–529, Prentice-Hall, 1987.

[21] E. Léty, *Une architecture de communication pour environnements virtuels distribués à grande échelle sur l'Internet*, Ph.D. thesis, Université de Nice Sophia Antipolis, décembre 2000.

[22] E. Léty and T. Turletti, "Issues in designing a communication architecture for large-scale virtual environments," in *Proceedings the 1st International Workshop on Networked Group Communication*, Pisa, Italy, November 1999.

[23] E. Léty, T. Turletti, and F. Baccelli, "Cell-based multicast grouping in large-scale virtual environments," Tech. Rep. RR-3729, INRIA, July 1999.

[24] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. WeiP. Tsuchiya, "Protocol independent multicast-sparse mode (pim-sm): Protocol specification," *RFC-2362*, June 1998.

[25] S. J. Rak and D. J. Van Hook, "Evaluation of grid-based relevance filtering for multicast group assignment," in *Proceedings 14th DIS workshop*, 1996.

[26] L. Zou, M. H. Ammar, and C. Diot, "An evaluation of grouping techniques for state dissemination in networked multi-user games," in *ICNP*, Toronto, Canada, 1999.

[27] M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham, and S. Zeswitz, "Npsnet : A network software architecture for large scale virtual environments," *MIT Presence 3(4)*, 1994.

[28] C. Carlsson and O. Hagsand, "Dive - a multi user virtual reality system," in *Proceedings IEEE VRAIS*, Seattle, Washington, September 1993.

[29] C. Greenhalgh, "Dynamic, embodied multicast groups in massive-2," Tech. Rep. NOTTCS-TR-96-8 1, University of Nottingham, 1996.

[30] J. W. Barrus, R. C. Waters, and D. B. Anderson, "Locales: Supporting large multiuser virtual environments," *IEEE Computer Graphics and Applications*, pp. 16(6):50–57, November 1996.

[31] H. Abrams, K. Watsen, and M. Zida, "Three-tiered interest management for large-scale virtual environments," in *Proceedings VRST*, Taipei, Taiwan, 1998.

[32] J. Dahman, J. R. Weatherly, and F. Kuhl, *Creating Computer Simulation Systems: An Introduction to The High Level Architecture*, Prenctice Hall, 1999.

[33] B. Cain, S. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, "Internet group management protocol, version 3," *RFC-3376*, October 2002.

[34] A. Gourdon, "V-eye: A virtual eye lsve application," http://www-sop.inria.fr/planete/software/V-Eye/, November 2002.

# B. ARTICLE SARC

Cette annexe contient un article publié dans la revue *EURASIP Journal on Applied Signal Processing, Vol. 2004, No. 2, pp. 158-175, February 2004*. Il décrit le protocole SARC présenté dans le chapitre 3.

in EURASIP Journal on Applied Signal Processing, Vol 2004, No. 2, pp. 158-175, February 2004

# Source and channel adaptive rate control for multicast layered video transmission based on a clustering algorithm

Jérôme Viéron[*] Thierry Turletti[†] Kavé Salamatian [‡] Christine Guillemot

INRIA
Campus de Beaulieu, 35042 Rennes Cedex, France

**Abstract**

This paper introduces Source-channel Adaptive Rate Control (SARC) a new congestion control algorithm for layered video transmission in large multicast groups. In order to solve the well-known feedback implosion problem in large multicast groups, we first present a mechanism for filtering RTCP receiver reports sent from receivers to the whole session. The proposed filtering mechanism provides a classification of receivers according to a predefined similarity measure. An end-to-end source and FEC rate control based on this distributed feedback aggregation mechanism coupled with a video layered coding system is then described. The number of layers, their rate and levels of protection are adapted dynamically to aggregated feedbacks. The algorithms have been validated with the NS2 network simulator.

**Keywords -** Multicast, Congestion control, Layered video, Aggregation, FGS.

## 1 Introduction

Transmission of multimedia flows over multicast channels is confronted with the receivers heterogeneity problem. In a multicast topology (multicast delivery tree in the $1 \rightarrow N$ case, acyclic graph in the $M \rightarrow N$ case), network conditions such as loss rate and queueing delays are not homogeneous in the general case. Rather, there may be local congestions affecting downstream delivery of the video stream in some branches of the topology. Hence, the different receivers are connected to the source via paths with varying delays, loss and bandwidth characteristics. Due to this potential heterogeneity, dynamic adaptation of multimedia flows over multicast channels, for optimized QoS of multimedia sessions, faces challenging problems. The adaptation of source and transmission parameters to the network state often rely on the usage of feedback mechanisms. However, the use of feedback schemes in large multicast trees faces the potential problem of feedback implosion. This paper introduces Source-channel Adaptive Rate Control (SARC), a new congestion control algorithm for layered video transmission in large multicast groups. The first issue addressed here is therefore the problem of aggregating heterogeneous reports into a consistent view of the communication state. The second issue concerns the design of a source rate control mechanism that would allow a receiver to receive the source signal with quality commensurate with the bandwidth and loss capacity of the path leading to it.

Layered transmission has been proposed to cope with receivers heterogeneity [1, 2, 3]. In this approach, the source is represented using a base layer, and several successive enhancement layers refining the quality of the source reconstruction. Each layer is transmitted over a separate multicast group and receivers decide the number of groups to join (or leave) according to the

[*]Now with THOMSON multimedia R&D, 1 av Bellefontaine - CS 17616, 35576 Cesson-Sévigné, France. contact: jerome.vieron@thomson.net

[†]INRIA, 2004 route des Lucioles - BP 93,06902 Sophia Antipolis Cedex, France.

[‡]University of Paris VI, Paris, ,France.

quality of their reception. At the other side, the sender can decide the optimal number of layers and the encoding rate of each layer according to the feedback sent by all receivers. A variety of multicast schemes making use of layered coding for audio and video communication have been proposed, some of which rely on a multicast feedback scheme [3], [4]. Despite rate adaptation to the network state, applications have to face remaining packet losses. Error control schemes using FEC (*Forward Error Correction*) strongly reduce the impact of packet losses [5, 6, 7]. In these schemes, redundant information are sent along with the original information so that the lost data (or at least part of it) can be recovered from the redundant information. Clearly, sending redundancy increases the probability of recovering packets lost, but it also increases the bandwidth requirements and thus, the loss rate of the multimedia stream. Therefore, it is essential to couple the FEC scheme to the rate control scheme in order to jointly determine the transmission parameters (redundancy level, source coding rate, type of FEC scheme, etc.) as a function of the state of the multicast channel, to achieve the best subjective quality at receivers. For such adaptive mechanisms, it is important to have simple channel models that can be estimated in an on-line manner.

The sender, in order to adapt the transmission parameters to the network state, does not need reports of each receiver in the multicast group. It rather needs a partition of the receivers into homogeneous classes. Each layer of the source can then be adapted to the characteristics of one class or of a group of classes. Each class represents a group of homogeneous receivers according to discriminative variables related to the received signal quality. The clustering mechanism used here follows the above principles. A classification of receiver reports is performed by aggregation agents organized into a hierarchy of local regions. The approach assumes the presence of aggregation agents at strategic positions within the network. The aggregation agents classify receivers according to similar reception behaviors, and filter correspondingly the RTCP receiver reports. By classifying receivers, this mechanism solves the feedback implosion problem and at the same time provides the sender with a compressed representation of the receivers.

In the experiments reported in this paper we consider two pairs of discriminative variables in the clustering process: the first one constituted of the loss rate and the *goodput* and the second constituted of the loss rate and the throughput of a conformant TCP connection under similar loss and round trip time conditions. We show that approaches in which receivers rate-requests are only based on the *goodput* measure risk to lead to a severe sub-utilization of the network resources. To use a TCP throughput model, receivers have first to estimate their Round Trip Time (RTT) to the source. In order to do so we use the algorithm described in [4] jointly with a new application-defined RTCP packet, called *Probe-RTT*.

This distributed feedback aggregation mechanism is coupled with a video FGS layered coding system, to adapt dynamically the number of layers, the rate of each layer and its level of protection. Notice that the aggregation mechanism that has to be supported by the network nodes remains generic and can be used for any type of media. The optimization is performed by the sender and takes into account both the network aggregated state as well as the rate-distortion characteristics of the source. The latter allows to optimize the quality perceived by each receiver in the multicast tree.

The remainder of this paper is organized as follows. Section 2, provides an overview of related research on multicast rate and congestion control. Section 3 sets the main lines of SARC, our new hybrid sender/receiver driven rate control based on a clustering algorithm. The protocol functions to be supported by the receivers, and the receiver clustering mechanism governing the feedback aggregation are described respectively in section 4 and section 5. Section 6 describes the

multi-layer source and channel rate control and the multi-layered MPEG-4 Fine Grain Scalable source encoder [8, 9] that has been used in the experiments. Finally, experimental results obtained with the NS2 network simulator with various discriminative clustering variables (*goodput*, TCP-compatible throughput), including with the additional usage of forward error correction are discussed in section 7.

## 2 Related Work

Related work in this area focuses on error, rate and congestion control in multicast for multimedia applications. Layered coding is often proposed as a solution for rate control in video multicast applications over the Internet. Several - sender driven [10], receiver driven [11, 12], or hybrid schemes [13, 3, 14] - approaches have been proposed to address the problem of rate control in a multicast transmission. Receiver-driven approaches consist in multicasting different layers of video using different multicast addresses and let the receivers decide which multicast group(s) to subscribe to. RLM [11] and RLC [12] are two well-known receiver-driven layered multicast congestion control protocols. However, they both suffer from pathological behaviors such as transient periods of congestion, instability, and periodic losses. These problems mainly come from the bandwidth inference mechanism used [15]. For example, RLM uses *join-experiments* that can create additional traffic congestion, during transition periods corresponding to the latency for pruning a branch of the multicast tree. RLC [12] is a TCP-compatible version of RLM based on the generation of periodic bursts that are used for bandwidth inference on synchronization points indicating when a receiver can join a layer. Both the synchronization points and the periodic bursts can lead to periodic congestion and periodic losses [15]. PLM [16] is a more recent layered multicast congestion control protocol based on the generation of packet pairs to infer the available bandwidth. PLM does not suffer from the same pathological behaviors than RLM and RLC but requires a Fair Queuing network.

Battacharya & *al.* [17] present a general framework for the analysis of AIMD (Additive Increase Multiplicative Decrease) Multicast congestion control protocols. This paper shows that because of the so called "Path Loss Multiplicity Problem", unclever use of congestion information sent by receivers to sender, may lead to severe degradation and lack of fairness. This paper formalizes the multicast congestion control mechanism in two components : the Loss Indication Filter (LIF) and the rate Adjustement Algorithm. Our paper presents an implementation that minimises the Loss Multiplicity Problem by using a LIF which is implemented by a clustering mechanism (section 5.2) and a rate Adjustement Algorithm following the algorithm described in sections 4 and 6.

TFMCC [18] is an equation-based multicast congestion control mechanism that extends the TCP-friendly TFRC [19] protocol from the unicast to the multicast domain. TFMCC uses a scalable round-trip time measurements and a feedback suppression mechanism. However, since it is a single rate congestion control scheme, it cannot handle heterogeneous receivers and adapts its sending rate to the current limiting receiver.

FLID-DL [20] is a multi-rate congestion control algorithm for layered multicast sessions. It mitigates the negative impact of long IGMP leave latencies and eliminates the need for probe intervals used in RLC. However, the amount of IGMP and PIM SM control traffic generated by each receiver is prohibitive. WEBRC [21] is a new equation-based rate control algorithm that has been recently proposed. It solves the main drawbacks of FLID-DL using an innovative way to transmit data in waves. However, WEBRC such as FLID-DL are intended for reliable

download applications and possibly streaming applications but cannot be used to transmit real-time hierarchical flows such as H.263+ or MPEG-4.

A source adaptive multi-layered multicast - SAMM - algorithm based on feedback packets containing information on the estimated bandwidth available on the path from the source is described in [3]. Feedback mergers are assumed to be deployed in the network nodes to avoid feedback implosion. A mechanism based on *partial suppression* of feedbacks is proposed in [4]. This approach avoids the deployment of aggregation mechanisms in the network nodes, but on the other hand, the partial feedback suppression will likely induce a flat distribution of the requested rates.

MLDA [13] is a TCP-compatible congestion control scheme in which as in the scheme we propose, senders can adjust their transmission rate according to feedback information generated by receivers. However, MLDA does not provide a way to adapt the FEC rate in the different layers according to packet loss observed at receivers. Since the feedback only includes TCP-compatible rates, MLDA does not need feedback aggregation mechanisms and uses exponentially distributed timers and a *partial suppression* mechanism to prevent feedback implosion. However, when the receivers are very heterogeneous, the number of requested rates (in the worst case on a continuous scale) can potentially lead to a feedback implosion. Moreover, the partial suppression algorithm does not allow quantifying the number of receivers requesting a given rate in order to estimate how representative this rate is.

In [14], a rate based congestion and loss control mechanism for multicast layered video transmission is described. The strategy relies on a mechanism that aggregates feedback information in the networks nodes. However, in contrast with SAMM, the optimization is not performed in the nodes. Source and channel (FEC) rates in the different layers are chosen among a set of requested rates in order to maximize the overall PSNR seen by all the receivers. Receivers are classified according to their available bandwidth, and for each class of rate, two types of information are delivered to the sender: the number of receivers represented by this class and an average loss rate computed over all those receivers. It is supposed here that receivers with similar bandwidths have similar loss rates, which may not be always the case. In this paper, we solve this problem using a distributed clustering mechanism.

Clustering approaches have been already considered separately in [22] and [23]. In [22] a centralized classification approach based on k-means clustering is applied on a quality of reception parameter. This quality of reception parameter is derived based on the feedback of receivers consisting of reports including available bandwidth and packet loss. The main difference with our approach is that in our case the classification is made in a distributed fashion. Hence, receivers with similar bandwidths, but with different loss rates arc not classified within the same class. Therefore, with more accurate clusters, a better adaptation of the error-control process at the source level is possible. The global optimization performed is different and leads to improved performances. Moreover, [22] uses the RTCP filtering mechanism proposed in the RTP standard, i.e. they adapt the RTCP sending rate according to the number of receivers. However, when the number of receivers is large, it is not possible to get a precise snapshot of quality observed by receivers.

## 3 Protocol overview

This section gives an overview of the Source-channel Adaptive Rate Control (SARC) protocol proposed in this paper. Its design relies on a feedback tree structure, where the receivers are

organized into a tree hierarchy, and internal nodes aggregate feedbacks.

At the beginning of the session, the sender announces the range of rates (i.e. a rate interval $[R_{min}, R_{max}]$) estimated from the average rate-distortion characteristics of the source. The value $R_{min}$ corresponds to the bit rate under which the received quality would not be acceptable, and $R_{max}$ to the rate above which there is no significant improvement of the visual quality. This information is transmitted to the receivers at the start of the session. The interval $[R_{min}, R_{max}]$ is then divided into subintervals in order to only allow relevant values for layers rates. This quantization avoids having non quality discriminative layers.

After this initialization, the multicast layered rate control process can start. The latter assumes that the time is divided into feedback rounds. A feedback round comprises four major steps :

- At the beginning of each round the source announces the number of layers and their respective rates, via RTCP sender reports (SR). Each source layer is transmitted to an IP multicast group.
- Each receiver measures network parameters and estimates the bandwidth available on the path leading to it. The estimated bandwidth and the layer rates will trigger subscriptions or unsubscriptions to/from layers. Estimated bandwidth and loss rates are then conveyed to the sender via RTCP receiver reports (RR).
- Aggregation agents placed at strategic positions within the network classify receivers according to similar reception behaviors, i.e. according to a measure of distance between the feedback parameter values. On the basis of this clustering, these agents proceed with the aggregation of the feedback parameters, providing a representation of homogeneous clusters.
- The source then proceeds with a dynamic adaptation of the number of layers and of their rates in order to maximize the quality perceived by the different clusters.

The next sections describe in details each of the four steps.

## 4 Protocol functions supported by the receiver

Two bandwidth estimation strategies have been considered : the first approach measures the *goodput* of the path, and the second approach considered estimates the TCP-compatible bandwidth under similar conditions of loss rates and delays. This section describes the functions supported by the receiver in order to measure the corresponding parameters, and the multicast groups join and leave policy that has been retained. The bandwidth values estimated by the receivers are then conveyed to the sender via RTCP RR reports augmented with dedicated fields.

### 4.1 Goodput-based estimation

A notion of *goodput* has been exploited in the SAMM algorithm described in [3]. Assuming priority-based differentiated services for the different layers, the *goodput* is defined as the cumulated rate of the layers received without any loss. If a layer has suffered from losses, it will not be considered in the *goodput* estimation. The drawback of such a measure is that the estimated bandwidth will be highly dependent on the sending rates, hence it does not allow an accurate estimation of the link capacity. When no loss occurs, in order to best approach the link capacity, SAMM considers values higher than the *goodput* measured. Nevertheless, a loss rate of 0% is not

realistic on the Internet. Experiments have shown that this notion of *goodput* in a best-effort network in presence of cross traffic leads to estimated bandwidths decreasing towards zero during the sessions. Here, the *goodput* is defined instead as the rate received by the end system. A simple mechanism has been designed to try to approach the bottleneck rate of the link . If the loss rate is under a given threshold $T_{loss}$, the bandwidth value $B_t$ estimated at time $t$ is incremented as

$$B_t = B_{t-1} + \Delta \tag{1}$$

where $\Delta$ represents a rate increment and $B_{t-1}$ represents the last estimated value. Let $g_t$ be the observed *goodput* value at time $t$. Thus, when the loss rate becomes higher than the threshold $T_{loss}$, $B_t$ is set to $g_t$.

In the experiments we have taken $t_{loss} = 3\%$ and the $\Delta$ parameter increases similarly to the TCP increase, i.e. of one packet per round-trip time.

### 4.2 TCP-compatible bandwidth estimation

The second strategy considered for estimating the bandwidth available on the path relies on the analytical model of TCP throughput [24], known also as the TCP-compatible rate control equation. Notice however that the application of the model in a multicast environment is not straightforward.

#### 4.2.1 TCP throughput model

The average throughput of a TCP connection under given delay and loss conditions is given by [24]:

$$T = \frac{MSS}{RTT\sqrt{\frac{2p}{3}} + T_o min(1, 3\sqrt{\frac{3p}{8}})p(1+32p^2)}, \tag{2}$$

where $p$, $RTT$, $MSS$ and $T_o$ represent respectively the congestion event rate [19], the round-trip time, the Maximum Segment Size (i.e. maximum packet size) and the retransmit timeout value of the TCP algorithm.

#### 4.2.2 Parameters estimation

In order to be able to use the above analytical model, each receiver must estimate the RTT on its path. This is done using a new application-defined RTCP packet that we called (Probe-RTT). To prevent feedback implosion, only leaf aggregators are allowed to send Probe-RTT packets to the source. In case receivers are not located in the same LAN than their leaf aggregator, they should add the RTT to their aggregator; this can be easily estimated locally and without generating undesirable extra traffic. The source periodically multicast RTCP reports including the RTT computed (in ms) for the latest Probe-RTT packets received along with the corresponding SSRCs. Then, each receiver can update its RTT estimation using the result sent for its leaf agregator. The estimation of the congestion event rate $p$ is done as in [25] and the parameter $MSS$ is set to 1000 bytes.

#### 4.2.3 Singular receivers

In highly heterogeneous environments, under constraints of bounded numbers of clusters, the rate received by some end systems may strongly differ from their requests, hence from the TCP-compatible throughput value. The resulting excessively low values of congestion event rates

lead in turn to overestimated bandwidth values, hence to unstability. In order to overcome this difficulty, the TCP-compatible throughput $B_t$ at time $t$ is estimated as

$$B_t = min(T, max(S_{rate} + T_{rate}, B_{t-1})) \quad (3)$$

where $S_{rate}$ is the rate subscribed to and $T_{rate}$ is a threshold chosen so that the increase between two requests is limited (i.e. $T_{rate} = K * MSS/RTT$ with $K$ a constant). $B_{t-1}$ is the last estimated value of the TCP-compatible throughput. When the estimated throughput value $T$ is not reliable, the history used in the estimation of loss rates is re-initialized using the method described in [19]. We will see in the experimentation results that the above algorithm is still reactive and responsive to changes in network conditions.

#### 4.2.4 Slowstart

The slowstart mechanism adopted here differs from the approaches described in [19] and [18]. At the begining of the session or when a new receiver joins the multicast transmission tree, the requested rate is set to $R_{min}$. Then, after having a first estimation of $RTT$ and $p$, $T$ can be computed and the resulting requested rate $B_t^{slow}$ is given by

$$B_t^{slow} = max(T, g_t + K \times MSS/RTT) \quad (4)$$

where $g_t$ is the observed *goodput* value at time $t$ and $K$ is the same constant as the one used in section 4.2.3. The estimation given by (4) is used until we observed the first loss. After the first loss, the loss history is re-initialized taking $g_t$ as the available bandwidth and proceeding with Eq. (3).

### 4.3 Join/Leave policy

Each receiver estimates its available bandwidth $B_t$ and joins or leaves layers accordingly. However, the leaving mechanism has to take into account the delay between the instant a feedback is sent and the instant the sender adapts the layer rates accordingly. Undesirable oscillations of subscription may occur if receivers decide to unsubscribe a layer as soon as the TCP-compatible throughput estimated is lower than the current rate subscribed to. It is essential to let enough time for the source to adapt its sending rates, and then, only decide to drop a layer if the request has not been satisfied. It is why in order to still be reactive, we have chosen a delay of $K * RTT$ before leaving a layer except in the case where the loss rate becomes higher than a chosen acceptable bound $T_{loss}$ ($K$ is the same constant as the one used in section 4.2.3). These coupled mechanisms permit to avoid a waste of bandwidth due to IGMP (Internet Group Management Protocol) traffic.

### 4.4 Signalling protocol

The aggregated feedback informations (i.e. estimated bandwidth, loss rate) are periodically conveyed towards the sender in RTCP Receiver Reports(RR), using the RTCP report extension mechanism. The RR are augmented with the following fields:

- $EB$ : a 16 bits field which gives the value of the estimated bandwidth expressed in kbit/s.
- $LR$ : a 16 bits field which gives the value of the real loss rate.
- $NB$ : a 16 bits field which gives the number of clients requesting this rate (i.e. EB). This value is set to one by the receiver.

## 5 Aggregated feedback using distributed clustering

Multicast transmission has been reported to exhibit strong spatial correlations [26]. A classification algorithm can take advantage of this spatial correlation to cluster similar reception behaviors into homogeneous classes. In this way, the amount of feedback required to figure out the state of receivers can be significantly reduced. This will also helps in bypassing Loss Path Multiplicity problem explained in [17] by filtering out receivers report of losses. In our scheme, receivers are grouped into a hierarchy of local regions (see Fig. 1). Each region contains an aggregator that receives feedback, performs some aggregation statistics and send them in point-to-point to the higher-level aggregator (*merger*). The root of the aggregator tree hierarchy (called the *manager*) is based at the sender and receives the overall aggregated reports.

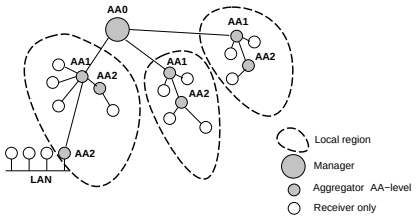



Figure 1: Multi-level hierarchy of aggregators

This architecture has a slight modification compared to the generic RTP architecture. Similar to the PIM-SSM context, RR receiver reports are not sent in multicast to the whole session, but are sent in point-to-point to a higher level aggregator. As these RTCP feedback are local to an aggregator region and will not cross the overall multicast tree, they may be set to be more frequent without breaking the 5% of the overall traffic constraint specified by the RTP standard.

### 5.1 Aggregators organization within the network

Aggregation agents (AAs) must be set up at strategic positions within the network in order to minimize the bandwidth overhead of RTCP receiver reports (RRs). Several approaches have been proposed to organize receivers in a multicast session to make scalable reliable multicast protocols [27]. We have chosen a multi-level hierarchical approach such as described in the RMTP [28] protocol in which receivers are also grouped into a hierarchy of local regions. However, in our approach, there are no designated receivers: all receivers send their feedback to their associated aggregator.

The root of the aggregator tree hierarchy (called the *Manager*) is based at the sender and receives the overall summary reports. The maximal allowed height of the hierarchical tree is set to 3 as recommended in [29]. In our approach, the overall summary report is a classification containing the number of receivers in each class and the mean behaviour of the class. The mechanism of aggregation is described in section 5.2.

In our experiments, aggregators are manually set up within the network. However, if extra router functionalities are available, several approaches can be used to automatically launch aggregators within the network. For example, we can implement the *aggregator* function using a

*custom concast* [30]. *Concast* addresses are the opposite of multicast addresses, i.e. they represent groups of senders instead of groups of receivers. So, a *concast* datagram contains a multicast group source address and a unicast destination address. With such a scheme, all receivers send their RRs feedback packets using the RTCP source group address and to the sender's unicast address, and only one aggregated packet is delivered to the sender. The *custom concast* signaling interface allows the application to provide the network the description of the merging algorithm function.

### 5.2 Clustering mechanism

The clustering mechanism is aimed toward taking advantage of the spatial and temporal correlation between receiver state of reception. Spatial correlation means that there is redundancy between reception behavior of neighbor receivers. This redundancy can be removed by compression methods. This largely reduce the amount of data required for representing feedback data sent by receivers. The compression is achieved by clustering similar (by a predefined similarity measure) reception behaviors into homogeneous classes. In this case, the clustering can be viewed as a vector quantization [31] that construct a compact representation of the receivers as a classification of receivers issuing similar receiver report. Moreover, for sender-based multicast regulation, only a classification of receivers is sufficient to apply adaptation decisions.

The clustering mechanism can also take advantage from time redundancy. For this purpose, classification of receivers should integrate the recent history of receivers as well as the actual receiver reports. Different reception states experienced by receivers during past periods, are treated as reports of different and heterogeneous receivers. By this way temporal variation of the quality of a receiver reception are integrated in the classification. A receiver that observes temporal variation may change its class during time.

In a stationary context, the classification would converge to a stable distribution. This stationary distribution will be a function of the spatial as well as the temporal dependencies. However, since over large time scales, the stationary hypothesis cannot be always validated, a procedure should be added to track variation of the multicast channel and adapt the classification to it. This procedure can follow a classical exponential weighting that drive the clustering mechanism to forget about far past time reports. In this weighting mechanism, the weight of clusters is multiplied by a factor ($\gamma < 1$) at the end of each reporting round, and clusters with weight below a threshold are removed.

Before describing the classification algorithm, several concepts should be introduced. First, we should choose the discriminative characteristic and the similarity (or dissimilarity) measure needed to detect similar reception behavior.

#### 5.2.1 Discriminative network characteristics

In the system presented in this paper, we have considered two pairs of discriminative variables: the first one constituted of the loss rate and the *goodput* (cf section 4.1) and the second constituted of the loss rate and a TCP-compatible bandwidth share measure (cf section 4.2). Both loss rate and bandwidth characteristics (*goodput* or TCP-compatible) are clearly relevant not only as network characteristics but also as video quality parameters.

#### 5.2.2 Similarity measure

Two kinds of measure should be defined : The similarity measure between two observed reports $x$ and $y$ ($d(x,y)$) and between an observed report $x$ and a cluster $C$ ($d(x,C)$). The former similarity measure can stand for the simple $L^p$ distance ($d(x,y) = \sqrt[p]{\sum_i (x_i - y_i)^p}$ ) or any other more sophisticated distance suitable to a particular application. The retained similarity measure used in this work is given by $d(x,y) = \max_i \frac{abs(x_i - y_i)}{dt_i}$ where $dt_i$ is a chosen threshold for the dimension $i$. The latter similarity measure is more difficult to apprehend. The simplest way is to choose in each cluster a representative $\hat{x}_C$ and to assign the distance $d(x,\hat{x}_C)$ to the distance between the point and the cluster ($d(x,C) = d(x,\hat{x}_C)$). We can also define the distance to cluster as the distance to the nearest or the furthest point of the cluster ($d(x,C) = \min_{y \in C} d(x,y)$ or $d(x,C) = \max_{y \in C} d(x,y)$). The distance can also be a likelihood derived over a model mixture approach. The type of measure used will impact over the shape of the cluster and over the classification.

#### 5.2.3 Classification Algorithm

Each cluster is represented by a representative point and a weight. The representative point can be seen as a vector the components of which are given by the discriminative variables considered in the clustering process.

The clustering algorithm is initialized with a maximal number of classes ($N_{\max}$) and a cluster creation threshold ($d_{th}$). Aggregation agents regularly receive RTCP reports from receivers and/or other aggregation agents in their coverage area as described in section 5.1. To classify the receiver reports in the different clusters we use a very simple Nearest Neighbor (NN) $k$-means clustering algorithm (see pseudo-code shown in Fig. 2). Even if this algorithm might be subject to largely reported deficiencies as false clustering, dependences to the order of presentation of samples and non-optimality which has lead researchers to develop more complex clustering mechanism as mixture modelling, we believe that this rather simple algorithm attain the goals of our approach which is to filter out receiver reports to a compact classification in a distributed, an asynchronous way. A new report joins the cluster that has the lowest Euclidean ($L^2$) distance to it, and updates the cluster representative by a weighted average of the points in the cluster. When a new point joins a cluster, it changes slightly the representative point which is defined as the cluster center and updates the weight of the cluster; afterwards the point is dropped to achieve compression. If this minimal distance is more than a predefined threshold, a new cluster is created. This bounds the size of the cluster. We also use a maximal number of clusters (or classes) which is fixed to 5, as it is not realistic to have more layers in such a layered multicast scheme.

At the end of each reporting round, the resulting classification is sent back to the higher level aggregation agent (i.e. the manager) in form of a vector of clusters representatives and of their associated weights and clusters are reset to a null weight. Clusters received by different lower level aggregation agents are classified following a similar clustering algorithm which will aggregate representative points of clusters, i.e. cluster center, with the given weight. This amounts to applying the Nearest Neighbor clustering algorithm to the representative points reported in the new coming receiver report.

At the higher level of the aggregators hierarchy, the clustering generated by aggregating lower level aggregator reports is renewed at the beginning of each reporting round.

As explained before, the classification of receivers should also integrate the recent history of

$d_{th}$ =*predefined threshold*
$N_{\max}$ =*maximal number of clusters (5)*
$\mathbf{r}$ = *received receiver report*
search for the nearest cluster $d(\mathbf{r}, \hat{C}) = \min_C d(\mathbf{r}, C)$
**if** $(d(\mathbf{r}, \hat{C}) \geq d_{th})$
  **if** (Number of existing cluster $< N_{\max}$)
    Add a new cluster $C_{\text{new}}$ and set $\hat{C} = C_{\text{new}}$
Recalculate representative of cluster $\hat{C}$,
$\hat{x}_{\hat{C}} = \frac{weight(\hat{C})\hat{x}_{\hat{C}} + \mathbf{r}}{weight(\hat{C})+1}$
Increment the weight of cluster $\hat{C}$

Figure 2: NN clustering algorithm.

$w_{\min}$ =*predefined cluster suppression threshold*
$\gamma$ =*memory weight*
At the beginning each reporting round
  **for** all clusters $C$
    %*Weight the current normalized cluster by* $\gamma$
    $weight(C) = weight(C) * \gamma$
    **if** $weight(C) < w_{\min}$
      Remove cluster $C$
  Aggregate new normalized reports
  Send aggregate reports to the sender

Figure 3: Aggregation algorithm at highest level with memory weighting.

receivers. This memory is introduced into the clustering process by using the cluster obtained during past reporting round as an *a priori* in the highest level of the aggregator hierarchy.

Nevertheless, since, over large time scales, the stationary hypothesis cannot be always validated, a procedure must be added to ensure that we forget about far past time reports, to not bias the cluster representative by out-of-date reports. This is handled by an exponential weighting heuristic: at each reporting round the weight of a cluster is reduced by a constant factor (see Fig. 3). If the weight of a cluster falls below a cluster suppression threshold level, the cluster is removed.

#### 5.2.4 Cluster management

The clustering algorithm implements three mechanisms to manage the number of clusters: a cluster addition, a cluster removal and a cluster merge mechanisms. The cluster addition and the cluster removal mechanisms have been described before. The cluster merging mechanism aims to reduce the number of clusters by combining two clusters that have been driven very close to each other. The idea behind this mechanism is that clusters should fill up uniformly the space of possible reception behaviors. The cluster merging mechanism merges two clusters that have a distance lower than a quarter of the cluster creation threshold ($d_{th}$). The distance between the two clusters is defined as the weighted distance of the cluster representatives. The merging threshold is chosen based on the heuristic that: 1) $d_{th}$ defines the fair diameter of a cluster and 2) two clusters that are distant by $\frac{d_{th}}{4}$ may create by merging a cluster of diameter smaller than

$d_{th}$. The cluster merging mechanism replaces the two clusters with a new cluster represented by a weighted average of the two cluster representatives and a weight corresponding to the sum of the two clusters.

The combination of these three mechanisms of cluster management creates a very dynamic and reactive representation of the reception behaviour observed during the multicast session.

## 6 Layered source coding rate control

The feedback channel created by the clustering mechanism offers periodically to the sender information about the network state. More precisely, this mechanism delivers a loss rate, a bandwith limit and the number of receivers within a given cluster. This information is in turn exploited to optimize the number of source layers, the coding mode, the rate and the level of protection of each layer. This section first describes the media and FEC rate control algorithm that takes into account both the network state and the source rate-distortion characteristics. The Fine Grain Scalable (FGS) video source encoding system used and the structure of the streaming server considered are then described.

### 6.1 Media and FEC rate-distortion optimization

We consider in addition the usage of forward error correction. In the context of transmission on the Internet, error detection is generally provided by the lower layer protocols. Therefore, the upper layers have to deal mainly with erasures, or missing packets. The exact position of missing data being known, a good correction capacity can be obtained by systematic MDS codes [32]. An (n,k) MDS - Maximal Distance Separable - code takes k data packets and produces $n-k$ redundant data packets. The MDS property allows to recover up to $n-k$ losses in a group of $n$ packets. the effective loss probability $P_{eff}(k)$ of an MDS code, after channel decoding, is given by

$$P_{eff}(k) = P_e \left( \sum_{j=0}^{k-1} \left( \begin{array}{c} n-1 \\ j \end{array} \right) P_e^{n-1-j} (1-P_e)^j \right), \tag{5}$$

where $P_e$ is the average loss probability on the channel. One question to be solved is then, given the effective loss probability, how to split in an optimal way the available bandwidth for each layer between raw and redundant data. This amounts to finding the level of protection (or the code parameter $k/n$) for each layer.

The rate for both raw data and FEC (or equivalently the parameter $k/n$) are optimized jointly as follows. For a maximum number of layers $L$ supported by the source, the number of layers, their rate and level of protection are chosen in order to maximize the overall PSNR (Peak Signal to Noise Ratio) seen by all the receivers. Note that the rates are chosen in the set of $N$ requested rates (feedback information). This can be expressed as

$$(\Omega_1, \ldots, \Omega_l) = \arg \max_{(\Omega_1,\ldots,\Omega_l)} G, \tag{6}$$

Where $\Omega_i = (r_i, \frac{\kappa_i}{n}), i = 1, \ldots, l$ with $r_i$ representing the cumulated source and channel rate and $\frac{\kappa_i}{n}$ the level of protection for each layer $i$. The quality measure $G$ to be maximized is defined as

$$G = \sum_{j=1}^{N} \left( \sum_{i=1}^{l} \mathrm{PSNR}(\Omega_i).\mathbf{P}_{\mathbf{eff_{j,i}}} \right).C_j \tag{7}$$

$$\text{where } l = \arg \max_{k \in [1,\ldots,L]} \left\{ \sum_{i=1}^{k} r_i \leq R_j \right\}. \tag{8}$$

The terms $R_j$ and $C_j$ represent respectively the requested rate and the number of receivers in the cluster $j$. The term PSNR($\Omega_i$) denotes the PSNR increase associated with the reception of the layer $i$. Note that the PSNR corresponding to a given layer $i$ depends on the lower layers. The term $\mathbf{P_{eff_{j,i}}}$ denotes the probability, for receivers of cluster $j$, that the $i$ layers are correctly decoded and can be expressed as

$$\mathbf{P}_{eff_{j,i}} = \prod_{k=1}^{i} \left(1 - \bar{p}_{eff_{j,k}}\left(\frac{\kappa_k}{n}\right)\right), \tag{9}$$

where $\bar{p}_{eff_{j,k}}$ is the effective loss probability observed by all the receivers of the cluster $j$ receiving the $k$ considered layers. The values PSNR($\Omega_i$) are obtained by estimating the rate-distortion $D(R)$ performances of the source encoder on a training set of sequences. The model can then be refined on a given sequence during the encoding process, if the coding is performed in real-time, or stored on the server in the case of streaming applications.

The upper complexity bound, in the case of an exhaustive search, is given by $\frac{L!}{N!(N-L)!}$ where $L$ is the maximum number of layers and $N$ the number of clusters. However, this complexity can be significantly reduced by first sorting the rates $R_j$ requested by the different clusters. Once the rates $R_j$ have been sorted, the constraint given by Eq.(8) allows to limit the search space of the possible combinations of rate $r_i$ per layer. Hence, the complexity of an exhaustive search within the resulting set of possible values remains tractable. For large values of $L$ and $N$, the complexity can be further reduced by using dynamic programming algorithm [33].

Notice that here we have not considered the use of hierarchical FEC. The FEC used here (i.e. MDS codes) are applied on each layered separately. Only their rates $k_i/n$ are optimized jointly. The algorithm could be extended by using layered FEC as described in [34].

### 6.2 Fine Grain Scalable source

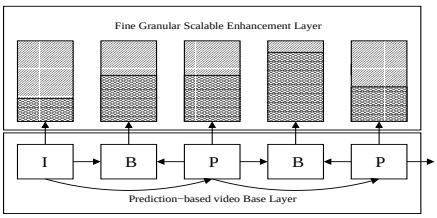



Figure 4: FGS video coding scalable structure.

The layers are generated by an MPEG-4 Fine Granular Scalability (FGS) encoder [8, 9]. FGS has been introduced in order to cope with the adaptation of source rates to varying network bandwidth in the case of streaming applications with pre-encoded streams. Indeed, even if classical scalable (i.e. SNR, spatial, temporal) coding schemes provide elements of response to

the problem of rate adaptation to network bandwidth, those approaches suffer from limitations in terms of adaptation granularity. The structure of the FGS method is depicted in Fig. 4. The base layer is encoded at a rate denoted $R_{BL}$, using a hybrid approach based on a motion compensated temporal prediction followed by a DCT-based compression scheme. The enhancement layer is encoded in a progressive manner up to a maximum bit-rate denoted $R_{EL}$. The resulting bitstream is progressive and can be truncated at any points, at the time of transmission, in order to meet varying bandwidth requirements. The truncation is governed by the rate-distortion optimization described above, considering the rate-distortion characteristics of the source. The encoder compresses the content using any desired range of bandwidth $[R_{min} = R_{BL}, R_{max}]$. Therefore, the same compressed streams can be used for both unicast and multicast applications.

### 6.3 Multicast FGS streaming server

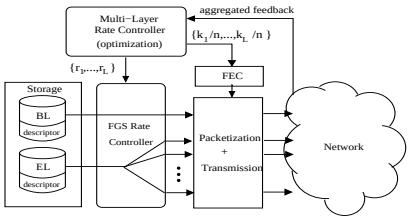



Figure 5: Multicast FGS streaming server.

The experiments reported in this paper are done assuming an FGS streaming server. Fig. 5 shows the internal structure of the multicast streaming system considered including the layered rate controller and the FEC module. For each video sequence pre-stored on the server, we have two separate bitstreams (i.e. one for BL and one for EL) coupled with its respective descriptors. These descriptors contain various informations about the structure of the streams. Hence, it contains the offset (in bytes) of the beginning of each frame within the bitstream of a given layer. The descriptor of the base layer contains also the offset of the begining of a slice (or video packet) of an image. The composition timestamp (CTS) of each frame used as the presentation time at the decoder side is also contained in the descriptor.

Upon receiving a new list $(r_0, r_1, \ldots, r_L)$ of rate constraints, the FGS rate controller computes a new bit budget per frame (for each expected layer) taking into account the frame rate of the video source. Then, at the time of transmission, the FGS rate controller partitions the FGS enhancement layer into a corresponding number of "sub_layers". Each layer is then sent to a different IP multicast group. Notice that, regardless of the number of FGS enhancement layers that the client subscribes to, the decoder has to decode only one enhancement layer (i.e. the "sub_layers" of the enhancement layer merge at the decoder side).

### 6.4 Rate control signalling

In addition to the value of the RTT computed for the *Probe-RTT* packets, the RTCP sender reports periodically sent include information about the sent layers i.e., their number, their rate) and their level of protection, according to the following syntax:

- $NL$ : an 8 bits field which gives the number of enhancement layers.
- $BL$ : a 16 bits field which gives the rate of the base layer.
- $EL_i$ : a set of 16 bits fields which give the rate of the enhancement layer i, $i \in 1, \ldots, NL$.
- $k_i$ : a set of 8 bits fields conveying the rate of the Reed-Solomon code used for the protection of layer i, $i \in 0, \ldots, NL$ [1].

## 7 Experimental results

The performance of the SARC algorithm has been evaluated considering various sets of discriminative clustering variables using the $NS$ 2 version 2.1$b$6 network simulator.

### 7.1 Analysis of fairness

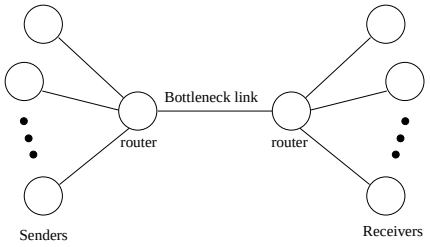


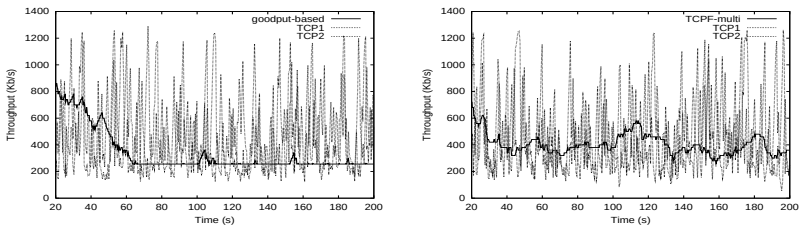

Figure 6: Simulation Topology (bottleneck).

The first set of experiments aimed at analyzing the fairness of the flows produced against conformant TCP flows. Fairness has been analyzed using the single bottleneck topology shown in Fig. 6, where a number of sending nodes are connected to as many receiving nodes via a common link with a bottleneck rate of 8 Mb/s and a delay of 50ms. The video flows controlled by the above algorithm are competing with 15 TCP conformant flows. Fig. 7-(a) depicts the respective throughput of one video flow controlled with the *goodput* measure and of two out of the 15 TCP flows. Fig. 7-(b) depicts the throughputs obtained when using the TCP-compatible rate equation. As expected, the flow regulated with the *goodput* measure does not compete fairly with the TCP flows (cf. Fig. 7-(a)). In presence of cross-traffic at high rate, the estimated bandwidth decreases regularly to reach the lower bound $R_{min}$ that has been set to 256 Kb/s. The average thoughput of the flow regulated with the TCP-compatible measure matches closely the average TCP throughput with a smoother rate (cf. Fig. 7-(b)).

### 7.2 Loss rate and PSNR performances

The second set of experiments aimed at measuring the PSNR and loss rate performances of the rate control mechanism, with the two measures (*goodput*, TCP-compatible measure), without

[1] Here we consider Reed-Solomon codes of rates $k/n$. The value of $n$ is fixed at the beginning of the session and only the parameter $k$ is adapted dynamically during the session. However, we could also easily consider to adapt the parameter $n$, therefore the syntax of the SR packet would have to be extended accordingly.




(a) (b)

Figure 7: Respective throughputs of two TCP flows and of one rate controlled flow with (a) a measure of goodput; (b) the TCP-compatible measure.

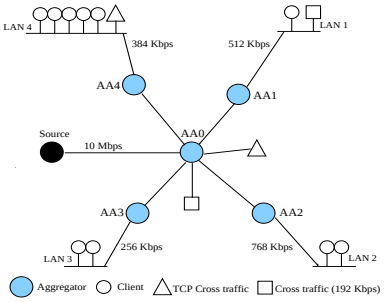



Figure 8: Simulated topology.

and with the presence of FEC. We have considered the multicast topology shown in Fig. 8. The periodicity of the feedback rounds is set to be equal to the maximum RTT value of the set of receivers. The sequence used in the experiments, called "Brest" [2], has a duration of $300s$ (25 Hz, 6700 frames). The rate-distortion characteristics of the FGS source is depicted in Fig. 9. The experiments depicted here are realized with the MoMuSys MPEG-4 version 2 video codec [9].

### 7.2.1 Testing scenario

Given the topology of the multicast tree, we have considered a source representation on three layers, each layer being transmitted to an IP-multicast address. The base layer is encoded at a constant bit rate of $256kb/s$. The overall rate (base layer plus two enhancement layers) ranges from $256Kbit/s$ up to $1Mb/s$. At $t = 0$ each client subscribes to the three layers with respective initial rates of $R_{BL} = 256kb/s$, $R_{EL1} = 100kb/s$ and $R_{EL2} = 0kb/s$. During the session, the video stream has to compete with point-to-point UDP cross-traffic with a constant bit rate of $192kb/s$ and with TCP flow. These competing flows contribute to a decrease of the links bottleneck. The activation of the cross-traffic between clients represented by "squares" on Fig. 8, in the time interval $[100s, 200s]$, limits the bottleneck of the corresponding link (i.e. LAN 1's client) down to

[2] courtesy of Thomson Multimedia R&D France

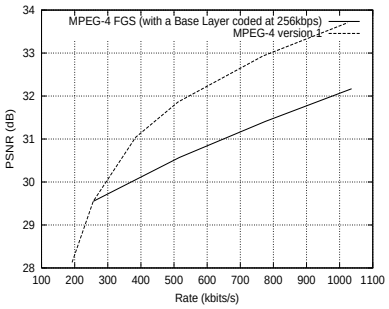



Figure 9: Rate-distortion model of the FGS video source.

$320kb/s$. Similarly, competing TCP traffic is generated between clients denoted by "triangles" in the interval $[140s, 240s]$ leading to a bottleneck rate of the link (i.e. LAN 4's clients) down to $192kb/s$ during the corresponding time interval.

The first test aimed at showing the benefits for the quality perceived by the receivers of an overall measure that would also take into account the source characteristics (and in particular the rate-distortion characteristics) versus a simple optimization of the overall *goodput*. Thus, we compare our results with the SAMM algorithm proposed in [3]. The corresponding mechanism is called SAMM-like in the sequel.

Our algorithm relying on the rate-distortion optimization has then been tested with respectively the *goodput* and the TCP-compatible measures, in order to evidence the benefits of the TCP-compatible rate control in this layered multicast transmission system. In the sequel these approaches are respectively called GB-SARC (Goodput-Based Source Adaptive Rate Control) and TCPF-SARC (TCP-Friendly Source Adaptive Rate Control). The constant $K$ is set to 4 in the experiments. In addition, in order to evaluate the impact of the FEC we have considered the TCP-compatible bandwidth estimation both without and with FEC (TCPF-SARC+FEC) for protecting the base layer. When FEC is not applied, the $k_i$ parameter of each layer is set to $n$ (i.e. 10 in the experiments).

### 7.2.2 Results

Fig. 10 and 11 show the results obtained with the SAMM-like algorithm. It can be seen that the SAMM-like approach does not permit an efficient usage of the bandwidth. For example, the LAN 2's client (with a link with a bottleneck rate of 768 kb/s) has not received more than 300 kb/s on its link. Similar obervations can be done with the receivers of the other LANs. Notice also that if the rate had not been lower bounded by an $R_{min}$ value, the *goodput* of the different receivers would have converged to a very small value. In addition to the highly suboptimal usage of bandwidth, the approach suffers from a very unstable behavior in term of subscriptions and unsubscriptions to multicast groups.

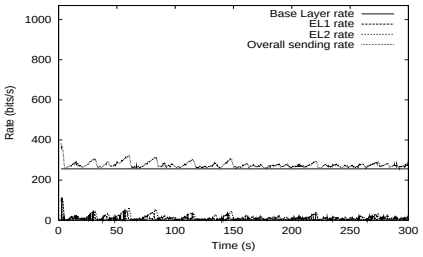
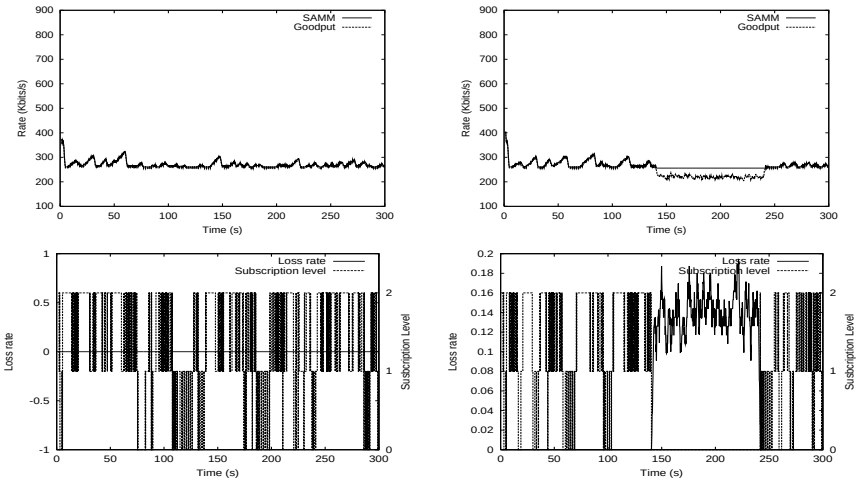



Figure 10: Rate variations for each layer of the FGS video source with the SAMM-like approach.




(a) (b)

Figure 11: SAMM-like throughput vs real goodput measure, Loss rate and Subscription level obtained for (a) a LAN 2's client (link 768kb/s), (b) a LAN 4's client (link 384kb/s).

Fig. 12, 14 and 16 show the rate variations of the different layers of the FGS source over the session, obtained respectively with the GB-SARC, TCPF-SARC and TCPF-SARC+FEC methods. Fig. 13, 15 and 17 depict the throughput estimated with these three methods versus the real measures of *goodput*, the loss rate, the number of layers received and PSNR values observed for two representative clients (i.e. LAN 2 with a bottleneck rate of $768Kb/s$ and LAN 4 with a bottleneck rate of $384Kb/s$).

Fig. 12 and 13, with the GB-SARC agorithm, show that the rate control taking into account the PSNR (or rate-distortion ) characteristics of the source leads to a better bandwidth utilization that the SAMM-like approach. In addition, the throughput estimated follows closely the bottleneck rates of the different links. Moreover, the number of irrelevant subscriptions and unsubscriptions to multicast groups is strongly reduced. However, the loss rates observed remain high. For example, the LAN 4's client observe an average loss rate of 30% between 240s and 300s. This is due to the fact that during this time interval the receiver of LAN 1 (bottleneck rate of $512kb/s$) has subscribed to the first enhancement Layer (EL1), hence the rate of this layer is higher than the bottleneck rate of LAN 4's clients. In this case the GB-SARC algorithm does not permit a reliable bandwidth estimation for the LAN 4's clients. As expected, the quality of the received video suffers from the high loss rates and the obtained PSNR values are relatively low. Finally, another important drawback is that during the corresponding period the rate constraints given to the FGS video streaming server are very unstable (see Fig. 12).

With the TCPF-SARC algorithm (cf Fig. 14 and 15), the sending rates of the different layers follows closely the variations of the bottleneck rates of the different links. This leads to stable sessions with low loss rates and with a restricted number of irrelevant subscriptions and unsubscriptions to multicast groups. The comparison of the PSNR curves in Fig. 15 reveals a gain of at least 2db for LAN 2 with respect to LAN 4. This evidences the interest of such multilayered rate control algorithm in a multicast heterogeneous environment. Notice that the peaks of instantaneous loss rates observed result from a TCP-compatible prediction which occasionally exceeds the bottleneck rate. Also, in Fig. 15-(b) the loss rate observed over the time interval $[140s, 240s]$ remains constant and relatively high. This comes from the fact that, in presence of competing traffic, the bottleneck rate available for the video source is lower than the rate of the base layer which in the particular case of an FGS source is maintained constant in average (e.g. $256Kb/s$).

The FEC permits to improve slightly the PSNR performances, especially for the receivers of LAN4 (cf. Fig. 17-(b)). It can be seen on Fig. 16 that the usage of FEC however leads to a bit more unstable behavior, i.e. to higher rate fluctuations of the different layers of the FGS source.

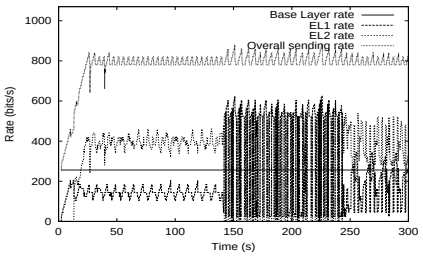
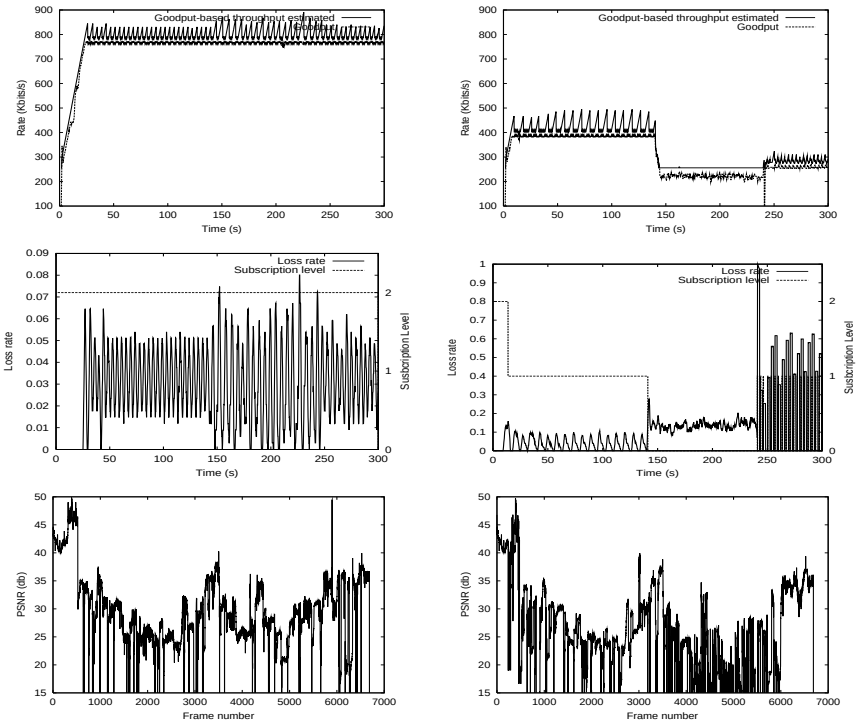



Figure 12: Rate variations for each layer of the FGS video source with the GB-SARC approach.




(a) (b)

Figure 13: GB-SARC throughput vs real goodput measure, Loss rate, Subscription level and PSNR obtained for (a) a LAN 2's client (link 768kb/s), (b) a LAN 4's client (link 384kb/s).

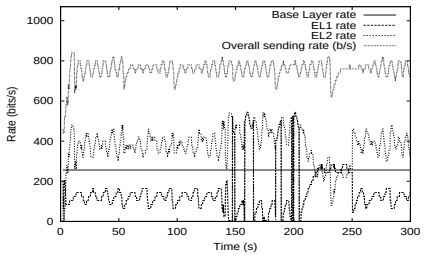
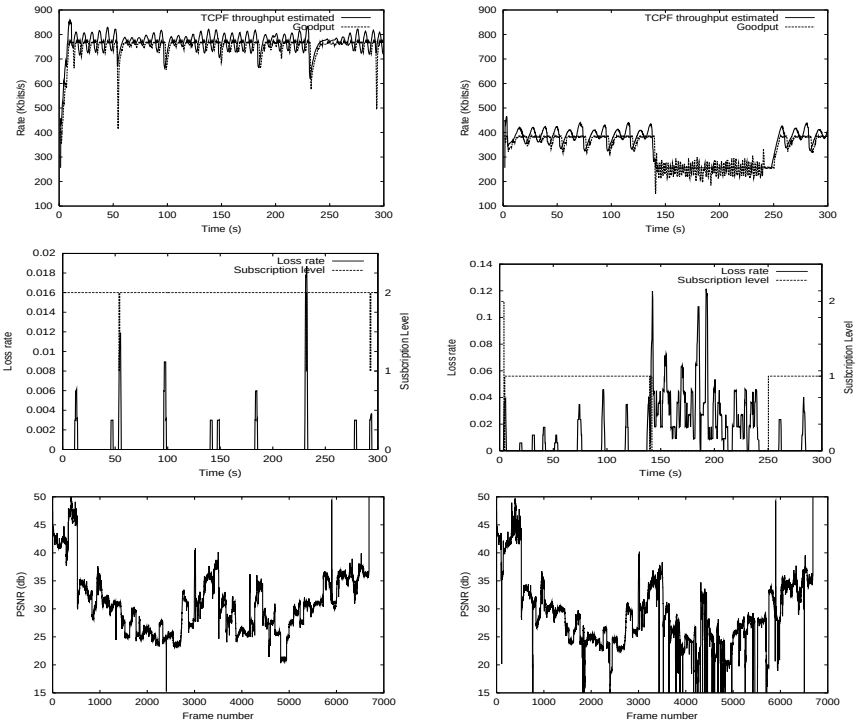



Figure 14: Rate variations for each layer of the FGS video source with the TCPF-SARC approach.




(a) (b)

Figure 15: TCPF-SARC throughput vs real goodput measure, Loss rate, Subscription level and PSNR obtained for (a) a LAN 2's client (link 768kb/s), (b) a LAN 4's client (link 384kb/s).

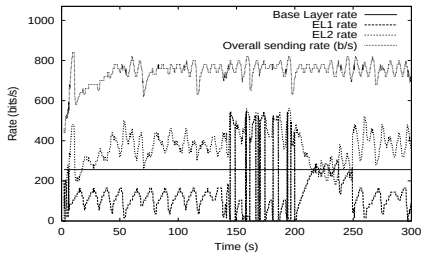



Figure 16: Rate variations for each layer of the FGS video source with the TCPF-SARC approach and additionnal FEC.

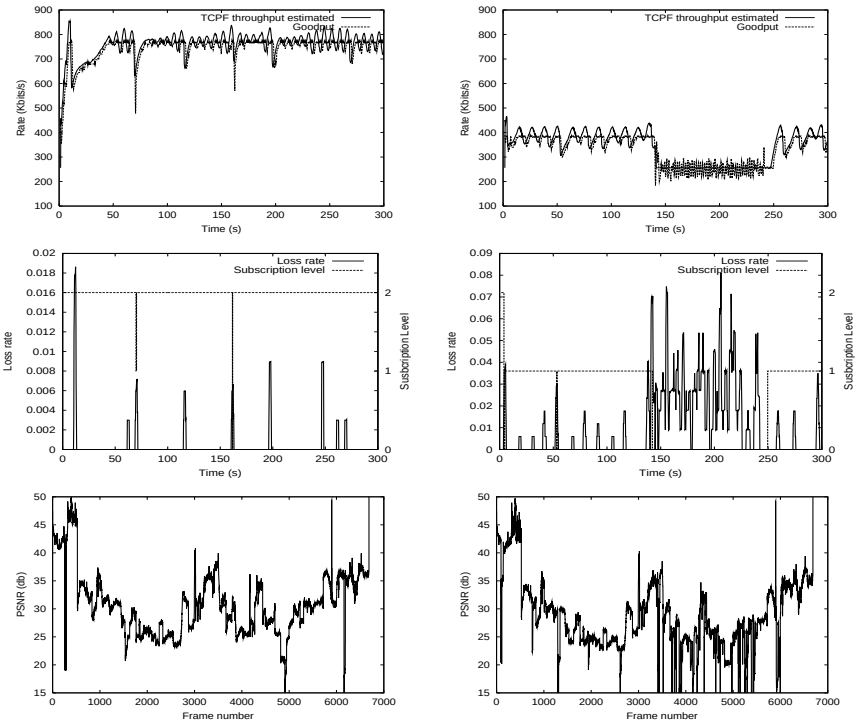



Figure 17: TCPF-SARC throughput with FEC vs real goodput measure, Loss rate, Subscription level and PSNR obtained for (a) a LAN 2's client (link 768kb/s), (b) a LAN 4's client (link 384kb/s).

## 8 Conclusion

In this paper, we have presented a new multicast multilayered congestion control protocol called SARC (Source-channel Adaptive Rate Control). This algorithm relies on an FGS layered video transmision system in which the number of layers, their rate, as well as their level of protection, are adapted dynamically in order to optimize the end-to-end QoS of a multimedia multicast session. A distributed clustering mechanism is used to classify receivers according to the packet loss rate and the bandwidth estimated on the path leading to them. Experimentation results show the ability of the mechanism to track fluctuation of the available bandwidth in the multicast tree, and at the same time the capacity to handle fluctuating loss rates. We have shown also that using loss rate and TCP-compatible measures as discriminative variables in the clustering mechanism leads to higher overall PSNR (hence QoS) performances than using the loss rate and *goodput* measures.

## References


[1] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of Conference of the Special Interest Group on data COMMunnication, ACM SIGCOMM'96*, pages 117–130, Stanford, CA, August 1996.

[2] T. Turletti, S. Fosse-Parisis, and J.C. Bolot. Experiments with a layered transmission scheme over the internet. Technical Report RR-3296, INRIA Sophia-Antipolis, 1998.

[3] B. J. Vickers, C. Albuquerque, and T. Suda. Source adaptive multi-layered multicast algorithms for real-time video distribution. *IEEE/ACM Transactions on Networking*, 8(6):720–733, December 2000.

[4] D. Sisalem and A. Wolisz. MLDA: A tcp-friendly congestion control framework for heterogeneous multicast environments. Technical report, GMD FOKUS, Berlin, Germany, 2000.

[5] Y. Wang and Q.F. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5):974–997, May 1998.

[6] J.C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive fec-based error control for internet telephony. In *Proceedings of the Conference on Computer Communications, IEEE Infocom'99*, pages 1453–1460, NY, March 1999.

[7] K. Salamatian. Joint source-channel coding applied to multimedia transmission over lossy packet network. In *Proceedings Packet Video Workshop, PV'99*, New York City, NY, USA, April 1999.

[8] H. Radha and Y. Chen. Fine granular scalable video for packet networks. In *Proceedings of Packet Video Workshop, PV'99*, Columbia University, NY, USA, April 1999.

[9] MObile MUltimedia SYStems (MoMuSys) Software. *MPEG-4 video verification model 4.1*, December 2000.

[10] J.C. Bolot, T. Turletti, and I. Wakeman. Scalable feedback control for multicast video distribution in the internet. In *Proceedings of Conference of the Special Interest Group on data COMMunnication, ACM SIGCOMM'94*, pages 58–67, London, UK, September 1994.

[11] S. McCanne, M. Vetterli, and V. Jacobson. Low-complexity video coding for receiver-driven layered multicast. *IEEE Journal on Selected Areas In Communications*, 15(6):983–1001, August 1997.

[12] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of the Conference on Computer Communications, IEEE Infocom'98*, pages 996–1003, San Francisco, USA, March 1998.

[13] D. Sisalem and A. Wolisz. Mlda: A tcp-friendly congestion control framework for heterogeneous multicast environments. In *Proceedings of International Workshop on Quality of Service, IWQoS'00*, Pittsburgh, PA,USA, June 2000.

[14] X. Hénocq, F. Le Léannec, and C. Guillemot. Joint source and channel rate control in multicast layered video transmission. In *Proceedings of SPIE International Conference on Visual Communication and Image Processing, VCIP'2000*, pages 296–307, June 2000.

[15] A. Legout and E. W. Biersack. Pathological behaviors for RLM and RLC. In *Proceedings of International Conference on Network and Operating System Support for Digital Audio and Video, NOSSDAV'00*, pages 164–172, Chapel Hill, North Carolina, USA, 2000.

[16] A. Legout and E. W. Biersack. PLM: Fast convergence for cumulative layered multicast transmission schemes. In *Proceedings of ACM SIGMETRICS'00*, pages 13–22, Santa Clara, CA, USA, 2000.

[17] S. Bhattacharyya, D. Towsley, and J. Kurose. The loss path multiplicity problem in multicast congestion control. In *Proceedings of the Conference on Computer Communications, IEEE Infocom'00*, New York, NY,USA, March 1999.

[18] J. Widmer and M. Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of Conference of the Special Interest Group on data COMMunnication, ACM SIGCOMM'01*, San Diego, USA, August 2001.

[19] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of Conference of the Special Interest Group on data COMMunnication, ACM SIGCOMM'00*, pages 43–56, Stockholm, Sweden, August 2000.

[20] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shave. FLID-DL: Congestion control for layered multicast. In *Proceedings of the Second International Workshop on Networked Group Communication, NGC'00*, pages 71–81, Stanford, CA, USA, November 2000.

[21] M Luby and V. Goyal. Wave and equation based rate control building block. *IETF Internet Draft draft-ietf-rmt-bb-webrc-01*, June 2002.

[22] Q. Guo, Q. Zhang, W. Zhu, and Y.-Q. Zhang. A sender-adaptive and receiver-driven layered multicast scheme for video over internet. In *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'01*, Sydney, Australia, May 2001.

[23] K. Salamatian and T. Turletti. Classification of receivers in large multicast groups using distributed clustering. In *Proceedings of Packet Video Workshop, PV'01*, Taejon, Korea, May 2001.

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling tcp thoughput: a simple model and its empirical validation. In *Proceedings of Conference of the Special Interest Group on data COMMunnication, ACM SIGCOMM'98*, University of British Columbia, Vancouver, Canada, August 1998.

[25] J. Viéron and C. Guillemot. Real-time constrained tcp-compatible rate control for video over the internet. *To be published in IEEE Transactions On Multimedia*, 2003.

[26] M. Yajnik, J. Kurose, and D. Towsley. Packet loss correlation in the mbone multicast network. In *Proceedings IEEE Global Internet Conference*, London,UK, November 1996.

[27] B. N. Levine, S. Paul, and J. J. Garcia-Luna-Aceves. Organizing multicast receivers deterministically by packet-loss correlation. In *Proceedings of the sixth ACM international conference on Multimedia*, Bristol, UK, 1998.

[28] S. Paul, K.K. Sabnani, J.C. Lin, and S. Bhattacharyya. Reliable multicast transport protocol (rmtp). *IEEE Journal On Selected Areas in Communications*, 15(3):407–421, April 1997.

[29] R. El-Marakby and Hutchison D. Scalability improvement of the real-time control protocol (rtcp) leading to management facilities in the internet. In *Proceedings of the third IEEE Symposium on Computers and Communications, ISCC'98*, pages 125–129, Athens, Greece, June 1998.

[30] K.L. Calvert, J. Griffioen, B. Mullins, A. Sehgal, and S. Wen. Concast: Design and implementation of an active network service. *IEEE Journal on Selected Area in Communications (JSAC)*, 19(3):720–733, March 2001.

[31] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantiser design. *IEEE Transactions on Communications*, COM-28:84–95, January 1980.

[32] F.J. Mac Williams and N.J.A. Sloane. *The Theory of Error Correcting Codes*. Amsterdam, 1977.

[33] D. Koo. *Elements of optimization*. Springer-Verlag, 1977.

[34] D. Tan and A Zakhor. Video multicast using layered FEC and scalable compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3), March 2001.

# C. ARTICLE FHCF

Cette annexe technique contient un article qui va être publié dans le journal ACM/Kluwer MONET. Il décrit l'algorithme FHCF présenté dans le chapitre 4.

# FHCF: A Simple and Efficient Scheduling Scheme for IEEE 802.11e Wireless LAN

Pierre Ansel, Qiang Ni*and Thierry Turletti
Planète Project, INRIA Sophia Antipolis, FRANCE

**Abstract**

The IEEE 802.11e medium access control (MAC) layer protocol is an emerging standard to support quality of service (QoS) in 802.11 wireless networks. Some recent works show that the 802.11e hybrid coordination function (HCF) can improve significantly the QoS support in 802.11 networks. A simple HCF referenced scheduler has been proposed in the 802.11e which takes into account the QoS requirements of flows and allocates time to stations on the basis of the mean sending rate. As we show in this paper, this HCF referenced scheduling algorithm is only efficient and works well for flows with strict constant bit rate (CBR) characteristics. However, a lot of real-time applications, such as videoconferencing, have some variations in their packet sizes, sending rates or even have variable bit rate (VBR) characteristics. In this paper we propose FHCF, a simple and efficient scheduling algorithm for 802.11e that aims to be fair for both CBR and VBR flows. FHCF uses queue length estimations to tune its time allocation to mobile stations. We present analytical model evaluations and a set of simulations results, and provide performance comparisons with the 802.11e HCF referenced scheduler. Our performance study indicates that FHCF provides good fairness while supporting bandwidth and delay requirements for a large range of network loads.

**Keywords:** IEEE 802.11e, WLAN, medium access control (MAC), quality of service (QoS)

## 1 Introduction

IEEE 802.11 wireless LAN (WLAN) [1] has gained a great success for data applications in hotspots, enterprizes, university campuses, hospitals, etc. To share the wireless medium, the 802.11 standard defines two access methods at medium access control (MAC) layer: the mandatory contention-based distributed coordination function (DCF) and the optional point coordination function (PCF).

The explosive growth of multimedia applications in the recent years arose the requirement of Quality of Service (QoS) support such as guaranteed delay, jitter and bandwidth for these applications. However, the original IEEE 802.11 WLAN standard has been mainly designed for data applications and does not provide any QoS support for multimedia applications [2]. To enhance the QoS support of 802.11 WLAN, the IEEE 802.11 standard committee is working on a new standard, called 802.11e [3]. A new medium access method called Hybrid Coordination Function (HCF) has been proposed in the 802.11e draft, which combines a contention-based enhanced DCF access mechanism (called EDCA) and a controlled channel access mechanism (called HCCA) in a single function. Recent performance evaluations of 802.11e HCF [4] show that HCF is more flexible than DCF and PCF and that it can improve the QoS support in 802.11 WLAN. In order to meet the negotiated QoS requirements, the QoS-enhanced AP (QAP) needs to schedule efficiently downlink and uplink frame transmissions. As wireless channel is time-varying and since a lot of multimedia applications have variable bit rate (VBR) characteristics, designing a good HCF scheduling algorithm is a challenging topic. To the best of our knowledge, research issues of 802.11e HCF scheduling algorithm has not yet received much attention. Only a few papers [5, 6] have addressed the problem of 802.11e HCF scheduling algorithm.

*Qiang Ni, the corresponding author. He is now with the Hamilton Institute, National University of Ireland Maynooth (NUIM), Co. Kildare, Ireland. Tel: +353-17086463, Fax: +353-17086269, E-mail: Qiang.Ni@ieee.org.

In order to understand the impact of 802.11e HCF scheduling algorithm on the delay performance, we first derive a mathematical model in this paper, which shows the relationship between polling interval, queue length, and delays. Based on this analytical model, we propose a simple and efficient scheduling algorithm, FHCF, that aims to be fair for different kinds of multimedia flows and compatible with current IEEE 802.11e standard. The performance of the FHCF scheme is evaluated through computer simulations and compared with the performance of the IEEE 802.11e HCF scheme.

The rest of this paper is organized as follows: Section 2 introduces the basic 802.11e HCF scheduling algorithm. Section 3 establishes a mathematical relationship between delay, polling interval and queue length in the context of 802.11e HCF scheduling scheme. This relationship is the basis of the FHCF scheme whose principles are explained in Section 4. Section 5 details the FHCF implementation in the NS-2 simulator and Section 6 compares performance of the FHCF scheduling scheme with the standard HCF scheme. Finally, Section 7 concludes the paper.

## 2 The 802.11e HCF scheduling algorithm

A simple HCF scheduling algorithm is proposed as a reference design [3] to take into account QoS requirements of different types of traffic. We provide in this section a brief introduction of the 802.11e HCF referenced scheduler, for details about the 802.11e standard and its QoS enhancement mechanisms please refer to our survey paper [2]. In 802.11e HCF, each QoS-enhanced station (QSTA) that requires a strict QoS support is allowed to send QoS requirement packets to the QAP while QAP can allocate the corresponding channel time for different QSTAs according to the requests. Figure 1 shows an example of the new IEEE 802.11e beacon interval, which is composed of alternated modes of contention period (CP) and optional contention-free period (CFP). Contrary to the 802.11 PCF scheme, the 802.11e HCF scheme can operate during both CP and CFP. During the CP, the QAP can start several contention-free bursts, called Controlled Access Periods (CAPs) at any time to control the channel. An important new feature is the concept of transmission opportunity (TXOP), which refers to an instance during which a given QSTA has the right to send packets. Thus a QSTA can initiate multiple transmissions as long as its TXOP has not expired. The aim of introducing TXOP is to limit the time interval during which a QSTA is allowed to transmit frames. Each QSTA can have up to 8 different priority traffic streams (TSs). Basically, each TS first sends a QoS request frame to the QAP containing the mean data rate of the corresponding application, the MAC Service Data Unit (MSDU) size and the maximum required service interval (RSI). Using these QoS requests, the QAP determines first the minimum value of all the RSIs required by the different traffic which apply for HCF scheduling. Then it chooses the highest submultiple value of the 802.11e beacon interval duration (duration between two beacons) as the selected service interval (SI), which is less than the minimum of all the maximum RSIs. Thus, an 802.11e beacon interval is cut into several SIs and QSTAs are polled accordingly during each selected SI. The selected SI refers to the time between the start of successive TXOPs allocated to a QSTA, which is the same for all the QSTAs.

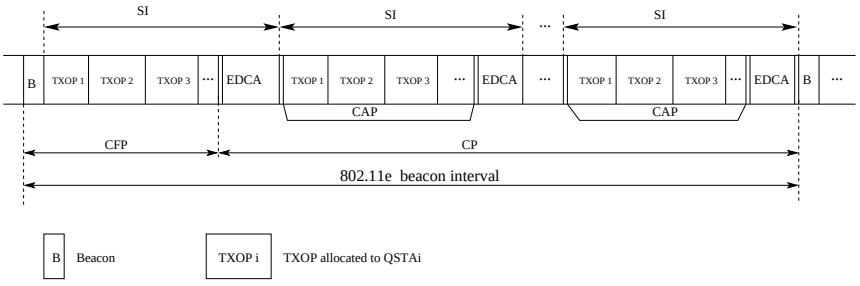



Figure 1: Structure of the 802.11e beacon interval used in the HCF scheduling algorithm [3]

As soon as the selected SI is determined, the QAP evaluates all the TXOPs allocated to the different TSs of the QSTAs which apply for HCCA. The TXOP should correspond to the duration required to transmit all the packets arriving during a SI in a TS queue. Let $N_i$ be the number of packets arriving in the TS queue $i$ for a QSTA during a SI:

$$N_i = \lceil \frac{\rho_i SI}{M_i} \rceil, \quad (1)$$

where $\rho_i$ is the application data rate and $M_i$ the MSDU size for TS queue $i$. Then the different TXOPs $T_i$ are computed as follows:

$$T_i = N_i \cdot (\frac{M_i}{R} + 2SIFS + ACK), \quad (2)$$

where $R$ is the physical (PHY) layer transmission rate, and $ACK$ is the duration to transmit an acknowledgement packet. This simple HCF scheduling algorithm can be efficient if traffic is strictly CBR. However, when real-time applications such as videoconferencing generate VBR traffic, this scheme may cause the average queue length to increase and possibly packets to be dropped. Even if the mean sending rate of the application is lower than the rate specified in its QoS requirements; peaks of sending rate may not be absorbed by TXOPs allocated according to the QoS requirements. A more flexible scheme that adapts to fluctuating rates is then necessary. This motivated the design of our FHCF scheduling scheme, which is based on the mathematical relationship between parameters and corresponding observations that we obtain in the following section.

## 3 Relationship between queue length, polling interval, and delays

Let us consider several simplifications to establish an analytical relationship between SI and delay parameters: First, we denote by $R_{eff}$ the effective data throughput corresponding to the actual amount of data packets transmitted per time unit. Note that the PHY layer of 802.11 wireless networks can adapt its transmission mode according to varying conditions of the PHY channel. Variation of PHY layer data rates can be achieved by choosing different modulation and coding schemes. In this work, we do not consider such rate adaptation options in 802.11 [1]. $R_{eff}$ is computed according to the maximum PHY data rate but is less than the maximum PHY data rate because of PHY and MAC layers' overheads. Second, we suppose that data packets are transmitted continuously (without taking packet fragmentation into account). For the delay analysis, we consider the following two cases: 1) The queue is empty at the end of TXOP allocation. This is the ideal case assumption for the IEEE 802.11e HCF scheme, which holds only for CBR-like traffic types; 2) The queue is not empty at the end of TXOP allocation, which is more realistic than the ideal case in real wireless networks.

### 3.1 Empty queues at the end of the TXOP

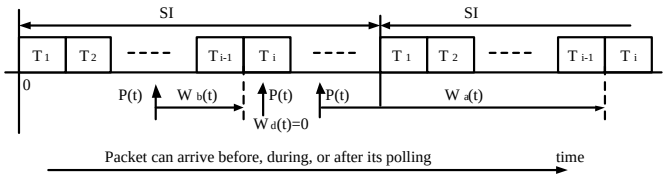



Figure 2: Timing relationships for delay analysis in ideal case (queueing delays not shown)

In this case, the queue of a polled QSTA is assumed to be zero at the end of its TXOP allocation, which is true if traffic is a CBR one. Under such an assumption, all the packets arriving in a SI interval will be serviced either in the same SI or in the next SI, and no later than the next SI. As shown in Figure 2, in

order to predict the delay for a data packet $P$ that arrives in the TS queue with polling order $i$ at time $t$ [1], two different cases should be considered: the packet $P$ can be transmitted in the following TXOP allocation ($T_i$) of that TS queue, or the packet has to be queued until another TXOP allocation in the next SI if there are some other packets before $P$ in the $TS-i$ queue. In this paper, all 802.11e schedulers do not actively drop a data packet if it is too old in the queue (i.e. the packet delay is larger than the delay bound of that traffic application). Hence, the delay for the data packet can be calculated as the sum of the packet queueing delays ($Q$) for sending out all the queued packets before $P$ in this SI interval (via the allocated TXOP $T_i$), and the waiting time delays ($W$) which is equal to the duration between the packet arrival time and the time the queue is polled. Notice that in current 802.11e MAC scheduling framework, each TS queue can only be polled once each SI interval. Those packets arriving after the polling may not be transmitted in this SI interval and thus has to wait for the next polling interval. Considering that the delay expressions are different if data packets arrive *before*, *during*, or *after* the QSTA is polled by the QAP. We denote respectively by $Q_b(t)$, $Q_d(t)$, and $Q_a(t)$ the packet queueing delays if the packet $P$ arrives before, during and after its polling at time $t$. Similarly, we denote respectively by $W_b$, $W_d$, and $W_a$ the packet waiting time delays while the packet arrives before, during and after its polling at time $t$. Finally, the delay for the packet $P$ can be expressed as:

$$\begin{aligned} d_i(t) &= (W_b(t)+Q_b(t))\chi_{[0,\sum_{j=1}^{i-1}T_j]}(t) \\ &\quad +(W_d(t)+Q_d(t))\chi_{(\sum_{j=1}^{i-1}T_j,\sum_{j=1}^{i}T_j]}(t) \\ &\quad +(W_a(t)+Q_a(t))\chi_{(\sum_{j=1}^{i}T_j,SI]}(t) \end{aligned} \tag{3}$$

with:

$$W_b(t)=\sum_{j=1}^{i-1}T_j-t\ , \qquad Q_b(t)=\frac{q_i^0M_i}{R_{eff}}+\frac{\rho_i t}{R_{eff}} \tag{4}$$

$$W_d(t)=0\ , \qquad Q_d(t)=\frac{q_i^0M_i}{R_{eff}}+\frac{\rho_i t}{R_{eff}}-(t-\sum_{j=1}^{i-1}T_j) \tag{5}$$

$$W_a(t)=SI-t+\sum_{j=1}^{i-1}T_j\ , \qquad Q_a(t)=\frac{\rho_i}{R_{eff}}(t-\sum_{j=1}^{i}T_j) \tag{6}$$

where $q_i^0$ represents the initial number of data packets in the $TS-i$ queue at the beginning of the SI, and we denote

$$\chi_{[i,j]}(t)=\begin{cases} 1 \text{ if } t\in[i,j] \\ 0 \text{ else} \end{cases}\ ,$$

$$\chi_{(i,j]}(t)=\begin{cases} 1 \text{ if } t\in(i,j] \\ 0 \text{ else} \end{cases}\ .$$

By using Equations (4)-(6), Equation (3) can be rewritten as:

$$\begin{aligned} d_i(t) &= (\sum_{j=1}^{i-1}T_j-t+\frac{q_i^0M_i}{R_{eff}}+\frac{\rho_i t}{R_{eff}})\chi_{[0,\sum_{j=1}^{i}T_j]}(t) \\ &\quad +(SI-t+\sum_{j=1}^{i-1}T_j+\frac{\rho_i}{R_{eff}}(t-\sum_{j=1}^{i}T_j))\chi_{(\sum_{j=1}^{i}T_j,SI]}(t). \end{aligned}$$

Note that delay is discontinuous at the end of the TXOP since those data packets arriving after the end of the TXOP have to wait for the next TXOP in the following SI.

[1] The initial time (time zero) is set at the beginning of the SI.

### 3.2 Non-empty queues at the end of the TXOP

When traffic is a VBR one, the queue at the end of the TXOP may not be empty as assumed in Section 3.1. In this case, we have to consider $q_i^e$, the non-zero queue length of the TS at the end of the TXOP $T_i$.

Further, different from the case in Section 3.1, packets can be queued later than the next SI interval since all the queues can be nonzero and delays are accumulative. Hence, delay can be discontinuous even during the TXOP duration in this case and it can be expressed as:

$$d_i(t) = (kSI + \sum_{j=1}^{i-1} T_j - t + \frac{q_i^0 M_i}{R_{eff}} + \frac{\rho_i t}{R_{eff}})\chi_{[0,t^d]}(t) + ((k+1)SI - t + \sum_{j=1}^{i-1} T_j + \frac{\rho_i}{R_{eff}}(t - t^d))\chi_{(t^d,SI]}(t),$$

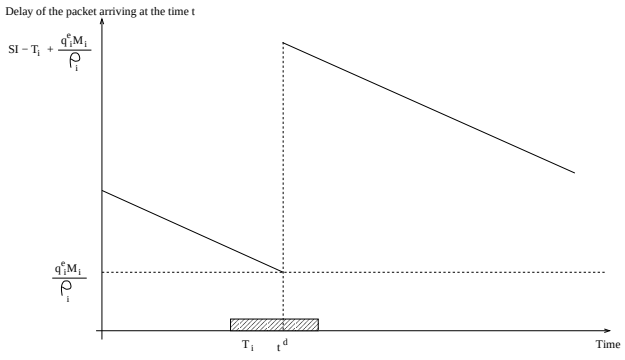



Figure 3: Theoretical delay versus time for the best case

where the value of $k$ depends on *which SI interval* the data packet can be serviced. In the *best case* if the packet can be transmitted no later than the next SI interval, $k = 0$, see Figure 3. In other cases, we have $k > 0$. $t^d$ denotes the time when delay discontinuity happens, and it corresponds to the arriving time of the first packet which can not be transmitted during the current TXOP duration:

$$t^d = \sum_{j=1}^{i} T_j - \frac{q_i^e M_i}{\rho_i}.$$

Suppose that the best case is satisfied ($k = 0$), we can obtain the maximum delay, which is denoted by $D_i$ and is achieved at the time $t^d$ as follows:

$$D_i = \max_t d_i(t) = SI - T_i + \frac{q_i^e M_i}{\rho_i}. \quad (7)$$

Equation (7) shows that the delay is higher than the ideal one in Section 3.1 if some packets remain in the queue at the end of the poll. In Section 3.1, queue is assumed empty at the end of the poll, and thus $D_i$ is almost equal to the SI duration since $T_i << SI$ which means packet delays are bounded by the SI in the ideal case. However, in other cases than ideal case and best case, the 802.11e HCF scheduler will not work well and packet delays will completely uncontrolled if traffic is highly variable. We will validate this observation through computer simulations in Section 6.

Moreover, we can express $T_i$ in order to link the queue size at the beginning of the polling $q_i^b$ with the queue size at the end of the polling $q_i^e$. Indeed, if $\frac{q_i^b M_i}{R_{eff}}$ represents the time to send the $q_i^b$ packets in the

queue before polling, $\frac{\rho_i}{R_{eff}}T_i$ the time to send the packets arrived in the queue during the TXOP, and $\frac{q_i^e M_i}{R_{eff}}$ the time to send the packets still in the queue at the end of the TXOP, we have the following relation

$$T_i = \frac{q_i^b M_i}{R_{eff}} + \frac{\rho_i}{R_{eff}}T_i - \frac{q_i^e M_i}{R_{eff}},$$

with $q_i^b = q_i^0 + \frac{\rho_i}{M_i}\sum_{j=1}^{i-1} T_j$ the queue length just before polling. Using (7), the maximum delay in the best case can be expressed as:

$$\begin{aligned} D_i &= SI + \frac{q_i^b M_i}{\rho_i} - \frac{R_{eff}}{\rho_i}T_i \\ &= SI(1 + \sum_{j=1}^{i-1}\frac{T_j}{SI} - \frac{R_{eff}}{\rho_i}\frac{T_i}{SI}) + \frac{q_i^0 M_i}{\rho_i}. \end{aligned} \quad (8)$$

(8) shows that we can control the maximum delay $D_i$ by two different ways: On the one hand, the QAP can reduce the delay by increasing the value of $T_i$ which is allocated to the $i$-th TS , since the number of packets remaining in the queue at the end of the TXOP ($T_i$) decreases when the allocated TXOP increases. From (1) and (2), we can note that $T_i/SI$ is independent from the SI duration, so we are able to reduce the maximum delay by reducing the SI duration. However, this increases the number of polls and overheads increase at the same time. On the other hand, we can control the maximum delay if we are able to control the queue length before polling $q_i^b$, whose value can be measured by the data rate of the TS. When the flow is not a constant bit rate traffic, $q_i^b$ may vary considerably. Actually, this is the main motivation of our FHCF scheme. One advantage of our FHCF scheme is that fairness for flows with the same priority can be achieved without any additional cost because they obtain the same maximum delay which is linked to the selected SI duration.

# 4 The FHCF Scheme

Basically, the FHCF scheme is composed of two schedulers: the QAP scheduler and the node scheduler. The QAP scheduler estimates the varying queue length for each QSTA before the next SI and compares this value with the ideal queue length, see Figure 4. The QAP scheduler uses a window of previous estimation errors for each TS in each QSTA to adapt the computation of the TXOP allocated to that QSTA. Then, the node scheduler located in each QSTA can redistribute the unused time among its different TSs because the TXOP is not allocated to a particular flow but all flows of a QSTA.

## 4.1 QAP scheduler

First, the QAP scheduler has to compute the ideal queue length of the TS queue $i$ for each QSTA at the beginning of the next SI:

$$q_i^{ideal} = \frac{\rho_i \cdot (SI - \sum_{j=1}^{i} N_j \cdot (\frac{M_j}{R_{eff}} + 2SIFS + ACK))}{M_i}. \quad (9)$$

In this paper, the ideal queue length refers to the queue size at the beginning of the next SI which was zero at the end of the current TXOP. The ideal queue length evolution assumption is used by the IEEE 802.11e HCF referenced scheduling scheme [3], which is valid only when the sending rate of the application is strictly CBR.

Second, when a QSTA sends a QoS data packet, the QAP uses the QoS control field of the IEEE 802.11e header to record its queue length $q_i^e$ at the end of TXOP. Let $t_i^e$ be the corresponding time at the end of the current TXOP. Note that $t_i^e$ is also recorded by the QAP scheduler. Using these information, the QAP scheduler is able to estimate $q_i^{est}$, the queue length of the $i$-th TS at the beginning of the next SI as follows:

$$q_i^{est} = \frac{\rho_i(SI - t_i^e)}{M_i} + q_i^e. \quad (10)$$

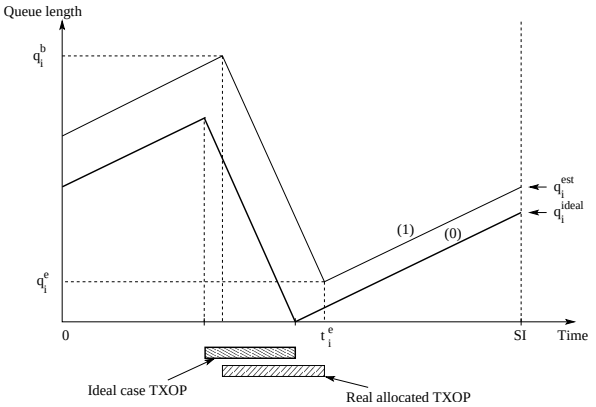



Figure 4: Queue length evolution for a TS: (0). Ideal queue length case; (1). Estimated queue length evolution

Since the sending rate and packet size of the application vary, the above simple method for queue length estimation is not always accurate. To solve this problem, the FHCF scheme proposes to use a window of $w$ already known real queue length measurements (or called history information) to tune the estimation. It means at the $n$-th SI, the QAP will have at its disposal the $w$ last estimation errors

$$\begin{aligned}
\Delta_i^{n-1} &= q_i^{b,real}(n-1) - q_i^{b,est}(n-1) \\
\Delta_i^{n-2} &= q_i^{b,real}(n-2) - q_i^{b,est}(n-2) \\
&\ldots \\
\Delta_i^{n-w} &= q_i^{b,real}(n-w) - q_i^{b,est}(n-w).
\end{aligned}$$

Then, just before the $n$-th SI, the QAP estimates $E[|\Delta_i(n)|]$, the expected value of the absolute value of the difference between $q_i^{b,real}(n)$ (the real queue length at the beginning of TXOP of the $i$-th TS) and the estimation of this queue length, $q_i^{b,est}(n)$ as follows:

$$E[|\Delta_i(n)|] \simeq \frac{\sum_{j=n-w}^{n-1} |\Delta_i(n)|}{w}. \tag{11}$$

In the Appendix, we show that this corrective term is close to the standard deviation of a traffic flow if we suppose that the sending rate of that flow follows a Gaussian distribution. Moreover, from a hardware point of view, it is easier to compute the absolute value of the difference than a estimation of the standard deviation.

Thus, by using Equation (11), the QAP scheduler is able to improve its estimation of the queue length at the beginning of the next polling of the TS queue as follows

$$q_{i,new}^{b,est}(n) = q_i^{b,est}(n) + E[|\Delta_i(n)|].$$

Third, the QAP compares the real queue length estimation to the ideal queue length[2] at the beginning of the next SI. It computes the number of additional packets, $DN_i^{est}$, which is the difference between the

[2]The ideal queue length was zero at the end of the current TXOP allocation.

estimated queue length and the ideal case:

$$DN_i^{est} = q_{i,new}^{b,est}(n) - q_i^{b,ideal}(n) = q_i^{est}(n) - q_i^{ideal}(n) + E[|\Delta_i(n)|]$$

where $q_i^{est}(n)$ and $q_i^{ideal}(n)$ are given by (10) and (9). Then, the QAP computes the additional required time $t_i^{est}$ (which may be positive or negative) for each TS of each QSTA and reallocates the corresponding TXOP duration according to the estimation of $DN_i^{est}$:

$$t_i^{est} = DN_i^{est} \cdot (\frac{M_i}{R_{eff}} + 2SIFS + ACK). \tag{12}$$

Then it computes $T_P$, the sum of all the positive values, $T_N$, the absolute value of the sum of all the negative values, and $T^r$, the remaining time of HCCA duration after allocating all the TXOPs computed in one SI using the ideal case. If $T_P - T_N > T^r$, it means that the scheduler is not able to allocate all the time it expected according to the estimations, and that the additional time $t_i^{est}$ have to be reduced. In order to be fair for all the flows, the scheduler reduces each positive additional time with a percentage $\beta$ (chosen negative to correspond to a reduction) of $t_i^{est}$. On the other hand, each negative additional time is increased by the same percentage $\beta$ of $t_i^{est}$, where $\beta$ is expressed as:

$$\beta = -\frac{(T_P - T_N) - T^r}{T_P + T_N}. \tag{13}$$

Finally, the *effective additional time* $t_i^{add}$ allocated to the $TS - i$ queue is equal to:

$$t_i^{add} = \begin{cases} (1+\beta)t_i^{est} & \text{if} \quad t_i^{est} \geq 0 \\ (1-\beta)t_i^{est} & \text{if} \quad t_i^{est} < 0. \end{cases} \tag{14}$$

When it is time for the QAP to poll a QSTA, the QAP scheduler computes the sum of all the normal TXOPs and the effective additional time allocated to the different TSs in a QSTA.

Note that our estimator in the *second step* is quite generic and works well for most kinds of traffic types. The key idea of our estimator and corresponding FHCF scheduler is, in order to adapt to the fluctuations of sending rates of different QSTAs, the QAP scheduler tries to allocate the service rate for different QSTAs which is equal to the mean arrival rate plus the expected value of the absolute deviation. More precisely speaking, we tune the TXOP allocation according to the mean sending rate plus this deviation, and then traffic spikes can be absorbed. From Chebyshev's Inequality theorem [13], we know that this estimator represents the average difference from the mean value, and is a measure of how spread out the random variables are. Although the standard deviation estimator could be another choice, it introduces higher computational complexities, and thus we decided to choose this simple estimator, which also provides reasonable accuracy.

How to choose the value of $w$ is a tradeoff between complexity and accuracy: with a larger value of $w$, a more accurate queue length estimation is normally obtained, but with an increasing complexity. In our work, the window size $w$ has been empirically set to 5 through simulation results.

### 4.2 Node scheduler

The node scheduler also plays a very important role since it has to redistribute the additional allocated time to the different TSs within a node. It performs almost the same computations as the QAP. Suppose that the number of active TS of a given polled QSTA is $p$ ($1 \leq p \leq 8$). First the node scheduler in this QSTA computes $N_i$, the number of packets to transmit in the $i$-th TS, and the time required to transmit a packet according to its QoS requirements (packet size, data rate). Second, according to its allocated TXOP $T$ and the number of packets the QSTA should transmit from each TS, it evaluates the remaining time $T^r$ that can be reallocated:

$$T^r = T - \sum_{i=1}^{p} N_i \cdot (\frac{M_i}{R_{eff}} + 2SIFS + ACK).$$

Since each QSTA knows exactly its own TS queue sizes at the beginning of the polling, it is able to estimate more precisely its queue sizes at the end of the TXOP and consequently the required additional

time per TS. Using this information, the node scheduler performs the same computations as the QAP scheduler (see (12), (13) and (14)). The difference is that the coefficient $\beta$ may be positive if the QSTA has more time than required to send all its packets or may be negative if, on the contrary, the remaining time is less than required. Thus, just after the CF-Poll reception, the QSTA can redistribute additional time to its different TSs with the option to add more time to each TS if $\beta$ is positive.

# 5 FHCF Implementation in NS-2 Simulator

## 5.1 Basis of the FHCF scheme

Our NS implementation of 802.11e HCF/FHCF/EDCA [9] is based on Stanford University's early version of NS-2 codes of EDCA/HCF [6]. We have added numerous new features according to the latest IEEE 802.11e standard draft [3]. In Stanford's original NS implementation, 8 kinds of different traffic classes (TC) of EDCA can be both used in HCCA and EDCA mode. However, according to the new 802.11e standard [3], TC queues for EDCA and TS queues for HCCA should be separated in order to prevent packets from EDCA queues to be transmitted through TS queues of HCCA and thus guarantee the delay performance in HCF controlled channel access. In our implementation, if a traffic with priority $i$ is accepted in HCCA, it becomes a Traffic Stream (TS) and all the packets for this class can only be transmitted in HCCA mode. In this way, it is possible to evaluate the performance of the different schedulers only using the HCCA mode. We also implemented the part of queue length estimation in HCCA since Stanford's original implementation is based on the use of old queue length information without any estimations.

Moreover, in their implementation beacon frames could be delayed since the medium was sometimes used at the time the beacon frame was scheduled. We have fixed the problem of beacon frame delay in the simulator by adding a beacon timer in every QSTA, to check before each transmission if there is enough time to transmit packets without disturbing the beacon sending.

## 5.2 TSPEC negotiation

As mentioned before, the purpose of the FHCF scheme is to improve the HCF scheduling algorithm by adapting it to fluctuating flows and by providing fairness. In this optic, we only consider HCCA traffic in the remainder of the paper. If a QSTA wants to send a certain flow, it will only use the HCCA mode, not the EDCA mode.

According to the 802.11e standard [3], a QSTA has to send a QoS request frame to the QAP whenever it wants to transmit packets of a certain TS in HCCA. In our implementation, QSTAs first start to transmit packets in EDCA in order that the QAP records source and priority of traffics. Thus, the first packet sent from a certain TC plays the role of a QoS request frame and the QAP considers this packet as a QoS request frame. If this flow is accepted, the QAP schedules it. Then, the QSTA is informed at the reception of the next CF-Poll which contains both the TXOP duration and an 8-bit integer indicating the accepted traffics. If the $j^{th}$ bit is one, it means that the $TC - j$ is taken into account by the scheduler to plan packets transmission during the HCCA mode.

As regards QoS parameters of the TSPEC negotiation, the nominal MSDU size, the mean sending rate and the maximum acceptable SI are determined statically at the beginning of the simulation for each among the 8 TCs of different priorities. This does not change the spirit of the scheduler which can also be adapted to a real TSPEC negotiation with different QoS requirements for TSs of the same priority.

# 6 Simulation Experiments

In order to evaluate the performance of the QAP scheduler and the node scheduler, two kinds of simulation topologies are used. The first one contains 18 mobile QSTAs and 1 QAP with only one TS per QSTA (see Section 6.1), which is designed to evaluate the performance of the QAP scheduler. The second topology is composed of 6 QSTAs and 1 QAP (see Section 6.2), each one with three different priority TSs to evaluate the performance of the node scheduler. For all the simulations, the destination of all the flows is the QAP (which is node 0 in our case): This allows us to compare fairly end-to-end delays among the different flows.

### 6.1 Scenario 1

In the first Scenario, six QSTAs send a high priority on-off audio traffic ($64kb/s$), six others QSTAs send a VBR video traffic ($200kb/s$ of average sending rate) with medium priority and the remaining six QSTAs send a CBR MPEG4 [10] video traffic ($3.2Mb/s$) with low priority. Table 1 summarizes the different traffics used for this simulation. We model the audio flow by on-off sources with parameters corresponding to a typical phone conversation [11]. The transport protocol is UDP. Audio flows are mapped to the $6^{th}$ TS of the MAC layer whereas VBR H.261 and CBR MPEG4 video flows are respectively mapped to the $5^{th}$ and $4^{th}$ TS. The different VBR flows have been obtained with the VIC [7] videoconferencing tool using the H.261 coding and QCIF format for typical "head and shoulder" video sequences. We made 6 trace files[3]: the mean sending rate was close to $200kb/s$ with a mean packet size of $660bytes$ and a mean interarrival time of $26ms$. A simple analysis of the trace files shows that the sending rate distribution follows a Gaussian law and its mean value belongs to a window of $80kb/s$ around the mean value of $200kb/s$, and the mean packet size between 600 and $700bytes$. Packet sizes of these flows belong to a large range of values between 20 and $1024bytes$[4]. The RSI request of audio traffic is set to $50ms$ and the RSIs of both VBR and CBR video traffic types are set to $100ms$ and beacon interval is $500ms$. According to the algorithm in 802.11e, the selected SI will be $50ms$. The PHY and MAC layer parameters used in the simulation are summarized in Table 2.

Table 1: Description of the different traffic streams

| Node | Application | Arrival period ($ms$) | Packet size ($bytes$) | Sending rate ($kb/s$) |
|---|---|---|---|---|
| $1 \rightarrow 6$ | Audio | 4.7 | 160 | 64 |
| $7 \rightarrow 12$ | VBR video | $\simeq 26$ | $\simeq 660$ | $\simeq 200$ |
| $13 \rightarrow 18$ | MPEG4 video | 2 | 800 | 3200 |

Table 2: PHY and MAC layer parameters

| | |
|---|---|
| SIFS | $16\mu s$ |
| DIFS | $34\mu s$ |
| ACK size | $14bytes$ |
| PHY rate | $36Mb/s$ |
| Minimum PHY rate | $6Mb/s$ |
| Slot time | $9\mu s$ |
| CCA time | $4\mu s$ |
| MAC header | $38bytes$ |
| PLCP header length | $4bits$ |
| Preamble length | $20bits$ |

#### 6.1.1 Comparison of throughput

Throughput curves on Figure 5 show that both FHCF and the standard HCF schemes succeed in providing the required throughputs for all the flows. For VBR H.261 flows, throughput is more fluctuating with FHCF than with standard HCF scheme since FHCF is able to adapt all the allocated TXOPs according to the queue length evolutions. For CBR MPEG4 flows, both FHCF and HCF provide a constant throughput. As regards audio flows, the throughput is sometimes equal to zero since it matches burst periods and idle periods.

[3]The video trace files are available from [9].
[4]By default, VIC uses a MTU size of $1024bytes$.

### 6.1.2 Comparison of the number of dropped packets

Table 3 and Figure 6 show that with the standard HCF scheme, no CBR MPEG4 packets are lost[5]. However, 513 VBR packets are lost with the HCF scheme. Please notice that in this work, packets are considered as lost only when reaching the maximum queue length limit and hence dropped by the queue buffers. They are never dropped because the delays are larger than the applications' delay bounds. The queue overflow can occur in the following two cases: First, the peak sending rates of some VBR flows may be much higher than the mean sending rate specified in their requirements, and second, the sending rate is fluctuating heavily during time. This confirms, as explained in Section 2, that HCF has not been designed for this kind of flows but rather for CBR traffic type. On the contrary, FHCF succeeds in having no dropped packets since the TXOPs allocated to the different QSTAs are adapted to the real queue lengths on the basis of their nominal QoS requirements.

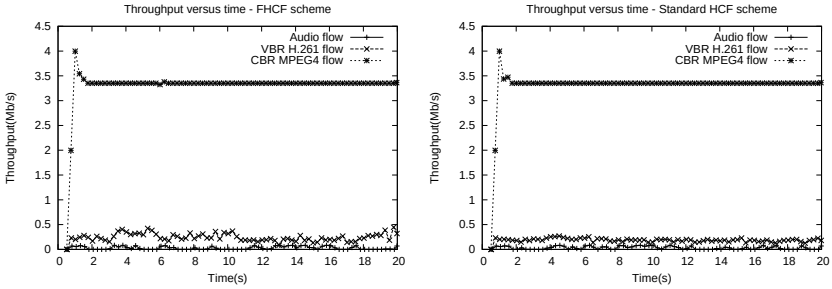



Figure 5: Throughput versus time for FHCF and HCF

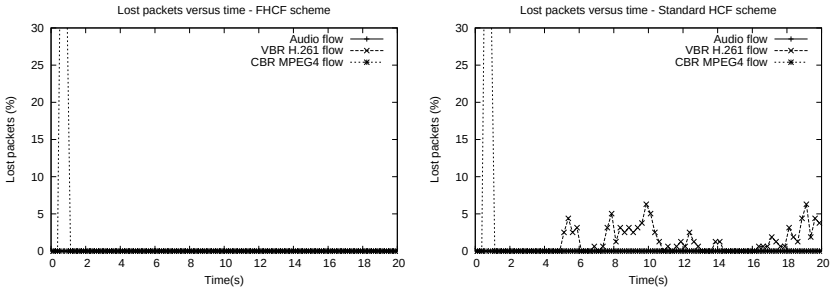



Figure 6: Dropped packets versus time for FHCF and HCF

Table 3: Number of drops for each kind of flow with FHCF and HCF

| | Audio | VBR video | CBR MPEG4 video |
|---|---|---|---|
| FHCF | 0 | 0 | 204 |
| Standard HCF | 0 | 513 | 204 |

[5]Some packet drops occur only at the beginning of the simulation because all the flows start at time 0 and the network is congested at the starting time. However, it does not depend on the scheduling/queueing policy and thus we did not consider them as losses.

### 6.1.3 Analysis of latency distributions

Figure 7 shows that with the FHCF scheme, all the flows have a maximum latency which are equal to the selected SI of the flows (chosen equal to $50ms$), whereas with the HCF scheme, the packet delays of VBR flows are completely uncontrolled. It is because the allocated TXOPs according to the mean rate are too small and thus the delays are very high. We observe on the same figure that the latency distribution curve of the VBR flow has a stair shape. If we analyze the trace file of the VBR flow represented on the different curves, we note that the interarrival time of packets is not $26ms$ (see Table 1) but precisely $33ms$ since sometimes two packets arrive at the same time[6] and the interarrival time is then higher than the mean value. Because packets arrive regularly and the interarrival time is not changing during simulation time, delays of packets are not regularly distributed between 0 and $50ms$.

Figure 8 represents the mean latency for each QSTA (sending only one traffic). We observe that for the FHCF scheme, all the flows have a mean latency almost equal to $25ms$. This means (see Figure 3) that the queues at the end of the TXOPs allocated to each QSTA are almost always empty ($q_i^e \simeq 0$). Effectively, the mean latency on a SI, which can be expressed as:

$$\frac{1}{2}(SI - T_i + 2\frac{q_i^e M_i}{R_{eff}}), \tag{15}$$

is equal to $25ms$ ($T_i << SI$).

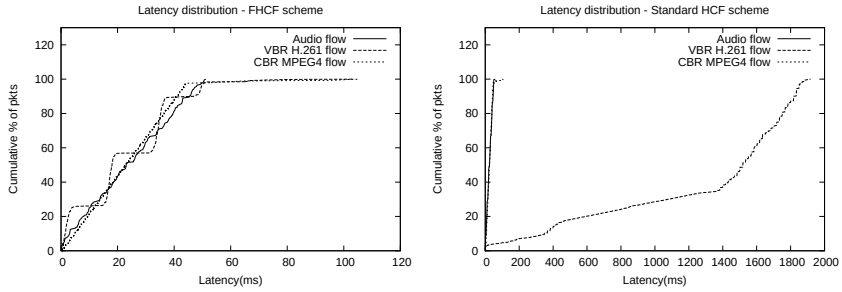



Figure 7: Latency distribution for FHCF and HCF

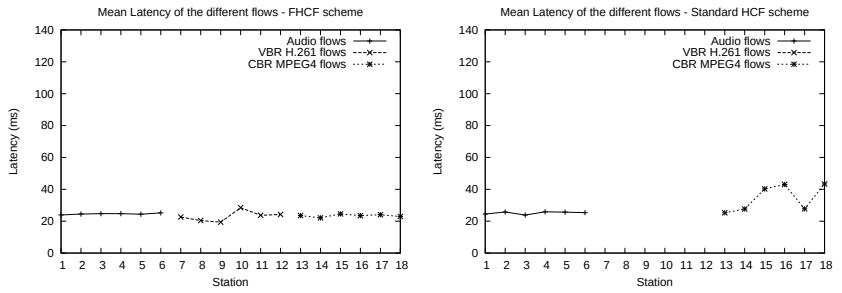



Figure 8: Mean latency for each flow with FHCF and HCF

[6]This difference is due to VIC fragmentation.

As regards the standard HCF scheme, the delays of the VBR flow are completely uncontrolled (see Figure 7) because the queue lengths are increasing during time. In this case, TXOPs are not long enough to absorb remaining bursts and the queue sizes are endless filling up because queues with large capacities are used. Note that the standard HCF scheme may be efficient if TXOPs are allocated according to the maximum sending rate of the VBR flows but, in this case, fewer flows than with FHCF can be accepted in HCCA. In our example of VBR flows, the gain with FHCF is between 14% and 37% depending on the flow.

### 6.2 Scenario 2

In Scenario 2 (see Table 4), each QSTA sends three audio, VBR H.261 and CBR MPEG4 video flows simultaneously through three different MAC-layer priority classes. This topology aims at evaluating the behaviors of the different TSs in the same QSTA and with the same priority TS in different QSTAs. The HCCA load has been changed by increasing the packet size of the CBR MPEG 4 traffic from 600$bytes$ (2.4$Mb/s$) to 1000$bytes$ (4$Mb/s$) using a 100$bytes$ increment and keeping the same inter-arrival period of 2$ms$. This is a time-consuming way to increase load because CBR video packets need more time to be transmitted, while another way to increase the network load is to increase the number of QSTAs. We chose the first way to increase the traffic load.

To compare fairness in terms of delay between the same kinds of traffics for the different schemes, we use Jain's fairness index [12]:

$$J = \frac{(\sum_{i=1}^{n} d_i)^2}{n \sum_{i=1}^{n} d_i^2}$$

where $d_i$ is the mean delay of the flow $i$ and $n$ is the number of flows for which the fairness index is computed.

Figures 9 and 10 show respectively the mean delays and the fairness of several types of flows obtained with the various schemes for different loads of the network (see Table 4).

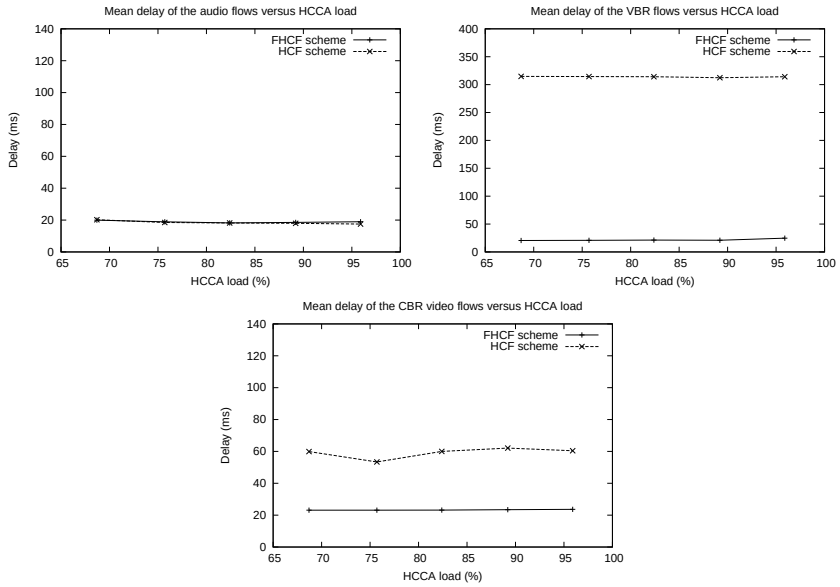



Figure 9: Mean delays versus load

Table 4: Description of different traffic streams

| Node | Application | Arrival period ($ms$) | Packet size ($bytes$) | Sending rate ($kb/s$) |
|---|---|---|---|---|
| $1 \rightarrow 6$ | Audio | 4.7 | 160 | 64 |
| $1 \rightarrow 6$ | VBR video | $\simeq 26$ | $\simeq 660$ | $\simeq 200$ |
| $1 \rightarrow 6$ | CBR video | 2 | $600 \rightarrow 1000$ | $2400 \rightarrow 4000$ |

#### 6.2.1 Audio and VBR H.261 video flows

Figure 9 shows that with FHCF, delay curves are almost horizontal lines which means that delays do not strongly depend on the network load. For the same reason as in Scenario 1, the delays of VBR flows with the standard HCF scheme are very high (the mean delays for the VBR flows are almost $300ms$).

As shown in Figure 10, Jain's fairness index between audio flows obtained with the HCF scheme and the FHCF scheme, is very high. The reason is that they both allocate TXOPs by excess to these audio flows. Concerning the VBR flows, FHCF is always fairer than HCF.

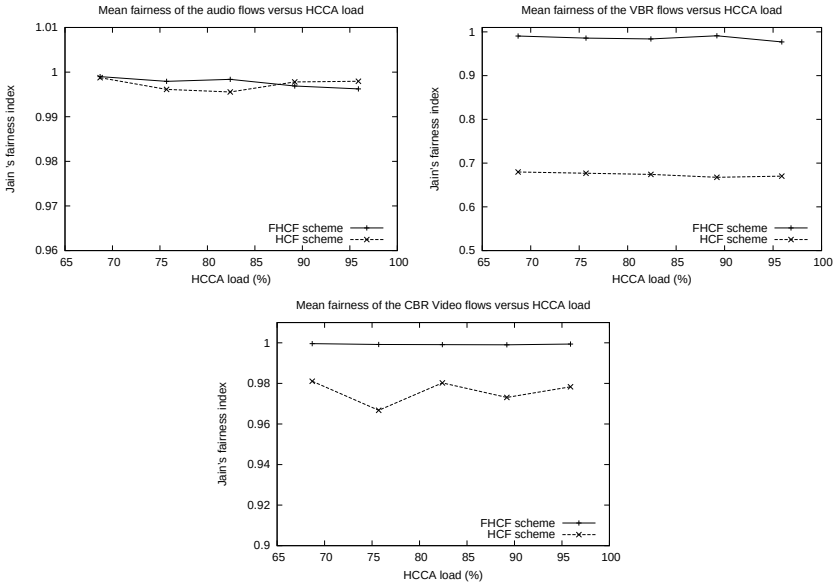



Figure 10: Fairness versus load

#### 6.2.2 CBR MPEG4 video flows

In our simulations, CBR flows are the most responsible for the network load. Figure 9 shows that the mean delays of both FHCF and standard HCF schemes are not affected by the traffic load, while the delay of FHCF is smaller than that of HCF. As seen in Figure 10, we observe that FHCF is fair between the different

CBR flows on a large range of loads since node schedulers succeed in redistributing time among the different TSs up to a very high network load (96%). On the other hand, with HCF fairness performance is poor because both schedulers are not able to absorb traffic fluctuations.

Figure 11 shows that the total throughput increases linearly both with standard HCF and FHCF schemes, while the total throughput is almost the same for the two schemes.

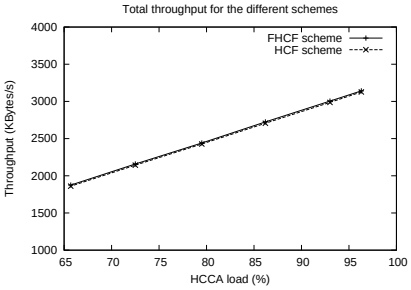



Figure 11: Total throughput

## 7 Conclusion

In this paper, we have derived an analytical model which explains the relationship between the 802.11e HCF polling interval, queue length, and delays. Based on it, we have proposed the FHCF scheduling scheme for the upcoming 802.11e MAC layer standard, with the aim of supporting fluctuating rates and/or packet sizes with QoS requirements. To allocate TXOPs, FHCF uses the mean sending rate of VBR applications plus estimated absolute deviations instead of the maximum sending rate usable for the standard HCF scheme. In this way, FHCF provides better QoS performance if a mean sending rate is used by the HCF scheduler, or it saves around 14% to 37% of the time allocated to the VBR flows if a maximum sending rate is used by the scheduler. Consequently, more traffic flows can be transmitted with good quality in HCCA. Moreover, the FHCF scheme is shown to achieve a higher degree of fairness among different multimedia flows than the 802.11e HCF scheme. Future work includes the design of adaptive and robust scheduling algorithms for error-prone IEEE 802.11e wireless channels and robust admission control mechanisms for 802.11e wireless networks.

## Acknowledgements

The authors would like to thank Dr. Chadi Barakat (INRIA), Prof. Sunghyun Choi (Seoul National University, Korea), Dr. Matthew Sherman (AT&T Shannon Laboratory, USA), the guest editors, and other anonymous reviewers for providing valuable comments to improve the quality of the paper. This work has been supported by the French ministry of industry in the context of the national project RNRT-VTHD++.

## References

[1] IEEE 802.11 WG: IEEE Std 802.11-1999, Part 11: Wireless LAN MAC and physical layer specifications. Reference number ISO/IEC 8802-11:1999 (E), (1999).

[2] Ni Q., Romdhani L., and Turletti T.: A survey of QoS enhancements for IEEE 802.11 wireless LAN. Wiley Journal of Wireless Communications and Mobile Computing (JWCMC), John Wiley & Sons Publisher, **4 (5)** (2004) 547–566.

[3] IEEE 802.11 WG: IEEE 802.11e/D4.1, Wireless MAC and physical layer specifications: MAC enhancements for QoS. (February 2003).

[4] Mangold S., Choi S., May P., et al.: IEEE 802.11e wireless LAN for Quality of Service. Proceeding of European Wireless, **1** (2002) 32–39.

[5] Grilo A., Macedo M., and Nunes M: A scheduling algorithm for QoS support in IEEE 802.11e networks. IEEE Communication Magazine, **10** (2003) 36–43.

[6] Garg P., Doshi R., Greene R., et al.: Using IEEE 802.11e MAC for QoS over wireless. IEEE IPCCC, (2003).

[7] McCanne S., Jacobson V.: VIC: a flexible framework for packet video. ACM Multimedia, (1995).

[8] ITU-T Recommendation H.261: Video codec for audiovisual services at $p \times 64$ kb/s. (1993).

[9] Ansel P., Ni Q., and Turletti T.: FHCF: A fair scheduling scheme for 802.11e WLAN", INRIA Research Report No 4883, July 2003. Implementation and NS simulation codes available from *"http://www-sop.inria.fr/planete/qni/fhcf/"*.

[10] ISO/IEC JTC1/SC29/WG11: MPEG4 coding of audio visual objects: visual. (1998).

[11] Soni P. M., Chockalingam A.: Performance analysis of UDP with energy efficient link layer on Markov fading channels. IEEE Transactions on Wireless Communications, **1** (2002).

[12] Jain R.: The art of computer systems performance analysis. John Wiley & Sons publisher, (1991).

[13] Jacod J., Protter P.: Probability essentials. Springer publisher, (2003).

## Appendix

The key idea of our estimator (see Equation 11) is that, in order to allow the QSTAs to adapt to the fluctuations of sending rates, the QAP scheduler tries to estimate the standard deviation by using the expected value of estimation errors.

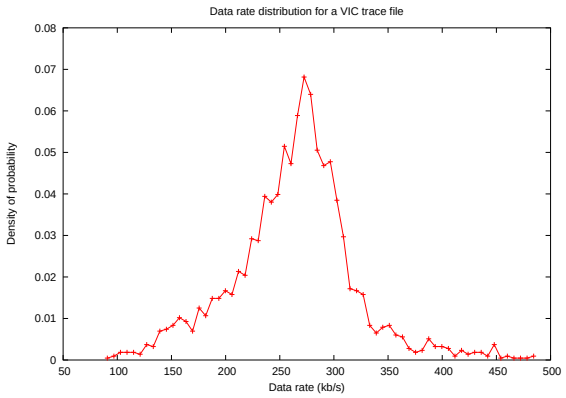



Figure 12: Distribution of the sending rate of a real VBR video traffic

Figure 12 shows that for a typical videoconferencing VBR video traffic, the sending rate almost follows a Gaussian distribution and packets have a fluctuating size. Thus, $\Delta_i(n)$ also follows the same Gaussian distribution with an expected value of 0. Then, the probability for $\Delta_i$ to be N packets is:

$$P_{\Delta_i}(N) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{N^2}{2\sigma^2}\right).$$

To simplify calculations, suppose that data arrive in a continuous bit stream at the TS without being cut into packets and consider $\delta_i$ the difference between the real amount of data and the estimated amount of data. Like $\Delta_i$, $\delta_i$ also follows a Gaussian law. The density of probability for $\delta_i$ is:

$$P_{\delta_i}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right)$$

where $y$ is the amount of additionnal data.

Consequently, the density of probability of $|\delta_i|$ is equal to:

$$P_{|\delta_i|}(y) = \begin{cases} 2P_{\delta_i}(y) \text{ if } y \geq 0 \\ 0 \text{ if } y < 0. \end{cases}$$

The expected value of the variable $|\delta_i|$ is equal to:

$$E(|\delta_i|) = \int_0^\infty \frac{2y}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right) dy = \sqrt{\frac{2}{\pi}}\sigma$$

Thus this corrective term is close to the standard deviation in the context of a Gaussian distribution of the sending rates($\sqrt{\frac{2}{\pi}}$ times the standard deviation). While the standard deviation estimation might provide tighter envelop estimates, it is more complicated and introduces higher computation complexity, and thus we decide to choose this simple estimator (Equation 11), which also provides reasonable accuracy.

However, this estimator is quite generic and works well for most kinds of traffic types.

# D. ARTICLE MORSA

Cette annexe contient un article qui va paraître dans la revue *EURASIP Journal on Wireless Communications and Networking, Special Issue on Ad Hoc Networks : Cross-Layer Issues*. Il décrit un mécanisme de sélection de mode de transmission pour réseaux locaux sans fil 802.11 qui prend en compte les caractéristiques de l'application. Ce mécanisme a été présenté dans le chapitre 5.

# An Evaluation of Media-Oriented Rate Selection Algorithm for Multimedia Transmission in MANETs

Mohammad Hossein Manshaei, Thierry Turletti
Planète Project, INRIA
2004 Route des Lucioles, BP-93
06902 Sophia-Antipolis Cedex, France
E-mail:{manshaei,turletti}@sophia.inria.fr

Thomas Guionnet
Temics Project, IRISA-INRIA
Campus de Beaulieu
35042 Rennes Cedex, France
E-mail: Thomas.Guionnet@irisa.fr

***Abstract*— Current wireless LANs treat multimedia flows and classical data flows alike. Typically, the same error control mechanisms are used for video flows which are generally error-tolerant but delay-sensitive, and TCP flows, which are error-intolerant and delay-insensitive. The performance of multimedia applications can be significantly improved by some degree of cross-layer awareness. In this paper, we focus on the optimization of real time multimedia transmission over 802.11 based ad-hoc networks. In particular, we propose a simple and efficient cross layer mechanism for dynamically selecting the transmission mode that considers both the channel conditions and characteristics of the media. This mechanism called MORSA (Media-Oriented Rate Selection Algorithm) targets loss-tolerant applications such as audio/video conferencing or VoD that do not require full reliable transmission. We provide an evaluation of this mechanism for MANETs using simulations with ns and analyze the video quality obtained with a fine grain scalable video encoder based on a motion-compensated spatio-temporal wavelet transform. Our results show that the proposed cross-layer approach achieves up to 4 Mbps increase in throughput and that the routing overhead decreases significantly. The transmission of a sample video flow over an 802.11a wireless channel has been evaluated with MORSA and compared with the traditional approach. Significant improvement is observed in throughput, latency and jitter while keeping a good level of video quality.**

***Index Terms*— Ad hoc networks, Cross-Layer optimization, IEEE 802.11 Wireless LAN, MANETs, Mode-selection algorithms.**

## I. INTRODUCTION

With recent performance advancements in computer and wireless communications technologies, mobile ad hoc networks (MANETs) are becoming an integral part of communication networks. The emerging widespread use of real-time voice, audio and video applications generates interesting transmission problems to solve over MANETs. Many factors can change the topology of MANETs such as the mobility of nodes or the changes of power level. For instance, power control done at the physical (PHY) layer can affect all other nodes in MANETs, by changing the levels of interference experienced by these nodes and the connectivity of the network, which impacts routing. Therefore, power control is not confined to the physical layer, and can affect the operation of higher level layers. This can be viewed as an opportunity for cross-layering design and poses many new and significant challenges with respect to wired and traditional wireless networks. As soon as we want to optimize data transmission according to both the characteristics of the data and to the varying channel conditions, a cross-layering approach becomes necessary. Numerous cross layer protocols have already been proposed in the literature [1], [2], [3], [4], [5]. They focus on the interactions between the application, transport, network and link layers. With the recent interest on software radio designs [7], it becomes possible to make the PHY layer as flexible as the higher layers. Adaptive and cross layering interactions can now affect the whole stack of the communication protocol. Consequently, the classical OSI approach of providing a PHY layer as reliable as possible independently of the type of data transmitted becomes questionable.

In this paper, we focus on the optimization of real time multimedia transmission over 802.11 based MANETs. In particular, we propose a simple and efficient cross layer protocol which dynamically adjusts the transmission mode, i.e., the physical modulation, rate and possibly the Forward Error Correction (FEC). This protocol called MORSA (Media-Oriented Rate Selection Algorithm) is convenient for loss-tolerant (LT) applications such as video or audio codecs that do not require 100% transmission reliability (i.e., a certain level of packet loss rate (PER) or bit error rate (BER) can be concealed at the receiver). Contrary to mail and file transfer applications, several multimedia applications, such as audio and video conferencing or video on demand (VoD) can tolerate some packet loss. For example, an MPEG video data flow can contain three different types of packet, Intra-Picture (I) frames, Prediction (P) frames and Bi-prediction (B) frames. I-frames are more important for the overall decoding of the video stream, because they serve as reference frames for P- and B-frames. Therefore, the loss of an I-frame has a more drastic impact on the quality of the video playback than the loss of other types of frames. In this respect, the frame loss requirement of I-frames is more stringent than those of P- and B-frames. Furthermore, as described in Section VI, some multimedia applications implement their own error control mechanisms [15] [16], making it inefficient to provide full reliability at the link layer.

MORSA takes into account both the intrinsic characteristics

of the application and varying conditions of the channel. It selects the highest possible transmission rate while guaranteeing a specific bit error rate: the selected transmission mode varies with time depending on the PER or BER tolerance and on the signal-to-noise ratio (SNR) measured at the receiver. We show in this paper that by adaptively selecting the transmission mode according to both loss tolerance requirements of the application and varying channel conditions, the application-layer throughput can be significantly increased and more stability can be achieved in ad hoc routing. Finally, we evaluate the quality of a sample video transmitted over a wireless 802.11a channel using MORSA and compare it with the quality obtained when we do not take into account characteristics of the application (i.e. using the standard approach). Our results show that MORSA can reach a comparable video quality than the one obtained with the standard mechanism while using only a very low ($5\%$) FEC overhead at the application level instead of the physical layer FEC ($50\%$ or $25\%$). This significantly decreases transmission delay of the application.

Throughout this paper, we assume that wireless stations use the Enhanced Distributed Channel Access (EDCA), proposed in the IEEE 802.11e [22] to support different levels of QoS. We have modified the NS simulation tool to evaluate the overall system efficiency when considering the interaction between layers in the protocol stack.

The rest of this paper is structured as follows. In Section II, we overview the salient features of the MAC and PHY layers in the 802.11 schemes. We also review some of the automatic rate selection algorithms that were proposed in the literature. In Section III we present related work about cross layer protocols in ad hoc networks. The MORSA scheme and a possible implementation within a 802.11-compliant device are discussed in Section IV. Simulation results with ns are analyzed in Section V. We evaluate quality of a sample video transmission over a wireless channel in Section VI. Finally, the conclusion is presented in Section VII.

## II. BACKGROUND

Today, three different PHY layers are available for the IEEE 802.11 WLAN as shown in Table I.

The performance of a modulation scheme can be measured by its robustness against path loss, interferences and fading that causes variations in the received SNR. Such variations also cause variations in the BER, since the higher the SNR, the easier it is to demodulate and decode the received bits. Compared to other modulations schemes, BPSK has the minimum probability of bit error for a given SNR. For this reason, it is used as the basic mode for each PHY layer since it has the maximum coverage range among all transmission modes. As shown in Figure 1, each packet may be sent with two different rates [17]: its PLCP (Physical Layer Convergence Protocol) header is sent at the basic rate while the rest of the packet might be sent at a higher rate. The higher rate, used to transmit the physical-layer payload, which includes the MAC header, is[2] stored in the PLCP header.

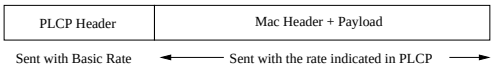



Fig. 1. Data rates for packet transmission.

The receiver can verify that the PLCP header is correct (using CRC or Viterbi decoding with parity), and uses the transmission mode specified in the PLCP header to decode the MAC header and payload. The mode with the lowest rate is used to transmit the PLCP header. Transmission mode selection can be performed manually or automatically in each station. A number of rate selection algorithms have been proposed in the literature. They include the Auto Rate Fallback (ARF) [19], the Receiver-Based Auto Rate (RBAR) [18] and Miser [20] schemes. RBAR tries to select the best mode (i.e. the mode which the highest rate) based on the received SNR, while ARF uses a simple ACK-based mechanism to select the rate. MiSer is a protocol based on the 802.11a/h standards whose goal is to optimize the local power consumption. While all these automatic rate selection mechanisms try to adapt the transmission mode according to the channel conditions, we are not aware of any protocol that considers characteristics of the application.

Since MORSA is based on RBAR, we detail the latter here. In RBAR, the sender chooses a data rate based on some heuristic (e.g., the most recent rate that was used to successfully transmit a packet), and then stores the rate and the packet size into the Request to Send (RTS) control packet. Stations that receive the RTS can use the rate and packet size information to calculate the duration of the requested reservation. They update their Network Allocation Vectors (NAVs) to reflect the reservation. While receiving the RTS, the receiver uses the current channel state as an estimate of the channel state when the upcoming packet is supposed to be transmitted. The receiver then selects the appropriate rate with a simple threshold-based mechanism and includes this rate (along with the packet size) in a Clear to Send (CTS) control packet. Stations that overhear the CTS calculate the duration of the reservation and update their NAVs accordingly. Finally, the sender responds to the CTS by transmitting the data packet at the rate selected by the receiver. Note that nodes that cannot hear the CTS can update their NAVs when they overhear the actual data packet by decoding a part of the MAC header called the *reservation subheader*. Further information concerning RBAR, including implementation and performance issues in 802.11b is available in [18].

## III. RELATED WORK

Several cross layer mechanisms such as mechanisms for TCP over wireless links [1], [5], power control [6], medium access

TABLE I 3

CHARACTERISTICS OF THE VARIOUS PHYSICAL LAYERS IN THE IEEE 802.11 STANDARD.

| *Characteristic* | *802.11a* | *802.11b* | *802.11g* |
|---|---|---|---|
| *Frequency* | 5 GHz | 2.4 GHz | 2.4 GHz |
| *Data Rates* | 6, 9, 12, 18, 24, 36, 48, 54 Mbps | 1, 2, 5.5, 11 Mbps | 1, 2, 5.5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48, 54 Mbps |
| *Modulation* | BPSK, QPSK, 16 QAM, 64 QAM | BPSK, QPSK, CCK | BPSK, QPSK, 16 QAM, 64 QAM, CCK |
| *FEC Rate* | 1/2, 2/3, 3/4 | NA | 1/2, 2/3, 3/4 |
| *Basic Transmission Mode* | BPSK, 6 Mbps, FEC 1/2 | BPSK, 1 Mbps | 802.11a (6 Mbps) or 802.11b (1 Mbps) basic modes |

control [2], QoS providing [8], video streaming over wireless LANs [9], and deployment network access point [1] have been proposed.

The Mobileman European project [12] introduced inside the layered architecture the possibility that protocols belonging to different layers can cooperate by sharing network status information while still maintaining separation between the layers in protocol design. The authors propose applying triggers to the network status such that it can send signals between layers. In particular, This cross-layering approach addresses the security and cooperation, energy management, and quality-of-service issues.

The effect of such cross layer mechanisms on the routing protocol, the queuing discipline, the power control algorithm, and the medium access control layer performance have been studied in [2].

A cross-layer algorithm using MAC channel reservation control packets at the physical layer is described in [4]. This mechanism improves the network throughput significantly for mobile ad hoc networks because the nodes are able to perform an adaptive selection of a spectrally efficient transmission rate.

[9] describes a cross-layer algorithm that employs different error control and adaptation mechanisms implemented on both application and MAC layers for robust transmission of video. These mechanisms are Media Access Control (MAC) retransmission strategy, application-layer Forward Error Correction (FEC), bandwidth-adaptive compression using scalable coding, and adaptive packetization strategies. Similarly a set of end-to-end application layer techniques for adaptive video streaming over wireless networks is proposed in [10]. In [11], the Adaptive Source Rate Control (ASRC) scheme is proposed to adjust the source rate based on the channel conditions, the transport buffer occupancy and the delay constraints. This cross layer scheme can work together with hybrid ARQ error control schemes to achieve efficient transmission of real-time video with low delay and high reliability. However, none of these algorithms have tried to adapt the physical layer transmission mode in 802.11 WLANs. More examples could be cited, but we are not aware of any cross layer algorithm that takes into account the physical layer parameters (e.g., PHY FEC) as explained in Section II.

It should be noted that standardization efforts are in progress to integrate various architectures. The important co-design of the physical, MAC and higher layers have been taken into account in some of the latest standards like 3G standards (CDMA2000), BRAN HiperLAN2, and 3GPP (High Speed Downlink Packet Access) [1]. IEEE has also considered a cross-layer design in the study group on Mobile Broadband Wireless Access (MBWA).

## IV. CROSS LAYER MODE SELECTION PROTOCOL

This section describes the MORSA mechanism and discusses implementation issues.

### *A. Algorithm Description*

As we already mentioned, real-time multimedia applications can be characterized by their tolerance to a certain amount of packet loss or bit errors. These losses can be ignored (if they are barely noticeable by human viewers) or compensated at the receiver using various error concealment techniques. In our scheme, the sender is able to specify its loss tolerance (LT) such that the receiver uses both this information and the current channel conditions to select the appropriate transmission mode (i.e., rate, modulation, and FEC level). More precisely, the sender includes the LT information in each RTS packet to allow the receiver to select the best mode. The LT information is also included in the header of each data packet such that the receiver can decide whether or not to accept a packet. While receiving the RTS, the receiver uses the information concerning the channel conditions along with the information related to LT to select the best data rate for the corresponding packet. The selected rate is then transmitted along with the packet size in the CTS back to the sender, and the sender uses this rate to send its data packets. When a packet arrives at the receiver side, if the receiver is able to decode the PLCP header, it can identify the BER tolerance for the encoded payload. If the packet can tolerate some bit errors, it has to be accepted even if its payload contains errors. As will be shown later, our mechanism makes it possible to define new transmission modes that do not use FEC but that exhibit comparable throughput performance.

To take into account both the SNR and the LT information, we have modified the RBAR threshold[1] mechanism. For 802.11a, we assume that the receiver uses FEC Viterbi decoding. The upper bound on the probability of error provided in [13], [20] is used under the assumption of binary convolutional

[1]These thresholds are used to select the best transmission mode in the receiver.

coding and hard-decision Viterbi decoding. Specifically, for a packet of length L (bytes) the probability of packet error can be bound by:

$$P_e(L) \leq 1 - (1 - P_u)^{8L} \quad (1)$$

where the union bound $P_u$ of the first-event error probability is given by

$$P_u = \sum_{d=d_{free}}^{\infty} a_d \cdot P_d \quad (2)$$

With $d_{free}$, the free distance of the convolutional code, $a_d$ the total number of error events of weight[2] d and $P_d$, the probability that an incorrect path at distance $d$ from the correct path is chosen by the Viterbi decoder. When hard decision decoding is applied, $P_d$ is given by Equation 3, where $\rho$ is the probability of bit error for the modulation selected in the physical layer[3].

$$P_d = \begin{cases} \sum_{k=(d+1)/2}^{d} \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k} & \text{if d is odd} \\ \frac{1}{2} \cdot \binom{d}{d/2} \cdot \rho^{d/2} \cdot (1-\rho)^{d/2} + \sum_{k=d/2+1}^{d} \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k} & \text{if d is even} \end{cases} \quad (3)$$

Figure 2 shows an example of the modifications made for the SNR threshold in RBAR with and without the media-oriented mechanism. Commonly, a BER at the physical layer smaller than $10^{-5}$ is considered acceptable in wireless LAN applications. By using theoretical graphs of BER as function of the SNR for different transmission modes on a simple additive white Gaussian noise (AWGN) channel (see Figure 2), we can compute the minimum SNR values required. Now if a particular application can tolerate some bit errors (e.g. a BER up to the $10^{-3}$ as shown in Figure 2), the receiver can select the highest rate for the following data transmission corresponding to this SNR. For example in Figure 2, when the SNR is equal to $5dB$, the receiver can select a $9Mbps$ data rate instead of a 6 $Mbps$ data rate if it is aware that the application can tolerate a BER less that $10^{-3}$.

We have calculated the thresholds using Equations 1, 2, and 3 for an application that can tolerate up to $10^{-3}$ BER (See Table III in Section V.). The receiver can use arrays of thresholds that are pre-computed for different LTs.

In the following sections, we describe how such a mechanism can be implemented in 802.11-based WLANs.

### *B. Implementation issues*

We propose to implement MORSA with the help of the EDCA protocol [21], [23]. EDCA is one of the features that has been proposed by IEEE 802.11e to support QoS in WLANs

[2]We have used the $a_d$ coefficients provided in [14].

[3]In this paper we use Additive White Gaussian Noise (AWGN) channel model.

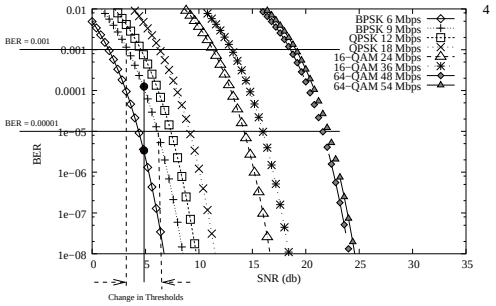



Fig. 2. Bit error rate (BER) versus SNR for various transmission modes (802.11a).

[22]. In this protocol each QoS-enhanced station (QSTA) has 4 queues to support up to 8 User Priorities (UP). Figure 3 shows the QoS control field that is added to the MAC header in the 802.11e specification [22]. Bits 6 and 7 of this header can be used to indicate the loss tolerance information. Table II shows a possible meaning for these two bits in our media-oriented mechanism that should be defined in the process of connection setup. LT information is sent to the receiver by adding one byte to the RTS packets as illustrated in Figure 4.

| Bit 0-3 | Bit 4 | Bit 5 | Bit 6-7 | Bit 8-15 |
|---|---|---|---|---|
| *Traffic ID* | *Schedule Pending* | *Ack Policy* | *Reserved* | *TXOP duration* |

Fig. 3. QoS control field in the 802.11e.

TABLE II
LOSS TOLERANCE CLASSIFICATION.

| Bit 6-7 | Application Sensitivity |
|---|---|
| 00 | No tolerance in payload |
| 01 | Low loss tolerance in payload |
| 10 | Medium loss tolerance in payload |
| 11 | High loss tolerance in payload |

| BYTES | 2 | 2 | 6 | 6 | 1 | 4 |
|---|---|---|---|---|---|---|
| | Frame Control | Rate & Length | Dest Address | Source Address | Tolerance Information | FCS |

Fig. 4. Modifications to the RTS header.

To make our mechanism operational, it is crucial to let the packets with corrupted payload reach the receiver's application layer. As such, some modifications of the standard are necessary. First, the CRC at the MAC layer should no more cover the payload but only the MAC, IP, UDP, and possibly the RTP headers. Second, the optional UDP checksum must be

disabled, as described in the UDP Lite proposal [25]. UDP Lite is a lightweight version of UDP with increased flexibility in the form of a partial checksum. The coverage of the checksum is specified by the sending application on a per-packet basis. This protocol can be profitable for MORSA. Furthermore, to make our mechanism more robust against bit errors, the headers of the different layers (MAC, IP, UDP, and RTP) have to be sent with the basic rate (See Figure 5). This is somewhat similar to the reservation sub-header used in [18] as explained in Section II. The corresponding bandwidth overhead is investigated in the next section.

## V. Simulation Results

Our simulations are based on the simulation environment described in [26] which uses the ns-2 network simulator, with extensions from the CMU Monarch project [27] to simulate multi-hop wireless ad hoc networks. In order to obtain more realistic results, Cisco Aironet 1200 Series parameters are used in our simulations [28]. Further details about the simulation environment are available in [26].

Note that in the following simulations, CTS and RTS control packets and PLCP headers are sent with a BPSK modulation, a FEC rate equal to 1/2 and a 6 Mbps data rate. All throughput shown in the following figures exclude the MAC and PHY headers; they are denoted as goodput for the remainder of the paper.

To evaluate the perceived quality for the user using our protocol, we have taken an example of video application that can tolerate 0.1% of bit errors (See Section VI-B). Thus, we have investigated the throughput performance of MORSA when the BER is equal to $10^{-3}$ in the following simulations. Of course other values of the BER can be chosen to perform simulations with similar results.

In our simulation, we assume that bit errors in a packet are distributed according to a binomial distribution. This is an acceptable assumption since the position of the bit errors are not taken into account by ns-2. In Section VI, we will provide more precise models for the distribution of bit errors in our data stream. Let $n$ represent the number of bit errors in a packet of $N$ bits, and $p$ be the probability of bit error. The probability of having less than $L$ bit errors can be calculated by:

$$P(n \leq L) = \sum_{i=0}^{L} \binom{N}{i} \cdot p^i \cdot (1-p)^{N-i} \quad (4)$$

We first evaluate our mechanism in a simple ad hoc network that contains two wireless stations. These wireless stations communicate on a single channel. Station A is fixed and station B moves toward station A. Station B moves in 5m increments over the range of mobility (0m - 200m) and is held fixed for a $60s$ transmission of CBR data towards station A. 30000 CBR packets of size 2304 bytes (including physical layer FEC) are sent in each step.

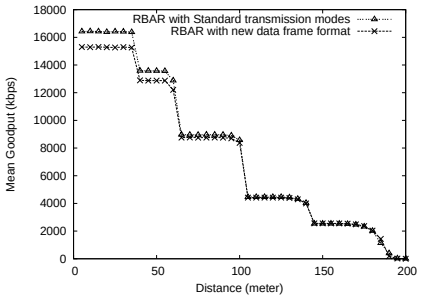



Fig. 7. Overhead of the modified frame format.

Figure 6 shows the mean goodput of this single CBR connection between two wireless stations versus the distance between them for different transmission modes with and without media-oriented mechanism[4].

Since no payload FEC is used in our media-oriented protocol, the mean goodput is increased significantly compared to the standard transmission modes. For example, we can observe that the media-oriented mechanism achieves a 4 Mbps mean goodput improvement at the highest rate mode. However, this has a cost in coverage range: in the same example, it is 50 meters less. It should be noted that if an application can tolerate more bit errors, the coverage range will be larger than for the standard transmission modes [23].

We have also evaluated the extra bandwidth overhead of the modified frame format. This overhead is caused by having to send the MAC header at the basic mode and by the additional byte in the RTS packet. Figure 7 compares the mean throughput for the traditional RBAR and for RBAR with the modified frame format. The worst-case overhead at the maximum rate is about 1 Mbps, but the coverage range does not change much compared to the standard specification.

To evaluate the performance of RBAR under different mode selection mechanisms, we need to calculate arrays of thresholds for each mechanism (see Section IV). Table III shows these threshold values for RBAR and MORSA[5]. These results show that if we can tolerate loss we will be able to send data with a higher rate.

Figure 8 illustrates the performance of RBAR and MORSA. Since the standard mode selection mechanism can achieve the maximum coverage range and the media-oriented mechanism obtains the maximum mean goodput, we have defined a new media-oriented mode selection mechanism called *hybrid transmission mode selection* or *H-MORSA*, to achieve both objectives at the same time (see Figure 9). The five PHY

[4]Based on our simulation study for 802.11a, we have selected five efficient transmission modes out of the 8 possible transmission modes in 802.11a [26].

[5]For a SNR smaller than these values, data will be sent with the basic mode which is 6 Mbps.

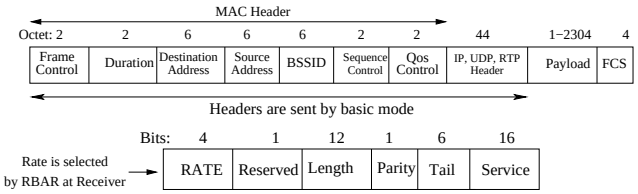



Fig. 5. Proposed Frame format.

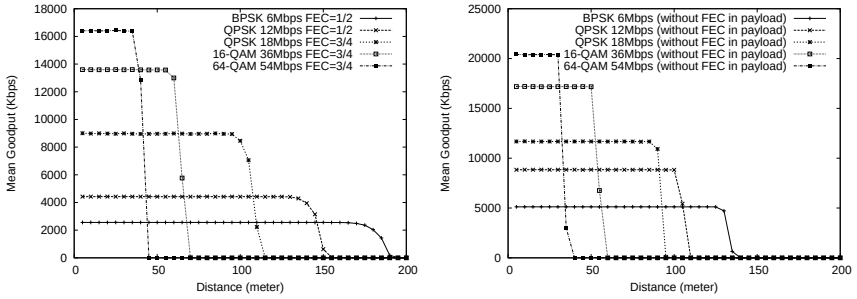



Fig. 6. Mean goodput versus distance for Standard transmission modes $(left)$ and Media-oriented with 0.1% bit errors $(right)$.

TABLE III
SNR(DB) THRESHOLD VALUES TO SELECT THE BEST TRANSMISSION MODE

| | Standard (With FEC) | Media-oriented (No LT) | Media-oriented (0.1% LT) |
|---|---|---|---|
| 12 Mbps | 0.68 | 6.12 | 4.94 |
| 18 Mbps | 4.75 | 7.37 | 6.18 |
| 36 Mbps | 11.39 | 14.22 | 13.5 |
| 54 Mbps | 17.29 | 21.58 | 20.3 |

transmission modes that are used for the hybrid mode selection mechanism do not use FEC.

Then, we evaluate the two media-oriented mechanisms (MORSA and H-MORSA) in ad hoc networks. Figure 10 shows an example of network configuration for 20 nodes which are commonly used for ad hoc network evaluation [18], [29], [27]. In our simulation, each ad hoc network consists of 20 mobile nodes that are distributed randomly in a 1500x300 meter arena. The speed at which nodes move is uniformly distributed between $0.9v$ and $1.1v$, for different speeds of $v$. We use the following speed values 2, 4, 6, 8 and 10 $m/s$. The nodes choose their path randomly according to a random waypoint mobility pattern. The same movement patterns are used in all experiments whatever the mean node speed. For

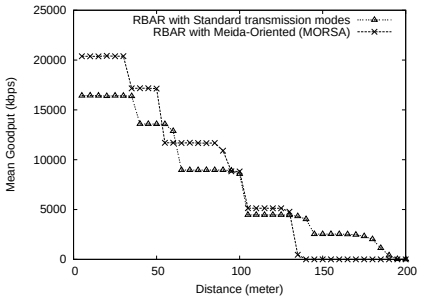



Fig. 8. RBAR performance for standard and media-oriented protocols (MORSA).

example, if node A moves from point $a$ to point $b$ with a speed of 2 m/s, it will take the same route with 4, 6, 8 and 10m/s in the other scenario patterns but with different delays. All the results are based on an average over 30 simulations with 30 different scenario patterns.

In each simulation, a single UDP connection sends data between two selected nodes. Other nodes can forward their packets in the ad hoc network. The data is generated by a CBR source at saturated rate. In other words, there are always

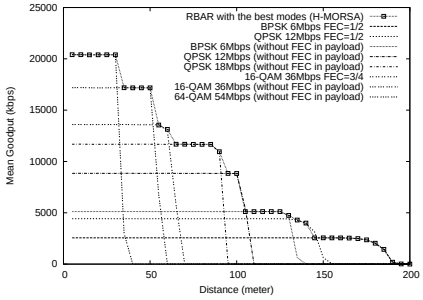



Fig. 9. RBAR performance using standard or media-oriented protocol (H-MORSA).

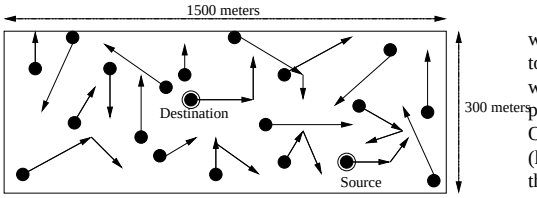



Fig. 10. Example of ad hoc network topology scenario.

packets to send during the whole simulation time. Unlike in the simple network topology with 2 nodes where we used static routing, here the Dynamic Source Routing (DSR) [29] protocol has been used. DSR is a simple and efficient routing protocol designed specifically for use in multi-hop ad hoc networks. It should be noted that routing packets are sent using the basic transmission mode like the RTS, CTS, and ACK control packets.

We use three automatic mode selection mechanisms defined in our previous simulations (see Figures 8 and 9). In the standard mode selection mechanism (RBAR) and hybrid mode selection mechanism (H-MORSA), we may have a hop in the route between source and destination that uses a physical FEC equal to 1/2. Thus, we have to use packets with a payload length equal to 1152 bytes for these simulations. However, with MORSA we are able to send packets with 2304 bytes since no physical layer FEC is used in this mechanism.

Figure 11 shows the mean goodput of a single CBR connection versus different mean node speeds. For an application that can tolerate a BER of $10^{-3}$, the mean goodput is about 25% higher when we take into account the application's characteristics.

Figure 12 shows the number of delivered bits for 30 scenario patterns[6] with mean speed equal to $2m/s$. In the scenarios

[6]Scenarios are sorted by the number of delivered bits obtained with the standard mode selection mechanism.



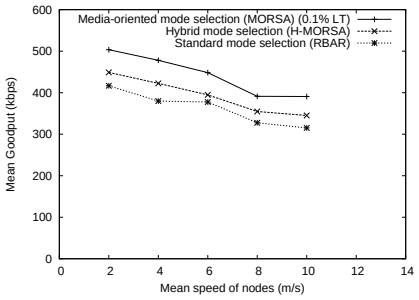



Fig. 11. Performance comparison for a single CBR connection in a multihop network, with and without MORSA.

where the number of delivered bits is zero, DSR was not able to find a route between the source and the destination during the whole simulation time. As expected, in most of the scenario patterns MORSA can deliver more data bits to the receiver. One interesting observation is that in some scenario patterns (less than 15% of them) the number of delivered bits with the standard RBAR and H-MORSA is more than the one in MORSA. The rationale behind this is that DSR packets can be sent with the maximum coverage range in the standard and the hybrid mode selection mechanisms. As a result, the source can find a route to the destination faster than MORSA. Thus, the number of delivered packets in the standard RBAR and the H-MORSA is more than that of MORSA (e.g., scenario number 20).

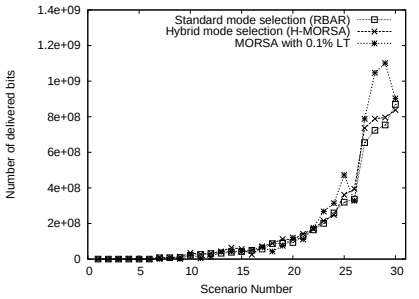



Fig. 12. Number of delivered bits to the application ($speed = 2m/s$).

We have also evaluated the overhead of the DSR routing protocol in different cases. The DSR algorithm has two different phases called *route discovery* and *route maintenance* to manage the routes in ad hoc networks. In *route Discovery*, ad hoc nodes need to find a route between the source and the destination. This is performed only when the source attempts to send a packet to the destination and does not already know

a route. In *route maintenance*, DSR detects changes in the network topology such that the source can no longer use the current route to destination. This can occur if a link along the route is not usable anymore.

Figure 13 shows the number of routing overhead packets generated by DSR, which have been sent in ad hoc networks according to different mean speed of the nodes. In order to evaluate this overhead, we have considered all DSR routing packets that should be sent before making a connection and during data transmission. So this overhead includes *route discovery* and *route maintenance* overheads. These results show that routing overhead decreases significantly when we use MORSA. We believe this is a consequence of having more stable connection when MORSA is used.

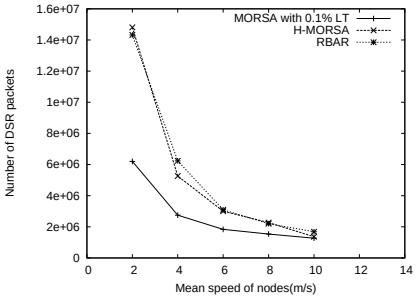



Fig. 13. DSR routing overhead in multihop network.

We have done different simulations to evaluate the performance of our mechanism in the presence of interference for ad hoc networks. For these simulations, 20 nodes are distributed in an area of 500x100 meters which is 9 times smaller than previous simulation scenarios. In this simulation, 6 UDP connections are set up between 12 different nodes. Data is generated by CBR sources at a saturation rate. The first source starts data transmission at time 3.12 and the last one at 25.12. For this simulation, nodes are fixed and DSR does not need to use *route maintenance*. The results are averaged over 30 different scenario patterns. Figure 14 shows the performance of MORSA in these experiments. Clearly, MORSA outperforms the standard mode selection (RBAR) and hybrid mode selection (H-MORSA) mechanisms. This is because the media-oriented mechanism considers the application's characteristics and does not use FEC at the physical layer when the channel condition is good.

## VI. Evaluation of Video Quality

Simulation results in ns-2 have shown a significant improvement in throughput when considering the loss requirements of the application to select the transmission mode. In this section, we evaluate the effectiveness of the proposed media-oriented



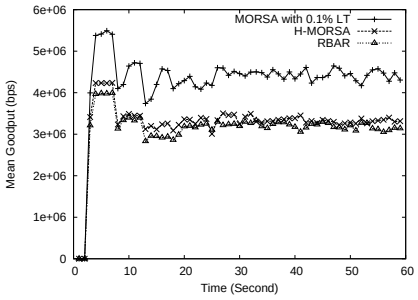



Fig. 14. Performance comparison for a several CBR connection in multihop network, with and without media-oriented mechanism.

mechanism using the simulation of a video transmission over a 802.11a wireless channel. Our previous observations about the performance of the media-oriented mechanism can be further justified by the evaluation of the video quality obtained at the receiver when we employ the media-oriented mechanism. In the following sections, we describe a wireless channel model that can estimate the position and the length of burst error bits in 802.11a. Then, we present a video application that can tolerate a BER equal to $10^{-3}$ by using an application level FEC whose overhead is only $5\%$. Finally, we compare the transmission delay and the video quality (Peak Signal-to-Noise Ratio) with standard and media-oriented transmission mechanisms.

### A. 802.11a Channel Model

Wireless channel models can be divided into two main groups: *memoryless models* and *models with memory*. In memoryless models, corrupted bits are produced by a sequence of independent trials. Each trial has the same probability $p$ of producing a correct bit and probability $q = 1-p$ of producing a bit error. However, in a real communication environment, links have memory and errors often occur in isolated bursts because of multipath fading, impulsive noise, or switch transients. A classic method to model a wireless channel with memory is using a Markov chain. In this model, the probability of bit error depends on the state of the model. We have considered in this section a model with memory, which is based on the model proposed in [33] for 802.11a WLANs.

In the 802.11a physical layer, the data field shall be encoded with a standard convolutional encoder of different coding rate $R$ = $1/2$, $2/3$, or $3/4$, depending on the data rate. The $1/2$ convolutional encoder uses the generator polynomials, $G_0 = 1338$ and $G_1 = 1718$, and simple puncturing is applied to derive higher convolutional rates [24]. Regarding convolutional decoding, it is usually implemented using the Viterbi algorithm.

In this paper, we use the derivation for distribution of error events obtained in these convolutional codes at the output of the

Viterbi decoder. We estimate the position and the length of bit errors at the output of the decoder with this method. We use asymptotic bounds to analyze the distribution of error event lengths at the output the Viterbi decoder. We also consider the relationship between the error probability of a random convolutional code and the error probability of a particular block code (termed *code termination* technique and presented in [34]). The tail of the distribution that is otherwise difficult to estimate with classical techniques can be estimated with this method.

Then, we use the error event length distribution and the distribution of errorless periods to derive a simple model which describes the residual error at the output of the soft decision Viterbi decoder. In the next section, we use this model to compute the distribution of corrupted bits for different transmission modes.

*B. Video Encoder*

The concept of fine grain scalability (FGS) has been introduced in order to allow for dynamic rate adaptation to varying bandwidth and receiver capabilities. Compression solutions based on motion-compensated spatio-temporal signal decomposition have thus gained attention as viable alternatives to classical predictive techniques for scalable video representation. The video codec that has been used in the experiments reported here, referred to as WAVIX in the sequel, has been developed in this framework.

A group of frames (GOF) is fed into the coding system. In order to fine tune the bit rate allocated to the motion fields, the block matching motion estimation makes use of a rate-constrained adaptive tree structure. The block size is thus adapted to local motion characteristics in a rate-distortion sense. The rate here refers to the bit rate allocated to encode the motion vectors and the distortion relates to the prediction error. The estimation itself, to save computation time, relies on a hierarchical approach. The motion vectors obtained in the first steps of the quadtree are used to initialize the search in the subsequent steps. The motion vectors are then predictively coded. The predictor is given by the median value of neighboring vectors. The prediction error is then coded using Huffman codes.

The GOF is fed to the motion-compensated temporal transform which is based on a two-taps Haar wavelet transform. The temporal decomposition is applied iteratively on pairs of images within the GOF. The advantage of the Haar wavelet transform is to achieve a fairly good temporal energy compaction with a limited number of motion fields (8 motion fields for a 3-stage temporal decomposition of a group of 8 images). Each temporal subband is then further decomposed by a bi-orthogonal 9-7 wavelet filter in the horizontal and vertical direction. In the experiments, 3 levels decompositions are being used. The subbands resulting from the spatio-temporal decomposition are then quantized with a uniform quantizer and encoded with a context-based bit plane arithmetic coding as used in the JPEG-2000 standard [35]. The algorithm optimizing the truncation points in a rate-distortion sense handles groups of spatio-temporal subbands. The truncation point rate-distortion optimization leading to quality layers is well suited to fine tune the rate allocated to the texture information, hence to support fine grain scalability.[9]

An inter-GOF temporal prediction is also used as an option in the above coding system. The inter-GOF temporal prediction leads to GOFs of type INTRA and of type INTER. The inter-GOF temporal prediction requires one additional motion field. This temporal prediction and corresponding motion estimation is realized in a closed loop. The closed-loop prediction is done by taking as reference information the corresponding image coded at a lower rate, as used in a base layer of a scalable representation. A more detailed description of this video codec can be found in [31].

Arithmetic codes are widely used in coding systems due to their high compression efficiency. They are however very sensitive to bit errors. A single bit error may lead to a complete de-synchronization of the decoder. In order to make the WAVIX codec robust to errors, an error-resilient arithmetic codes decoding technique [30] has been integrated in the video decoder. The technique consists in exploiting the residual redundancy in the bitstream by using soft-decision decoding procedures. The term *soft* here means that the decoder takes in input and supplies not only binary (*hard*) decisions but also a measure of confidence (a probability) on the bits. One can thus exploit the so-called *excess-rate* (or sub-optimality of the code), to reduce the catastrophic *de-synchronization* effect of VLC decoders, hence to reduce the residual symbol error rates. This amounts to exploiting inner codeword redundancy as well as the remaining correlation within the sequence of symbols (remaining inter-symbol dependency).

In practice, the decoding algorithm can be regarded as a soft-input soft-output sequential decoding technique run on a tree. The complexity of the underlying Bayesian estimation algorithm growing exponentially with the number of coded symbols, a simple, yet efficient, pruning method is integrated. It allows the user to limit the complexity within a tractable and realistic range, at a limited cost in terms of estimation accuracy.

In order to increase the re-synchronization capability, a *soft synchronization* mechanism has been added. This mechanism relies on both the use of *soft synchronization* markers and of forbidden symbols. The *soft synchronization* markers are patterns, inserted in the symbol stream at some known positions, which serve as anchors for favoring the likelihood of correctly synchronized decoding paths. This *soft synchronization* idea augments the auto-synchronization power of the chain at a controllable loss in information rate. The forbidden symbols, when used, provide additional error detection and correction capability [32].

The bitstream generated by WAVIX is split into motion vectors and texture. The texture is encoded with the EBCOT

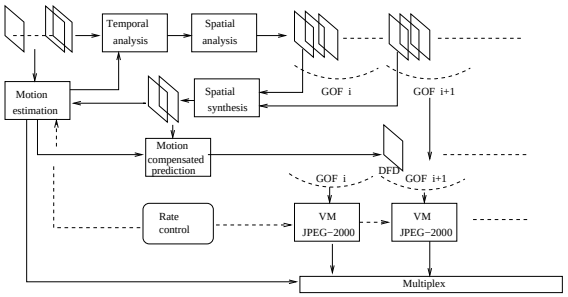



Fig. 15. WAVIX structure

algorithm. Hence it has the same properties as a JPEG 2000 bitstream. The corresponding bitstream is separated into header and entropy coded data. The header contains high level information, like GOF sizes, and provides a description of the structure of the entropy coded data. As this information is essential to the decoder, it is protected by a Reed-Solomon block code with high redundancy (127/255 for instance).

### C. Multimedia Transmission over 802.11a Wireless Channel

In this section, we evaluate the quality of the video bitstream at the receiver side when the media-oriented mechanism is used. In our experiments, the WAVIX video encoder is configured to encode a sample of 300 CIF video frames. The video bit rate is $2Mbits/s$ and each frame is a YUV image[7]. The number of frames in each GOF is 8. The activation of the WAVIX error resilience options corresponds to the addition of a 127/255 Reed-Solomon block code for header protection and of synchronization markers as explained in Section VI-B. The overhead of the header protection represents about $5.2\%$ of the video stream while the overhead of the synchronization markers is negligible.

The transmission delay is calculated by considering the number of retransmissions and the value of the backoff timer [17]. The retransmission limit is defined in the IEEE 802.11 MAC standard specification with the help of the two following counters: The Short Retry Count(SRC) and the Long Retry Count (LRC). These counters are incremented and reset independently. The SRC is incremented every time an RTS fails and LRC is incremented when data transmission fails. Both the SRC and the LRC are reset to $0$ after a successful data transmission. Data frames are discarded when SRC (LRC) reaches dot11ShortRetryLimit (dot11LongRetryLimit). The default values for dot11ShortRetryLimit and dot11LongRetryLimit are $7$ and $4$ respectively.

Furthermore, we consider the backoff timer period after each retransmission. For each retransmission, we select a random backoff which is drawn from a uniform distribution over the interval $[0, CW]$. In each retransmission, $CW$ is updated to either $2 \times (CW + 1) - 1$ or its maximal value $aCW_{max}$. Let $\bar{T}_{backoff}(i)$ denote the average backoff interval after $i$ consecutive unsuccessful transmission attempts. It can be calculated by [36]:

$$\bar{T}_{backoff}(i) = \begin{cases} \frac{2^i(aCW_{min}+1)-1}{2} \cdot aSlotTime & 0 \leq i \leq 6 \\ \frac{aCW_{max}}{2} \cdot aSlotTime & i \geq 6 \end{cases} \quad (5)$$

Where $aCW_{min}$, $aCW_{max}$ and $aSlotTime$ are $15$, $1023$ and $9\mu s$ for IEEE 802.11a WLANs[24].

We have chosen 4 SNR corresponding to 4 different transmission modes (see Table III). Using the 802.11a channel model described in Section VI-A, we can find the distribution of bit errors for each SNR and transmission mode at the output of Viterbi decoder. The bit errors are distributed over the packets of length $1000bytes$.

In the standard transmission mode, we only accept packets without corrupted bits. The error resilience options of the application layer are not employed for the standard transmission mechanism. However, we activate the WAVIX error resilience options and we accept packets with corrupted payload for the media-oriented mode selection mechanism.

Figures 16-19 show the PSNR, transmission delay, and interval jitter performance for 4 transmission modes with both the standard and the media-oriented mechanisms. Table IV also shows the overall duration of the transmission for this video stream. As expected, the media-oriented mechanism (with $LT = 0.1\%$ and $5.2\%$ FEC overhead at the application layer) significantly decreases the overall duration of the transmission (See Table IV).

We made the following observations from Figures 16-19. The packet transmission time is almost fixed with the media-oriented mechanism while it continuously changes with the number of retransmissions using the standard mechanism. When the media-oriented mechanism is used, the PSNR of the decoded video is equivalent to the standard transmission

[7]The foreman CIF ($352 \times 288$ pixels) video sequence has been used.

TABLE IV 11

TRANSMISSION TIME COMPARISON FOR VIDEO TRANSMISSION WITH AND WITHOUT MEDIA-ORIENTED MECHANISM

| Modulation | Data Rate (Mbps) | FEC Rate | SNR (dB) | Transmission duration for Standard (s) | Transmission duration for Media-oriented (s) |
|---|---|---|---|---|---|
| BPSK | 6 | 1/2 | -1.6 | 8.00 | 6.92 |
| QPSK | 12 | 1/2 | 1.3 | 4.14 | 3.57 |
| 16-QAM | 36 | 3/4 | 8.5 | 1.09 | 0.96 |
| 64-QAM | 54 | 3/4 | 17.3 | 0.81 | 0.72 |

mode, except for the *drops* that correspond to GOFs where errors occur. In this case, error resilience options allow us to decode the GOFs with the best achievable visual quality. The corrupted frames exhibit a lower quality, but their visual content is preserved. When the PSNR remains above 30 dB, the degradation is generally unnoticeable for a human viewer. When the PSNR falls as low as 25 dB, the decoded frames are severely degraded but still acceptable by a human viewer. The impact of errors on the visual quality depends on the characteristics of the current frame (in particular, the number and positions of errors, and the video content). In applications involving real time constraints, as for instance visiophony or streaming, it may be preferable to receive a degraded frame rather than losing it entirely or slowing down the video playback because of packets retransmission.

Another observation from the PSNR calculation is that after 4 consecutive retransmissions, (i.e. when a packet is lost for good), the standard transmission mechanism can not decode the rest of the video frame (e.g., this occurs at the frame number 220 in the Figure 16). However, this problem can be solved at the transmitter side with a more intelligent packetization scheme, or by adding resynchronisation patterns within the data flow. Nonetheless, in case of packet drop, the visual content of a full GOF may be lost.

Figures 16-19 also show the jitter for the standard and the media-oriented mode selection mechanisms. First, it is obviously and logically correlated to transmission delay. In the media-oriented mechanism, the jitter is much less important than with the standard mode. This is a very desirable property in the case of video transmission. Having a constant time interval between packets arrivals is equivalent to having a constant time slot available to decode each GOF. Therefore, complexity can be managed easily without the need for excessive buffering.

We have simulated the same scenarios for 10 different channel characteristics (different distributions of corrupted bits over data flow) in order to calculate the confidence interval of the PSNR with the media-oriented transmission mode. For each transmission rate, the 95% confidence intervals on the mean PSNR are computed. The intervals for the various rates are displayed by horizontal lines as shown in Figure 20. The results show an acceptable PSNR in all transmission modes.

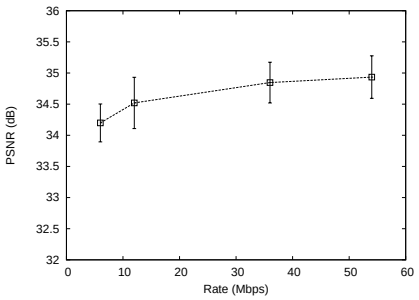



Fig. 20. 95% confi dence intervals of PSNR for different transmission modes with media-oriented mode selection mechanism.

Fig. 21. A sample of video stream at the receiver, transmitted by Media-oriented algorithm with 0.1% bit errors $(left)$ (SNR=1.3, Rate = 12 Mbps), original video stream $(right)$.

## VII. CONCLUSION

In this paper, we have presented a novel cross layer mechanism in MANETs to select the best transmission mode which takes into account some characteristics of the application. This mechanism, which we believe to be easy to implement in actual devices, uses information from the physical channel and the loss tolerance requirements of the application to select the optimal PHY rate, modulation and FEC transmission parameters. We have proposed new transmission modes which do not use FEC and which significantly increase the application throughput. Ns-based simulation results in ad hoc networks show that our mechanism achieves up to $4Mbps$ increase in throughput in MANETs. The gain obtained from the application point of view has been evaluated with the help of the WAVIX video encoder, which can tolerate a BER equal to $10^{-3}$ with only 5% of FEC overhead at the application level. The results show significant improvements in throughput, latency and jitter.

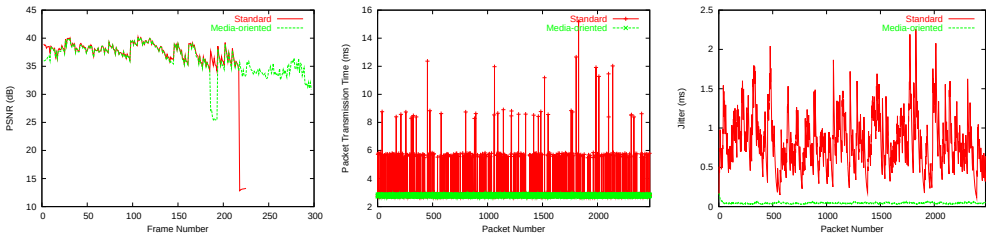



Fig. 16. PSNR, transmission delay, and jitter comparison (SNR = $-1.6dB$, 6 Mbps, FEC=1/2, BPSK).

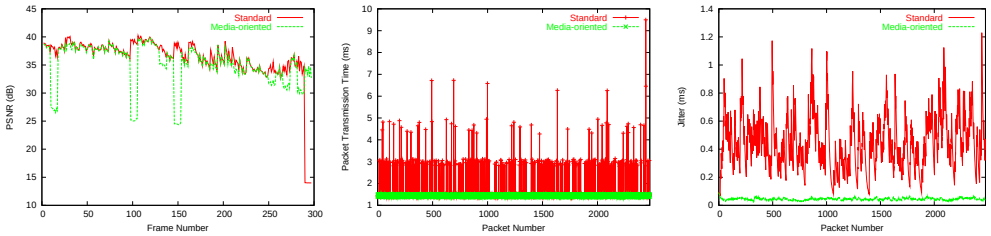



Fig. 17. PSNR, transmission delay, and jitter comparison (SNR = $1.3dB$, 12 Mbps, FEC=1/2, QPSK).

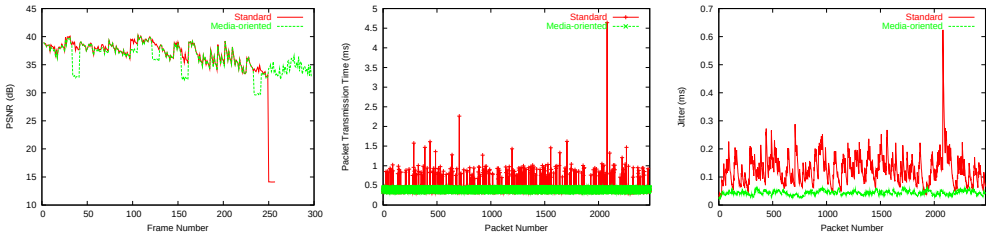



Fig. 18. PSNR, transmission delay, and jitter comparison (SNR = $8.5dB$, 36 Mbps, FEC=3/4, 16-QAM).

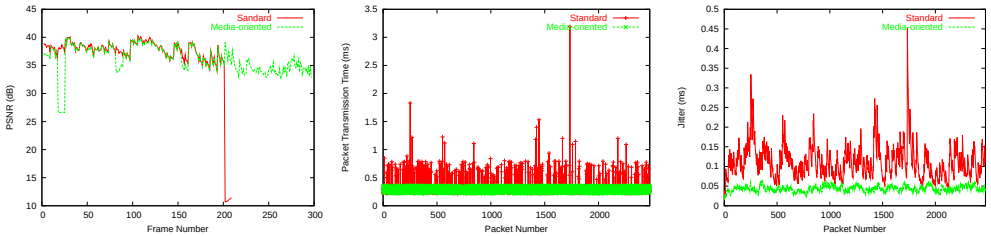



Fig. 19. PSNR, transmission delay, and jitter comparison (SNR = $17.3dB$, 54 Mbps, FEC=3/4, 64-QAM).

## ACKNOWLEDGMENTS

The authors wish to thank Marwan Krunz (University of Arizona, USA) for the many helpful discussions on protocol design during his visit at INRIA. The authors would also like to thank Kave Salamatian and Ramin Khalili (LIP6, FRANCE) for their help in channel modelling for 802.11a WLANs. Finally, we are grateful to Christine Guillemot and Mathieu Lacage (INRIA, FRANCE) for their critical comments on improving the quality of the paper.


## REFERENCES


[1] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks", *IEEE Communications Magazine*, Vol. 41, No. 10, October 2003, pp. 74-80.

[2] S. Toumpis, "Capacity and Cross-Layer Design of Wireless Ad Hoc Networks.", *PhD thesis*, Department of Electrical Engineering of Stanford University, USA, July 2003.

[3] A. Safwat, H. Hassanein, and H. Mouftah, "Cross-Layer Designs for Energy-Efficient Wireless Ad hoc and Sensor Networks,", *22nd IEEE International Performance, Computing, and Communications Conference*, Phoenix, Arizona, USA, April 2003.

[4] W. H. Yuen, H. Lee and T. D. Andersen, "A Cross Layer Networking System for Mobile Ad Hoc Networks,", *IEEE PIMRC'02* at Lisbon, Portugal, September 2002.

[5] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks", *ACM Wireless Networks*, Vol. 8, No. 2, March 2002.

[6] V. Bhuvaneshwar, M. Krunz, and Alaa Muqattash, "A Cross-Layer Power Aware Protocol for Mobile Ad Hoc Networks", *Proc. of ICC'04*, Paris, June 2004.

[7] J. Mitola, "The Software Radio Architecture", *IEEE Communications Magazine*, May 1995

[8] U.Kozat, I. Koutsopoulos, and L. Tassiulas, "A Framework for Cross-layer Design of Energy-efficient Communication with Qos Provisioning in Multi-hop Wireless Networks", *Proc. of Infocom'04*, Hong Kong, China, March 2004.

[9] S. Krishnamachari, M. VanderSchaar, S. Choi, and X. Xu, "Video Streaming over Wireless LANs: A Cross-Layer Approach", *Proc. IEEE Packet Video'03*, Nantes, France, April 2003.

[10] Y. Shan and A. Zakhor, "Cross Layer Techniques for Adaptive Video Streaming Over Wireless Networks", *International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 2002.

[11] H. Liu and M. El Zarki, "Adaptive source rate control for real-time wireless video transmission", *ACM Mobile Networks and Applications*, Volume 3, Issue 1, June 1998.

[12] M. Conti, G. Maselli, G. Turi, S. Giordano, "Cross-Layering in Mobile Ad Hoc Network Design", *Computer Magazine*, Vol. 37, No. 2, February 2004.

[13] M. B. Pursley and D. J. Taipale, "Error Probabilities for Spread-Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding", *IEEE Transactions on Communications*, No l., COM-35, No. 1, January 1987.

[14] P. Frenger, "Multi-Rate Codes and Multicarrier Modulation for Future Communication System", *PhD thesis*, Chalmers University of Technology, Goteborg, Sweden, 1999.

[15] H. Jegou and C. Guillemot, "Source Multiplexed Codes for Error-prone Channels", *IEEE ICC'03*, May 2003.

[16] T. Guionnet, "Codage robuste par descriptions multiples pour transmission sans fil d'information multimédia", *PhD thesis*, University of Rennes, 2003.

[17] IEEE 802.11 WG, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,", *Standard Specification*, IEEE, 1999.

[18] G. Holland, N. Vaidya and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks", *Mobicom'01*, Rome, Italy, July 2001.

[19] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band", *Bell Labs Technical Journal*, Summer 1997.

[20] D. Qiao, S. Choi et. al. , "MiSer: an optimal low-energy transmission strategy for IEEE 802.11a/h", *Proc. of Mobicom'03*, Pages 161-175, 2003.

[21] Q. Ni, L. Romdhani, T. Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN", *Wireless Communication and Mobile Computing Journal* (JWCMC), 2004.

[22] IEEE 802.11 WG, "Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)", *IEEE 802.11e/Draft* 4.2, February 2003.

[23] M. H. Manshaei, T. Turletti, and M. Krunz, "A media-oriented transmission mode selection in 802.11 wireless LANs", *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, USA, March 2004.

[24] IEEE 802.11 WG, Part 11a, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", High-speed Physical Layer in the 5 GHz Band, *Standard Specification*, IEEE, 1999.

[25] L.A. Larzon, M. Degermark, and S. Pink, "UDP Lite for Real Time Applications", *Technical Report* 1999-01, HP Laboratories Bristol, April 1999.

[26] M.H. Manshaei, T. Turletti, "Simulation-Based Performance Analysis of 802.11a WLANs", *Proc. of IST*, Iran, August 2003.

[27] "The Rice University Monarch Project, Mobile Networking Architectures", http://www.monarch.cs.rice.edu/

[28] "Cisco Aironet 1200 Series Access Point Hardwara Installation Guide", available in http://www.cisco.com

[29] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", in *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[30] T. Guionnet, C. Guillemot,"Soft Decoding and Synchronization of arithmetic Codes: Application to Image Transmission Over Noisy Channels", *IEEE Trans. Image Processing*, Vol.12, No.12, December 2003, pp.1599-1609.

[31] J. Vieron, C. Guillemot,"Low rate FGS video compression based on motion-compensated spatio-temporal wavelet analysis", *Proc. of the SPIE Intl. Conference on Visual Communication and Image Processing*, VCIP'03, July 2003.

[32] I. Kozintsev, J. Chou, K. Ramchandran,"Image Transmission Using Arithmetic Coding Based Continuous Error Detection", *Data Compression Conference*, Snowbird, UT, pp. 339-348, March 1998.

[33] R.Khalili and K. Salamatian,, "A new analytic approach to evaluation of Packet Error Rate in Wireless Networks", *Research Report*, RP-LIP6-2004-10-50, LIP6-CNRS, October 2004.

[34] G. David Forney Jr., "Convolutional codes ii. maximum-likelihood decoding", *Information and Control*, Vol. 25, No. 3, pp. 407-421, 1974.

[35] D.S. Taubman, and M.W. Marcellin, "JPEG2000: Fundamentals, Standards and Practice", *Kluwer Academic Publishers*, Boston, 2002.

[36] D. Qiao, S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation", *Proc. of the IEEE ICC'01*, Finland, June 2001.

# RÉSUMÉ

Fin 2004, un quart des foyers Européens étaient connectés à l'Internet haut débit. Avec le faible coût des machines toujours plus puissantes, de nombreuses applications multimédias ont pu être élaborées pour satisfaire la demande croissante du grand public. Le besoin d'adaptation des protocoles de communication sous-jacents est essentiel pour ce type d'applications. Les protocoles doivent pouvoir passer à l'échelle et s'adapter aux caractéristiques hétérogènes de ces nouvelles applications. Parallèlement, les transmissions sans fil ont connu un essor sans égal, permettant un accès à l'Internet de n'importe quel endroit. La multiplicité des technologies d'accès (GPRS, UMTS, WIFI, WiMAX, Bluetooth, etc.) et la grande variabilité des caractéristiques des canaux de transmission sans fil ont encore accru ce besoin d'adaptation.

Dans ce document d'habilitation, je présente quatre contributions qui mettent en relief le besoin d'adaptation des protocoles de communication. La première concerne un protocole de communication robuste au facteur d'échelle élaboré pour des applications d'environnements virtuels qui mettent en jeu un grand nombre de participants. La seconde décrit un algorithme pour contrôler la transmission de vidéo hiérarchique vers un ensemble hétérogène de récepteurs sur Internet. Les deux contributions suivantes portent sur la transmission sans fil. Je décris un mécanisme de différenciation de services efficace pour transmettre des flots multimédias à débit variable dans les réseaux IEEE 802.11e, ainsi qu'un mécanisme d'adaptation intercouches pour la transmission multimédia dans les réseaux WIFI.



**Mots-clés** : contrôle de transmission multimédia, Environnements Virtuels à Grande Échelle (EVGE), IEEE 802.11, IEEE 802.11e, mécanismes d'interaction intercouches, support de différenciation de services pour réseaux locaux sans fil.



# ABSTRACT

At the end of 2004, a quarter of European households had high speed access to the Internet. Since high speed connectivity with powerful PCs enable multimedia access, a large number of various multimedia applications have been designed to satisfy the general demand of the public. This huge number of potential users and the diversity of characteristics of these applications force the design of adaptive communication protocols. In the same time, multiple wireless access technologies have been developed enabling Internet access anywhere. The diversity of wireless access technologies (GPRS, UMTS, WIFI, WiMAX, Bluetooth, etc.) along with the high variability of wireless channels characteristics have increased further the adaptation needs.

In this manuscript, I describe four contributions that emphasize the need of adaptation for communication protocols. The first one is a scalable communication protocol for large scale virtual environments. The second contribution is an adaptive rate control algorithm for multicast layered video transmission over the Internet. The last two contributions concern wireless networking. I describe an efficient scheduling scheme for VBR multimedia transmission over IEEE 802.11e WLANs and a cross-layer adaptive mechanism to select the physical transmission mode of IEEE 802.11 devices for multimedia flows.



**Keywords** : Congestion Control, Cross-Layer Interaction Mechanisms, QoS support for WLAN, IEEE 802.11, IEEE 802.11e, Large Scale Virtual Environments (LSVE), Transmission Control for Multimedia.