



**HAL**  
open science

# Une architecture de communication pour environnements virtuels distribués à grande-echelle sur l'internet

Emmanuel Léty

► **To cite this version:**

Emmanuel Léty. Une architecture de communication pour environnements virtuels distribués à grande-echelle sur l'internet. Réseaux et télécommunications [cs.NI]. Université de Nice Sophia Antipolis, 2000. Français. NNT: . tel-00406512

**HAL Id: tel-00406512**

**<https://theses.hal.science/tel-00406512>**

Submitted on 22 Jul 2009

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

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

# THÈSE

Présentée à

**L'Université de Nice-Sophia Antipolis**

pour obtenir

le titre de : DOCTEUR EN SCIENCES  
Mention INFORMATIQUE

par

Emmanuel LÉTY

Laboratoire d'Accueil : INRIA - Sophia Antipolis

Projet de Recherche : Rodéo/Planète

École Doctorale : Sciences et Technologies de l'Information et de la Communication

Titre de la thèse :

## **Une Architecture de Communication pour Environnements Virtuels Distribués à Grande-Echelle sur l'Internet**

soutenue le 4 Décembre 2000 devant la commission d'examen

M. Wolfgang	EFFELSBURG	Rapporteurs
M. Eric	HORLAI	
M. François	BACCELLI	Directeurs
M. Thierry	TURLETTI	
M. Daniel	KOFMAN	Examineur
M. Lionel	BOUCHARD	Membre Invité



## Remerciements

Je tiens tout d'abord à remercier M. Wolfgang EFFELSBURG, Professeur à l'Université de Mannheim en Allemagne, et M. Eric HORLAIT, Professeur à l'Université Pierre et Marie Curie (Paris VI), qui m'ont fait l'honneur d'avoir accepté d'être les rapporteurs de ma thèse.

J'exprime ma profonde gratitude à M. François BACCELLI, Directeur de Recherche INRIA et Responsable scientifique du projet ENS/INRIA MCR, qui a été mon directeur de thèse. Je lui suis particulièrement reconnaissant pour les conseils judicieux qu'il a pu m'apporter au cours de cette thèse.

Je voudrais remercier M. Daniel KOFMAN, Maître de Conférence à l'ENST Paris, et M. Lionel BOUCHARD, Responsable du projet MP4NET aux Laboratoires d'Electronique Philips, qui se sont intéressés à mes travaux et ont bien voulu participer à mon jury de thèse.

Mes remerciements les plus vifs vont tout particulièrement à M. Thierry TURLETTI, Chargé de Recherche à l'INRIA-Sophia Antipolis, qui a dirigé mes travaux. La confiance qu'il m'a témoignée, le temps qu'il m'a consacré, et les conseils qu'il m'a donnés tout au long de mes travaux, m'ont largement aidé à mener à bien cette thèse.

Je n'oublie pas M. Walid DABBOUS, Directeur de Recherche à l'INRIA, qui a mis à ma disposition tous les moyens nécessaires en m'accueillant au sein du projet RODEO devenu PLANETE.

J'adresse également mes chaleureux remerciements à M. Antoine CLERGET, ex-Thésard à l'INRIA et actuellement Responsable Recherche et Développement dans la Jeune-Pousse UDCAST, pour ses précieux conseils et sa bonne humeur qui m'ont encouragé durant ces trois années.

Enfin, je tiens à remercier tous ceux qui m'ont soutenu, et plus particulièrement mes parents Monique et Gérard qui m'ont accompagné par la pensée tout au long de cette thèse. J'ai également une pensée toute particulière pour ma grand-mère décédée cette année. Par son exemple, elle a su contribuer à sa façon à l'aboutissement de ce travail que je lui dédie aujourd'hui.

Emmanuel LÉTY



## Résumé

Le modèle IP multipoint actuel rend particulièrement difficile le déploiement à grande-échelle d'Environnements Virtuels Partagés (EVPs) sur l'Internet. Cependant les concepteurs de ce type d'applications désirent permettre la connexion simultanée d'un nombre toujours plus important d'utilisateurs. Nous avons défini SCORE, une architecture de communication permettant d'améliorer la scalabilité et de déployer à grande-échelle des EVPs. Dans cette architecture, nous avons élaboré des mécanismes de filtrage et des protocoles au niveau de la couche transport, s'appuyant sur plusieurs agents et sur un ensemble de groupes multipoints. Afin d'évaluer les coûts introduits par SCORE ainsi que sa capacité à améliorer les performances, nous avons réalisé plusieurs séries d'expérimentations. Nos résultats montrent les situations dans lesquelles notre architecture de communication permet d'offrir un haut degré de scalabilité.

**Mots clés:** Environnements Virtuels distribués à Grande-Echelle, Grands Groupes Multipoint, Découpage en Cellules, Architecture de Communication

## Summary

The current IP multicast model has a lot of imperfections to handle Large-Scale Virtual Environments (LSVE) applications. Today, many of these applications have to handle an increasing number of participants and deal with the difficult problem of scalability. We have designed SCORE, a communication architecture that enables such applications to run on the Internet today. We propose to use a filtering mechanism done at the transport-layer, using multiple multicast groups and agents. We implemented this architecture to show its feasibility, then performed several experimentations to evaluate its cost and its capacity to improve the performance in LSVEs. Our results show the situations, where our communication architecture is able to maintain a high level of scalability.

**Keywords:** Large-Scale Virtual Environments, Multiple Multicast Groups, Cell-based Grouping, Communication Architecture



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Définitions . . . . .	3
1.1.1	Scalabilité . . . . .	3
1.1.2	Consistance . . . . .	4
1.2	Besoins des Environnements Virtuels Distribués . . . . .	4
1.2.1	Scalabilité . . . . .	4
1.2.2	Interactivité . . . . .	5
1.2.3	Consistance . . . . .	6
1.2.4	Fiabilité . . . . .	6
1.3	Approche pragmatique . . . . .	7
1.4	Contributions originales . . . . .	8
<b>2</b>	<b>Etat de l'Art</b>	<b>9</b>
2.1	Applications multipoints . . . . .	9
2.1.1	Modèle . . . . .	9
2.1.2	Routage multipoint inter-domaine . . . . .	10
2.1.3	Allocation d'adresses multipoints . . . . .	14
2.1.4	Vers un nouveau modèle multipoint . . . . .	17
2.1.5	Fiabilité et service de transport . . . . .	18
2.2	Environnements Virtuels . . . . .	21



2.2.1	Architectures de communication . . . . .	22
2.2.2	Filtrage de l'information . . . . .	24
2.2.3	Synchronisation . . . . .	29
<b>3</b>	<b>Problématique</b>	<b>31</b>
3.1	Interactivité et Scalabilité . . . . .	31
3.2	Hétérogénéité . . . . .	32
3.3	Constitution des groupes . . . . .	34
3.4	Limites du multipoint . . . . .	37
3.4.1	Routage . . . . .	38
3.4.2	Latence de l'abonnement . . . . .	39
3.4.3	Taille de session . . . . .	41
3.4.4	Contrôle d'accès . . . . .	42
<b>4</b>	<b>MiMaze3D</b>	<b>45</b>
4.1	Motivations . . . . .	45
4.2	Architecture de MiMaze3D . . . . .	48
4.3	Identification des problèmes de scalabilité . . . . .	49
4.4	Conclusion . . . . .	53
<b>5</b>	<b>Découpage de l'espace virtuel</b>	<b>55</b>
5.1	Principes du découpage en cellules . . . . .	55
5.2	Compromis sur la taille de cellule . . . . .	57
5.3	Modèles et Simulations . . . . .	60
5.3.1	Participants statiques . . . . .	61
5.3.2	Participants dynamiques . . . . .	68
5.4	Conclusion . . . . .	70

<b>6</b>	<b>SCORE: une architecture de communication scalable pour EVGEs</b>	<b>73</b>
6.1	Introduction . . . . .	73
6.2	Métrie de satisfaction . . . . .	74
6.3	Les agents . . . . .	77
6.3.1	Définition des agents . . . . .	77
6.3.2	Association cellules-groupes multipoints . . . . .	79
6.4	Communications dans SCORE . . . . .	87
6.4.1	Une architecture multi-agent doublement distribuée . . . . .	87
6.4.2	Communication participants-participants . . . . .	90
6.4.3	Communication participants-agent . . . . .	92
6.4.4	Communication agent-agent . . . . .	97
6.5	Connexion au monde virtuel . . . . .	98
6.6	Mise à jour des abonnements . . . . .	100
6.6.1	Définition . . . . .	100
6.6.2	Mécanisme de mise à jour des abonnements . . . . .	100
6.7	Algorithme de découpage . . . . .	102
6.8	Conclusion . . . . .	104
<b>7</b>	<b>Expérimentations de SCORE</b>	<b>107</b>
7.1	Développement . . . . .	107
7.1.1	Introduction et paramètres de lancement . . . . .	107
7.1.2	Présentation du cas étudié . . . . .	109
7.1.3	Description du participant . . . . .	110
7.1.4	Hétérogénéité et Dynamicité . . . . .	111
7.2	Environnement expérimental . . . . .	113
7.2.1	Plate-forme d'expérimentation . . . . .	113
7.2.2	Génération et utilisation des traces . . . . .	114
7.3	Les différentes situations étudiées . . . . .	115

7.4	Mise en évidence de l'amélioration de performance liée à SCORE . . . . .	117
7.4.1	Comparaison des satisfactions entre le cas statique et le cas dynamique pour différents niveaux d'hétérogénéité entre les participants . . . . .	117
7.4.2	Comparaison des satisfactions pour deux distributions différentes des ca- pacités . . . . .	122
7.4.3	Comparaison des satisfactions en fonction de la période de redécoupage et de la dynamique . . . . .	125
7.4.4	Comparaison du trafic de données reçu avec et sans SCORE . . . . .	129
7.5	Mesure et estimation du surcoût lié à SCORE . . . . .	131
7.5.1	Comparaison du trafic de signalisation reçu et de contrôle émis . . . . .	131
7.5.2	Comparaison de la fréquence d'abonnements en fonction de la période de redécoupage . . . . .	133
7.6	Utilisation des groupes multipoints dans SCORE . . . . .	135
7.6.1	Nombre de groupes multipoints utilisés par l'agent . . . . .	135
7.6.2	Comparaison du nombre de groupes multipoints souscrits par participant	136
7.6.3	Distribution des participants dans les groupes multipoints . . . . .	140
7.7	Conclusion . . . . .	142
<b>8</b>	<b>Conclusion</b>	<b>143</b>
8.1	Résumé de nos travaux . . . . .	143
8.2	Travaux futurs . . . . .	145

# Table des figures

2.1	Architecture d'Allocation d'Adresses Multipoints (MAAA) . . . . .	15
2.2	Architecture Clients-Serveur . . . . .	22
2.3	Architecture Distribuée Point-à-Point . . . . .	23
2.4	Architecture Distribuée Multipoint . . . . .	23
2.5	Découpage en Cellules dans le DIS . . . . .	25
4.1	MiMaze3D . . . . .	47
4.2	L'architecture de MiMaze3D . . . . .	48
4.3	Qualité perçue par un participant type . . . . .	51
4.4	Temps nécessaire au traitement des dernières positions reçues . . . . .	52
4.5	Pourcentage de trafic utile reçu . . . . .	53
5.1	Différentes Tailles de cellule . . . . .	57
5.2	Nombre Moyen de Groupes abonnés par Participant . . . . .	64
5.3	Pourcentage de Trafic Superflu . . . . .	65
5.4	Trafic Moyen Entrant (en nombre de sources) par Participant . . . . .	66
5.5	Trafic Maximal Entrant (en nombre de sources) / Participant avec $p = 0.95$ . . . . .	66
5.6	Temps de séjour moyen par cellule . . . . .	70
6.1	Découpage Statique de l'EV en Start-Zones . . . . .	80
6.2	Découpage Statique des Start-Zones en Zone-Units . . . . .	80

6.3	Découpage Dynamique des Start-Zones en Zones (Ex.1)	82
6.4	Découpage Dynamique des Start-Zones en Zones (Ex.2)	83
6.5	Découpage Dynamique des Zones en Cellules	83
6.6	Différentes tailles de cellule en fonction de la densité de participants	85
6.7	Une Architecture Multi-Agent Doublement Distribuée	88
7.1	Exemples de découpage en Zones avec 9 Start-Zones et 4 Zone-Units/Start-Zone	109
7.2	Enchaînement des étapes au niveau du participant	111
7.3	Interface graphique de lancement des participants en mode sans réception	114
7.4	Les différentes tailles de zone d'intérêt expérimentées	116
7.5	Comparaison des Satisfactions avec $V = 100$ , $PR = 1s$ et $I\text{Area} \geq 0.25$	120
7.6	Comparaison des Satisfactions avec $V = 100$ , $PR = 1s$ et $I\text{Area} \leq 0.16$	121
7.7	Comparaison des satisfactions pour deux distributions différentes des capacités	124
7.8	Comparaison des satisfactions pour $PR = 1s$ ou $6s$ , $V = 10$ ou $100$ et avec $I\text{Area} \geq 0.49$	126
7.9	Comparaison des satisfactions pour $PR = 1s$ ou $6s$ , $V = 10$ ou $100$ et avec $0.09 \leq I\text{Area} \leq 0.36$	128
7.10	Trafic de données entrant (Nb Sources/s) par participant ( $V = 10$ )	129
7.11	Trafic de signalisation reçu et de contrôle émis (en paquets/s) avec $V = 10$	130
7.12	Trafic de signalisation reçu et de contrôle émis avec $PR = 6s$	132
7.13	Fréquence d'abonnements avec $V = 10$	133
7.14	Fréquence d'abonnement avec $PR = 6s$	134
7.15	Nombre de groupes multipoints utilisé par l'agent avec $C \in [20, 40]$	135
7.16	Evolution du nombre de groupes souscrits par participant avec $V = 10$ et $I\text{Area} \geq$ $0.25$	138
7.17	Evolution du nombre de groupes souscrits par participant avec $V = 10$ et $I\text{Area} \leq$ $0.16$	139
7.18	Distribution des participants dans les groupes multipoints $V = 10$ et $PR = 1s$	141

# Chapitre 1

## Introduction

Au croisement de la réalité virtuelle et des réseaux de communication, les Environnements Virtuels Partagés (**EVP**) soulèvent un intérêt sans cesse croissant depuis ces dernières années. En effet, l'émergence de nouvelles technologies, aussi bien dans le domaine de l'image de synthèse, de la robotique ou des réseaux, ont rendu possible l'arrivée de cette nouvelle forme d'application.

Le développement rapide de la puissance de calcul des micro-ordinateurs, l'apparition sur le marché de cartes graphiques toujours plus performantes, ainsi que le développement et la standardisation de bibliothèques graphiques de haut niveau (OpenGL, Direct3D, VRML), ont également permis le déploiement de nombreuses applications "grand-public" de réalité virtuelle. Parmi celles-ci, nous pouvons citer de nombreuses applications multimédia, où l'utilisateur est plongé dans un monde virtuel tantôt imaginaire (ex : jeux vidéo), tantôt reproduction en trois dimensions d'un lieu existant (ex : musées, monuments historiques, zones touristiques, etc..).

Depuis déjà quelques années, le nombre d'ordinateurs connectés à l'Internet croît exponentiellement. Auparavant les applications de réalité virtuelle étaient destinées à un simple et unique utilisateur, mais l'avènement de l'Internet de part le monde a rendu aujourd'hui possible la création de mondes virtuels partagés. Un nouveau besoin est né : celui de se regrouper en com-

munauté virtuelle, afin de tirer partie de tous les bénéfices possibles de cette nouvelle avancée technologique. Les environnements virtuels partagés représentent donc un nouveau type d'application dont la vocation est de répondre à ces nouveaux besoins de regroupement virtuel. Dans ces environnements virtuels, les participants peuvent interagir, s'affranchissant ainsi des distances physiques qui les séparent dans le monde réel. Le réseau représente donc une ressource de toute première importance pour les EVPs. En effet, c'est sur celui-ci que vont s'échanger les différents messages et informations de mises à jour entre participants.

Les environnements virtuels sont, par nature, des applications interactives multi-utilisateur présentant de fortes contraintes temporelles. Une architecture distribuée, sans serveur, où chaque participant transmet ses informations en multipoint, permet de résoudre les problèmes de "scalabilité" liés au traitement centralisé et à la convergence du trafic vers un noeud du réseau. Cependant les temps de traitement de chaque participant et l'utilisation de la bande passante du réseau pour l'acheminement des paquets doivent être contrôlés afin de satisfaire les besoins d'interaction temps-réel de ce type d'application.

Au cours de cette thèse, nous allons nous pencher sur les applications distribuées interactives où un grand nombre de participants sont impliqués (quelques milliers voire dizaines de milliers). Or, pour un très grand nombre de participants et lorsque l'ensemble des informations transmises sur le réseau transite vers l'ensemble des participants, le trafic reçu par chaque participant devient très important. Ceci rend difficile le traitement temps-réel de l'intégralité des paquets reçus et augmente le risque de congestion sur la liaison réseau reliant le participant à l'Internet (souvent de faible capacité dans le cas de l'utilisation d'un modem).

Il est à noter également que chaque participant interagit à un instant donné avec un nombre limité d'autres participants (une dizaine tout au plus). Par conséquent, une grande partie de l'information qui lui parvient ne lui est d'aucune utilité. Cette information inutile entraîne un

accroissement du temps de traitement et de la demande en CPU, ce qui peut affecter les performances de l'application. De plus, cette information inutile transite sur le réseau et représente un coût en terme de bande passante. Le filtrage de l'information apparaît donc indispensable dans la résolution de ce problème. Pour cela, l'architecture de communication de ce type d'application doit être modifiée afin de supporter de nouveaux protocoles et mécanismes plus adaptés à la gestion des grands groupes. Il est nécessaire d'étudier des politiques de gestion des centres d'intérêt, de regroupement dynamique de l'information au sein de groupes multipoints et de filtrage de l'information.

## 1.1 Définitions

Afin de faciliter la lecture de cette thèse, nous utiliserons désormais l'abréviation française **EVGE** pour désigner les Environnements Virtuels Partagés (EVPs) distribués à Grande-Echelle sur l'Internet. Ce type d'application est cependant le plus souvent désigné par l'abréviation anglo-saxonne **LSVEs** (Large-Scale Virtual Environments) dans le monde de la recherche internationale.

### 1.1.1 Scalabilité

Le terme *scalabilité* dérive du verbe anglais "to scale" qui signifie "mettre à l'échelle". On considère qu'une architecture de communication pour EVGE est *scalable*, si elle rend possible le déploiement sur l'Internet d'EVPs pouvant accueillir un nombre arbitrairement grand de participants, sans impliquer pour autant d'importantes modifications dans sa structure et dans les mécanismes qu'elle met en œuvre. Le problème de la scalabilité dans les EVGEs revient donc à définir une architecture de communication sous-jacente, permettant une mise à l'échelle efficace, lorsque le nombre de participants varie de plusieurs ordres de grandeur (d'une dizaine à plusieurs milliers de participants). Cette efficacité se traduit par l'ordre de complexité des algorithmes utilisés, et par une consommation constante ou tout au plus linéaire des ressources au niveau du réseau et des machines hôtes des utilisateurs lorsque le nombre de participants augmente (nombre



de messages de contrôle échangés, occupation de la mémoire, charge du CPU, etc..).

### 1.1.2 Consistance

La *consistance*, ou encore la *cohérence* au sein d'un EVP, représente le degré de similitude sur un axe spatial et à temps fixe des actions affichées par les participants. Par exemple, si une entité E effectue une action qui modifie l'état de la session des participants P1 et P2, alors la consistance veut que les participants P1 et P2 affichent le résultat de l'action de l'entité E au même instant. Si  $t_1$  et  $t_2$  désignent respectivement les instants réels (sur une horloge absolue) d'affichage de cette même action au niveau des participants P1 et P2, alors la différence entre ces deux temps permettra de donner une estimation sur le degré de consistance de l'application.

## 1.2 Besoins des Environnements Virtuels Distribués

Ce chapitre introduit les différentes caractéristiques et besoins des environnements virtuels sur Internet.

### 1.2.1 Scalabilité

Dans les EVPs, chaque utilisateur est à la fois receveur et émetteur d'information. On fait communément référence à ce type d'applications en les qualifiant de "plusieurs-vers-plusieurs", par opposition aux applications "un-vers-plusieurs" qui ne contiennent qu'une seule source et plusieurs destinataires (ex: diffusion TV). Cependant la complexité des applications "plusieurs-vers-plusieurs" est sans cesse croissante. En effet, les concepteurs d'EVPs désirent permettre la connexion simultanée d'un nombre toujours plus grand d'utilisateurs. De plus, chacun d'entre eux est bien souvent la source de plusieurs flots de données de contenus et de types différents. Toutefois, chaque participant n'interagit à tout instant qu'avec un nombre limité de participants et n'est donc intéressé de recevoir que le trafic associé à ces derniers. Lorsque le nombre de participants augmente, le pourcentage d'information que désire recevoir un participant diminue (si on considère la totalité d'information émise par l'ensemble des participants).

Parvenir à transmettre à chaque utilisateur uniquement les informations qui l'intéresse, permettrait le déploiement à grande-échelle des environnements virtuels partagés. La conception et l'élaboration d'une telle architecture de communication reste encore un domaine actif de la recherche [BBP98][JMC99]. En l'absence d'une architecture de communication satisfaisant de telles conditions, les EVPs restent jusqu'à aujourd'hui limités par le nombre d'utilisateurs qu'ils peuvent simultanément accueillir. En effet, la transmission de tous les paquets de données émis par chaque participant vers tous les autres participants, peut avoir des conséquences désastreuses à plusieurs niveaux, se répercutant de fait sur la qualité perçue par les utilisateurs. A l'intérieur du réseau, cela peut entraîner une saturation des tampons mémoires au niveau des routeurs et une intensification des congestions sur les liens de transmission. Au niveau des machines hôtes, les effets peuvent se traduire par un débordement des files d'attente et une utilisation inutile des ressources CPU. De plus, l'utilisation d'un protocole de transmission fiable en point à point entre les utilisateurs (ex: TCP) ne peut qu'amplifier le phénomène en cas de pertes de paquets, en retransmettant des données bien souvent inutiles.

### 1.2.2 Interactivité

Un EVP 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 voir en temps-réel chacune de leurs actions. Cependant, la technologie actuelle ne permet pas de traiter l'information et de la communiquer instantanément. Dans le meilleur des cas, les réseaux de fibres optiques atteignent 2/3 de la vitesse de la lumière, ce qui donne par exemple un délai minimum de 140 millisecondes entre l'Europe et l'Australie. Ce délai 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 exemple de gêne occasionnée par un délai trop long est observable dans les téléphones par satellite où le délai de propagation est de 250 millisecondes. Le standard **DIS**(Distributed Interactive Simulation) [Soc95] recommande de fixer ce délai d'interaction au maximum de 150

millisecondes. D'après les sociétés qui commercialisent des jeux distribués sur réseau, la qualité du jeu commence à se dégrader quand ce délai dépasse les 200 millisecondes. Pour d'autres types d'EVPs que les jeux en ligne, le besoin d'interactivité peut être cependant encore plus faible. En conséquence, des mécanismes de contrôle de transmission adaptés aux différents besoins d'interactivité doivent être mis en œuvre. Ceux-ci ont pour rôle de masquer les effets du réseau sur l'application afin de préserver l'interactivité.

### **1.2.3 Consistance**

En tant qu'applications interactives distribuées, les EVPs doivent fournir à chaque participant une perception consistante du monde virtuel à tout instant. Les besoins en terme de consistance pour un EVP donné, sont de plus souvent liés à ses besoins en terme d'interactivité. Cependant, l'Internet est un réseau hétérogène non fiable, et chaque participant va subir de façon variable, inégale et imprévisible les effets du réseau (délais, gigue et pertes de paquets). Ainsi, chaque participant reçoit un sous-ensemble différent des paquets transmis sur le réseau, chaque paquet étant lui-même reçu avec des délais différents. Une autre conséquence des délais variables du réseau entre participants se traduit par le fait que l'ordre de réception des paquets au niveau d'un participant ne correspond pas toujours à l'ordre d'émission de ces mêmes paquets. Or certains événements doivent être transmis de manière ordonnée, afin de respecter un ordre causal [BMR96].

### **1.2.4 Fiabilité**

En raison des différences de nature des données transmises entre participants d'un EVP, différents protocoles de transport peuvent être utilisés. Certaines données peuvent être transmises avec des débits variables, s'adaptant aux ressources disponibles à un instant donné dans le réseau (contrôle de congestion) ou au niveau des récepteurs (contrôle de flux). Les messages audio ou vidéo transmis par les participants en sont un bon exemple. Cependant, les EVPs, de part leur complexité, ont presque toujours besoin de permettre une communication fiable sur certains types

de données. Dans les mondes virtuels, c'est notamment le cas au cours d'interactions critiques, souvent brèves, lors de collisions par exemple. L'échange de messages textuels entre participants, via l'utilisation d'un tableau blanc partagé par exemple, ne peut également se faire que si une fiabilité quasi totale est garantie. Cependant, Internet est un réseau hétérogène et n'offre aucune garantie sur la qualité des transmissions dans son protocole de base IP (Internet Protocol). Il n'y a donc aucune garantie sur le fait que les messages seront délivrés aux participants (et encore moins dans l'ordre), et aucun engagement concernant les délais de transmission. De plus, la caractéristique temps-réel des EVPs interdit la plupart du temps l'utilisation d'un protocole de retransmission au niveau de la couche transport. En effet, les échanges d'acquittements et les délais de retransmission rendent le plus souvent obsolète l'information retransmise.

### 1.3 Approche pragmatique

Afin d'identifier les problèmes de scalabilité rencontrés dans les EVGEs et de proposer des solutions adéquates, nous avons choisi de procéder par étapes. La première de ces étapes consiste à partir d'un exemple concret d'environnement virtuel partagé, et d'observer le comportement de l'application lorsque le nombre de participants augmente. Une telle approche permet tout d'abord de démontrer l'existence d'un réel problème de scalabilité lorsque qu'aucune technique de filtrage d'information n'est mise en œuvre. Elle permet également, de part son caractère pragmatique et expérimental, de se sensibiliser au problème de la scalabilité, d'identifier ses composantes, et de mesurer son impact sur les performances de l'application au niveau de l'utilisateur final. Dans une seconde étape, une réflexion peut se dégager au regard des résultats obtenus, et des solutions peuvent être proposées et simulées. Puis, il convient de démontrer la faisabilité de ces solutions en les implémentant et en les intégrant dans une architecture de communication pour EVGEs. Enfin, l'objectif final ne peut être atteint qu'après avoir procédé à leur évaluation, toujours dans un cadre expérimental, afin d'analyser leur efficacité ainsi que leurs limites.

## 1.4 Contributions originales

Les contributions de nos travaux concernent la caractérisation et l'étude du point de vue de la communication, de la scalabilité dans les environnements virtuels distribués à grande-échelle sur l'Internet. Nous avons caractérisé l'ensemble de la problématique liée à la mise en réseau à grande-échelle d'applications interactives de réalité virtuelle. Nous avons réalisé et évalué MiMaze3D, un jeu 3D multi-utilisateurs basé sur l'utilisation d'un unique groupe multipoint, afin d'identifier les problèmes de scalabilité rencontrés dans les EVGEs. A partir de cette première étude, nous proposons une méthode de découpage de l'espace virtuel en cellules de taille identique. Nous caractérisons les paramètres impliqués dans la détermination d'une taille de cellule appropriée. Nous décrivons une méthode basée sur la théorie des processus de points dans le plan, pour déterminer une taille de cellule de manière à limiter le trafic reçu par participant ainsi que le nombre de franchissements de cellules par unité de temps. Au vu de ces résultats, nous avons défini et implémenté SCORE, une nouvelle architecture de communication permettant d'améliorer la scalabilité et de déployer à grande-échelles des EVPs. Dans cette architecture, nous avons élaboré des mécanismes et des protocoles s'appuyant sur plusieurs agents et sur un ensemble de groupes multipoints. Afin d'évaluer les coûts introduits par SCORE ainsi que sa capacité à améliorer les performances dans les EVGEs, nous avons réalisé plusieurs séries d'expérimentations. Nos résultats montrent les situations dans lesquelles notre architecture de communication permet d'offrir un haut degré de scalabilité aux EVGEs.

# Chapitre 2

## Etat de l'Art

### 2.1 Applications multipoints

#### 2.1.1 Modèle

L'objectif du routage multipoint est de permettre un service efficace de communications pour les applications désireuses d'émettre des données identiques vers de multiples destinataires, et ceci sans créer une surcharge au sein du réseau. Par conséquent au niveau de chaque routeur une seule copie d'un paquet multipoint entrant est transmise sur le lien de sortie, au lieu d'avoir autant de copies que de récepteurs accessibles par ce même lien. Le modèle IP multipoint fut proposé pour la première fois en 1991 dans la thèse de Steeve Deering [Dee91]. Dans ce modèle, IP multipoint est associé avec la notion de **groupe**, identifié par une adresse IP de classe D et composé d'un ensemble de participants. Dans les réseaux IP, on définit comme **membre** d'un groupe, un participant désirant recevoir les données émises sur ce groupe et ayant préalablement effectué une procédure d'abonnement pour ce groupe. Par conséquent une source n'a pas besoin d'être membre du groupe, et donc de s'abonner, afin d'émettre sur celui-ci et d'atteindre ainsi tous les membres du groupe. Il est également à noter que durant une session multipoint, de nouveaux membres peuvent s'abonner tandis que d'anciens membres peuvent au contraire se désabonner. Plusieurs mécanismes sont donc nécessaires afin de rendre possible le déploiement

d'IP multipoint :

- Des protocoles de routage du trafic multipoint, déployés au sein du réseau, entre les différents routeurs supportant le multipoint.
- Une gestion des abonnements et des désabonnements. Celle-ci se fait par l'intermédiaire du protocole IGMP (Internet Group Management Protocol), entre les machines hôtes et les routeurs présents dans leur voisinage immédiat.
- Un mécanisme d'allocation d'adresses multipoint

### 2.1.2 Routage multipoint inter-domaine

Actuellement, l'Internet est constitué d'une interconnexion de réseaux, administrés par divers organismes. Un ensemble de réseaux dépendant d'un même organisme, est appelé **domaine** ou **système autonome**. Cependant, le **MBone** (Multicast Backbone), le réseau multipoint virtuel de l'Internet, s'est déployé depuis une dizaine d'années sans tenir compte de la hiérarchie existante entre ces domaines et des différentes politiques de routage du trafic entre ceux-ci. Les environnements virtuels partagés, en tant qu'applications "plusieurs-vers-plusieurs", illustrent tout particulièrement les problèmes qui peuvent découler de la topologie actuelle du MBone. En effet, les participants sont potentiellement répartis dans les différents domaines de l'Internet. Les groupes multipoints auxquels ils s'abonnent et se désabonnent dynamiquement ont, par conséquent, un impact direct sur le routage du trafic entre les différents systèmes autonomes de l'Internet. Le MBone est donc confronté à un double problème, à la fois de scalabilité du routage inter-domaine et de gestion des diverses politiques de routage intra-domaine.

Au regard des différents protocoles de routage multipoint déployés sur l'Internet, il est intéressant de constater qu'un grand nombre d'entre eux nécessite une diffusion en broadcast de certains types d'information. Les protocoles de routage multipoint **DVMRP** (Distance Vector Multicast Routing Protocol) [Pus98] et **PIM-DM** (Protocol Independent Multicast - Dense Mode) inondent périodiquement le réseau de paquets de données. **MOSPF** (Multicast Extensions to

Open Shortest Path First) [Moy94] transmet à tous les routeurs les informations concernant les abonnements aux groupes multipoint, afin qu'ils puissent construire les arbres de diffusion multipoint associés. Des protocoles comme **CBT** (Core Based Tree) [BFC93] et **PIM-SM** (Protocol Independent Multicast - Sparse Mode) [DEF<sup>+</sup>96] sont plus adaptés aux grands-domaines car chaque membre du groupe doit envoyer un message explicite d'abonnement à un routeur jouant le rôle de point de rendez-vous (appelé "Core Router" ou "Rendez-Vous Point"). Toutefois, le mécanisme d'association des groupes multipoints avec leur point de rendez-vous n'est pas trivial. De plus, les tables de routage et de transfert présentes au sein des routeurs ne sont pas de capacité infinie. Le nombre croissant de groupes multipoints dont les arbres se répandent sur tout l'Internet, peut rapidement aboutir à des tables de routage de plus en plus grandes si aucun nouveau mécanisme ne permet l'établissement d'un routage hiérarchique du multipoint, comme c'est déjà le cas pour le point-à-point.

Pour permettre la communication point-à-point de paquets entre les domaines de l'Internet tout en prenant en compte les différentes politiques de routage et les accords définis entre eux, le protocole **BGP** (Border Gateway Protocol) [LR91] est utilisé au sein des **routeurs-frontières** de chaque domaine. Les routeurs-frontières des domaines limitrophes s'échangent des messages de mise à jour via une connexion TCP, afin de calculer des chemins de bout-en-bout de type vecteurs de distance entre les domaines. Une solution à court terme concernant l'évolution du multipoint inter-domaine, réside en partie dans l'extension de BGP afin de transporter différents types de routes, dont le multipoint. Grâce à cette nouvelle version, appelée **MBGP** (Multiprotocol Extensions to BGP4) [Ra95][BCKR98], les routeurs multipoints n'auront plus à connaître l'entière topologie du Mbone, mais uniquement la topologie dans leur domaine et le chemin par lequel atteindre les autres domaines.

MBGP répond donc en partie au problème du routage multipoint inter-domaine, dont les EVPs ont si grand besoin (bien que celui-ci soit totalement transparent au niveau de la machine hôte).



Cependant, bien que MBGP permette de déterminer par quel routeur et par quel domaine un paquet doit être routé afin d'atteindre les autres participants du groupe, ce protocole ne participe aucunement à la construction des arbres multipoints inter-domaine. Deux protocoles permettent actuellement de répondre à ce besoin: **MSDP** (Multicast Source Discovery Protocol) [FRL<sup>+</sup>00] et **BGMP** (Border Gateway Multicast Protocol) [TEM98]. Ces deux protocoles viennent en complément de PIM-SM. En effet, PIM-SM (Sparse Mode) est actuellement le seul protocole utilisé pour le routage multipoint destiné aux groupes dont les sources sont éparpillées.

Avec MSDP, un groupe multipoint peut avoir plusieurs "Points de Rendez-vous" (**PR**), en disposant d'un PR par domaine impliqué. Chaque domaine possédant un PR permet donc l'établissement d'un arbre de routage partagé par l'ensemble des récepteurs présents dans le domaine pour ce groupe. A l'intérieur d'un domaine, chaque source émet ses paquets vers le groupe en encapsulant le paquet multipoint dans un paquet point-à-point en direction du PR. Dans PIM-SM, chaque récepteur envoie un message d'abonnement au PR, tandis que les sources lui envoient un message d'enregistrement (PIM-SM classique). Ainsi, le PR peut à tout moment envoyer un message de contrôle en direction d'une des sources, afin d'établir une nouvelle branche sur l'arbre multipoint, et permettre ainsi à la source d'émettre directement en multipoint, supprimant ainsi le surcoût lié à l'encapsulation.

Un problème important subsiste alors puisque le PR d'un domaine n'a aucune information concernant les sources du groupe présentes dans les autres domaines. MSDP vient résoudre ce problème délicat en permettant l'établissement d'une connexion TCP entre les PRs. Lorsqu'une source s'enregistre auprès de son PR, ce dernier transmet aussitôt à tous les PRs voisins un message **SA** (Source Active) contenant l'adresse IP de la nouvelle source. Les PRs voisins recevant un message SA, retransmettent à leur tour ce même message à leurs voisins, et ainsi de suite. Dans chaque domaine où se trouvent des récepteurs pour le groupe multipoint, le PR associé au domaine pour ce groupe, envoie un message de contrôle vers la nouvelle source afin

d'établir une nouvelle liaison multipoint avec la source.

MSDP est une solution qui peut être déployée à court terme. De plus, elle satisfait certaines des attentes des **FAI** (Fournisseur d'Accès Internet) jusque-là réticent à laisser passer le trafic multipoint, en rendant ce trafic plus facilement administrable entre les différents systèmes autonomes de l'Internet.

Cependant, MSDP présente d'importantes faiblesses en terme de scalabilité et paraît particulièrement mal adapté aux EVPs. En effet, la mise en place des arbres multipoints inter-domaine par MSDP introduit d'importants délais et représente un coût non négligeable en terme de messages de contrôle échangés. Avant qu'une nouvelle source puisse atteindre les membres du groupe présents dans d'autres domaines, de nombreux paquets de données seront émis sans pour autant parvenir aux membres "éloignés". Dans les EVPs, l'utilisation de plusieurs groupes multipoints est aujourd'hui nécessaire afin de regrouper les participants qui interagissent entre eux (voir Section 2.2.2.1). Les EVPs sont des applications dynamiques et chaque participant est donc potentiellement amené à s'abonner et se désabonner fréquemment à plusieurs groupes multipoints. Cette caractéristique risque d'aggraver les problèmes de scalabilité de MSDP en augmentant sensiblement le surcoût généré par le protocole, tout en réduisant l'interactivité de l'application. Enfin, MSDP qui est un protocole de routage, se place pourtant au dessus de TCP, qui lui est un protocole de transport, violant par la-même le modèle des couches IP. MSDP prend ainsi des allures de solutions ad-hoc, visant plutôt une réponse à court terme des problèmes de routage multipoint inter-domaine.

Une deuxième solution à été proposée afin de permettre la construction d'arbres multipoints inter-domaine: BGMP. Cette fois-ci, le protocole préconise la construction d'arbres multipoints bidirectionnels partagés entre les domaines, mais en n'utilisant qu'un unique PR par groupe. BGMP doit donc permettre "l'élection" d'un PR parmi l'ensemble des PRs présents dans les différentes zones autonomes de l'Internet. Pour se faire, BGMP se base sur l'existence a priori

d'un mécanisme d'allocation d'adresses multipoints entre les domaines. Ainsi, le PR sélectionné appartiendra au domaine qui détient l'adresse du groupe. Cette proposition est justifiée par le fait que le domaine qui détient l'adresse d'un groupe est la plupart du temps le domaine dans lequel se trouve le plus de sources. C'est le cas par exemple pour les applications de vidéo conférence, où il est effectivement plus judicieux de placer la racine de l'arbre multipoint le plus près possible de la source principale. De cette façon, les temps de communication sont réduits entre la source et ses destinataires. Concernant les applications "plusieurs-vers-plusieurs" où chaque participant est à la fois source et destinataire, l'argument précédent n'a plus vraiment de sens. En revanche, BGMP évite l'envoi de messages de contrôle entre plusieurs PRs, et apparaît par conséquent comme une solution plus scalable que MSDP. De plus, en s'appuyant sur des mécanismes d'allocation d'adresses multipoints, BGMP soulève un sérieux problème qui existe encore dans l'Internet: Comment permettre à un EVP de réserver dynamiquement des plages d'adresses multipoints sans entrer en conflit avec des adresses multipoints déjà utilisées par d'autres applications (EVPs ou autres)?

### **2.1.3 Allocation d'adresses multipoints**

Aujourd'hui, en l'absence d'IPv6, chaque groupe multipoint est désigné par une adresse IPv4 de classe D, dont l'espace d'adressage est limité à 28 bits. Il existe donc un peu plus de 268 millions d'adresses multipoints disponibles. Si très peu d'adresses multipoints sont utilisées à un instant donné, et si les groupes multipoints s'étendent pour la plupart sur un seul domaine, une allocation aléatoire d'adresses multipoints est suffisante. En revanche, si on suppose un déploiement massif d'applications "plusieurs-vers-plusieurs" à grande-échelle, et les EVPs en sont un bon exemple, la probabilité de conflit dans les allocations d'adresses multipoints peut devenir réellement problématique. Plusieurs mécanismes permettant de résoudre les problèmes d'allocation d'adresses multipoints ont été proposés. On peut cependant les séparer en deux familles: Les mécanismes venant s'ajouter au dessus du modèle multipoint actuel [Dee91], et les propositions récentes beaucoup plus radicales, nécessitant une véritable réforme du modèle actuel. Dans tous

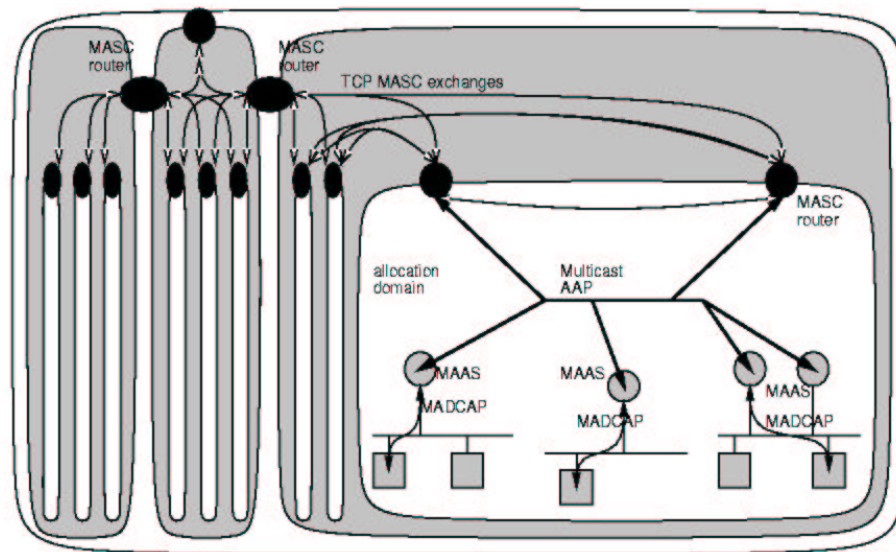


FIG. 2.1 – Architecture d'Allocation d'Adresses Multipoints (MAAA)

les cas, un mécanisme d'allocation d'adresses multipoints a une double mission à remplir : éviter les conflits d'adressage bien sûr, mais également désigner d'une certaine manière le domaine de l'Internet qui "détient" une adresse multipoint, comme responsable du groupe associé.

Les propositions basées sur le modèle multipoint actuel, sont regroupées dans une architecture unique, **MAAA** (Multicast Address Allocation Architecture) proposée dans [KRT<sup>+</sup>98] en association avec le protocole BGMP. Cette architecture peut être décomposée en 3 parties. Chacune de ces parties représente un niveau hiérarchique au sein de l'architecture, avec un protocole qui lui est associé (voir la Figure 2.1<sup>1</sup>) : **MASC** (Multicast Address-Set Claim), **AAP** (Address Allocation Protocol) et **MADCAP** (Multicast Address Dynamic Client Allocation Protocol) [HPS99].

MADCAP est utilisé par un "client multipoint" (typiquement une machine hôte de l'Internet), afin de formuler une requête en vue d'obtenir des adresses multipoints. Cette requête est émise

1. Figure réalisée par les auteurs de MAAA.

vers un serveur d'allocation d'adresses multipoints appelé **MAAS** (Multicast Address Allocation Server), présent dans le même domaine. **AAP** est utilisé par les serveurs **MAAS** d'un même domaine (donc géré par une même administration) afin de s'informer mutuellement des ensembles d'adresses multipoints (appartenant à des plages prédéfinies) demandées et utilisées. Enfin le protocole **MASC** est utilisé par les routeurs situés aux frontières des domaines afin de réaliser une allocation d'adresses multipoints inter-domaine. Les adresses multipoints allouées par les serveurs **MAAS** font par conséquent partie de l'ensemble des adresses allouées à un niveau plus haut (au niveau du domaine) par **MASC**.

**AAP** est également utilisé par les routeurs **MASC** (border gateway) qui annoncent périodiquement à leur domaine les ensembles d'adresses multipoints disponibles pour tout le domaine. Les paquets **ASA** (Address-Set-Announce) sont envoyés par les routeurs **MASC** vers les serveurs **MAAS**. Les paquets **ACLM** (Address-CLaiM) sont envoyés en multipoint par les **MAAS** dans un même domaine afin de faire la demande d'une liste d'adresses (pas encore utilisées au moment de la demande) pour une période de temps définie par une date de début et de fin d'allocation. Après avoir effectué cette demande, le serveur **MAAS** attend quelques secondes pour s'assurer qu'il n'y a pas de problèmes (adresse déjà utilisée ou ne faisant plus partie de l'ensemble d'adresses allouées par les routeurs **MASC**). Régulièrement les serveurs **MAAS** s'échangent en multipoint des paquets **AIU** (Address-In-Use) contenant les adresses en cours d'utilisation, et envoient des rapports aux routeurs **MASC** du domaine à l'aide de paquets **ASRP** (Address-Space-Report).

Un autre mécanisme, **GLOB** [ML], plus simple que **MASC**, consiste à allouer statiquement aux domaines des plages d'adresses multipoints. Le numéro du domaine est alors encodé dans une partie de l'adresse. Toutefois, ce mécanisme ne permet à chaque domaine de ne disposer que d'un nombre très limité d'adresses multipoints. De plus, il ne résout en rien les problèmes d'allocation intra-domaine. Les protocoles **AAP** et **MADCAP** peuvent donc être encore utilisés.

#### 2.1.4 Vers un nouveau modèle multipoint

Face à un modèle multipoint de plus en plus complexe et alourdi par les nombreux protocoles inter-domaine, de nombreux membres de la communauté multipoint de l'**IETF** (Internet Engineering Task Force) ont commencé à dénoncer les limitations du modèle multipoint actuel. Deux changements majeurs ont donc été proposés: **EXPRESS multicast** (EXPLICITLY REquested Single-Source) [HC99] et **Simple Multicast** [PLB<sup>+</sup>98], tout deux regroupés sous le titre de **RAMA** (Root Addresses Multicast Architecture). Nous nous intéresserons ici plus particulièrement à Simple Multicast car Express Multicast a été essentiellement conçu pour les applications ayant une source unique, ce qui n'est évidemment pas le cas des EVPs. Soulignons toutefois les caractéristiques très attractive d'Express Multicast qui permet, entre autres, la tarification et le contrôle d'accès au sein de groupes multipoints.

Dans Simple Multicast, l'idée principale consiste à identifier un groupe multipoint non plus uniquement par son adresse de classe D (1110x) mais par le couple adresse multipoint M - adresse C du "core" (ou Point de Rendez-vous) de l'arbre multipoint. L'autre idée, qui est indépendante de la première, est de construire un arbre bidirectionnel comme dans CBT ou BGMP. Afin de satisfaire la première idée, les paquets de données doivent transporter C et M, ce qui peut être fait de deux manières différentes: Soit en définissant une nouvelle option dans l'en-tête IP qui contiendrait C et M, soit en définissant un nouveau protocole et en le spécifiant dans le champ "protocole" de l'en-tête IP. Dans ce dernier cas, la charge utile est encapsulée dans le nouveau protocole contenant C et M dans l'en-tête. Dans le premier cas, l'adresse de destination contiendrait l'adresse du prochain routeur Simple Multicast, permettant ainsi le "tunneling" des paquets sur l'Internet existant, et le déploiement progressif de Simple Multicast. Le principale avantage de Simple Multicast est le mécanisme trivial d'allocation d'adresses multipoints. En effet, chaque "core" peut allouer l'ensemble des 28 bits d'adresses multipoints sans avoir à se coordonner avec les autres routeurs de l'Internet (contrairement à MASC).

### 2.1.5 Fiabilité et service de transport

On peut distinguer trois niveaux de fiabilité pour un envoi en multipoint :

- L’envoi *ordinaire*, sans garanties.
- L’envoi en *recommandé*, où la couche transport assure la retransmission des données manquantes aux destinataires qui le demandent.
- L’envoi en *recommandé avec accusé de réception*, où la couche transport s’assure que les participants ont bien reçu les données et en informe ensuite l’application. Il doit exister dans le groupe, de manière centralisée ou distribuée, une “conscience” des membres du groupe.

Considérons par exemple l’EVP constitué d’une grande foire virtuelle dans laquelle s’échangent, se vendent ou s’achètent des articles. Suite à la mise en vente d’un produit, un ensemble d’acheteurs parmi les participants présents entrent dans une phase d’enchères. La présentation audio ou vidéo du produit, ou le déplacement des avatars dans l’environnement virtuel est transmise sans fiabilité (envoi “ordinaire”). Lorsqu’une enchère est émise, elle est envoyée en “recommandé” aux observateurs présents, mais en “recommandé avec accusé de réception” aux acheteurs qui prennent part aux enchères. On souhaite en effet s’assurer que ces derniers ont bien pris en compte l’enchère avant de proposer de nouvelles enchères.

Au niveau protocole, ceci se traduit par différents types de contrôles d’erreurs :

- *Sans contrôle d’erreur* (Ordinaire)
- *Fiabilité orienté redondance* (Protégé). On se protège d’un certain taux de perte de bout en bout à l’aide de codes correcteurs d’erreurs. En introduisant de la redondance dans les données transmises, les récepteurs sont capables de récupérer un certain nombre de pertes de paquets.
- *Fiabilité orienté récepteur* (Recommandé). Lorsque qu’un récepteur détecte un paquet manquant dans la séquence, il envoie un **NACK** (Negative Acknowledgment) pour demander

une retransmission.

- *Fiabilité orienté émetteur* (Accusé de réception). En l’absence d’acquittement **ACK** d’un des récepteurs, le paquet manquant est retransmis.
- *Hybride*. Le récepteur envoie des **NACK** lorsqu’un trou dans la séquence est détecté. Un **ACK** est demandé après certains paquets de la séquence.

Le problème de la scalabilité est également essentiel dans la conception des protocoles multipoints. Il se décline sous deux formes. Les participants de l’EVP doivent être capables de gérer, d’une part la masse des données reçues au niveau applicatif, et d’autre part l’ensemble des messages de contrôle (demandes de retransmission, acquittements, gestion de groupe, etc...) au niveau transport. La fiabilité orientée récepteur est intrinsèquement plus scalable que la fiabilité orientée émetteur, puisque les demandes de retransmission ne sont transmises qu’en cas de perte de paquets. Ceci explique en partie pourquoi on ne peut pas simplement étendre TCP au multipoint. De plus, la connaissance des membres du groupe n’est pas nécessaire. Toutefois, en cas de perte partagée, l’émetteur n’est toujours pas protégé d’une possible “implosion” de messages de contrôle. Une technique utilisée pour gagner en scalabilité consiste à distribuer autant que possible les fonctionnalités du protocole, comme la retransmission de paquets manquants.

Au fur et à mesure que le nombre de récepteurs  $R$  augmente, la probabilité  $\hat{p} = 1 - (1 - p)^R$  que l’un d’entre eux perde un paquet augmente également. Les premiers protocoles, comme Xpress Transport Protocol (**XTP**) [SDW92], ont adopté une approche *centralisée* pour la retransmission. Bien qu’envoyant les retransmissions en multipoint, ils se trouvent confrontés, dans des environnements partagés de très grande taille, à un nombre très important de retransmissions. Une bonne partie de la bande passante est alors gaspillée. Des méthodes ont été proposées pour améliorer la scalabilité. Une première idée consiste à répartir le travail de retransmission. Dans Reliable Broadcast Protocol (**RBP**) [CM84], un sous-ensemble du groupe joue à tour de rôle de récepteur primaire ayant la responsabilité des retransmissions de paquets émis par les autres participants. Les récepteurs qui souhaitent une retransmission en font la demande au récepteur



primaire correspondant qui leur renvoie le paquet manquant en point à point, pour éviter d'encombrer le groupe. De nombreux protocoles ont également introduit la notion de *récupération locale*. Comme pour RBP, on désigne des récepteurs privilégiés ou d'autres entités, chargés des retransmissions. Ils ont ici la responsabilité des retransmissions sur une région locale uniquement. Dans Log-based receiver-reliable multicast (**LBRM**) [HSC95], ces entités sont des "serveurs de retransmission" choisis de manière statique. Reliable Multicast Transport Protocol (**RMTP**) [PSLt97] étend cette structure de retransmission à une *structure en arbre*. L'arbre multicast est découpé en sous arbres et un Designated Receiver (DR) est responsable de chaque sous arbre. Cette structure en arbre augmente encore la scalabilité du protocole, toutefois le problème de la construction de cet arbre reste un problème délicat.

D'autres techniques ont également été proposées dans le cadre de la fiabilité orienté récepteur, comme la technique visant à réduire le nombre de requêtes de retransmission. L'idée est simple : il est inutile de demander la retransmission d'un paquet si un autre récepteur l'a déjà fait. Les NACKs sont donc transmis en multipoint à tout le groupe, et non seulement à la source ou à l'hôte responsable des retransmissions. Un récepteur qui entend cette requête annule alors éventuellement la sienne. Les retransmissions sont ensuite envoyées en multipoint. Pour éviter que tous les récepteurs n'envoient leur requête en même temps, ce qui rendrait la suppression de NACKs inefficace, on retarde cette dernière d'une durée fixée aléatoirement. C'est l'approche adoptée par Scalable Reliable Multicast (**SRM**) [MJV96]. La durée est fixée aléatoirement suivant une loi uniforme dans un intervalle qui varie dynamiquement en fonction des délais observés entre les membres du groupe. On peut préférer une loi exponentielle, qui permet de réduire, à durée d'attente moyenne constante, l'espérance du nombre de requêtes [NB97]. SRM utilise le même mécanisme pour faire de la "suppression de retransmission". Tout récepteur peut répondre à un NACK, et des temps de retardement sont ici encore ajustés de façon à réduire au maximum les retransmissions inutiles.

Dans les EVPs, où les délais de retransmissions ne sont souvent pas vraiment acceptable compte tenu du caractère interactif de l'application (et donc de l'utilité très courte dans le temps du contenu des paquets de données transmis), la fiabilité orienté redondance peut s'avérer très utile. Les codes correcteurs d'erreurs ont été utilisés à l'origine pour corriger des bits de données erronés transmis sur un média sujets aux erreurs. Ils se sont avérés utiles dans le contexte du multipoint fiable pour la récupération de paquets perdus. Un code correcteur d'erreur ou FEC (Forward Error Correction) consiste, à l'émission, à coder  $k$  paquets en  $n$ . Un récepteur qui reçoit  $k$  paquets parmi ces  $n$  peut reconstituer les paquets d'origine. Le gain en bande passante dans un contexte multipoint est alors important, et l'est d'autant plus que les pertes observées par les récepteurs sont indépendantes. Supposons qu'à la suite de la transmission des  $k$  premiers paquets, chaque récepteur ait perdu un paquet. En l'absence de FEC, il aurait fallu retransmettre tous les paquets manquants, alors qu'un seul paquet de redondance suffirait à chaque récepteur pour récupérer son paquet manquant. Cependant, un des inconvénients de l'utilisation de la FEC est le délai supplémentaire induit par le système codage/décodage. Il faut en effet attendre d'avoir reçu  $k$  paquets avant de pouvoir décoder et faire remonter les premiers paquets à la couche applicative. Dans le cas des EVPs, seuls les données sans fortes contraintes temps réel pourront bénéficier de la FEC.

## 2.2 Environnements Virtuels

Dans cette partie, nous présentons les solutions et les mécanismes permettant de répondre au problème de scalabilité dans les EVPs. Toutefois, nous n'aborderons pas dans cette section les mécanismes et protocoles associés aux services de transport dont le rôle est d'assurer la fiabilité. L'étude de la scalabilité pour ce type de service a été abordée dans la Section 2.1.5 et ne constitue pas l'étude principale de cette thèse.

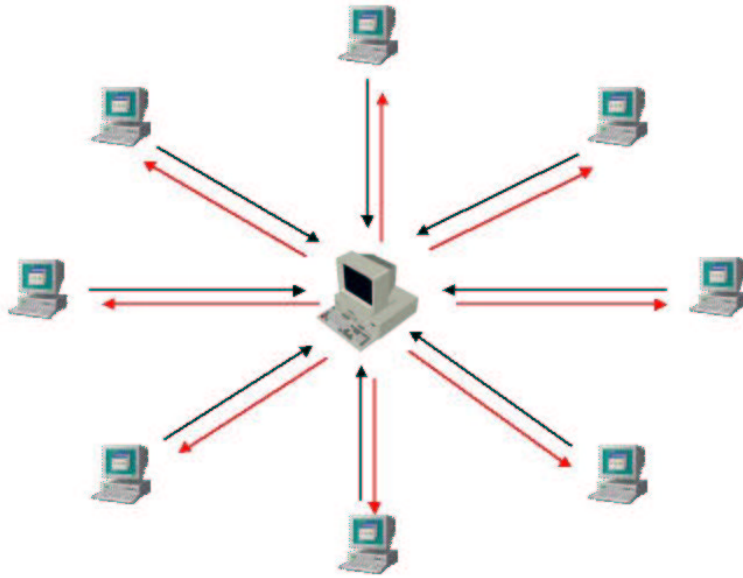


FIG. 2.2 – *Architecture Clients-Serveur*

### 2.2.1 Architectures de communication

Les environnements virtuels distribués peuvent s'appuyer sur deux types d'architecture: centralisée ou distribuée. Typiquement, le DIS utilise une architecture distribuée tandis que les applications commerciales actuelles se basent sur une architecture centralisée client/serveur (voir Figure 2.2). Cependant, au niveau de la scalabilité, l'architecture centralisée pose un problème majeur: l'intégralité du trafic converge vers le serveur. Dans le cas d'une architecture distribuée, chaque participant transmet ses actions aux autres participants. Si seule la communication en point-à-point est disponible, chaque participant doit transmettre ses informations en utilisant plusieurs connexions point-à-point (voir Figure 2.3). L'effet rafale du "multi-point-à-point" au niveau de chaque participant en émission, ne confère à l'architecture distribuée aucun réel avantage sur une architecture client-serveur. En revanche, l'utilisation conjointe d'une architecture distribuée avec un système de communication multipoint, représente une solution nettement plus efficace en terme de scalabilité (voir Figure 2.4).

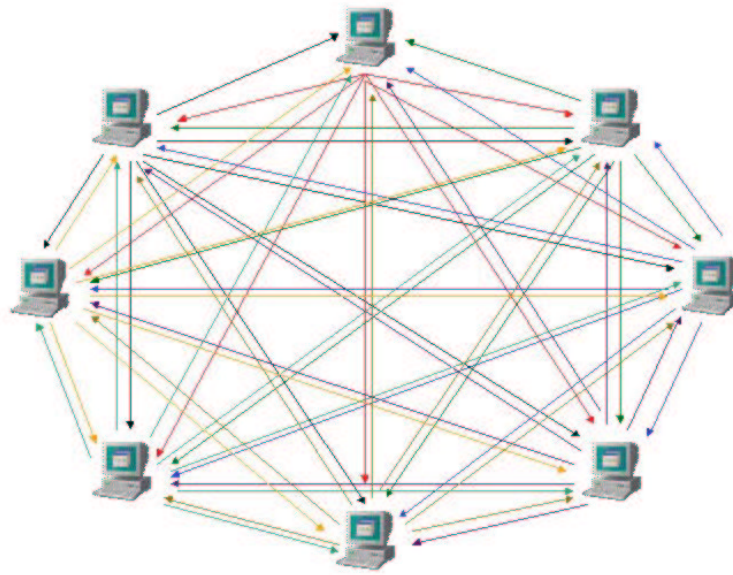


FIG. 2.3 – *Architecture Distribuée Point-à-Point*

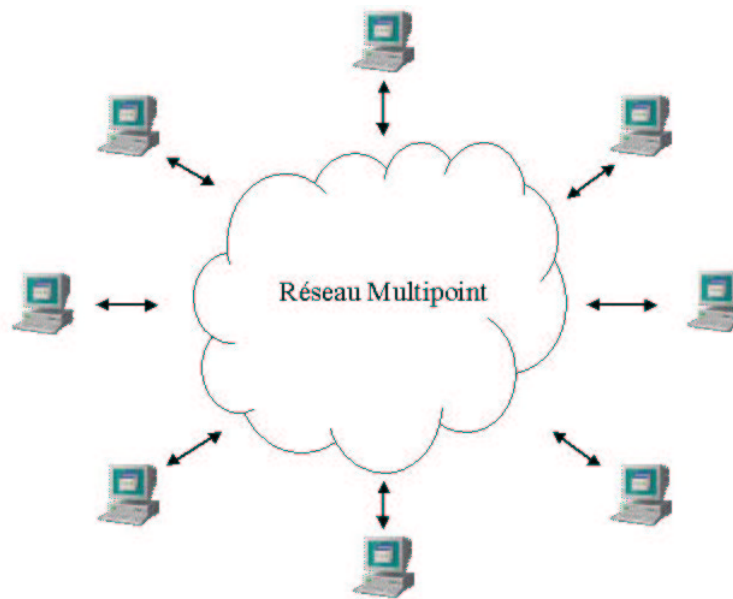


FIG. 2.4 – *Architecture Distribuée Multipoint*

## 2.2.2 Filtrage de l'information

Ce qui est important dans l'étude de la scalabilité, c'est de déterminer le ou les nœuds de communication au sein desquels les ressources ne sont pas suffisantes. Dans une approche centralisée, un goulot d'étranglement "naturel" se situe inévitablement au niveau du/des serveur(s) (architecture multi-serveur). Dans les architectures distribuées n'utilisant qu'un groupe multipoint unique, lorsque le nombre de participants augmente, le trafic destiné à chaque participant peut devenir trop important. Ceci peut se traduire par des pertes de paquet au niveau de certains nœuds du réseau et au niveau des machines hôtes des utilisateurs dont la connectivité réseau est bien souvent très limitée. Dans une architecture centralisée, le serveur joue le rôle de filtre en ne transmettant à chaque participant que l'information qui l'intéresse (cf. le trafic en provenance des autres participants avec lesquels il interagit). En revanche, dans une architecture distribuée, chaque participant doit filtrer l'information utile parmi l'ensemble des paquets reçus. Ce filtrage entraîne donc un accroissement du temps de traitement et de la demande en CPU, ce qui rend difficile le traitement temps-réel de l'intégralité des paquets reçus. Il est également à noter que l'information inutile reçue transite sur le réseau et représente un coût en terme de bande passante.

Deux approches sont actuellement proposées, une approche au niveau de la couche transport et une approche au niveau de la couche réseau. Dans le premier cas, l'architecture de communication est basée sur l'utilisation de plusieurs groupes multipoints. Des mécanismes de répartition dynamique des participants dans les différents groupes doivent donc être mis en œuvre (voir Section 2.2.2.1). La deuxième approche consiste à rajouter de nouveaux mécanismes au sein des routeurs afin de limiter la propagation des paquets vers le sous-ensemble de destinataires.

### 2.2.2.1 Approche couche transport

Depuis déjà quelques années, de nombreux centres de recherche de part le monde se sont attaqués à ce problème, en mettant au point divers mécanismes capable de filtrer l'information. Les premiers travaux entrepris autour de ce thème aboutirent à l'apparition de deux normes: **SIM-**

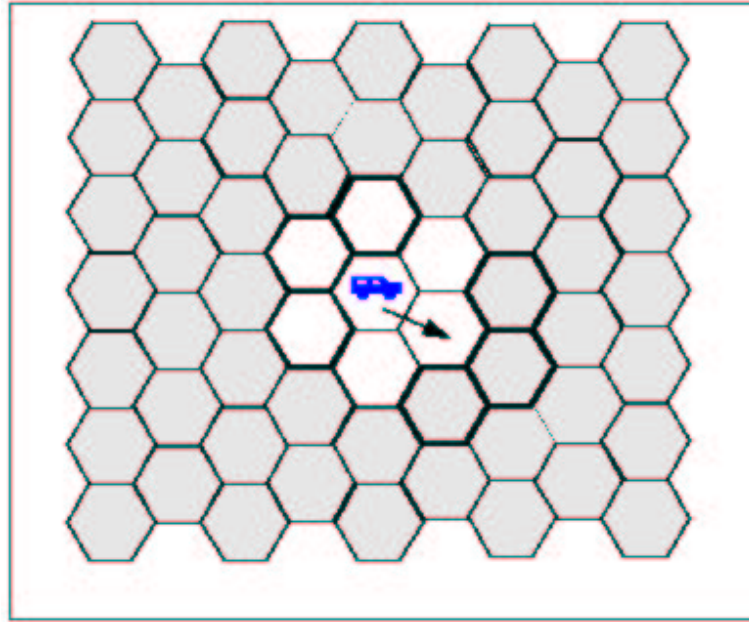


FIG. 2.5 – *Découpage en Cellules dans le DIS*

**NET** [Loc95] puis **DIS** (Distributed Interactive Simulation) [Soc95][BBM<sup>+</sup>94][SSM97], toutes deux très orientées vers la simulation militaire. Leur architectures distribuées furent toutefois à l'origine d'une des toutes premières thèses dans le domaine [Mac95]. Dans cette thèse, le problème de la scalabilité des **LSDS** (Large-Scale Distributed Simulations) est identifié. Les solutions proposées consistent à partitionner 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. Jusqu'à aujourd'hui, le réel problème consiste à déterminer une méthode réalisant ces associations, qui soit à la fois suffisamment générique, efficace et surtout scalable.

La première tentative fut implémentée dans **NPSNET** [MZP<sup>+</sup>94], qui utilise un découpage statique du monde virtuel en cellules (voir Figure 2.5). A 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. Une entité rejoint ou quitte un groupe

multipoint en envoyant un message d'abonnement ou de désabonnement aux autres membres du groupe. Le membre le plus ancien du groupe, appelé "Group Leader", est chargé de lui communiquer de façon fiable (via TCP) l'état de toutes les autres entités présentes dans la cellule associée à ce groupe [MZP<sup>+</sup>95]. 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. Dans [ZAD99], la stratégie de découpage en cellules et la stratégie de regroupement par classe d'entités sont comparées. Toutefois, aucune méthode permettant d'implémenter ces deux stratégies dans le cas réel n'est proposé. D'autres EVPs, comme **VREng** (Virtual Reality Engine) [Fel99], sont également basés sur des principes similaires à ceux implémentés dans NPSNET. Dans [RH96], une première étude est conduite afin de déterminer l'impact de la taille de cellules sur le trafic réseau. Cependant, les simulations ne sont réalisées que sur des exemples spécifiques de simulations militaires [RC98], rendant les résultats inexploitable dans le cas générique. D'autres travaux [AWZ98][LHnJR97][Pry99][MZ96] viennent également s'inscrire dans la continuité des techniques développées dans le DIS, mais s'efforcent essentiellement de faire ressortir plus clairement les besoins spécifiques aux EVPs.

Dans **DIVE** (Distributed Interactive Virtual Environment) [CH93] et **MASSIVE** (Model, Architecture and System for Spacial Interaction in Virtual Environments) [Gre96a][Gre96b][GBR99], plusieurs groupes multipoints sont également utilisés. Les environnements virtuels ont cette fois-ci décrits par une hiérarchie d'objets plutôt que par un découpage en cellules. Dans ces deux implémentations, plusieurs types de médias sont supportés et le concept d'**auras** [BGL97][BG97] est utilisé, généralisant ainsi la notion de centre d'intérêt d'une entité virtuelle (qui peut différer selon le type de média pour une même entité).

Diamond Park est un EVP basé sur une architecture multi-serveur appelée **SPLINE** (Scalable Platform for Large Interactive Networks Environments) [WAB<sup>+</sup>96]. Le monde virtuel est partagé en plusieurs zones ou *locales* à l'intérieur desquelles les communications se font en multipoint.

Chaque *locale* est représentée avec son propre système de coordonnées. Lorsqu'un participant change de *locale*, il obtient par l'intermédiaire des serveurs, une matrice lui permettant de changer de systèmes de coordonnées. Le protocole spécifique **ISTP** (Interactive Sharing Transfer Protocol) [WAS97] est proposé par les auteurs de SPLINE, afin de permettre la diffusion des messages de contrôle entre les serveurs et les participants.

#### 2.2.2.2 Approche couche réseau

Le déploiement actuel du multipoint dans l'Internet ne permet pas la création de *sous-groupes* au sein d'un unique groupe multipoint. En conséquence, les paquets transmis sur un groupe multipoint sont acheminés vers l'ensemble des membres du groupe. La nouvelle version du protocole d'abonnement au groupe multipoint, **IGMPv3** (Internet Group Management Protocol Version 3) [CDT00], devrait permettre aux membres d'un groupe de préciser les différentes sources dont ils désirent recevoir ou ne pas recevoir le trafic. Toutefois, ceci ne répondra que très partiellement aux besoins de sous-adressage au sein de groupes multipoints. En effet, le filtrage de l'information à l'aide de ce protocole, nécessitera à tout instant la connaissance des autres membres potentiels du groupe, ainsi qu'une connaissance a priori du trafic qu'ils génèrent. Le caractère interactif et dynamique des EVPs, dû non seulement aux déplacements mais également aux arrivées et départs fréquents des participants, confirme le besoin de mécanismes de filtrage plus efficaces.

L'approche au niveau de la couche réseau consiste à rajouter de nouveaux mécanismes au sein des routeurs afin de limiter la propagation des paquets vers le sous-ensemble de destinataires susceptibles d'être intéressés par son contenu [SFL<sup>+</sup>00][LGLA97]. Dans ce cas, chaque paquet doit contenir des *meta-informations* précisant la nature et le contenu de l'information qu'il contient. Dans AIM (Addressable Internet Multicast architecture) [LGLA97], les auteurs proposent que ces meta-informations soient comparées à des labels assignés à chaque routeur de l'arbre multipoint sous-jacent. Trois catégories de label sont définis : les labels de positions, les labels de routage et les labels de flots. AIM peut s'appuyer sur n'importe quel type d'arbres de routage



(basés sur la source ou partagés), mais nécessite cependant qu'un nœud de cet arbre soit désigné au préalable comme *racine d'adressage*.

Ainsi les labels de positionnement assignés à chaque routeur de l'arbre multipoint représentent la position du routeur par rapport à la racine d'adressage du même arbre. Grâce à ces labels, chaque source ou récepteur du groupe multipoint possède une adresse au sein de l'arbre multipoint associé. Un algorithme assez simple est proposé et chaque routeur l'implémentant peut ainsi calculer la valeur de son label. 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 cette fois-ci maintenir une liste de labels désignant les flots en cours de réceptions ainsi que la liste des interfaces sur lesquelles faire suivre les paquets pour chacun de ces flots. Chaque paquet est associé à un flot par l'intermédiaire de l'application qui lui assigne un label en fonction de son contenu.

Une approche au niveau réseau pour les EVPs permet en outre d'éviter les temps de latence liés à l'établissement d'arbre multipoint et les coûts liés au maintien d'un grand nombre d'arbres multipoint. En effet, à partir d'un unique arbre multipoint déjà existant, des techniques de sous-division en groupes peuvent être facilement utilisées par les applications afin de restreindre leur trafic ou une partie de leur trafic vers un sous-ensemble de destinataires. Idéalement ces destinataires ne représenteraient uniquement ceux intéressés par le contenu des paquets émis. Cependant une telle approche souffre de deux inconvénients majeurs : le problème de l'organisation des groupes au niveau applicatif (ou comment regrouper les sources et les récepteurs entre eux) et son déploiement à grande-échelle. En effet, cette approche entraîne des modifications au sein même du réseau au niveau de chaque routeur, et sera par conséquent beaucoup plus lourde à être déployée. Elle ne peut par conséquent que constituer une réponse partielle au problème, et qui plus est, à long terme.

Il est cependant intéressant de ne pas opposer systématiquement une approche au niveau réseau avec une approche au niveau transport. Bien au contraire, ces deux approches sont plutôt complémentaires, et sans doute une solution hybride permettrait de profiter des avantages de l'une comme de l'autre. Aucun travail de recherche ne s'est encore penché sur une solution duale transport-réseau à notre connaissance.

### 2.2.3 Synchronisation

En fonction de la nature des événements transmis dans un certain flot de données, les EVPs ont des exigences diverses concernant l'ordre des messages reçus par l'application. Les mécanismes de synchronisation ont en outre, la fonction de maintenir un certain état de cohérence entre les participants. Lorsqu'un état de consistance stricte n'est pas nécessaire ou tout simplement irréalisable compte tenu des propriétés du réseau, un mécanisme permettant de garantir l'*ordre causal* entre les événements peut parfois s'avérer suffisant. Dans ce cas, l'utilisation d'un temps logique peut permettre de propager la causalité sans pour autant garantir la consistance (le fait que les participants aient la même vision du monde virtuel au même moment). Toutefois ces mécanismes se rapportent plutôt aux systèmes distribués et ne seront pas détaillés dans ce chapitre. En revanche, il est intéressant d'étudier les mécanismes permettant une synchronisation globale sur l'Internet. Aujourd'hui, la solution utilisée consiste à synchroniser, via le réseau, les horloges d'un ensemble de machines sur une source externe. Le protocole **NTP** (Network Time Protocol) [Mil92] peut alors être utilisé, en totale indépendance avec l'application, afin de réaliser la synchronisation des horloges.

Dans [GKD99], un système de communication est proposé afin de permettre à chaque participant de calculer un état du monde virtuel en temps-réel. Pour cela, un mécanisme de synchronisation distribuée, appelé mécanisme de synchronisation par seuil, doit être mis en place. Le principe de base de ce mécanisme consiste à retarder le calcul d'un état global du monde virtuel, du temps

équivalent au délai d'interaction  $D$  (voir Section 1.2.2). Ce mécanisme permet donc d'absorber les variations de délais de transmission entre participants, de façon similaire aux mécanismes de temporisation utilisant l'algorithme de "playout" dédiés aux flux audio/vidéo. Chaque paquet émis par un participant contient une estampille de temps indiquant la date exacte d'émission du paquet. Le protocole **RTP** (Realtime Transport Protocol) [SCFJ98][RS99], définit un format d'encapsulation standard pour différents types de données (Audio, Vidéo, etc..) et permet d'inclure ces informations de temps. Le mécanisme de synchronisation divise le temps en intervalles de longueur  $T$  (avec  $T \leq D$ ). Lorsqu'un participant calcule un état global du monde virtuel, il utilise l'information contenue dans tous les paquets dont la date d'émission est comprise entre  $D$  et  $D - T$  (relativement au temps courant). Ainsi, toutes les actions des participants générées dans le même intervalle de temps seront traitées simultanément au niveau de chaque utilisateur.

Bien entendu, ce mécanisme fonctionne si les délais de transmission restent inférieurs au temps d'interaction  $D$ . Dans le cas d'une perte ou d'un délai trop grand, des techniques d'extrapolation de trajectoire sont utilisées. Le *Dead-Reckoning* représente une telle technique. Elle fut initialement développée dans le domaine de l'aviation, pour estimer la position des avions à partir de la dernière position connue et du vecteur de déplacement. Cette technique est également utilisée dans le DIS pour limiter le débit des données transmises. Chaque participant à une simulation DIS maintient en local sa position courante, ainsi que sa position extrapolée à partir de la dernière position transmise sur le réseau. La position courante n'est transmise sur le réseau que si l'estimation obtenue par *Dead-Reckoning* dépasse un seuil donné. Des paquets de maintien de connexion (*Keep-Alive*) sont toutefois envoyés régulièrement sur le réseau par chaque participant, afin de signaler leur présence dans le monde virtuel. Toutes ces techniques doivent s'adapter aux conditions variables du réseau (taux de perte, délais), en modifiant dynamiquement leurs paramètres de manière à garantir la meilleure consistance possible à tout instant.

## Chapitre 3

# Problématique

### 3.1 Interactivité et Scalabilité

Après avoir défini dans les chapitres précédents les notions d'interactivité et de scalabilité, ainsi que les divers approches qui visent à répondre au problème particulier de la scalabilité dans les EVGEs, nous allons désormais traiter de la problématique consistant à répondre à ces deux critères à la fois. En effet, la nature même des EVGEs implique que l'architecture de communication sous-jacente soit capable de satisfaire les contraintes liées au grand nombre d'utilisateurs ainsi que les contraintes liées au caractère interactif de ce type d'applications. Les travaux de recherche ayant conduit à diverses propositions pour les EVGEs répondent le plus souvent à l'une ou l'autre de ces contraintes mais souffrent tout particulièrement de leur association. En effet, une architecture centralisée permet de part sa nature un meilleur contrôle des délais entre les participants mais la convergence du trafic qu'elle entraîne ne la rend pas scalable. Au contraire, les approches proposées au niveau transport préconisent l'utilisation d'un grand nombre de groupes multipoints et semblent répondre au besoin de filtrage de trafic dans les EVGEs, et par conséquent résoudre le problème de la scalabilité.

Cependant, ces approches se fondent sur deux hypothèses pas très réalistes, ce qui ne leur per-

met pas de garantir de façon scalable, l'interactivité de l'application : La première hypothèse consiste à considérer les groupes multipoints comme une ressource inépuisable et n'impliquant aucun coût particulier. Par exemple, le DIS préconise l'association d'un groupe multipoint par entité présente dans le monde virtuel, et ceci pour un nombre arbitrairement grand d'entités. Ainsi, les problèmes d'allocation d'adresses multipoints, de coût d'établissement et de maintien des arbres multipoint ne sont pas considérés. Nous détaillons dans la Section 3.4, les limitations de l'utilisation du multipoint. La deuxième hypothèse réside au niveau du gestionnaire de centre d'intérêt chargé de définir et d'identifier les centres d'intérêt dans l'EVGE et de réaliser le couplage entre ces centres d'intérêt et les groupes multipoints. En effet, les délais consécutifs aux calculs de regroupement des participants dans les groupes multipoints sont bien souvent ignorés ainsi que la latence due à la transmission de ces informations de couplage vers les participants. Toutefois, la forte dynamique et hétérogénéité du grand nombre de participants au sein d'un EVGE, nécessite la prise en compte de ces coûts dans l'architecture de communication, afin de garantir l'interactivité de l'application tout en restant scalable.

Les travaux présentés dans cette thèse s'appliquent donc à proposer une solution globale répondant à la fois au délicat problème de la scalabilité dans les EVPs, ainsi qu'au respect des contraintes d'interactivité intrinsèques à ce type d'application.

## 3.2 Hétérogénéité

Les EVPs distribués à grande-échelle (EVGE), peuvent contenir un très grand nombre de participants (voir Section 1.2.1), répartis sur plusieurs domaines de l'Internet. Cette caractéristique est à l'origine d'une très forte hétérogénéité entre les participants. Cette hétérogénéité se traduit en premier lieu par les différentes capacités des liens qui relient les participants à l'Internet, ainsi qu'aux différentes puissances de calcul des machines-hôtes. Cependant, cette hétérogénéité peut également s'exprimer en terme de quantité de détails désirés, de niveau de consistance demandé et de type de données préférées. Cette grande hétérogénéité entre les préférences des utilisateurs

doit donc être prise en compte par les EVPs afin que les ressources dont ils disposent soient utilisées le plus efficacement possible, aussi bien sur le plan quantitatif que qualitatif. Les mécanismes qui sont employés doivent donc trouver le juste compromis entre l'état rajouté dans les routeurs, la bande passante utilisée, et la satisfaction de l'utilisateur final.

Chaque participant d'un EVGE peut donc être 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. En conséquence, une architecture de communication pour EVGE se doit de prendre en compte l'hétérogénéité au sein des participants pour ces deux paramètres, et d'allouer au mieux les ressources dont elle dispose pour améliorer la qualité des informations que chaque participant reçoit. Au sein d'un environnement virtuel, ces paramètres peuvent également prendre des valeurs différentes pour un même participant au cours d'une même session. De plus, certains participants peuvent se désabonner à la session en cours tandis que de nouveaux participants peuvent se connecter avec de nouvelles exigences qualitatives et quantitatives sur le trafic qu'ils désirent recevoir. Cette dynamique figure donc au cœur de la problématique dans l'étude et la conception de l'architecture de communication.

Dans le cadre des applications de vidéo-conférences sur l'Internet, des techniques de codage en couches peuvent être utilisées pour répondre à une certaine forme d'hétérogénéité. L'idée consiste à partitionner les données en plusieurs couches et à transmettre les paquets dans chacune des couches à des débits croissants. Grâce à l'association de chacune des couches avec un groupe multipoint distinct, les destinataires peuvent réduire ou augmenter le débit d'information qu'ils désirent recevoir, en s'abonnant ou se désabonnant aux groupes multipoints correspondant [MJV96]. Dans le cas de la vidéo, la source va découper ses données en un certain nombre de flots spatialement ou temporellement complémentaires. Pour une complémentarité spatiale par exemple, le flot est composé d'une première couche dite de base qui représente une expression de basse qualité de l'image, et de couches dites complémentaires qui viennent progressivement

affiner la résolution de l'image.

Dans le cadre des EVGEs, on pourrait également envisager d'utiliser ces techniques de codage en couches afin de gérer une partie de l'hétérogénéité au niveau des participants. Cependant, elles ne répondent que partiellement au problème de filtrage de l'information et nécessitent encore davantage de groupes multipoints, ce qui peut parfois les rendre très coûteuses dans les EVGEs (voir Section 3.4) : En effet, contrairement aux applications de vidéo conférence où le nombre de participants émetteurs d'information est le plus souvent bien en dessous du nombre de destinataires, les EVPs sont des applications de type "plusieurs-vers-plusieurs" où chaque participant est à la fois source et récepteur. L'exemple du DIS est assez parlant : chaque entité du monde virtuel est associé à un groupe multipoint. Les participants doivent donc s'abonner aux groupes multipoint associés aux entités avec qui ils désirent interagir. Par exemple, lors d'une simulation d'exercice militaire où 15000 entités sont présentes, le nombre de groupes multipoints utilisés (et donc a fortiori d'adresses multipoint) serait égal à 15000 sans codage en couche, et à un multiple de 15000 si un codage en couche est utilisé.

### 3.3 Constitution des groupes

Définir une architecture de communication pour EVGEs est particulièrement complexe tant le nombre de paramètres à considérer et les besoins pour ce type d'applications sont importants. Une fois ces besoins et ces paramètres clairement identifiés, l'étape suivante consiste à réunir les outils nécessaires à l'élaboration d'une solution capable de fournir aux EVGEs une réponse globalement satisfaisante. Ainsi le multipoint apparaît aujourd'hui comme un outil essentiel à la résolution du problème de scalabilité dans les EVGEs. Grâce à ce nouvel outil, des techniques de filtrage de l'information peuvent donc être mise en place. Cependant à ce stade de la réflexion, les difficultés ne font que commencer. Tout d'abord, il convient d'évaluer les limites et les coûts liés à l'utilisation de ce nouvel outil. Il faut ensuite intégrer et pondérer ses coûts avec les autres paramètres définis par l'application. Enfin, et après avoir défini une métrique globale d'évalua-

tion, des techniques de regroupement des participants au sein de groupes multipoints doivent être mise en œuvre.

La première problématique que nous rencontrons ici revient à confronter l'ensemble des éléments rentrant en ligne de compte dans la définition d'une métrique globale d'évaluation afin de dégager une politique efficace de constitution des groupes :

- Améliorer la satisfaction des récepteurs.
- Réduire le nombre de duplication de paquets à l'émission (et par conséquent circulant au sein du réseau).
- Limiter les coûts liés au multipoint en limitant son utilisation, à savoir le nombre de groupes multipoints utilisés par l'architecture.

Il est important de constater que dans cette énumération n'apparaissent pas les coûts relatifs à la mise en œuvre de l'architecture de communication elle-même, et qui regroupent l'ensemble des messages de contrôle ainsi que les divers composants susceptibles d'être rajoutés par celle-ci (serveurs,agents,proxies,etc..).

Concernant le premier point, une nouvelle question se pose: comment définir la satisfaction au niveau des récepteurs? Une première approche consisterait à considérer qu'un récepteur n'est totalement satisfait que lorsque que l'ensemble des informations reçues l'intéresse, et que l'ensemble des informations l'intéressant a bien été reçu et traité. Une deuxième approche tenterait de prendre en compte les limitations propres au récepteur, c'est à dire la capacité du lien le connectant au reste de l'Internet ainsi que sa puissance de calcul. Dans ce cas, le récepteur peut avoir une satisfaction maximale, si l'ensemble des informations reçues l'intéresse même si l'ensemble des informations l'intéressant n'a pas été reçu et traité en temps-réel du fait de ses propres limitations. En effet, dans la situation où le récepteur désire recevoir et traiter plus de données qu'il n'en est capable, aucune politique de constitution de groupes ne pourrait y changer quelque chose. Dans une troisième approche, un participant sera totalement satisfait si l'ensemble des



informations l'intéressant a bien été reçu et traité. Par rapport à la première approche, on ne tient plus compte ici du fait que des informations inutiles aient également pu être reçues.

En ce qui concerne la limitation dans l'utilisation du multipoint, une problématique se pose de nouveau. On peut en effet considérer qu'on dispose d'un nombre limité de groupes multipoints et qu'il s'agit alors d'optimiser son utilisation. Au contraire, on peut poser des contraintes fixes sur la satisfaction minimale des récepteurs, et chercher les techniques de regroupements qui répondent à cette contrainte tout en minimisant le nombre de groupes multipoints utilisés. Quoiqu'il en soit, dans les deux cas la question du regroupement se pose : Il convient d'analyser les intérêts communs entre participants afin de les répartir efficacement dans des groupes multipoints. Ce concept, également appelé **similarité de préférences** dans [WKM00], peut entraîner des regroupements de sources ou de destinataires selon les cas. En effet, si un grand nombre de récepteurs est intéressé par le trafic émis par 5 sources particulières qu'ils ont en commun, alors il serait plus judicieux de regrouper ces 5 sources dans le même groupe multipoint. En revanche, si un grand nombre de sources désire transmettre des informations vers 5 récepteurs particuliers qu'elles ont en commun, alors le regroupement de ces 5 destinataires dans un même groupe multipoint s'avère plus judicieux.

Dans le cas des EVGEs, chaque participant est à la fois source et destination, et le centre d'intérêt de chaque participant peut être représenté par une zone virtuelle qui entoure son avatar (i.e. la représentation du participant dans le monde virtuel). Cette zone contient par définition l'ensemble des entités virtuelles avec qui le participant peut interagir. Par conséquent, on retrouve ici ce concept de similarité de préférence en observant les recouvrements entre les différentes zones d'intérêt des participants. La difficulté de constitution des groupes dans ce cas, revient à définir dynamiquement des groupes de participants pouvant interagir à un instant donné. Cependant si un participant A peut interagir avec deux participants B et C, cela n'implique pas forcément que les participants B et C peuvent interagir.

La constitution des groupes dans un EVGE n'est donc pas trivial, surtout si on considère le nombre limité de groupes alloués à l'application ainsi que le coût lié à la réorganisation des groupes en cours de session. Ainsi, si un algorithme de regroupement optimal des participants existait, celui-ci ne serait pas forcément le plus adéquat au sein des EVGEs. En effet, l'optimalité en terme de regroupement n'implique pas forcément que l'algorithme peut être réalisé avec un coût acceptable. La dynamique des paramètres pour ce type d'application nécessite en effet une redistribution périodique des membres au sein des groupes. Une fois de plus, si l'on souhaite limiter le surcoût introduit par les techniques de regroupement dynamique, on doit considérer une nouvelle contrainte dans le choix de cette technique. L'algorithme optimal entraînerait donc, de par son coût, une période de redistribution plus grande. Ceci reviendrait donc à accepter qu'entre deux instants distant d'une période dans le temps, la distribution des participants au sein des groupes ne soit pas optimale. Par conséquent la constitution des groupes doit être réalisée en tenant compte du niveau de dynamique du EVGE. De plus son évaluation ne doit pas seulement être effectuée aux instants précis de regroupement mais à tout instant au cours de la session. On peut par exemple mesurer les caractéristiques du gain (moyenne, variance, etc..) en terme de satisfaction au niveau des participants pour une session donnée avec un niveau de dynamique donné.

### 3.4 Limites du multipoint

Les travaux de recherche menés jusqu'ici dans le domaine de la scalabilité dans les EVGEs, ont clairement mis en évidence la nécessité de l'emploi de multipoint dans l'élaboration d'une éventuelle architecture de communication, capable de répondre aux besoins propres de ce type d'application. Dès lors, plusieurs propositions ont été décrites, basées sur le multipoint. Parfois, celle-ci ne s'appuie que sur l'utilisation d'un groupe multipoint unique[GKD99]. A l'opposé d'autres propositions recommandent l'utilisation sans limite du multipoint, sans se soucier des problèmes et des limitations inhérents à ce type de communication sur l'Internet. Dans cette

section, nous allons tenter de bien identifier ces problèmes et ces limites afin de parvenir à une vision la plus complète possible de la problématique de mise en réseau d'un environnement virtuel partagé sur l'Internet.

### 3.4.1 Routage

Une première limitation dans l'utilisation de plusieurs groupes multipoint découle directement des protocoles de routage multipoint. Ceux-ci permettent la construction des arbres multipoint reliant les membres de chaque groupe entre eux. Pour chacun des groupes, tous les routeurs présents sur l'arbre multipoint associé doivent maintenir un ensemble d'états. Ces états sont stockés dans les tables de routage et de suivis du trafic multipoint, présentes au sein des routeurs. Ils précisent les ports de sortie à partir desquels les paquets multipoints émis sur le groupe devront être dupliqués. Or il serait guère réaliste de considérer ces tables comme des ressources de capacité infinie. De plus, les protocoles de routage multipoint intra-domaine permettent la construction de deux types d'arbre : les arbres basés à la source (DVRMP, MOSPF, PIM-DM, PIM-SM), et les arbres partagés (CBT, PIM-SM). Dans le cas des arbres basés à la source, un état par source et par groupe doit être maintenu dans chaque routeur. En revanche dans les arbres partagés, un seul état par groupe est maintenu dans les routeurs, quelque soit le nombre de sources présentes dans le groupe. Or, dans les EVGEs, chaque participant est à la fois source et destinataire (cependant les techniques de regroupement de participants peuvent impliquer qu'un participant soit membre de plusieurs groupes sans pour autant que celui-ci ne soit émetteur dans chacun de ces groupes). Par conséquent, un grand nombre de groupes multipoints impliquerait un coût non négligeable au sein du réseau.

Une seconde limitation dans l'utilisation du multipoint apparaît dès que l'on prend en compte le caractère dynamique des EVGEs. En effet, dans la Section 3.3, nous avons présenté la problématique liée au besoin de réorganisation périodique des groupes de participants. Au cours d'une session, les participants sont donc amenés à s'abonner ou se désabonner à des groupes multi-

points. Ces deux actions entraînent un surcoût en terme de traitement au niveau des routeurs et en terme de bande passante au sein du réseau. Un message d'abonnement ou de désabonnement IGMP est transmis entre la machine-hôte du participant et son premier routeur multipoint, attaché à son sous-réseau. Cette information est ensuite propagée jusqu'au premier embranchement de l'arbre multipoint (dans le cas où il n'y avait pas d'autres participants présents sur le groupe parmi les autres utilisateurs du sous-réseau). Cette propagation a pour effet d'entraîner un surcoût de traitement au niveau de chaque routeur intermédiaire. Toutefois, l'importance de ce surcoût dépend fortement de la topologie de l'arbre multipoint existant. Plus le point d'embranchement est éloigné, plus le surcoût sera important. De plus, certains protocoles de routage multipoint comme DVRMP, s'appuie sur l'inondation périodique de messages dans le réseau, afin de mettre à jour l'arbre multipoint. Ce trafic à bien entendu un coût, et constitue un nouvel argument en faveur de la limitation du nombre de groupes multipoints dans les EVGEs.

### 3.4.2 Latence de l'abonnement

Un autre problème dans l'utilisation d'IP multipoint doit être également pris en compte. Il s'agit de la latence liée à l'abonnement, c'est à dire le temps qui sépare l'émission d'une requête d'abonnement IGMP, de la réception éventuelle d'un premier paquet circulant sur le groupe. L'utilisation d'un nombre important de groupes multipoints dans une architecture de communication pour EVGEs, implique nécessairement une plus grande fréquence d'arrivées et de départs dans chacun des groupes. Au contraire, si on utilise un groupe multipoint unique, ce problème n'a plus raison d'être, mais cette fois-ci l'architecture n'est plus scalable.

Dans la Section 3.4.1, on a vu que l'arrivée d'un nouveau participant sur un groupe multipoint a pour conséquence la mise à jour de certains états présents au sein des routeurs entre le participant et le point d'embranchement le plus proche sur l'arbre multipoint associé. Cette mise à jour n'a pas seulement un impact sur la bande passante et le traitement au niveau des routeurs, mais a également un impact en terme de latence d'abonnement. Or les EVGEs sont des applications in-

teractives avec des contraintes temporelles plus ou moins fortes selon l'application. Dans tous les cas, ce temps de latence doit être pris en compte et "masqué" à l'utilisateur final d'une manière ou d'une autre. Le caractère interactif de l'EVGE doit en effet être préservé tout au long de la session, et quelque soit le niveau de dynamicité des participants dans l'environnement virtuel. Ce temps de latence peut prendre des valeurs différentes en cours de session et selon les participants. En effet, si un autre participant présent sur le même réseau local, est déjà abonné au groupe multipoint, alors l'abonnement se fera presque instantanément. En revanche, un participant connecté en Australie et désirant rejoindre une session à laquelle seuls des Européens sont abonnés, percevra un temps de latence à l'abonnement pouvant atteindre plusieurs centaines de millisecondes.

La latence liée au désabonnement constitue également un problème qu'il convient de préciser : Si des techniques récentes [Riz98] permettent à un membre d'un groupe de se désabonner rapidement auprès de son routeur désigné sur son sous-réseau, il serait irréaliste d'imaginer qu'au niveau de l'arbre multipoint, cette opération soit instantanée. Selon les auteurs de [KHTK99], traitant de l'utilisation de plusieurs groupes multipoints dans le transport multipoint fiable, le routeur peut attendre quelques secondes avant de propager ce message de désabonnement vers le prochain routeur de l'arbre. Ceci implique que même lorsqu'un participant isolé (au niveau physique) se désabonne à un groupe multipoint, le sous-réseau sur lequel est présente la machine-hôte du participant, continue de recevoir les paquets émis sur le groupe. Par conséquent, une opération de désabonnement au niveau d'un participant permet au participant de filtrer rapidement le trafic qu'il reçoit et d'économiser ainsi les ressources de sa machine-hôte, sans pour autant économiser la bande passante du réseau (en tout cas pas de façon immédiate).

Le problème de la latence d'abonnement peut cependant avoir des effets beaucoup plus désastreux sur l'application que le problème de la latence de désabonnement. Le seul moyen permettant de résoudre ce problème consiste à forcer l'application à anticiper l'abonnement aux groupes multipoints susceptibles de l'intéresser après un temps égal à une latence d'abonnement.

Une double problématique se pose désormais : Comment estimer le plus précisément possible la latence d'abonnement (sachant qu'il convient toujours de la majorer dans son estimation) ? Quels mécanismes doit-on mettre en œuvre afin de réaliser la prédiction préalablement nécessaire à toute anticipation d'abonnement ? Au cours de cette thèse, nous traiterons essentiellement du deuxième point de cette problématique ; d'autres études s'étant déjà efforcées de répondre à la première partie [AA96].

### 3.4.3 Taille de session

Une architecture de communication capable de supporter un nombre arbitrairement grand de participants, doit au préalable avoir connaissance de ce nombre ou au moins d'une approximation suffisamment précise de ce nombre. De plus, et afin de s'adapter de façon scalable à la taille de la session en cours, l'architecture de communication doit périodiquement calculer sa valeur courante et prendre ainsi en compte son évolution au cours du temps. La répartition des participants au sein de plusieurs groupes multipoints rend cette opération encore plus difficile à réaliser. Cette problématique devient encore plus complexe si le nombre de groupes multipoints disponibles pour l'application est limité. En effet, dans ce cas de figure, un groupe multipoint, qui peut être perçu comme un outil de filtrage de l'information, devient une ressource limitée qu'il convient alors d'utiliser le plus efficacement possible. La connaissance de la taille de la session, regroupant l'ensemble des participants, est par conséquent une information essentielle, puisqu'elle permettra à l'architecture de communication sous-jacente, d'évaluer "la charge moyenne" des groupes multipoints, c'est-à-dire d'avoir une certaine connaissance du nombre moyen de sources par groupe. Il va sans dire que cette information devra être régulièrement communiqué à l'organe de l'architecture (centralisé ou distribué) chargé d'effectuer la constitution des groupes (voir Section 3.3).

Le modèle actuel de IP multipoint est basé sur un service dit "ouvert", et ne facilite pas vraiment le calcul de la taille d'une session au sein d'un groupe multipoint. En effet, - et certains argumenteront qu'il ne s'agit pas d'un bogue mais d'une caractéristique positive et propre à la

nature même du multipoint -, n'importe quelle machine-hôte peut émettre sur un groupe sans pour autant devenir préalablement membre du groupe. De plus, aucun mécanisme n'a été initialement prévu afin de renseigner les membres d'un groupe, du nombre de sources et/ou de récepteurs qu'il contient. Dans [FT99][LN], des techniques d'approximation de taille de session pour un groupe multipoint sont présentées. Ces techniques permettent de réduire au maximum le trafic de contrôle nécessaire au calcul de l'estimation de la taille de session. Toutefois, leur coût n'en reste pas moins non négligeable, ainsi que la latence introduite par l'envoi de ces messages de contrôle. L'utilisation de plusieurs groupes multipoints doit donc être pris en compte dans la problématique soulevée par le calcul de la taille de session. De nouveaux mécanismes plus adaptés à un ensemble de groupes de taille moyenne plutôt qu'à un groupe unique de grande taille devront par conséquent être mise en œuvre dans l'architecture de communication pour EVGE.

#### **3.4.4 Contrôle d'accès**

Nous traitons ici du contrôle d'accès au sein d'une session d'un environnement virtuel partagé. Nous nous plaçons ici au niveau de la couche réseau; des techniques applicatives de cryptage adaptées aux groupes multipoints dynamiques ont été proposées dans [CWSP]. Ce dernier point nous semble important d'être abordé dans la problématique générale d'utilisation du multipoint dans une architecture de communication pour EVGE. Il convient en effet de signaler qu'aucun mécanisme n'existe actuellement, au niveau routage multipoint, permettant d'assurer le contrôle d'accès d'un participant, source ou émetteur, dans un groupe. Redéfinir un nouveau modèle multipoint répondant à ce dernier point ne constitue assurément pas le sujet de cette thèse. En revanche, il nous paraît essentiel de l'aborder afin de proposer une solution partielle au problème dans notre architecture de communication. En effet, l'utilisation de plusieurs groupes multipoints, comme nous l'avons exposé dans les sections précédentes, nécessite des techniques de répartition des participants au sein des groupes. Or, la constitution des groupes représente précisément une des composantes d'une architecture de communication pour EVGE.

Dans certains cas, on peut supposer une répartition assez implicite des participants au sein des groupes. C'est par exemple le cas dans NPSNET (voir Section 2.2.2.1), où on a un découpage statique du monde virtuel en cellule avec un groupe multipoint associé par cellule. En revanche, on peut imaginer d'autres techniques de regroupement plus complexes, nécessitant à chaque participant d'obtenir une information explicite sur les groupes multipoints qu'il doit joindre afin de recevoir le trafic qui l'intéresse. Une architecture de communication mettant en œuvre de tels mécanismes pourrait par conséquent permettre aux EVGEs d'effectuer un certain contrôle d'accès au sein de leur session.





## Chapitre 4

# MiMaze3D

### 4.1 Motivations

Afin d'identifier les problèmes de scalabilité rencontrés dans les EVGEs et de proposer des solutions adéquates, nous avons choisi d'aborder le problème en partant de l'existant. Cette approche présente un double intérêt de part son caractère pragmatique : Premièrement, elle nous permet de motiver le besoin réel de scalabilité au sein des EVGEs, en observant le comportement d'un environnement virtuel multi-utilisateur lorsque le nombre de participants augmente. Deuxièmement, elle nous sensibilise, au travers des expérimentations d'un cas réel d'environnement virtuel partagé, au problème de la scalabilité, et nous permet d'en saisir les composantes, sa complexité, et de mesurer son impact sur les performances de l'application au niveau de l'utilisateur final. Cette première étape dans notre travail de recherche a pour objectif d'aboutir à une connaissance globale de la problématique, ainsi qu'à l'élaboration de solutions à la fois efficaces et réalistes en tenant compte des caractéristiques propres de l'Internet.

Parmi les diverses applications interactives multi-utilisateur existant au démarrage de nos travaux de recherche [GB95][CH93][LHM][LHMM97][WAB<sup>+</sup>96], notre choix s'est naturellement porté sur MiMaze (Multicast internet Maze), un jeu distribué sur l'Internet, basé sur le système de simula-

tion distribuée interactive (DIS). En effet, outre le fait que cette application avait déjà fait l'objet d'un thèse au sein de notre projet de recherche (Rodeo), elle présentait des caractéristiques très intéressantes de part son architecture de communication entièrement distribuée, l'utilisation du multipoint, et ses mécanismes de synchronisation. Nous avons donc poursuivi les travaux sur MiMaze afin de rajouter une composante 3D à l'application, dans le but d'étudier un exemple concret de monde virtuel partagé sur l'Internet. L'aboutissement de ces travaux tient donc dans une seule et nouvelle application, constituant à elle-seule un parfait exemple d'EVPs : MiMaze3D (voir Figure 4.1).

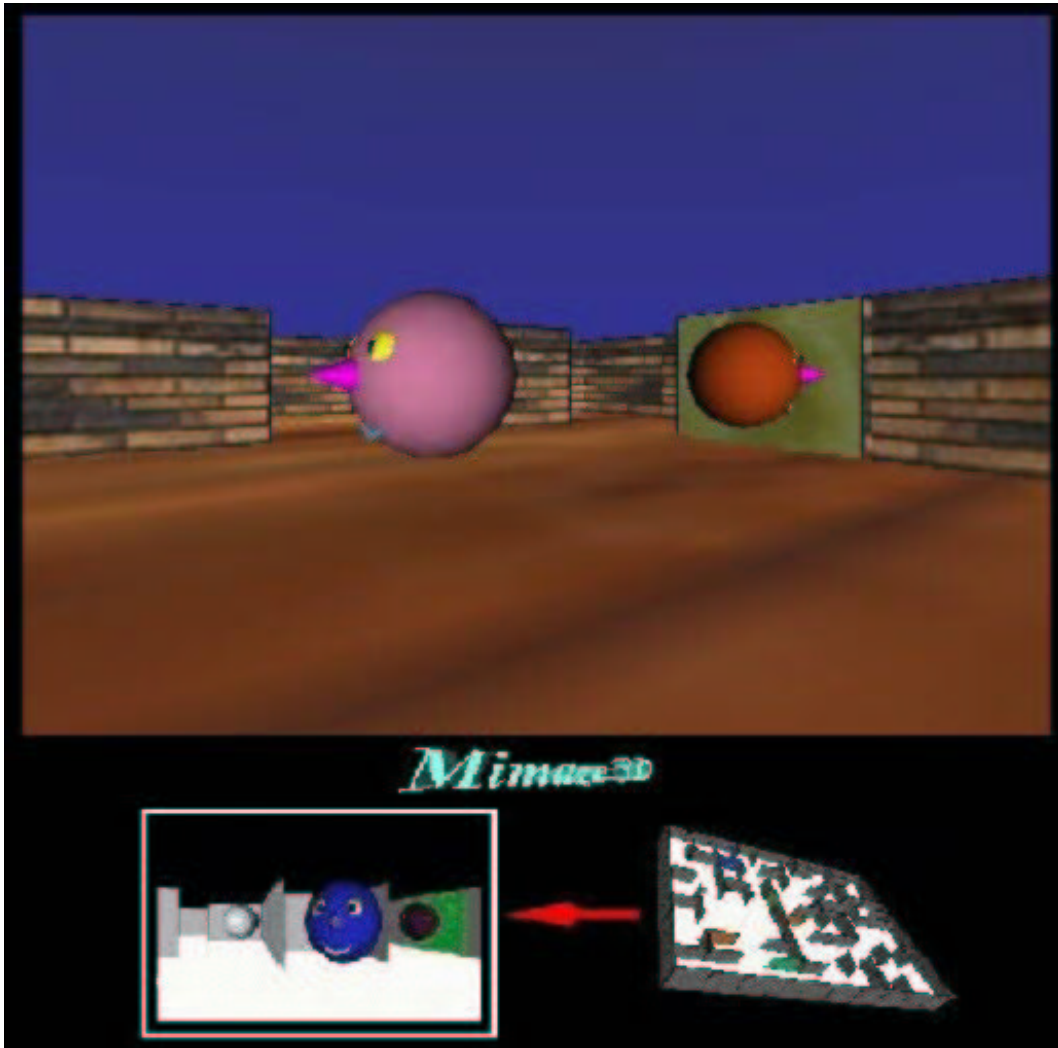


FIG. 4.1 - *MiMaze3D*

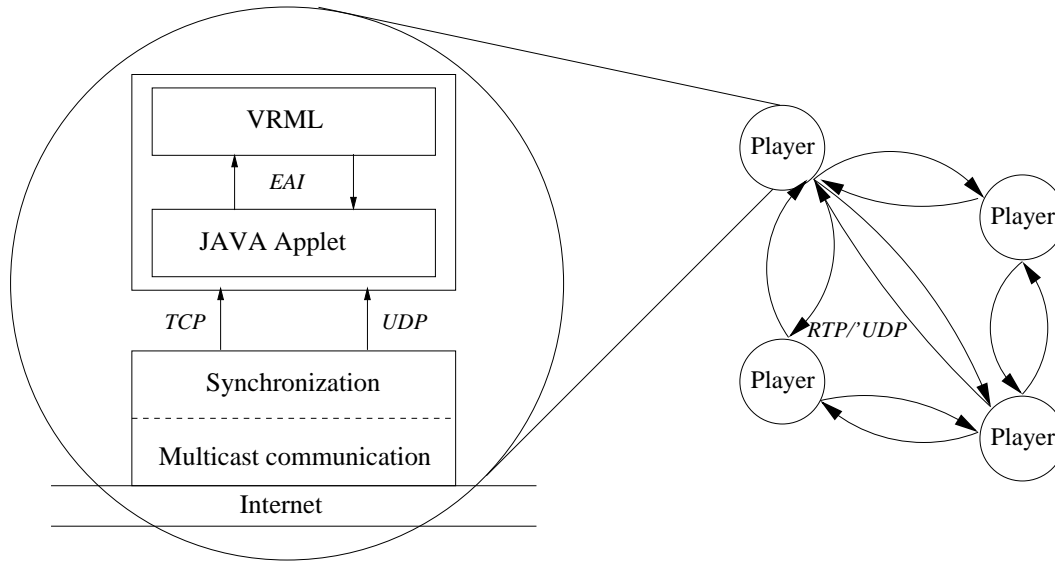


FIG. 4.2 – L'architecture de MiMaze3D

## 4.2 Architecture de MiMaze3D

MiMaze3D présente une architecture en couches. La couche la plus basse contient le module de communication (écrit en C++). Tous les participants communiquent entre eux en échangeant des paquets sur un groupe multipoint unique. Au niveau applicatif, un paquet est également appelé une *ADU* (Application Data Unit), et contient la position, la direction, la vitesse linéaire et la vitesse angulaire du participant. Le débit d'émission est le même pour chaque participant, à savoir 25 paquets par seconde. Au dessus du module de communication se trouve le coeur de l'application, avec ses mécanismes de synchronisation, d'extrapolation de trajectoire en cas de perte d'une ADU (dans certains cas, il peut également s'agir d'une interpolation, pour plus de détails concernant ces mécanismes, voir [GKD99]), et le calcul d'état. C'est au dessus de cette couche que nous avons implémenté la couche de rendu 3D.

Nous avons choisi VRML [VRM97][WH98] comme langage de description de scène 3D pour les multiples avantages qu'il nous offre :

- Indépendance de la plate-forme (Windows, Mac OS, UNIX, etc.),

- Standard ISO/IEC 14772 destiné à la description de mondes virtuels sur le Web,
- Extensibilité et inter-opérabilité,
- Support de graphique 3D, vidéo (MPEG-1) et son spatial.

L'architecture de MiMaze a donc été modifiée en conséquence. Une couche supplémentaire a été rajoutée au dessus de la couche de communication. Cette nouvelle couche peut-être découpée en 2 modules : une applet Java et un navigateur VRML, tout deux embarqués dans une même page HTML (voir Figure 4.2). L'applet Java joue le rôle d'interface entre le moteur de rendu et la couche de communication. D'un côté, elle utilise la librairie de l'EAI (External Authoring Interface) fournie par le navigateur VRML, afin d'instancier, de modifier et de supprimer les noeuds VRML de la scène 3D. D'un autre côté, elle est connectée à la couche de communication via deux sockets : une socket TCP par laquelle elle reçoit de façon fiable les informations d'arrivées et de départs de participants dans la session, et une socket UDP (sans fiabilité) par laquelle les positions et orientations des participants sont transmises. L'utilisation de ces deux sockets au coeur de MiMaze3D se justifie principalement par ces deux arguments : Tout d'abord, cela permet un fonctionnement totalement asynchrone entre le module de communication et le module de rendu. Mais également, cela nous permet d'isoler les deux modules sur deux machines différentes (sur le même réseau local). En conséquence, ces deux raisons nous permettent d'avoir un contrôle plus précis sur la découverte du goulot d'étranglement de l'application lors de l'étude de la scalabilité.

### 4.3 Identification des problèmes de scalabilité

Afin de mesurer la résistance au facteur d'échelle (i.e. la scalabilité) de MiMaze3D, une série d'expérimentations a été effectuée sur réseau local et sur le MBone, le Backbone Multipoint de l'Internet. La complexité des algorithmes présents dans MiMaze a été évaluée ainsi que les performances du jeu en fonction du nombre de participants. Dans MiMaze, le navigateur VRML représente le moteur de rendu de l'application. Par conséquent, la fréquence de rafraîchissement de la scène 3D dépend du nombre d'entités virtuelles à afficher, du navigateur VRML, de la puis-

sance de calcul de la machine-hôte ainsi que de la capacité de la carte graphique à accélérer le traitement temps-réel des primitives 3D de la librairie graphique utilisée par le moteur de rendu (OpenGL ou Direct3D). L'applet Java jouant le rôle d'interface entre le navigateur VRML et le module de communication, nous avons conçu un algorithme spécifique d'ordonnement dans l'applet, permettant de préserver "au mieux" les caractéristiques temps-réel de l'application : A chaque fois que la scène VRML est mise à jour, seules les dernières positions reçues sont utilisées. Par exemple, si pendant une période de rafraîchissement de la scène 3D, trois positions du même joueur A ont été reçues. seule la plus récente sera utilisée, les autres étant tout simplement ignorées. Ainsi, aucune accumulation inutile d'ADUs au niveau du module graphique ne peut avoir lieu, et venir perturber nos mesures de performance lorsque le nombre de participants devient grand.

Etant donné que l'ensemble des participants communique sur un seul groupe multipoint, chacun d'entre eux reçoit l'intégralité du trafic circulant sur le groupe. Par conséquent, le trafic reçu croît en  $O(N)$  où  $N$  est le nombre de participants. En revanche, ce qui est intéressant de mesurer, c'est l'évolution du pourcentage d'information utile reçue et l'évolution des performances de l'application. Nous mesurons les performances en nous basant sur deux critères : 1/ La fréquence de rafraîchissement du navigateur VRML qui reflète la qualité perçue par l'utilisateur final. 2/ Le temps de traitement de l'ensemble des derniers paquets reçus, c'est-à-dire les positions les plus récentes de tous les participants. Ce dernier critère nous renseigne sur la complexité des algorithmes présents au sein de l'application.

Nous avons généré des traces au niveau d'un participant, sur un PC Pentium Pro 200 Mhz équipé d'une carte accélératrice 3D Diamond Fire OpenGL 1000 Pro, sous Windows NT. Le navigateur VRML CosmoPlayer 2.1 a été retenu, celui-ci permettant de profiter de l'accélération graphique fournie par la carte graphique. Sur la Figure 4.4, nous pouvons observer que le temps de calcul nécessaire afin de traiter l'ensemble des derniers paquets reçus de tous les participants,

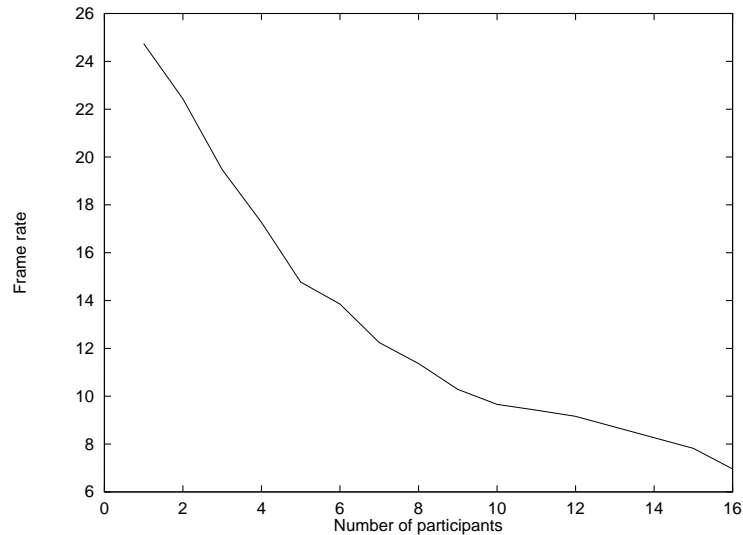


FIG. 4.3 – *Qualité perçue par un participant type*

augmente linéairement avec le nombre de participant. Cependant, la limite est vite atteinte car le calcul de rendu représente le coût principal au niveau CPU. En effet sur la Figure 4.3, on remarque qu'à partir de 10 participants, le taux de rafraîchissement du navigateur VRML passe en dessous de 10 trames par secondes, ce qui occasionne une gêne de plus en plus importante pour l'utilisateur. Ajouter plus de puissance CPU à notre machine d'expérimentation et utiliser des bibliothèques graphiques de plus bas niveau que VRML pourrait évidemment améliorer les performances mais ne résoudra en rien les problèmes de scalabilité. En effet, dans les EVGEs, le nombre de participants n'est pas de l'ordre de la dizaine, mais du millier voire même de la dizaine de milliers. La Figure 4.5 nous montre le pourcentage de trafic utile reçu en fonction du nombre de participants. Ce pourcentage est calculé en regardant le contenu de chaque ADU transmis au module graphique. Si, l'ADU contient la position d'une entité virtuelle pouvant interagir avec le participant, celui-ci est comptabilisé comme une information utile, sinon comme une information inutile. Nous avons pu observer au cours des sessions d'expérimentation que les participants se regroupaient par groupe de 3 ou 4 dans le labyrinthe virtuel. Dès qu'une entité virtuelle est "tuée" par une autre, elle réapparaît 1 seconde plus tard avec une nouvelle position tirée aléatoirement selon une loi de distribution uniforme dans le plan. Pour bien comprendre les résultats obtenus,



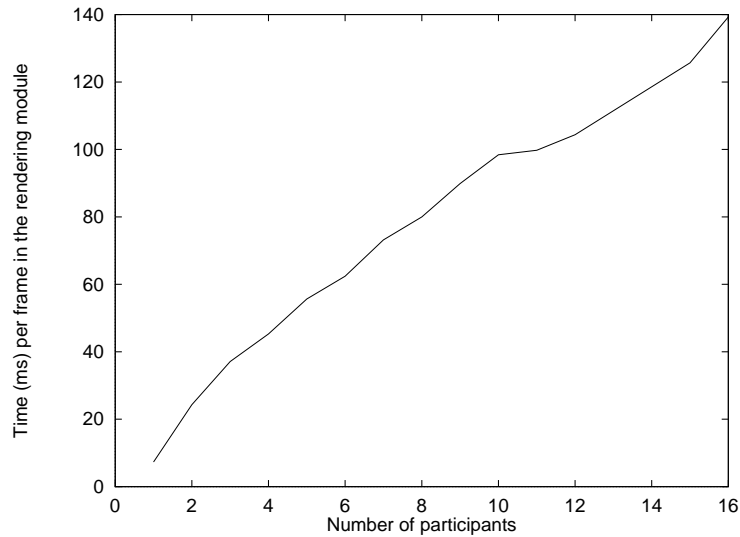


FIG. 4.4 – *Temps nécessaire au traitement des dernières positions reçues*

il faut savoir que la surface représentant le centre d'intérêt d'un participant occupe approximativement un dixième de la surface globale de l'environnement virtuel (le labyrinthe). Sur cette figure, on s'aperçoit qu'avec un nombre de participants compris entre 3 et 12, ce pourcentage est compris entre 18% et 25%. Ceci s'explique par le fait que la densité de participants dans le plan est telle que de petits groupes peuvent se former rapidement et assez naturellement. En revanche, lorsque le nombre de participants augmente, ce pourcentage chute progressivement, traduisant à la fois l'augmentation du nombre de petits groupes de participants en train d'interagir, et la stabilité du nombre de participants au sein de ces petits groupes. En effet, de part la nature et la dynamicité du jeu, il devient difficile pour un participant d'interagir simultanément avec plus de 4 autres entités virtuelles.

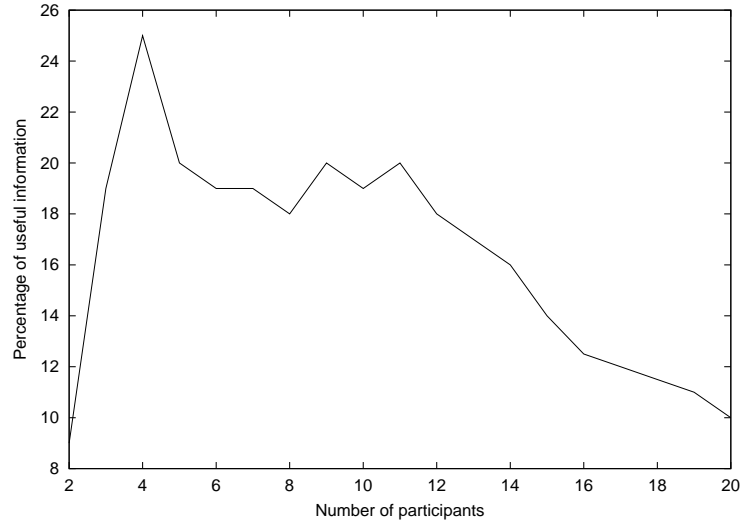


FIG. 4.5 – *Pourcentage de trafic utile reçu*

## 4.4 Conclusion

L'implémentation et les expérimentations de MiMaze3D nous ont permis d'identifier de façon précise les problèmes de scalabilité rencontrés aujourd'hui par tous les concepteurs de jeu vidéo multi-joueur. De plus, le cas particulier des jeux vidéo, où les contraintes d'interactivité sont extrêmement fortes (en comparaison avec un forum virtuel, ou un espace virtuel de bavardage de type "chat"), fait apparaître ce problème encore plus rapidement. L'architecture distribuée de MiMaze ne résout donc pas ce problème de scalabilité, même si en comparaison avec une architecture centralisée, un grand pas avait déjà été franchi grâce à l'utilisation du multipoint. Toutefois, il nous apparaît de manière évidente que l'utilisation d'un groupe multipoint unique n'est pas suffisant si l'on veut donner aux EVGEs la capacité d'accueillir de façon scalable un nombre croissant de participants.



## Chapitre 5

# Découpage de l'espace virtuel

L'utilisation de plusieurs groupes multipoints au sein d'une architecture de communication pour EVGEs nous fait entrer au cœur d'un nouveau problème qu'il convient de résoudre le plus efficacement possible : Comment découper l'espace virtuel d'un EVP afin de permettre aux participants de limiter le trafic qu'ils reçoivent au trafic émis par les autres participants avec lesquels ils peuvent interagir ? Quel est l'impact de ce découpage sur le trafic entrant au niveau de chaque participant ? Nous faisons l'hypothèse ici que chaque participant génère un flot unique de débit identique. Les réponses que nous apportons au problème peuvent cependant facilement s'adapter aux applications plus complexes qui nécessitent, par exemple, un codage en couches des données transmises.

### 5.1 Principes du découpage en cellules

La stratégie de regroupement basée sur le découpage en cellules consiste à répartir les participants dans plusieurs groupes multipoints, en fonction de leur intérêt commun. Avant de regrouper les participants dans ces groupes multipoints, il convient donc d'identifier pour chaque participant les données qui l'intéressent. Nous définissons donc ici le concept *d'intérêt du participant*, comme l'ensemble d'entités virtuelles avec lesquelles le participant peut interagir. Il est important de remarquer ici que les entités virtuelles situées dans le champ de vision du participant ne

représentent qu'une partie des entités présentes dans sa *zone d'intérêt*. Le découpage d'un environnement virtuel revient donc à découper celui-ci en cellules et à associer un groupe multipoint à chacune de ces cellules.

Au cours de leur déplacement dans l'environnement virtuel, les participants sont donc amenés à s'abonner et à se désabonner dynamiquement à ces groupes multipoint. Toutefois, on peut séparer ces groupes multipoints en deux parties : le groupe multipoint au sein duquel le participant est à la fois émetteur et récepteur, et les groupes multipoints au sein desquels le participant est uniquement récepteur. Tout au long de la session, chaque participant identifie la cellule dans laquelle il se trouve et s'abonne au groupe multipoint qui lui est associé. Ce groupe multipoint est le seul groupe multipoint dans lequel le participant émet ses données. Le participant est donc à la fois source et récepteur dans ce groupe. Afin de recevoir les données en provenance des autres participants avec qui il peut interagir, chaque participant doit également s'abonner à tous les groupes multipoints associés aux cellules qui intersectent sa zone d'intérêt. Cette zone d'intérêt représente l'espace virtuel de "proximité" qui entoure le participant. Les entités virtuelles présentes à l'extérieur de cet espace ne peuvent pas, par définition, interagir avec le participant. En ce qui concerne ces derniers groupes multipoints, le participant est cette fois-ci uniquement récepteur de données.

De la même manière, lorsqu'un participant se déplace, celui-ci est amené à se désabonner des groupes multipoints associés aux cellules qui n'intersectent plus sa zone d'intérêt. Cette stratégie de découpage en cellules est donc particulièrement adaptée aux environnements virtuels représentant des espaces virtuels euclidiens, avec une métrique implicite de distance virtuelle similaire à celle du "monde réel". Cependant, la difficulté principale dans ce découpage est de trouver la taille de cellule la plus appropriée. En effet, opter pour des cellules de petites tailles a pour conséquence d'augmenter le surcoût associé aux abonnements et désabonnements dynamiques, tandis qu'opter pour une plus grande taille de cellule entraîne une augmentation du trafic inutile

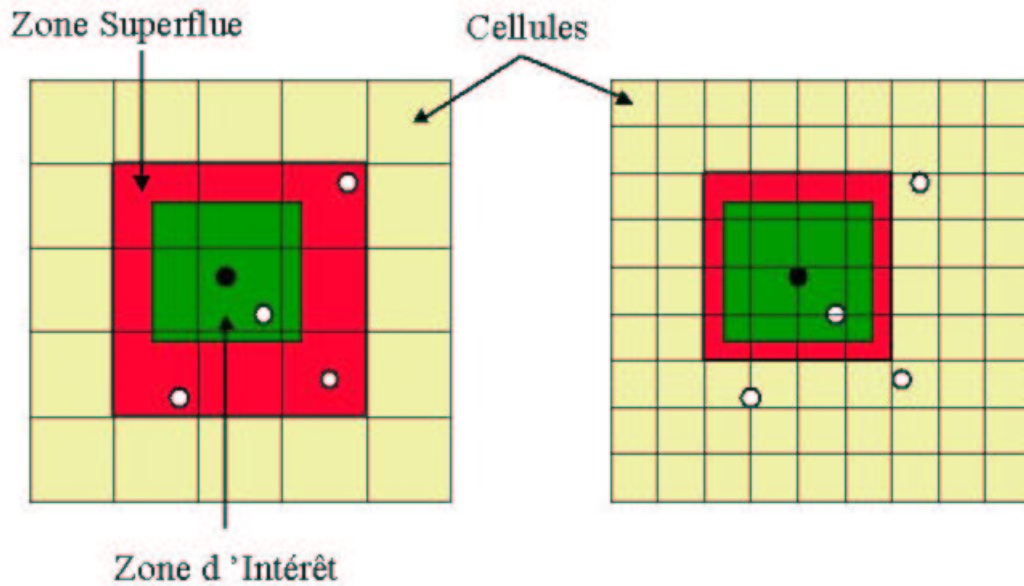


FIG. 5.1 – *Différentes Tailles de cellule*

reçu par participant [HRC94]. Dans la section suivante, nous examinons en détail les divers paramètres entrant en ligne de compte dans la sélection du meilleur compromis de taille de cellule. Afin de simplifier l'analyse, nous faisons ici l'hypothèse que l'ensemble des cellules de l'EVGE ont une taille identique.

## 5.2 Compromis sur la taille de cellule

Deux approches peuvent être considérées dans le calcul de la meilleure taille de cellule au sein des EVGEs. La première approche consiste à pré-calculer une taille statique de cellule pour toute la durée de la session. La seconde approche revient à calculer et re-estimer périodiquement la taille de cellule en cours de session, en prenant cette fois-ci en compte les paramètres dynamiques de l'EVGE [LTB99]. Afin de motiver l'utilisation de l'une ou l'autre de ces approches, il convient dans un premier temps d'identifier les divers paramètres impliqués dans l'estimation de la meilleure taille de cellule, puis, dans un second temps, d'examiner l'impact de la dynamique de ces paramètres (i.e. l'évolution dans le temps des valeurs prises par ces paramètres) sur la

taille de cellule la plus appropriée. Les paramètres suivants doivent être identifiés afin de trouver la taille de cellule la plus appropriée :

- **Le nombre de groupes multipoints disponibles** est un paramètre essentiel. En effet, dans la Section 3.4, nous avons expliqué pourquoi il ne serait pas réaliste de faire l’hypothèse que les EVGEs puissent disposer d’un nombre illimité de groupes multipoints. Il est en effet beaucoup plus réaliste de considérer qu’un ensemble fini et par conséquent limité de groupes multipoints est affecté par EVGE. Ainsi, le nombre de groupes multipoints disponible par l’application représente un paramètre important à prendre en compte dans le calcul de la taille de cellule puisqu’il donne une borne inférieure à celle-ci. Etant donné que le nombre de groupes multipoints utilisés est inversement proportionnel à la taille de cellule, allouer un plus petit nombre de groupes multipoints reviendrait à opter pour une plus grande taille de cellule. L’utilisation de l’ensemble des groupes multipoints disponibles, reviendrait en effet à définir une taille de cellule égale au ratio entre la surface totale de l’environnement virtuel et le nombre de groupes multipoints disponibles.
- **La capacité en réception des participants** représente à la fois la capacité des liens qui relient les participants à l’internet, et la puissance de calcul disponible par participant. Typiquement, ce paramètre limite la quantité de trafic que les participants peuvent supporter et traiter en réception. En faisant l’hypothèse que chaque utilisateur génère la même quantité de trafic en émission, le trafic entrant pour un participant augmente linéairement avec le nombre total de sources présentes au sein des groupes multipoints auxquels celui-ci s’est abonné. En d’autres termes, la quantité de trafic entrant pour un participant est fonction d’une nombre d’entités virtuelles présentes dans les cellules intersectant (et parfois totalement incluses) dans sa zone d’intérêt. Cependant, certaines de ces entités peuvent être situées en dehors de cette zone d’intérêt bien qu’à l’intérieur d’une cellule intersectant la zone d’intérêt du participant. Le ratio entre ce nombre d’entités "inintéressantes" et le nombre total de sources présentes dans les groupes multipoints souscrits, représente le pourcentage de trafic entrant *superflu*. Par conséquent, la taille de la cellule et plus par-

ticulièrement, le ratio entre la taille de la cellule et la taille de la zone d'intérêt, ont un impact direct sur la quantité de trafic entrant superflu. Or, les EVGEs sont partagés par un grand nombre de participants dont les connectivités réseaux et les puissances de calcul peuvent être très hétérogènes. Par conséquent, la limitation du trafic superflu entrant (grâce à l'utilisation de tailles de cellule plus petites) peut avoir un impact plus ou moins important sur leur perception qualitative de l'environnement virtuel, selon leur capacité à recevoir et traiter en temps-réel un nombre important d'entités virtuelles (on parle ici du trafic généré par ces entités).

- **La densité de participants** par unité de surface est égale au ratio entre le nombre de participants et la taille de l'environnement virtuel. Dans la stratégie de découpage en cellules, la zone d'intérêt est approximée par le plus petit ensemble de cellules recouvrant la zone d'intérêt. Nous proposons le concept de *zone superflue*, en opposition avec le concept déjà existant de zone d'intérêt, en définissant la zone superflue comme la différence entre la zone recouverte par cet ensemble de cellules, et la zone d'intérêt, voir la figure 5.1. La densité de participants dans l'environnement virtuel a donc non seulement un impact sur le nombre moyen de participants présents à l'intérieur de la surface d'intérêt mais également sur le nombre de moyen de participants présents à l'intérieur de la zone superflue (on se place ici au niveau d'un participant quelconque). En d'autres termes, la densité de participants, la taille de la surface d'intérêt et la taille de cellule sont trois paramètres influant directement sur la quantité de trafic superflu entrant pour un participant. Une plus petite taille de cellule permet une meilleure approximation de la surface d'intérêt en diminuant la zone superflue. En fonction de la densité de participants dans l'environnement virtuel, cet meilleure approximation aura par conséquent un impact plus ou moins important sur la réduction du trafic superflu.
- **La vitesse des participants** intervient également dans la stratégie de découpage en cellules car elle permet d'estimer le surcoût généré lorsque les participants franchissent de nouvelles cellules et s'abonnent aux groupes multipoints associés. De même, la fréquence



à laquelle les participants sont amenés à se désabonner des groupes multipoints dépend non seulement de la taille des cellules mais également de la vitesse des participants. La vitesse permet également d'évaluer le temps moyen qu'un participant passe par cellule, en fonction de la taille de cellule. De plus, nous avons expliqué dans la Section 3.4.2. que l'abonnement à un groupe multipoint est une opération qui peut prendre un délai non négligeable. Or les participants ne doivent en aucun cas souffrir de ce délai car le caractère interactif de l'application doit à tout pris être préservé. Ainsi, la solution consiste à anticiper l'abonnement d'un temps égal à une latence d'abonnement. Pour cela, chaque participant doit s'abonner à tous les groupes multipoints associés aux cellules dans lesquelles il peut se rendre pendant un temps égal à la latence de l'abonnement. Par conséquent, la vitesse des participants ainsi que la taille des cellules ont un impact direct sur le nombre de groupes multipoints qu'ils doivent joindre par anticipation et donc sur le trafic IGMP émis et le surcoût généré au sein du réseau.

### 5.3 Modèles et Simulations

Dans cette section, nous allons présenter des modèles de zone d'intérêt et de participants basés sur des processus ponctuels aléatoires inspirés de l'approche de géométrie aléatoire proposées dans [BZ99]. Ce modèle nous permet d'évaluer diverses valeurs moyennes particulièrement intéressantes et d'adresser certains problèmes relatifs à la mobilité des participants dans un environnement virtuel. Nous avons réalisé une série de simulations à partir des résultats de ce modèle. Ces simulations nous ont ensuite permis de mieux appréhender la problématique posée par la détermination de la taille de cellule la plus appropriée, dans le cas d'une stratégie de regroupement de participants basée sur un découpage en cellules de l'environnement virtuel.

### 5.3.1 Participants statiques

Dans cette partie, nous commençons par restreindre le problème en posant les hypothèses suivantes:

- Les cellules forment une grille carrée, régulière et infinie dans le plan.
- Les participants sont statiques et répartis dans le plan selon un processus de Poisson ponctuel et homogène d'intensité  $\lambda$  [Kin93].
- La zone d'intérêt d'un participant forme un carré de surface  $r^2$ , centré sur le participant.

Par la suite, nous utiliserons le terme *IArea* pour faire référence à cette surface.

Nous dénotons par  $s$  la taille des cellules (i.e. la distance qui sépare verticalement et horizontalement deux frontières adjacentes d'une même cellule) et par *CellArea* la surface des cellules  $s^2$ .

Nous nous concentrons dans un premier temps sur la distribution du nombre  $M$  de cellules intersectant la zone d'intérêt d'un participant typique, et sur  $N$ , le nombre de participants présents dans ces cellules. Soit  $\lfloor x \rfloor$ , la fonction qui retourne la partie entière du nombre réel  $x$ , c'est à dire le plus grand entier inférieur ou égale à  $x$ . Soient

$$k = k(r, s) = \lfloor \frac{r}{s} \rfloor \quad (5.1)$$

$$p = p(r, s) = \frac{r}{s} - \lfloor \frac{r}{s} \rfloor. \quad (5.2)$$

On a donc  $0 \leq p < 1$ . Nous démontrons ci-dessous que

1. La loi de  $M$  suit une distribution discrète sur les trois entiers  $(k + 1)^2$ ,  $(k + 1)(k + 2)$  et  $(k + 2)^2$ , avec les paramètres suivants:

$$P[M = (k + 2)^2] = p^2, \quad (5.3)$$

$$P[M = (k + 1)(k + 2)] = 2p(1 - p), \quad (5.4)$$

$$P[M = (k + 1)^2] = (1 - p)^2. \quad (5.5)$$

2. La fonction génératrice de  $N$  nous est donnée par la formule suivante:

$$\begin{aligned}
E[z^N] &= p^2 e^{-\lambda s^2 (k+2)^2 (1-z)} \\
&+ 2p(1-p) e^{-\lambda s^2 (k+1)(k+2)(1-z)} \\
&+ (1-p)^2 e^{-\lambda s^2 (k+1)^2 (1-z)}.
\end{aligned} \tag{5.6}$$

A partir des propriétés bien connues des processus de Poisson homogènes, la configuration vue par un participant type (que nous plaçons au point  $(\frac{r}{2}, \frac{r}{2})$  sans perte de généralité, de sorte que la surface  $I\text{Area}$  recouverte par la zone d'intérêt forme le carré  $[0, r] \times [0, r]$ ), est telle que la grille a un de ses points d'intersection en  $(X, Y)$ , où  $X$  et  $Y$  sont des variables aléatoires indépendantes uniformément distribuées sur l'intervalle  $[0, s]$ . Dans une configuration de la sorte, si  $X \leq x_0$ , où  $x_0$  est défini par la relation

$$x_0 = r - s \lfloor \frac{r}{s} \rfloor, \tag{5.7}$$

alors le nombre de cellules qui intersectent les côtés horizontaux de la zone d'intérêt est exactement égal à  $k + 2$  (voir la définition de  $k$  en (5.1)). Si  $X > x_0$ , ce nombre est désormais de  $k + 1$ . Il en va de même pour le nombre de cellules intersectant les côtés verticaux de la zone d'intérêt. En utilisant les propriétés d'indépendance et d'uniformité, nous obtenons avec la probabilité  $(\frac{x_0}{s})^2 = p^2$ , que le nombre de cellules intersectant la zone d'intérêt est égal à  $(k + 2)^2$ . Nous obtenons de façons similaires les probabilités pour (5.4) et (5.5).

Nous démontrons maintenant les résultats obtenus pour la fonction génératrice de  $N$ . Soit

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

où les variables aléatoires  $N_i$  représentent les nombres de participants présents dans les cellules  $C_i$  intersectant la zone d'intérêt. Chacune des ces variables suit une distribution Poissonienne de paramètre  $\lambda s^2$ . De plus, les variables  $N_i$  et  $M$  sont indépendantes. Par conséquent, nous pouvons appliquer ici la loi nous donnant la fonction génératrice d'une somme de variables aléatoires, qui

nous dit que la fonction génératrice de  $N$  est  $\psi(\phi(z))$  où  $\phi$  est la fonction génératrice de  $N_1$  et  $\psi$  la fonction génératrice de  $M$  [Fel71]. On a donc

$$\phi(z) = e^{-\lambda s^2(1-z)} \quad (5.8)$$

et

$$\psi(z) = p^2 z^{(k+2)^2} + 2p(1-p)z^{(k+1)(k+2)} + (1-p)^2 z^{(k+1)^2}, \quad (5.9)$$

ce qui nous permet d'obtenir immédiatement la formule (5.6).

Comme conséquences directes de ces formules, nous pouvons en déduire les expressions mathématiques suivantes:

1. La valeur moyenne de  $M$  et sa variance:

$$E[M] = p^2(k+2)^2 + 2p(1-p)(k+1)(k+2) \quad (5.10)$$

$$+(1-p)^2(k+1)^2 \quad (5.11)$$

$$\text{var } M = p^2(k+2)^4 + 2p(1-p)(k+1)^2(k+2)^2 \quad (5.12)$$

$$+(1-p)^2(k+1)^4 - (E[M])^2. \quad (5.13)$$

2. La valeur moyenne de  $N$ :

$$E[N] = \lambda s^2 E[M]. \quad (5.14)$$

3. La variance de  $N$ :

$$\begin{aligned} \text{var}(N) &= E[M]\text{var } N_1 + E[N_1]^2 \text{var } M \\ &= E[M]\lambda s^2 + \text{var } M \lambda^2 s^4. \end{aligned} \quad (5.15)$$

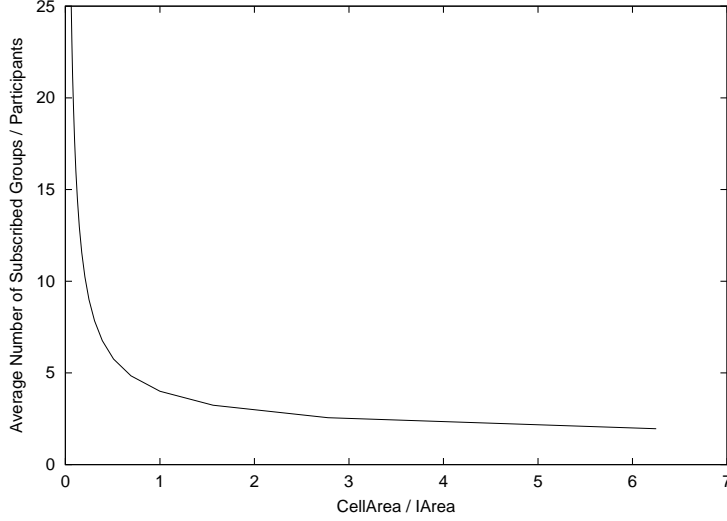


FIG. 5.2 – *Nombre Moyen de Groupes abonnés par Participant*

4. La probabilité que  $N$  est inférieur à une certain seuil  $n$ , où  $n$  est un entier non négatif:

$$\begin{aligned}
 P[N \leq n] &= p^2 g_n((k+2)^2) \\
 &\quad + 2p(1-p)g_n((k+1)(k+2)) \\
 &\quad + (1-p)^2 g_n((k+2)^2),
 \end{aligned} \tag{5.16}$$

où

$$g_n(m) = e^{-\lambda s^2 m} \sum_{i=0}^n \frac{(\lambda s^2 m)^i}{i!}. \tag{5.17}$$

Nous avons réalisé une série de simulations à partir de ce modèle. Dans un premier temps, nous analysons l'impact de  $CellArea$  et de l'intensité de participant  $\lambda$  sur le trafic  $N$  reçu par participant, le nombre moyen de groupes multipoints souscrits, et le pourcentage de trafic superflu reçu. Dans le but d'obtenir les résultats les plus exploitables possible, nous observons l'impact des variations du ratio entre la surface de la cellule  $CellArea$  et la surface d'intérêt  $IArea$  (i.e.,  $\frac{s^2}{r^2}$ ). Nous rappelons que nous faisons l'hypothèse ici que chaque participant génère le même débit de

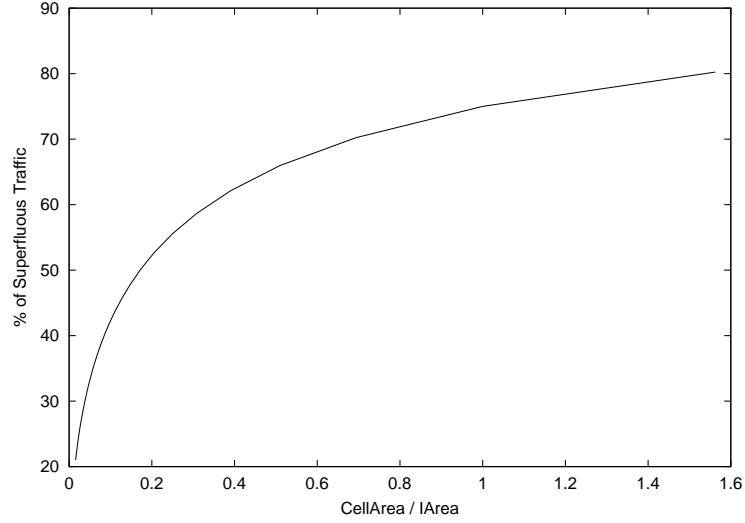


FIG. 5.3 – *Pourcentage de Trafic Superflu*

trafic. Même si cette hypothèse ne peut s’appliquer à tout type d’environnements virtuels, nous pensons que les résultats obtenus expriment assez clairement le compromis dans le choix de la taille de cellules en fonction de divers paramètres. De plus, si on prend le cas de jeux vidéo multi-utilisateur, cette hypothèse est bien souvent réaliste, tout du moins en moyennant le trafic généré.

La Figure 5.2 nous montre que le nombre moyen de groupes multipoints souscrits par participant  $E[M]$  diminue fortement lorsque  $CellArea$  est inférieur à  $IArea$  mais qu’il tend à prendre des valeurs de plus en plus proche (i.e. lorsque  $\frac{CellArea}{IArea}$  tend vers 1). En revanche, dès que la cellule devient plus grande que la zone d’intérêt, une augmentation de la cellule n’a que très peu d’effet sur le nombre moyen de groupe multipoints auxquels les participants sont abonnés.

Sur la Figure 5.3, nous avons tracé l’évolution du pourcentage moyen de trafic superflu reçu par participant, en faisant toujours varier le ratio  $\frac{CellArea}{IArea}$ . Etant donné que les participants sont répartis dans le plan selon un processus de Poisson ponctuel et homogène, ce pourcentage est égal à:  $1 - \frac{r^2}{E[M]s^2}$ . On observe sur cette courbe que lorsque  $CellArea$  est plus grand que  $IArea$ , plus de 70% du trafic est superflu. Ce pourcentage reste cependant acceptable si on le compare

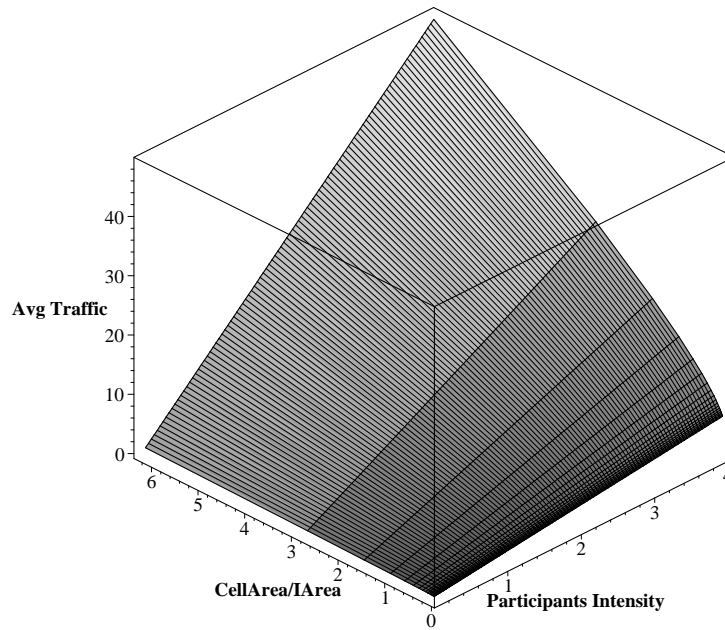


FIG. 5.4 – *Trafic Moyen Entrant (en nombre de sources) par Participant*

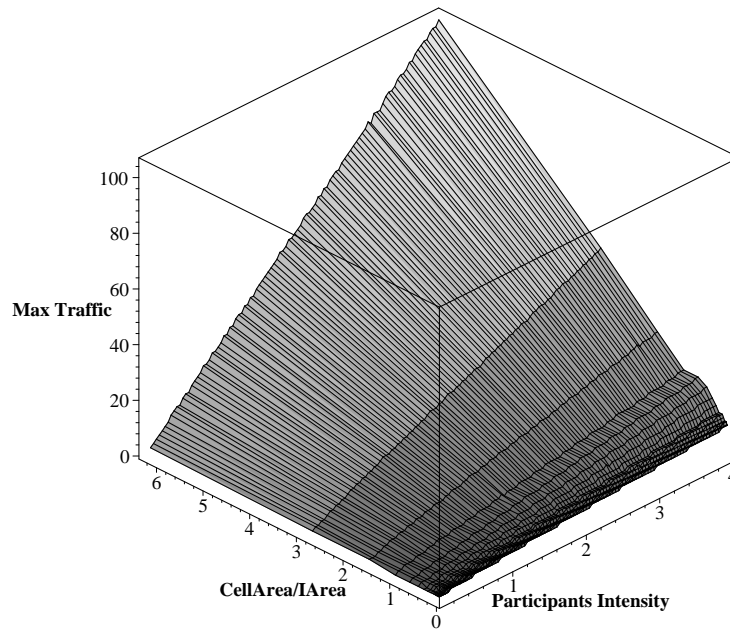


FIG. 5.5 – *Trafic Maximal Entrant (en nombre de sources) / Participant avec  $p = 0.95$*

à la situation où tout le trafic circule sur un groupe multipoint unique[LGD98]. Dans ce cas, et en reprenant les mêmes paramètres, on aurait moins de 1% de trafic utile à l'arrivée.

La Figure 5.4 nous donne le trafic moyen reçu par participant, en fonction de l'intensité de participants dans l'environnement virtuel, et en fonction du ratio entre *CellArea* et *IArea*. Nous exprimons ici l'intensité de participants par le nombre moyen de participants contenus dans une surface équivalente à la surface de la zone d'intérêt *ISize*, soit:  $\lambda r^2$ . Cette manière d'exprimer la densité (ou l'intensité si on se réfère à un processus de Poisson) de participants dans un environnement virtuel est très pratique. En effet, elle nous permet de faire varier *CellArea* indépendamment de la valeur de la densité. Les résultats affichés sur cette courbe nous montre que pour une valeur donnée d'intensité de participants, il est toujours possible de trouver le plus grand ratio possible entre *CellArea* et *IArea*, de telle sorte à ce que le trafic moyen entrant par participant reste en dessous d'un certain seuil. Avec une zone d'intérêt de taille fixe, cela revient à trouver la plus grande cellule possible, ou encore le plus petit nombre de groupes multipoints possible permettant de garantir qu'une certaine limite de trafic ne soit pas dépassé en moyenne. Le surcoût lié à l'utilisation du multipoint peut ainsi être réduit. Le trafic moyen nous est donné par la formule:  $\lambda s^2 E[M]$ .

Enfin, la Figure 5.5 nous donne sans doute les résultats les plus intéressants. En effet, si on veut préserver le plus souvent possible la satisfaction des participants au cours d'une session, il est plus judicieux de déterminer une taille de cellule appropriée, de sorte que le trafic entrant reste en dessous d'un certain seuil avec une forte probabilité. Cette probabilité peut être déterminée à l'avance par les administrateurs d'un environnement virtuel. Elle reflète le compromis existant entre le degré de satisfaction que l'on désire garantir aux utilisateurs, et le nombre de groupes multipoints nécessaires (et par conséquent le coût de l'architecture de communication adjacente). Sur la Figure 5.5, nous pouvons constater qu'il est toujours possible de trouver la plus grande taille de cellule possible, telle que le trafic entrant reste en dessous d'un seuil donné



avec une probabilité de 0.95. Nous remarquons également sur cette figure que pour une valeur donnée de *CellArea*, le trafic augmente linéairement avec l'intensité de participants.

### 5.3.2 Participants dynamiques

Dans cette section, nous introduisons un modèle de mobilité qui est compatible avec l'hypothèse que le processus ponctuel de participants est Poissonnien à tout instant. Ceci nous permet d'évaluer diverses valeurs moyennes particulièrement intéressantes, en relation avec la mobilité des participants dans un environnement virtuel. Nous étudierons plus particulièrement:

- Les entrées et sorties (i.e. abonnements et désabonnements) des groupes multipoints, que nous exprimons par l'intensité d'un processus ponctuel dans le temps, des franchissements de frontières des cellules associées par les participants en déplacement.
- Le temps de séjour moyen d'un participant type dans un groupe multipoint, soit le temps de séjour moyen passé dans la cellule correspondante.

Nous faisons toujours l'hypothèse que les participants sont initialement répartis dans le plan selon un processus de Poisson ponctuel d'intensité  $\lambda$ . Aucun participant n'entre ou ne quitte la session. Cependant, chaque participant se déplace dans le plan avec le mouvement aléatoire indépendant suivant: Une paire de variables aléatoires  $(V_i, \theta_i)$ , est associée avec le participant  $i$ , où  $V_i \in \mathbb{R}_+$  est la vitesse aléatoire d'un participant et  $\theta_i \in [0, 2\pi)$ , sa direction aléatoire. Nous faisons l'hypothèse que les paires  $(V_1, \theta_1)$  sont indépendantes, avec  $V_1$  de densité  $f$  sur  $\mathbb{R}_+$  et avec  $\theta_i$  uniformément distribuée sur  $[0, 2\pi)$ . Grâce au théorème de Kingman (see [Kin93], p. 61), le processus ponctuel donnant la position de tous les participants est encore un processus de Poisson ponctuel d'intensité  $\lambda$  à tout instant  $t$ . Par conséquent, les résultats de la section précédente sont toujours valides à tout ces instants.

Soit  $\sigma$  un segment de taille fixe  $u$ , que nous supposerons situé sur un axe horizontal sans perte de généralité. L'ensemble des participants ayant une paire de déplacement  $(v, \theta)$  qui franchissent  $\sigma$  entre les instants 0 et  $t$ , est initialement situé dans le parallélogramme de surface  $uvt|\sin \theta|$ .

L'ensemble des participants ayant une paire de déplacement  $[v, v + dv] \times [\theta, \theta + d\theta]$  est Poissonien avec l'intensité  $\lambda f(v)dv \frac{d\theta}{2\pi}$ . Par conséquent, le nombre moyen de participants franchissant  $\sigma$  entre les instants 0 et  $t$ , est égale à:

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

Considérons désormais une cellule type, soit le carré de périmètre  $4s$ . De ce qui précède nous obtenons l'expression suivante pour la valeur moyenne d'entrées-sorties de cette cellule par unité de temps:

$$H = \frac{8\lambda s E[V]}{\pi}. \quad (5.18)$$

Grâce au théorème de Kingman, nous pouvons toujours utiliser la loi de Poisson pour le nombre de participants dans cette cellule à tout instant. Sa valeur moyenne est:  $\lambda s^2$ . Comme l'intensité des entrées dans la cellule est  $\frac{H}{2}$ , la loi de Little nous permet d'obtenir l'expression suivante pour le temps de séjour moyen d'un participant dans un groupe multipoint type:

$$E[W] = \frac{2\lambda s^2}{H} = \frac{\pi s}{4E[V]}. \quad (5.19)$$

La Figure 5.6 nous montre que le temps de séjour moyen par cellule,  $E[W]$ , est fonction de la vitesse moyenne  $E[V]$  du participant. Nous exprimons ici la vitesse en cellules par seconde (on fait référence ici à la taille d'une arête de la cellule). Nous pouvons observer que le temps de séjour moyen décroît exponentiellement lorsque la vitesse moyenne tend vers 1 cellule par seconde. Ce résultat plaide en faveur d'une limitation de la vitesse (en cellule par seconde) dans les EVGEs, de sorte que le temps de séjour moyen soit un ordre de grandeur plus grand que le temps de latence d'un abonnement. En effet, chaque participant doit anticiper l'émission de ses requêtes d'abonnement IGMP pour les groupes multipoints associés aux cellules où il peut se rendre dans

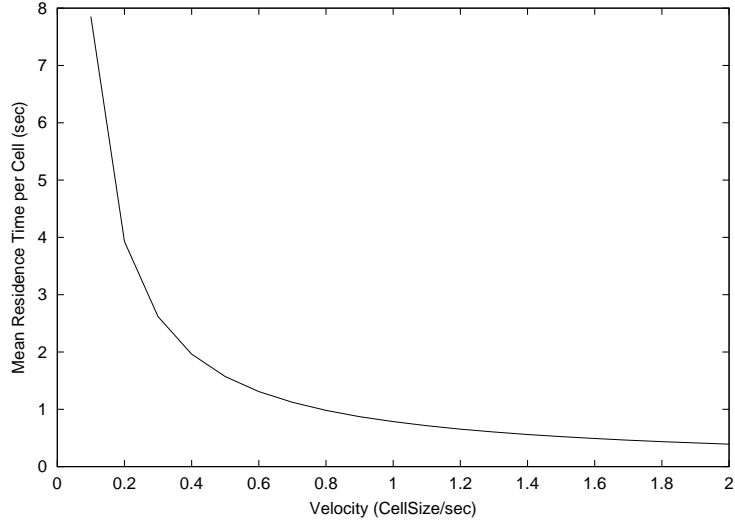


FIG. 5.6 – *Temps de séjour moyen par cellule*

un temps égal à la latence de l’abonnement. Par conséquent, la vitesse du participant ainsi que la taille de cellule ont un impact direct sur le nombre de groupes multipoints à joindre par anticipation, et donc sur le trafic IGMP généré.

## 5.4 Conclusion

Au cours de chapitre, nous avons présenté la stratégie de découpage en cellules d’un environnement virtuel. Nous avons identifié l’ensemble des paramètres qu’il est nécessaire de considérer afin de réaliser le découpage répondant le plus précisément possible aux contraintes et aux besoins de l’application. Enfin, nous avons étudié l’impact de la taille de cellule sur le trafic entrant au niveau des participants et sur l’utilisation des groupes multipoints associés aux cellules. Cette étude nous a permis de mesurer toutes les difficultés liées à la réalisation d’une architecture de communication multipoints pour les environnements virtuels pouvant accueillir un grand nombre de participants. En effet, ces participants peuvent présenter une très forte hétérogénéité dans leur capacités en traitement et en réception ainsi que dans leur intérêts concernant les données qu’ils désirent recevoir. Par conséquent, deux problèmes restent à résoudre au vu des résultats : Com-

ment gérer cette hétérogénéité et comment bâtir une architecture de communication capable de mettre en oeuvre les mécanismes liés à la stratégie de découpage en cellules?



## Chapitre 6

# SCORE : une architecture de communication scalable pour EVGEs

### 6.1 Introduction

Dans le chapitre précédent, nous avons montré que la taille de cellule devrait prendre des valeurs différentes en fonction de la densité de participants dans l'environnement virtuel et selon la quantité de trafic que les participants sont capables de recevoir et de traiter en temps-réel. Etant donné la forte hétérogénéité de capacité entre les participants d'un EVGE, ce résultat démontre clairement la nécessité d'un découpage dynamique de l'environnement virtuel en cellules de différentes tailles. De plus, chaque espace virtuel présente des caractéristiques qui lui sont propre. Par exemple, si un espace virtuel représente un immeuble de plusieurs étages avec plusieurs pièces et couloirs à chaque étage, il paraît difficile de réaliser un découpage sans tenir compte au préalable de ces caractéristiques architecturales. La répartition des participants à l'intérieur de l'espace virtuel doit également être considérée. Si deux participants se trouvent isoler dans la pièce A, tandis qu'une cinquantaine sont regroupés dans la pièce B, doit on découper ces deux pièces de la même manière? Quels critères doit on prendre en compte afin de définir la granularité la plus adéquate dans le découpage de l'espace virtuel [LT99]?

Dans ce chapitre, nous présentons SCORE, une architecture de communication multi-agent pour EVGEs basée sur l'utilisation d'un nombre limité de groupes multipoints. Nous avons implémenté cette architecture afin de démontrer sa faisabilité, d'évaluer ses performances, et d'en estimer son coût. Au cours de ce chapitre, nous détaillerons les différents composants de notre architecture, et tenterons d'en expliquer les mécanismes de la façon la plus complète possible. Le but de cette architecture est de permettre aux environnements virtuels d'être déployés de façon scalable sur l'Internet, et ce pour plusieurs milliers de participants hétérogènes. Nous tenterons également de démontrer pourquoi nous pensons que cette architecture peut fonctionner avec un nombre limité de groupes multipoints. Bien entendu, nous ne prétendons pas répondre à tous les problèmes dans toutes conditions. Notre approche se veut donc avant tout réaliste, c'est pourquoi nous tenterons d'en exposer également les limites. L'objectif de SCORE n'est pas de proposer un nouveau modèle pour IP multipoint, ou de proposer une approche au niveau réseau, en rajoutant de nouveaux mécanismes dans les routeurs. Nous présentons ici une solution au niveau de la couche transport, en supposant toutefois que chaque utilisateur est capable de recevoir et d'émettre du trafic multipoint. Afin de permettre à chaque participant de sélectionner l'information qu'il désire recevoir, nous proposons des mécanismes impliquant l'utilisation de plusieurs groupes multipoints. Chaque participant s'abonne et se désabonne dynamiquement aux groupes multipoints, en fonction de son intérêt pour le contenu des données en transit sur chaque groupe. Plusieurs travaux dans le domaine du multipoint fiable ont également préconisé l'utilisation de plusieurs groupes multipoints [KHTK99, FJL<sup>+</sup>97, HSC95]. Toutefois, nous ne présenterons pas ici de nouveaux protocoles de transmission multipoint fiable.

## 6.2 Métrique de satisfaction

Nous avons précédemment discuté sur la nécessité de définir une métrique de satisfaction des participants dans la Section 3.3 relative à la problématique de constitution des groupes. La situation idéale du point de vue de l'utilisateur final pourraient être définie comme celle où le trafic

entrant ne contient aucune information superflue. Cependant, cette situation n'est guère réaliste si on considère le coût du multipoint, et par conséquent la limitations du nombre de groupes multipoints disponibles par l'application. De plus, dans SCORE, nous faisons l'hypothèse qu'un certain nombre  $n$  de groupes multipoints a déjà été statiquement alloué à notre architecture de communication. Nous verrons plus tard comment intégrer une potentielle ré-allocation de nouveaux groupes en cours de session. Dans tous les cas, nous pouvons redéfinir ici la notion de "situation idéale", en tentant de faire "au mieux" avec le nombre de groupes multipoints dont on dispose initialement. Cette philosophie se retrouve d'ailleurs plus largement dans le monde IP, où l'on parle de service de type "*best-effort*" (le réseau fait de son mieux mais ne garantit rien).

Notre objectif consiste donc à donner aux utilisateurs la possibilité de filtrer le trafic entrant, en réduisant de façon significative la quantité de trafic superflu. Si on considère la notion d'intérêt au sein des environnements virtuels, définie par une zone d'intérêt entourant chaque participant, on aboutit à la constatation suivante: Plus une entité virtuelle a des chances d'interagir avec un participant, plus l'intérêt que porte le participant au trafic généré par cet entité devient important. La réciproque s'applique de la même manière. Moins une entité virtuelle a des chances d'interagir avec un participant, plus l'intérêt que porte le participant au trafic généré par cet entité devient faible, voire nul. Par conséquent, on peut affiner l'objectif précédemment défini en cherchant à limiter le plus que possible le trafic superflu à l'aide des ressources dont on dispose (les groupes multipoints), tout en permettant aux utilisateurs de filtrer de façon graduelle ce trafic. Ainsi le trafic superflu filtré en priorité, représentera le trafic généré par les autres participants avec lesquels la probabilité d'interagir est la plus faible. De même, si deux participants reçoivent un excédent trop important de trafic superflu, et que notre architecture de communication ne peut permettre qu'à l'un d'entre eux de "raffiner" son trafic entrant, alors le participant au niveau duquel le trafic superflu aura le plus d'impact négatif, sera préféré. Désormais, la prise en compte de l'hétérogénéité entre participants est intégrée à l'objectif dont se réclame notre architecture de communication.



Pour toutes les raisons que nous venons d'évoquer, nous proposons de définir notre métrique de satisfaction  $S$  par la formule suivante:

$$S = \frac{U_r}{\min(U_t, C)} \quad (6.1)$$

où  $U_r$  représente le débit d'*information utile* reçue et traitée. L'information utile (ou encore les données utiles) représente le trafic émis par les participants situés à l'intérieur de la zone d'intérêt d'un participant type.  $U_t$  représente le débit total d'information utile, reçue ou non reçue (limitation de la capacité du lien entrant), traitée ou non traitée (limitation de la puissance de calcul). Dans le cas d'une distribution des participants dans le plan selon un processus de Poisson ponctuel et homogène d'intensité  $\lambda$ , (voir Section 5.3),  $U_t$  est proportionnel à  $\lambda r^2$ . Enfin  $C$  est la capacité du participant à recevoir et à traiter l'information en temps-réel.  $C$  représente donc le débit maximal d'information que le participant est capable de supporter, limité par sa connectivité à l'Internet et par sa puissance de calcul.

Nous allons désormais tenter de traduire concrètement ce que signifie le choix d'une telle métrique, et de justifier son utilisation au sein de SCORE. Nous pouvons constater à la lecture de cette formule, que lorsqu'un participant reçoit et traite la totalité des informations utiles, sa satisfaction est maximale, quelque soit son débit entrant d'information superflue. Nous considérons donc ici que lorsque le trafic superflu entrant n'occasionne aucune perturbation sur la qualité perçue au niveau applicatif par l'utilisateur final, la satisfaction de ce dernier est maximale ( $S = 1$ ).

De plus, si pour un participant donné,  $U_r = C$ , alors  $S$  est également maximale. En effet, par définition, nous avons  $U_r \leq U_t$ . Compte tenu de l'égalité précédente, nous avons également  $C \leq U_t$ , donc  $\min(U_t, C) = C = U_r$ , et par conséquent  $S = 1$  même dans le cas où l'information utile n'a pas été reçue dans son intégralité. Nous justifions cette propriété par le fait

qu'aucune architecture de communication ne peut permettre à un participant de recevoir un débit d'information supérieur à sa capacité maximale. Si la taille de sa zone d'intérêt est telle que le trafic généré par les autres participants présents dans celle-ci dépasse cette capacité, alors aucune architecture de communication ne peut lui permettre de recevoir 100% du débit utile. En effet, même en approximant parfaitement sa zone d'intérêt grâce à des cellules de la plus petite taille possible, le débit entrant dépassera la limite définie par  $C$ . En revanche, des cellules de plus petites tailles lui permettront tout de même de diminuer son débit entrant d'information superflue.

Nous motivons le choix de la métrique de satisfaction  $S$  par le compromis nécessaire entre le débit d'information superflue entrant, l'état rajouté dans le réseau, et le surcoût associé aux abonnements et désabonnements dynamiques aux groupes multipoints. Avec cette métrique de satisfaction, le but de notre architecture ne consiste pas à s'adapter au récepteur le plus faible (en terme de connectivité réseau ou de puissance de calcul), mais de maximiser la satisfaction du participant ayant le plus petit  $S$ . Cette approche est souvent référencée par "l'équité max-min" (max-min fairness), et est décrite plus en détails dans [BG87].

## 6.3 Les agents

### 6.3.1 Définition des agents

SCORE est une architecture de communication multi-agent basée sur une stratégie de découpage en cellules utilisant plusieurs groupes multipoints. Les **agents** sont des machines ou des processus placés à différents endroits du réseau (e.g., sur le LAN d'un campus, hébergé chez un ou plusieurs ISP, etc.). Les administrateurs d'EVGEs ont la responsabilité de déployer graduellement ces agents en divers points stratégiques du réseau, c'est à dire à proximité des clients potentiels. Cette proximité s'exprime ici en terme de délai et non de distance physique.

Un des points essentiels de notre architecture réside dans le fait que ces agents ne reçoivent aucune des données que les participants se transmettent les uns aux autres. L'utilisation du terme "agents" plutôt que "serveurs" n'est donc pas fortuite. Les agents ne sont aucunement impliqués dans le calcul d'états, la synchronisation entre participants, ou dans la centralisation du trafic émis puis sa redistribution vers les participants concernés. Aucun trafic de données relatives au contenu des messages applicatifs (positions, orientations, messages textes, description, voix, images, etc..) ne converge vers ces agents. En ce sens, nos agents ne peuvent pas être comparés à des serveurs, de type serveur de jeux en-ligne par exemple; leur mission n'étant pas de calculer l'état global de l'environnement virtuel à tout instant, ni même de redistribuer un quelconque trafic.

Il est à noter que le modèle développé dans la Section 5.3 ne peut s'appliquer directement que si on suppose une distribution des participants dans l'EV selon un processus de Poisson ponctuel et homogène. Or, dans un EV réel, une telle distribution n'est guère réaliste. Toutefois, si nous découpons judicieusement l'EV en différentes **Zones** ou sous-parties, nous pouvons cette fois-ci nous rapprocher d'une telle distribution dans chacune des Zones, mais avec des intensités différentes pour chaque Zone. Le rôle des agents dans SCORE consiste donc à définir dynamiquement des Zones à l'intérieur de l'EV, en tenant compte de la répartition des participants, et à calculer des tailles de cellule appropriées en fonction des diverses densités de participants dans ces Zones. Les agents ont également la mission d'évaluer 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. Ces valeurs de satisfactions sont calculées très simplement, en utilisant le modèle Poissonien développé dans la Section 5.3 à l'intérieur de chaque Zone. Une fois ces satisfactions calculées, les agents peuvent de nouveau recalculer de nouvelles Zones (ou au contraire agréger certaines Zones) et modifier la taille des cellules dans les Zones où se trouvent les participants les moins satisfaits.

Ainsi, le rôle des agents consiste à réaliser un découpage dynamique de l'EV en cellules de tailles différentes, et à associer à chacune de ces cellules un groupe multipoint différent. En résumé, les agents ont donc comme mission d'utiliser au mieux les ressources dont ils disposent, à savoir les groupes multipoints, tout en répondant aux trois objectifs suivants:

- Permettre aux participants situés dans les régions virtuelles les plus peuplées, de mieux approximer leur zone d'intérêt grâce à des cellules de plus petites tailles, réduisant ainsi de manière significative le trafic superflu entrant.
- Permettre aux participants les moins satisfaits de réduire leur trafic superflu, augmentant ainsi leur satisfaction.
- Limiter le sur-coût lié aux messages de contrôle et de signalisation entre les participants et les agents.

### 6.3.2 Association cellules-groupes multipoints

L'association d'une cellule de l'environnement virtuel avec un groupe multipoint peut se décomposer en deux étapes. Premièrement, identifier et désigner avec précision la cellule dans l'environnement virtuel: sa taille, sa localisation, la région virtuelle dans laquelle elle se trouve, ainsi que les cellules avoisinantes et communicantes (i.e. il existe un chemin direct avec ces cellules). La deuxième étape consiste à associer une adresse IP multipoint unique à cette cellule. Nous désignons par **information de mise en correspondance**, le résultat obtenu à l'issue de ces deux étapes. Les agents, qui ont la responsabilité d'effectuer cette opération, doivent également permettre aux utilisateurs d'en connaître le résultat. Cependant, avant de communiquer ces informations de mise en correspondance aux utilisateurs, il est nécessaire de trouver un moyen d'identifier et de nommer les différentes cellules de l'environnement virtuel.

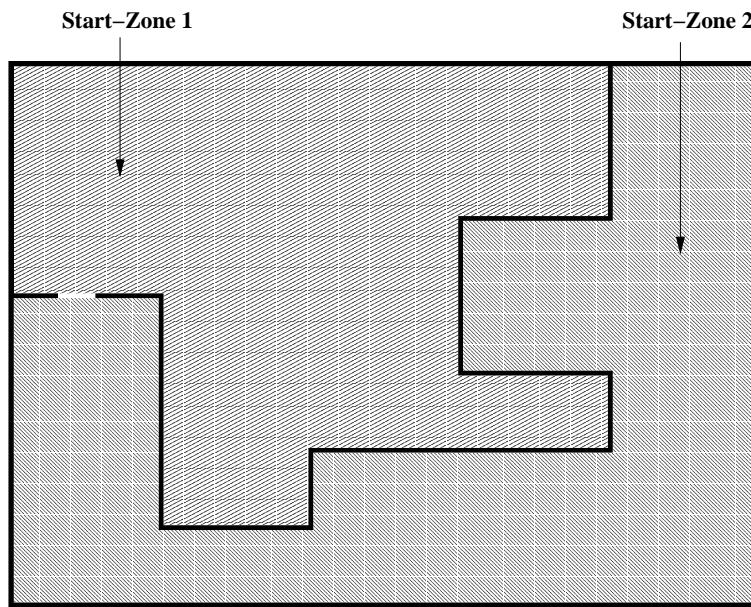


FIG. 6.1 – *Découpage Statique de l'EV en Start-Zones*

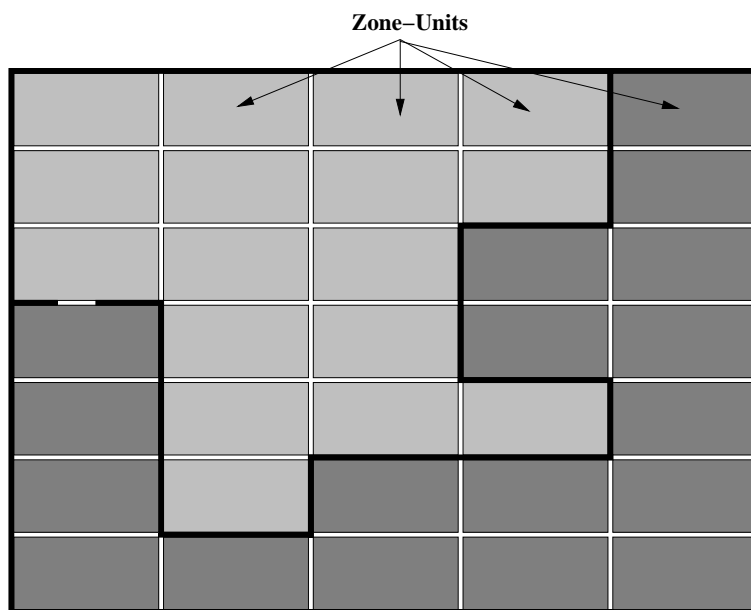


FIG. 6.2 – *Découpage Statique des Start-Zones en Zone-Units*

### 6.3.2.1 Désignation des cellules

Cette étape est particulièrement complexe de part la nature structurée d'un environnement virtuel et la dynamicité des tailles de cellule en cours de session. En effet, l'EV peut être composé de plusieurs étages, salles ou couloirs. Chacune de ses parties peut être isolée par des cloisons empêchant toute interaction entre des participants pourtant proche dans l'espace virtuel euclidien. Par conséquent, cette information spécifique à la structure de l'EV doit être pris en compte dans la procédure de désignation des cellules. De plus, dans une même salle virtuelle, le nombre de cellules recouvrant la salle peut changer en cours de session, entraînant une modification de la taille des cellules. Bien que parfaitement immobile, certains participants présents dans cette salle, vont changer de cellule courante après cette modification. Une nouvelle information de mise en correspondance devra donc leur parvenir afin qu'il puisse identifier la cellule dans laquelle il se trouve désormais et le groupe multipoint sur lequel il devront émettre leur trafic.

L'environnement virtuel est préalablement et statiquement découpé en plusieurs parties de grande taille, que l'on nommera **Start-Zone** dans la suite de la thèse. Ces Start-Zones sont en fait définies initialement en tenant compte de la nature intrinsèque et définitive de l'EV (Salle, Etage, Murs, etc..) (voir Figure 6.1). Chaque Start-Zone sera ensuite dynamiquement découpée en **Zones** en cours de session ; une Zone étant par définition une sous-partie de l'EV à l'intérieur de laquelle les cellules ont toutes la même taille (sauf éventuellement quelques cellules limitrophes). Bien entendu, deux Zones distinctes peuvent avoir des cellules de tailles différentes (voir Figure 6.5.

Afin de faciliter le découpage dynamique des Start-Zones en Zones, chaque Start-Zone est pré-découpée statiquement en **Zone-Units**, dont la taille est la taille minimum que peuvent prendre les Zones (voir Figure 6.2). Une fois que ce pré-découpage statique en Zone-Units est effectué, chaque Zone-Unit est référencée par l'identifiant de sa Start-Zone ainsi que par sa position dans celle-ci. Une matrice de probabilité d'interaction est alors construite entre les Zone-Units, prenant

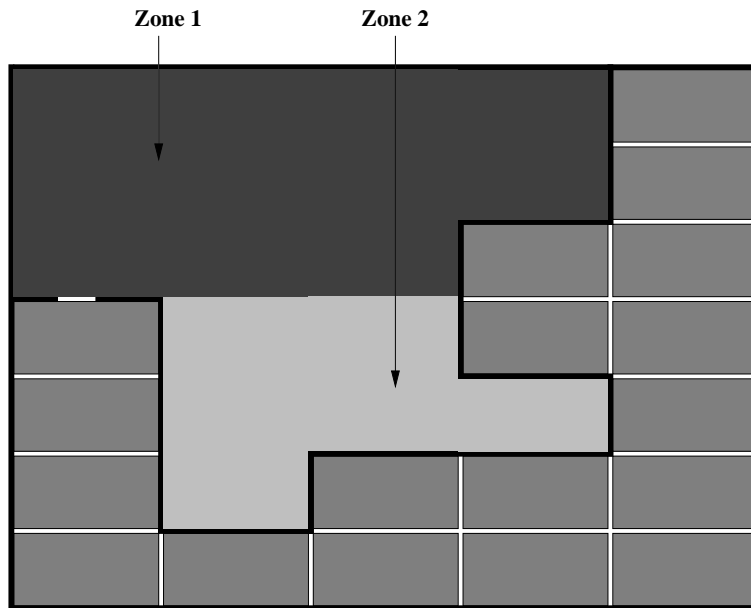


FIG. 6.3 – *Découpage Dynamique des Start-Zones en Zones (Ex.1)*

en compte la structure de l'EV. Les agents utilisent cette matrice pour définir dynamiquement de nouvelles **Zones**, afin de tenir compte du changement de la répartition des participants dans l'EV. Les agents désignent ces Zones par des ensembles constitués de  $n$  Zone-Units contiguës (avec  $n \leq 1$ ), appartenant à la même Start-Zone (voir Figures 6.4 et 6.3).

Grâce à ce découpage, une cellule peut être identifiée par les trois informations suivantes :

- La Zone dans laquelle elle se trouve.
- Sa position au sein de cette Zone.
- La taille des cellules dans cette Zone.

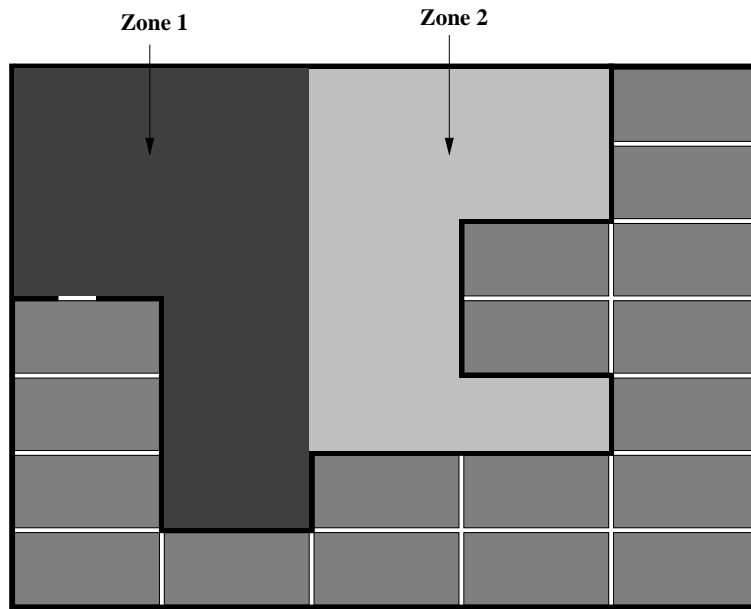


FIG. 6.4 – Découpage Dynamique des Start-Zones en Zones (Ex.2)

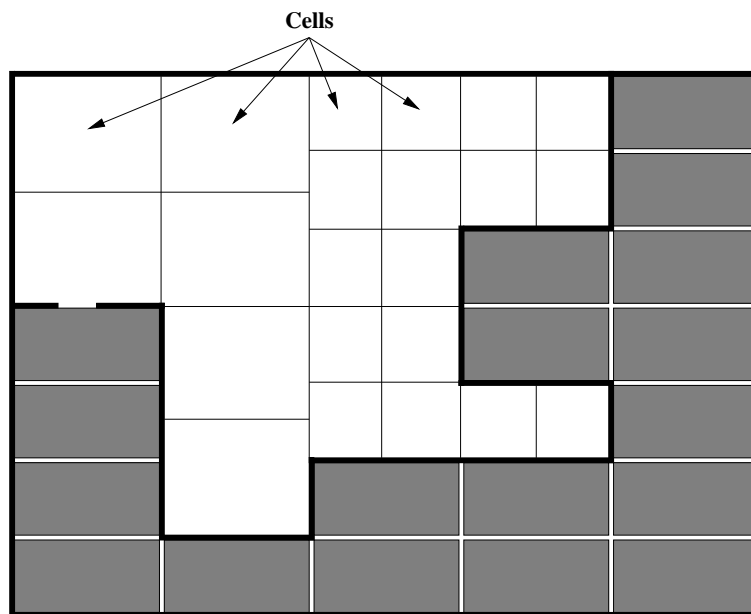


FIG. 6.5 – Découpage Dynamique des Zones en Cellules



### 6.3.2.2 Affectation des groupes multipoints

Avant toute connexion de participant, l'EV est prédécoupé en Start-Zones, et à chacune de ces Start-Zones est associé un groupe multipoint. En cours de session, quatre opérations successives sont réalisées par les agents :

- Découpage de chaque Start-Zone en plusieurs Zones, en fonction de la distribution des participants dans la Start-Zone.
- Calcul de la taille de cellule appropriée pour chaque Zone, en fonction des paramètres listés dans la Section 7.1.1 (voir Figure 6.6).
- Division de chaque Zone en cellules, en fonction de la taille de cellule précédemment calculée, et affectation d'une adresse IP multipoint à chaque cellule de chaque Zone.
- Diffusion vers les participants concernés des informations de mise en correspondance, afin que ceux-ci puissent s'abonner aux groupes multipoints associés aux cellules situées dans leur entourage.

Dans la suite de la thèse, nous ferons référence aux trois premières opérations en utilisant l'expression d'**algorithme de mise en correspondance**. Les agents ont connaissance de l'ensemble des informations de mise en correspondance pour toutes les cellules de l'EV. Lorsque les agents décident d'augmenter la taille de cellule pour une Zone de l'EV, ils peuvent "récupérer" certains groupes multipoints associés aux anciennes cellules de la Zone. Ces groupes multipoints peuvent ainsi être affectés à de nouvelles cellules dans des Zones où cette fois-ci une taille de cellules plus petite a été calculée. Il s'effectue ainsi un mécanisme de répartition des charges ou *load balancing*, entre le nombre de cellules des différentes Zones, et par conséquent des groupes multipoints qui leur sont associés. Le nombre total de groupes multipoints reste quant à lui constant si on considère l'environnement virtuel de façon globale.

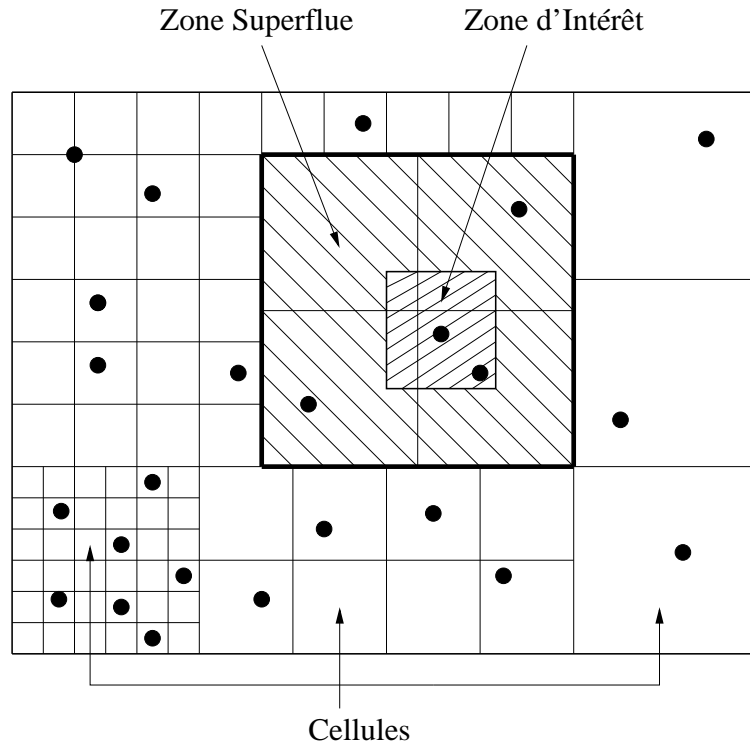


FIG. 6.6 – *Différentes tailles de cellule en fonction de la densité de participants*

### 6.3.2.3 Format des informations de mise en correspondance

Avant de rentrer dans ce qui constitue le coeur de notre architecture de communication, à savoir la description de ses protocoles de communication et des messages de signalisation échangés, nous allons nous intéresser au format des informations de mise en correspondance. Avant d'en définir la problématique et de proposer notre solution pour SCORE, il convient de rappeler deux points essentiels que toute architecture de communication doit considérer. Le premier d'entre eux concerne la bande passante utilisée, et dans certains cas réservée, au trafic de signalisation propre au bon fonctionnement de l'architecture. Cette bande passante doit être minimale et représenter un faible pourcentage de la bande passante totale utilisée par l'application. Dans tous les cas, le surcoût lié au trafic de signalisation, doit rester inférieur au gain, toujours en terme de coût, lié à la réduction du trafic de données superflues. Le deuxième point concerne la limitation du nombre de groupes multipoints disponibles pour l'application.

En effet, cette limitation peut être perçue à deux niveaux différents:

- Une limitation du nombre d'adresses IP multipoint disponibles. Nous avons précédemment discuté de ses raisons dans la Section 2.1.3.
- Une limitation du nombre de groupes multipoints disponibles pour des raisons liées au routage et à la dynamicité des participants au sein des groupes multipoints (voir Section 3.4).

Le premier point nous éclaire sur la nécessité de réduire au maximum la taille des informations de mise en correspondance transmises aux participants. De plus, chaque participant n'est intéressé que par un sous-ensemble des informations de mise en correspondance de l'EV. Ce sous-ensemble regroupe uniquement les informations de mise en correspondance concernant la région virtuelle dans laquelle le participant se trouve. Par conséquent, le trafic lié à la transmission de ces informations ne doit être diffusé que vers les intéressés dans la mesure du possible.

Le deuxième point n'a pas vraiment d'impact sur la problématique, mais plutôt sur les solutions envisageables pour y répondre. En effet, nous pouvons être confrontés à deux situations différentes avec des limitations plus ou moins contraignantes. Dans un premier cas, les deux limitations énoncées ci-dessus peuvent s'appliquer, tandis que dans un deuxième cas, la limitation sur le nombre d'adresses IP multipoint peut être ignorée (par exemple avec la mise en place d'IPv6).

Dans le cas où on peut s'affranchir de la limitation du nombre d'adresses multipoints disponibles, les informations de mise en correspondance pour les cellules d'une Zone peuvent être réduites à un couple de données: la taille de cellule pour cette Zone et un masque d'adressage pour les groupes multipoints associés. En revanche, la solution que nous proposons pour le premier cas, consiste à maintenir une liste des adresses de groupes multipoints disponibles au niveau des agents. Quand le nombre de cellules dans une Zone est diminué, les adresses IP multipoint des cellules supprimées sont ajoutées à cette liste. De la même façon, les agents vont "puiser"

dans cette liste à chaque fois qu'ils auront besoin d'affecter de nouvelles cellules à une Zone. Le problème avec cette solution est que rapidement les adresses des groupes multipoints d'une même Zone vont avoir des adresses IP non contiguës. Ceci implique que la transmission des informations de mise en correspondance pour les cellules de cette Zone, contienne l'ensemble et l'intégralité de ces adresses, ce qui ne va pas dans le sens d'une minimisation du trafic de signalisation. Des méthodes permettant d'agrèger certaines adresses contiguës peuvent cependant être utilisées. En plus de ces méthodes, des techniques de redistribution des adresses dans chaque Zone peuvent être périodiquement exécutées, et permettre ainsi une réduction conséquente des informations de mise en correspondance par Zone. Ces optimisations, similaires aux optimisations d'espace mémoire ou de gestion des secteurs d'un disque dur, ne seront pas développées dans cette thèse, car elles s'éloignent sensiblement de notre sujet.

## 6.4 Communications dans SCORE

### 6.4.1 Une architecture multi-agent doublement distribuée

SCORE est une architecture de communication dont l'objectif premier est de rendre scalable le déploiement des EVGEs sur l'Internet tout en prenant en compte l'hétérogénéité entre ses utilisateurs. Afin de répondre de façon réaliste à cet objectif, nous proposons l'utilisation de plusieurs groupes multipoints tout en considérant le coût et les limitations du multipoint. Les données générées par les participants sont donc émises sur ces différents groupes. A chaque groupe est attaché une cellule de l'EV comme une "étiquette" renseignant sur le contenu du trafic circulant sur celui-ci. Le calcul d'état, la synchronisation des données et les abonnements/désabonnements aux groupes multipoints sont laissés à la charge des participants. Ces groupes multipoints jouent donc le rôle d'outils de filtrage de trafic pour les participants.

Sur ce premier niveau d'architecture distribuée, nous insérons un second niveau de communication en ajoutant des agents. Leur mission consiste à associer des cellules de différentes tailles

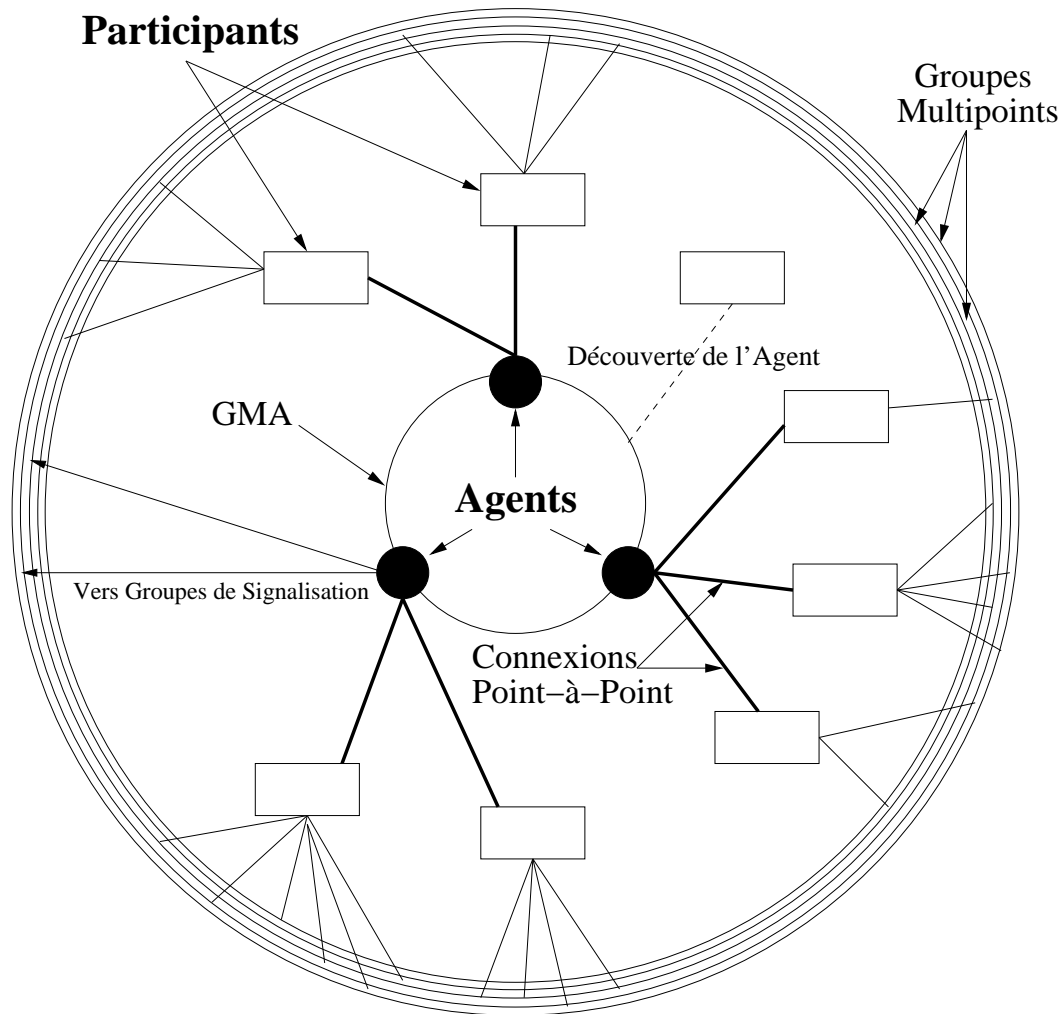


FIG. 6.7 – Une Architecture Multi-Agent Doublement Distribuée

aux groupes multipoints de façon à se rapprocher encore davantage de l'objectif précédemment énoncé. La présence des agents vient cependant ajouter un niveau de complexité à notre architecture. Les agents doivent transmettre aux participants les informations de mise en correspondance. En effet, ce sont ces derniers qui vont prendre la décision finale et décider quand s'abonner ou se désabonner aux groupes multipoints. Or, l'utilisation de plusieurs groupes de données dans notre architecture est précisément motivée par le besoin de ne pas transmettre l'intégralité du trafic vers tous les participants, et par la même de rendre scalable ce type d'application. Transmettre

l'intégralité des informations de mise en correspondance à tous les participants ne serait donc pas cohérent avec notre approche initiale.

Dans SCORE, nous utilisons la notion de découpage virtuel (voir Section 6.3.2), basée sur les concepts de cellule, Zone-Unit, Zone et Start-Zone, afin de permettre aux participants de réaliser non seulement un filtrage sur le trafic de données entrant mais également un filtrage sur le trafic de signalisation entrant (i.e., les informations de mise en correspondance). Tandis que les groupes multipoints, sur lesquels circulent les données générées par les participants sont associés aux cellules, des groupes multipoints de signalisation vont être associés aux Start-Zones. Cette association est cette fois-ci réalisée de façon statique. La correspondance entre les Start-Zones et les adresses des groupes de signalisation est donc connue par tous les participants dès leur connexion à l'EV. Le surcoût lié à l'utilisation de ces groupes est négligeable par rapport à celui lié aux groupes de données. Premièrement, le nombre de Start-Zones étant relativement faible par rapport au nombre de cellules, le nombre total de groupes disponibles pour l'application n'en est que faiblement diminué. Deuxièmement, la taille des Start-Zones étant beaucoup plus importante que la taille des cellules, la fréquence des abonnements et des désabonnements est par conséquent beaucoup plus faible. Enfin, une dernière remarque concerne le trafic de signalisation circulant sur ces groupes : Ce trafic est généré par les agents à un débit beaucoup plus faible (en théorie), que le trafic de données. Ce dernier point demande cependant plus d'explications. Nous reviendrons donc plus en détail là-dessus dans la suite de la thèse, les expérimentations réalisées venant également confirmer cette remarque.

L'architecture distribuée de SCORE ne se base pas sur la présence d'un ou plusieurs serveurs chargés de centraliser les données émises par les participants. Ceci constitue un aspect essentiel de la scalabilité de SCORE. Par conséquent, la présence d'agent dans SCORE ne doit pas entraîner une perte de scalabilité de part la convergence du trafic de signalisation émis par les participants vers un agent unique. Pour cette raison, SCORE est une architecture multi-agent *doublement*

*distribuée*. En effet, plusieurs agents sont impliqués dans notre architecture de communication, ce qui évite ainsi la convergence du trafic vers un goulot d'étranglement. De plus, les agents communiquent entre eux en émettant sur un groupe multipoint unique qui leur est réservé. De part son architecture doublement distribuée, SCORE permet donc d'accueillir simultanément un très grand nombre de participants. Notre architecture permet également l'ajout dynamique et transparent de nouveaux agents en cours de session. En effet, un nouvel agent a simplement besoin de connaître l'adresse du groupe multipoint réservé aux agents. Il lui suffit ensuite de s'annoncer sur ce groupe auprès des autres agents.

Sur la Figure 6.7, nous pouvons observer les différents niveaux de communication de notre architecture :

- Chaque participant s'abonne à un nombre  $n$  de groupes multipoints, avec  $n \geq 1$ , mais n'émet son trafic que dans un seul groupe.
- Chaque participant est connecté à un agent unique, via une connexion UDP point-à-point.
- Les agents communiquent entre eux sur un groupe multipoint unique auquel ils sont toujours abonnés : le Groupe Multipoint des Agents (*GMA*).
- Les nouveaux participants cherchant à se connecter, émettent des **paquets Hello** sur le groupe *GMA*.

#### 6.4.2 Communication participants-participants

Les participants n'ont pas de connaissance a-priori de la liste des autres participants connectés au LSVE. Aucune connexion point-à-point n'est établie entre les participants. De plus, la répartition physique des utilisateurs sur le réseau n'est pas une information nécessaire au bon fonctionnement de l'application. Dans SCORE, les utilisateurs n'effectuent aucune duplication de paquets à l'émission. Afin de transmettre son trafic vers les autres participants, chaque participant envoie ses données applicatives (position dans l'EV, orientation, état, etc..) dans un groupe multipoint unique. Ce groupe correspond au groupe associé à la cellule dans laquelle il se

trouve à un instant donné. En aucun cas, le groupe d'émission pour un participant ne dépend de la taille de sa zone d'intérêt, ou d'une information autre que sa position dans le monde virtuel. Par conséquent, le trafic circulant sur un groupe multipoint donné, correspond à l'ensemble des trafics générés par les participants situés dans la cellule associée à ce groupe.

Chaque participant est responsable de son abonnement aux groupes multipoints sur lesquels circule le trafic présentant un intérêt. Afin de définir la liste des groupes auxquels un participant doit s'abonner à un instant donné  $t$ , plusieurs paramètres rentrent en jeu :

- La zone d'intérêt du participant. Par définition, un participant n'est intéressé que par le trafic émis par les autres participants situés dans sa zone d'intérêt.
- La liste  $l1$  des cellules intersectant la zone d'intérêt du participant. Chaque cellule est identifiée par la Zone dans laquelle elle se trouve et par sa position dans cette Zone.
- La liste  $l2$  des cellules pouvant intersecter la zone d'intérêt du participant dans un temps égal à la latence de l'abonnement, et compte tenu de sa vitesse  $V$ .
- Les informations de mise en correspondance pour chacune des cellules appartenant aux listes  $l1$  et  $l2$  ainsi établies.
- La satisfaction  $S$  du participant à l'instant  $t - 1$ .

L'établissement des listes  $l1$  et  $l2$  est réalisée dynamiquement par chaque participant au cours de ses déplacements dans l'EV. La liste  $l2$  est particulièrement importante car chaque participant doit anticiper l'abonnement aux groupes multipoints afin de masquer la latence liée à l'abonnement multipoint (voir Section 3.4.2). Cette latence peut être définie comme une constante ou ré-estimée dynamiquement en cours de session, en donnant toujours une majoration. Dans tous les cas, l'interactivité de l'application doit être préservée tout au long de la session. En effet, lorsqu'une nouvelle entité virtuelle s'introduit dans la zone d'intérêt d'un participant, le trafic généré par celle-ci doit être instantanément "routé" vers le participant.

La satisfaction  $S$  du participant rentre également en ligne de compte dans la liste des groupes



auxquels celui-ci doit s'abonner. En effet, lorsqu'un participant est insatisfait, cela signifie qu'il ne parvient pas à recevoir ou à traiter en temps réel le débit total d'information utile. Dans cette situation, deux évolutions sont envisageables : Les agents peuvent décider de calculer une taille de cellule plus petite dans la Zone où se trouve le participant. Ainsi celui-ci sera plus en mesure de réduire le débit de trafic superflu entrant, et par conséquent, d'augmenter sa satisfaction. La deuxième évolution envisageable peut être, quant à elle, initiée du côté participant. En effet, lorsque celui-ci subit des pertes au niveau du trafic entrant, il peut décider de réduire judicieusement ce trafic de manière à diminuer les pertes sur les informations qui l'intéressent le plus. Pour effectuer cette opération, le participant peut décider de réduire volontairement sa zone d'intérêt, réduisant ainsi le nombre de cellules intersectant celle-ci, et par conséquent le nombre de groupes multipoints auxquels il est abonné. Ainsi, son trafic entrant sera diminué, réduisant ainsi les pertes, et le participant sera plus en mesure de recevoir et de traiter en temps réel le débit d'information utile (que l'on pourrait qualifier ici de débit d'information **le plus utile**).

La partie synchronisation des données est également laissée à la charge des participants. Ceux-ci peuvent donc rajouter une estampille de temps dans chaque paquet transmis, et resynchroniser les différents paquets reçus des autres participants. Des mécanismes de synchronisation distribués tout à fait adaptés à notre architecture sont décrits dans [GKD99]. Définir de nouveaux mécanismes de synchronisation dans le cadre de SCORE sort cependant du cadre de la thèse.

### 6.4.3 Communication participants-agent

Les participants sont amenés à s'abonner à deux catégories de groupes multipoints :

- Des **groupes de données** qui sont associés à chacune des cellules qui intersectent leur zone d'intérêt. Il est à noter qu'un participant ne transmet des données que dans un seul groupe multipoint, à savoir le groupe multipoint associé avec la cellule dans laquelle il se trouve. Il se comporte uniquement comme un récepteur pour tous les autres groupes de données auxquels il est abonné.

- Des **groupes de signalisation** qui sont associés aux Start-Zones et dans lesquels ils ne sont que des récepteurs. Les agents émettent les informations de mise en correspondance pour toute la Start-Zone, dans le groupe de signalisation qui lui est associé. Ces informations sont envoyées périodiquement dans chacune des Start-Zones (avec une période  $P = P_{Start-Zone}$ ), et contiennent les informations de mise en correspondance pour toutes les Zones appartenant à la Start-Zone (i.e., la taille de cellule pour chaque Zone et les adresses des groupes multipoints associées). Les agents se partagent la responsabilité d'émission dans les différentes Start-Zones.

Pour chacun de ces groupes, le participant doit anticiper l'émission de son message d'abonnement en tenant compte de sa vitesse et de la latence de l'abonnement.

Pour les groupes de signalisation, la période  $P_{Start-Zone}$  est également prise en compte afin que les participants puissent recevoir les informations de mise en correspondance avant que leur centre d'intérêt n'intersecte de nouvelles cellules appartenant à de nouvelles Start-Zones.

En plus d'être abonné à des groupes de signalisation en tant que récepteur uniquement, **chaque participant est connecté en mode point-à-point à un unique agent via UDP**. Cette connexion est établie entre chaque participant et son agent le plus proche (voir la Section 6.5). Le terme "connexion" est employé ici en faisant référence à une connexion au niveau session et non au niveau de la couche transport (où UDP est présenté comme un mode "non-connecté" à la différence de TCP). Nous avons choisi UDP plutôt que TCP pour des raisons de scalabilité au niveau de l'agent. En effet, un grand nombre de participants (jusqu'à plusieurs milliers) peut être rattaché à chaque agent. L'utilisation d'UDP permet de multiplexer les paquets sur un même port à l'arrivée au niveau de l'agent. Dans le cas de TCP, un nombre de *descripteurs de fichier* équivalent au nombre de participants connectés, est nécessaire au niveau du code de l'agent. Or, sous la plupart des systèmes d'exploitation, dont Solaris et Linux, le nombre de descripteurs de fichier par processus est limité. La mise en œuvre d'agents multi-processus entraînerait une perte en terme de scalabilité: création dynamique de nouveaux processus, mémoire partagée,

et coûts supplémentaires de traitement de l'information. De plus, les informations transmises des participants vers les agents n'ont pas besoin d'un service de transport fiable. En revanche, des mécanismes de contrôle de flux doivent être mise en œuvre afin d'éviter aux agents d'être submergé de messages en provenance des participants auxquels ils sont rattachés. Mais avant de rentrer plus en détail dans ce mécanisme, il convient tout d'abord de présenter les messages de contrôle émis par les participants vers leur agents.

Afin de permettre aux agents de calculer les densités de participant dans chaque Zone, de définir de nouvelles Zones en rassemblant ou désassemblant plusieurs Zone-Units contigus, et d'évaluer périodiquement la satisfaction de chaque participant, plusieurs informations doivent être contenues dans ces messages :

- L'identité unique du participant codée sur 4 octets. Cette information est obligatoire car elle permet aux agents de détecter de nouvelles arrivées et de nouveaux départs au sein de l'EV.
- La Zone-Unit courante où se trouve le participant. Dans la Section 6.3.2.1, on désigne une Zone-Unit en précisant le numéro de sa Start-Zone ainsi que sa position dans celle-ci. Nous avons codé cette information sur 8 octets.
- La taille courante de sa zone d'intérêt codée sur 2 octets.
- Sa capacité  $C$  (voir Section 6.2) à recevoir et à traiter le trafic entrant en temps réel, codée sur 2 octets.

Chaque participant transmet ce message (16 octets de charge utile en point-à-point vers son agent), à chaque fois qu'il franchit une nouvelle Zone-Unit de l'EV. Ainsi chaque agent connaît à chaque instant l'emplacement de ses participants connectés, c'est-à-dire leur répartition au sein des différentes Zone-Units de l'EV.

En ce qui concerne la capacité  $C$  transmise par les participants, celle-ci peut être définie de deux différentes manières. Une première possibilité consiste à définir cette capacité de façon sta-

tique au niveau du participant, en lui laissant définir son *profil utilisateur*: modem, LAN, T1, etc.. pour la partie connectivité réseau, et différents types de processeurs et de cartes graphiques pour la partie puissance de calcul temps-réel. L'avantage évident dans ce cas est bien entendu la simplicité de calcul de  $C$ . En revanche, calculer  $C$  de cette manière ne permet pas de tenir compte de l'état courant du réseau en amont du participant, en cas de congestion par exemple. Une deuxième possibilité consisterait donc à ré-estimer dynamiquement la capacité  $C$  en cours de session, en fonction du taux de perte perçu au niveau de chaque participant. Toutefois, la détermination de ce taux de perte dans le cas d'abonnements/désabonnement dynamiques à plusieurs groupes multipoints n'est pas triviale. Nous n'aborderons pas cette problématique dans cette thèse, qui constitue en elle-seule un sujet de recherche à part entière.

Nous exposons désormais notre mécanisme de contrôle de flux utilisé entre les participants et leur agents respectifs. Les participants sont tenus d'envoyer à un débit minimum les messages de contrôle vers leur agent, afin que ceux-ci puissent détecter rapidement une déconnexion, et mettre à jour le nombre de participants dans le monde virtuel et ses différentes Zones. En fonction du nombre de participants connectés, les agents informent les participants de la période minimale  $P_{min}$  de séparation entre deux messages de contrôle. Cette période est transmise avec les informations de mise en correspondance envoyées par les agents dans les groupes de signalisation associés aux Start-Zones. A chaque émission de message de signalisation, chaque participant ré-initialise une horloge (ou timer). Si le participant n'a pas franchi de nouvelle Zone-Unit pendant un temps égale à la période fixée par les agents, il envoie à son agent un paquet de **maintien de connexion** (ou Keep-Alive) et réinitialise à nouveau son horloge. Ce paquet très court contient uniquement le numéro d'identifiant du participant ainsi que sa Zone-Unit courante. En fonction du débit de trafic de contrôle entrant et de leur capacité à supporter la charge de travail qu'il leur est impartie, les agents peuvent modifier dynamiquement la valeur de  $P_{min}$ , et contrôler ainsi le débit entrant.

De leur côté, les agents maintiennent la liste des participants connectés ainsi que la dernière date de réception de paquet pour chacun d'entre eux. Si, pour un participant  $P_i$ , aucun paquet n'a été reçu pendant un temps égale à  $n$  fois la période minimale supposée séparer deux messages de contrôle, alors le participant  $P_i$  est supprimé de la liste, et le nombre de participants dans l'EV et la Zone-Unit est réduit de 1. Notons qu'avec un  $n$  petit, la non-réception de paquets n'est pas nécessairement dû à une déconnexion, et peut s'expliquer par un taux de perte trop important dans le réseau. C'est pourquoi, lorsque l'agent considère un participant déconnecté, il ne le supprime pas totalement de sa mémoire mais le place dans une liste de participants déconnectés. Ainsi, si un message de contrôle en provenance du participant supposé déconnecté est finalement reçu, celui est automatiquement réintroduit dans la liste initiale sans avoir à repasser par une éventuelle procédure de connexion (voir la Section 6.5 sur ce sujet). De même, les compteurs de participants pour la Zone-Unit et pour le monde virtuel sont de nouveau incrémentés. Nous verrons plus loin dans ce chapitre, l'importance de ces compteurs au niveau des agents, afin que ceux-ci est la vision la plus exacte possible de la répartition de participants dans l'EV, et ainsi mettre à jour dynamiquement les tailles de cellule dans les différentes Zones.

Nous avons donc proposé ici un mécanisme de contrôle de flux, permettant également une certaine fiabilité concernant le nombre de participants connectés à un instant  $t$ . En revanche, aucune fiabilité n'est garantie ici en ce qui concerne les paquets de changement de Zone-Unit transmis par les participants. Notre architecture de communication n'est cependant pas affectée outre mesure par ce type de pertes. En effet, nous verrons dans la Section 6.7 que le nombre de participants par Zone-Unit n'est pas important en tant que tel. En revanche, ce sont plutôt les variations de participants entre les différentes Zone-Units qui rentrent en ligne de compte dans le traitement des informations de mise en correspondance. Or on peut supposer ici que les pertes des paquets de contrôle émis par les participants vont affecter "équitablement" les compteurs de participants des Zone-Units côté agent, d'autant plus que le goulot d'étranglement responsable de ces pertes se trouve proche de l'agent.

#### 6.4.4 Communication agent-agent

Au cours du chapitre 5, nous avons discuté de l'impact de la densité de participants et du ratio entre la taille de cellule et la taille de la zone d'intérêt, sur la quantité de trafic entrant au niveau des participants. Suite aux résultats obtenus, nous avons proposé de mettre en œuvre SCORE, une nouvelle architecture de communication prenant en compte à la fois la répartition non-uniforme des participants dans l'EV et l'hétérogénéité entre ces participants. De plus, SCORE doit répondre également aux contraintes de scalabilité qui apparaissent lors du déploiement à grande-échelle des EVs. Pour cette raison, nous avons conçu SCORE d'une manière doublement distribuée. D'une part, le calcul d'état et la synchronisation sont effectués de façon distribuée entre les participants. D'autre part, le trafic de contrôle émis par les participants ne converge pas vers un agent unique, mais est réparti entre plusieurs agents. Dans cette section, nous ne présentons que la partie communication entre les agents. La Section 6.7 explique de manière plus détaillée l'utilisation de ces informations par les agents.

Cependant, et afin de permettre aux agents de calculer les diverses densités de participants dans les différentes Zone-Units du monde virtuel, les agents doivent à leur tour s'échanger un certain nombre d'informations. En effet, chaque agent ne connaît que le nombre de participants qui lui sont connectés ainsi que leur position (la Zone-Unit) dans laquelle ils se trouvent. Afin de s'échanger ces informations, les agents communiquent en multipoint sur un groupe particulier qui leur est réservé : le groupe *GMA* (voir Figure 6.7). Quatre informations sont utilisées par les agents afin de réaliser le découpage de l'EV en Zones et en cellules :

- Le nombre total de participants connectés à l'EV. Cette information permet à chaque agent de calculer le nombre moyen de participants par groupe multipoint.
- Le nombre de participants dans chaque Zone définie à un instant  $t$  par les agents dans l'EV. Cette information permet d'évaluer la densité de participants dans chacune des Zones.
- Le nombre de participants dans chaque Zones-Unit. Cette information permet aux agents de reformer de nouvelles Zones en tentant toujours de s'approcher d'une distribution uniforme

de participants dans chaque Zone.

- La liste des Zone-Units dans lesquelles se trouvent les participants ayant la satisfaction  $S$  la plus faible. Ainsi les agents ont la possibilité de définir de nouvelles Zones et peuvent calculer une taille de cellule plus petite dans les Zones où se trouvent les participants insatisfaits.

La première information nécessaire aux agents est le nombre total de participants connectés. Nous avons vu dans la Section 3.4.3, que l'estimation de la taille de session peut s'avérer particulièrement problématique. Toutefois, notre architecture de communication nous permet ici d'obtenir ce paramètre très facilement. En effet, chaque agent va diffuser sur le groupe *GMA*, le nombre exact de participants (qui lui sont connectés) par Zone-Unit. Un unique paquet sera émis par agent, agrégeant ainsi l'ensemble des paquets de contrôle émis par les participants vers leur agents respectifs. A la réception des paquets en provenance de tous les autres agents, chaque agent est ainsi capable de calculer les trois premières informations de notre liste précédemment définie. En ce qui concerne la liste des Zone-Units dans lesquelles se trouvent les participants ayant la satisfaction  $S$  la plus faible, les agents vont procéder en deux étapes. Tout d'abord, chaque agent évalue la satisfaction de chacun de ses participants connectés, puis calcule la satisfaction minimale par Zone-Unit. Enfin, dans un deuxième temps, chaque agent va diffuser sur le groupe *GMA* la satisfaction minimale de chaque Zone-Unit. Ainsi, après la réception de tous les paquets émis par les autres agents, chaque agent pourra établir la liste désirée.

## 6.5 Connexion au monde virtuel

Avant de débiter une session, nous supposons ici que chaque participant a déjà téléchargé la description du monde virtuel en local sur sa machine. Cette description comprend l'emplacement des Start-Zones et des Zone-Units qui les composent. En revanche, la seule adresse multipoint connue est l'adresse du groupe multipoint des agents (le groupe *GMA*). Aucune information de mise en correspondance n'est communiquée avant le début de la connexion.

Afin d'entamer une procédure de connexion dans l'EV, un nouveau participant doit en premier lieu se rattacher à un agent. Dans notre architecture les participants découvrent leur agent en émettant des paquets "Hello" sur le groupe *GMA*. Cette découverte peut être effectuée de deux manières :

- Le participant émet un paquet Hello sur le groupe *GMA* avec un TTL fixé à 1. S'il reste sans réponse après un temps *Hello – Wait*, il renouvelle cette opération en augmentant le TTL (en suivant les différentes valeurs discrètes de TTL fixées pour le multipoint). Dès qu'un agent détecte un paquet Hello, il récupère l'adresse source de l'émetteur et lui transmet en point-à-point son adresse IP afin qu'une connexion puisse s'établir.
- Le participant émet un paquet Hello sur le groupe *GMA* avec un TTL fixé à 255. Tous les agents répondent en transmettant leur adresse IP à l'émetteur. Le premier paquet reçu par le participant contient l'adresse de l'agent avec lequel le RTT est le plus faible.

La première méthode a l'avantage d'éviter que chaque participant désirant se connecter, "inonde" tous les agents avec un paquet "Hello" et reçoive autant de réponses que d'agents. En revanche, l'agent avec lequel la connexion se fait n'est pas forcément l'agent le plus proche en terme de délai (RTT). La deuxième méthode de découverte permet en revanche la découverte de l'agent le plus proche (toujours en terme de délai). Une solution hybride entre les deux est également envisageable.

Dès que le participant a récupéré l'adresse IP d'un agent, il peut initier une procédure de connexion dans le monde virtuel en passant éventuellement par une procédure d'authentification. A la suite de cette procédure, le participant reçoit de son agent la liste des adresses multipoints associées aux Start-Zones et sa position dans l'EV. Le participant peut alors écouter sur le groupe multipoint de signalisation associé à sa Start-Zone, afin de récupérer les informations de mise en correspondance qui l'intéressent, et commencer ainsi à se déplacer dans l'EV.



## 6.6 Mise à jour des abonnements

### 6.6.1 Définition

Afin d'informer les participants des groupes multipoints auxquels ils doivent s'abonner pour pouvoir interagir avec les autres participants situés autour d'eux, les agents doivent faire face à deux situations. La première situation arrive lorsqu'un participant se déplace dans l'EV et qu'il est sur le point d'entrer dans une nouvelle Start-Zone. Dans ce cas, le participant doit anticiper son abonnement au groupe multipoint de signalisation associé à cette Start-Zone, pour lui permettre de recevoir à temps les informations nécessaires. La seconde situation se produit lorsque les agents décident de changer les informations de mise en correspondance dans une Start-Zone. Il peut s'agir par exemple du changement de la taille de cellule dans une Zone, de la création de nouvelles Zones, ou encore de l'agrégation de deux Zones contiguës. Dans ce cas, les participants situés dans la région virtuelle concernée, doivent mettre également à jour leur abonnement aux groupes multipoints de données, tout en continuant d'interagir les uns avec les autres. Cette opération particulièrement critique doit par conséquent être effectuée de façon complètement transparente pour l'utilisateur final.

### 6.6.2 Mécanisme de mise à jour des abonnements

Ce mécanisme doit permettre aux participants de mettre à jour leurs abonnements aux groupes multipoints en tenant compte des nouvelles informations de mise en correspondance. Ce mécanisme doit remplir les conditions suivantes :

- Préserver l'interactivité entre les participants pendant la mise à jour.
- Eviter toute augmentation soudaine de trafic de données entrant au niveau des participants, qui serait due à une duplication éventuelle de paquets dans les anciens et les nouveaux groupes multipoint.
- Limiter le surcoût en terme de trafic de signalisation lié à ce mécanisme.

Voici la liste des opérations successives qu'il est nécessaire de réaliser pour effectuer une mise à jour des abonnements :

- Dès qu'un participant reçoit une nouvelle information de mise en correspondance, il doit s'abonner aux nouveaux groupes multipoints associés aux nouvelles cellules qui intersectent sa zone d'intérêt.
- Dans la Section 6.4, nous avons expliqué que les agents diffusent périodiquement dans les groupes multipoints de signalisation, les informations de mise en correspondance concernant les Start-Zones associées. Cependant, lorsque les agents décident de changer les informations de mise en correspondance dans une Start-Zone donnée, ils doivent augmenter temporairement leur débit d'émission dans le groupe de signalisation associé.
- Le participant attend d'avoir reçu  $n$  nouvelles informations de mise en correspondance avant d'émettre dans les nouveaux groupes multipoints. Cependant, si le participant commence à recevoir des données dans un des nouveaux groupes multipoints, il émet à son tour dans les nouveaux groupes et arrête d'émettre dans les anciens.
- Lorsqu'un participant n'a pas reçu de paquets sur aucun des anciens groupes multipoints pendant un temps  $T_{no-receive}$ , il se désabonne à ces groupes.

Il est important de noter ici que les participants s'abonnent aux nouveaux groupes multipoints avant de procéder au changement effectif de groupes d'émission. Deux raisons sont à l'origine de cette opération : Premièrement, attendre que tous les participants concernés aient bien reçu la nouvelle information de mise en correspondance avant de procéder aux changements. Deuxièmement, attendre que les nouveaux arbres de transmission multipoints ait le temps de s'établir dans le réseau, afin que les premiers paquets émis sur les groupes associés soient bien diffusés à tous les membres. Ce mécanisme permet également de synchroniser entre les participants le changement effectif de groupes d'émission, même si des paquets contenant les informations de mise en correspondance ont été perdus avant d'atteindre certains participants.

## 6.7 Algorithme de découpage

Le même algorithme de découpage est utilisé par tous les agents. Il consiste à définir dynamiquement des Zones et à calculer dynamiquement une taille de cellule appropriée dans chaque Zone en fonction de la répartition des participants et de leur satisfaction. Bien entendu le nombre de cellules dans une Zone est inversement proportionnel à leur taille. Par conséquent, un nombre limité de groupes multipoints entraîne une certaine limitation dans la taille minimum des cellules (on rappelle qu'à chaque cellule est associé un unique groupe multipoint). La taille minimum d'une cellule est donc fonction de plusieurs paramètres comme le nombre de groupes multipoints disponibles, la vitesse maximale des participants, la taille moyenne de la zone d'intérêt des participants, et la latence maximale de l'abonnement multipoint. On fait l'hypothèse que les agents ont déjà connaissance de ces paramètres en début de session. Cette hypothèse est par ailleurs réaliste puisque la vitesse maximale des participants et la taille moyenne de leur zone d'intérêt sont des paramètres dépendant de l'application, et par conséquent connus a priori par les administrateurs d'EVEs. Si on se place au niveau global en considérant l'EV dans son intégralité, le nombre de cellules reste constant (à 1 ou 2 près). En revanche, l'affectation des groupes dans chacune des Zones change dynamiquement, prenant ainsi en compte la dynamique intrinsèque à ce type d'application.

Afin de permettre aux agents d'effectuer de façon simple le découpage de l'EV, nous ne considérons que des cellules rectangulaires, avec un nombre entier de cellules par Zone. Tout au long de la session, les agents calculent périodiquement la densité moyenne de participants par groupe multipoint, en divisant le nombre de participants connectés avec le nombre de groupes multipoints disponibles pour l'application. Nous ferons désormais référence à cette densité par l'expression **seuil de redécoupage**. Au fur et à mesure des arrivées, départs, et déplacements des participants dans l'EV, les agents calculent également les densités de participants par cellule dans chacune des Zones. Par exemple, si la Zone  $Z_i$  contient  $p$  participants et que celle-ci est constituée de  $c$  cellules, la densité de participants dans  $Z_i$  est égale à  $\frac{p}{c}$ .

L'algorithme de découpage consiste en la réalisation de trois étapes successives :

- Un premier calcul permet de définir une taille de cellule dans chaque Zone, en ne tenant compte que de la répartition des participants dans l'EV (et non de leur satisfaction). Ce calcul est réalisé en comparant la densité de participants par cellule dans chaque Zone avec le seuil de redécoupage.
- Dans une seconde étape, les participants les moins satisfaits sont identifiés, ainsi que leur répartition dans l'EV. Si une concentration de participants insatisfaits est détectée dans une partie d'une Zone seulement, cette Zone est découpée en deux nouvelles Zones afin d'isoler ces participants; sinon la Zone reste inchangée. Une taille de cellule plus petite est ensuite calculée dans les Zones contenant les participants les moins satisfaits. Ces participants seront alors en mesure d'effectuer une meilleure approximation de leur zone d'intérêt. Ils pourront ainsi réduire leur trafic superflu ce qui aura pour effet d'augmenter leur satisfaction.
- Enfin une troisième opération est exécutée périodiquement mais moins fréquemment que les deux précédentes. Au cours de cette opération, les agents peuvent décider d'agréger deux Zones contiguës, si la taille de cellule est identique dans ces deux Zones, et si celles-ci appartiennent à la même Start-Zone. En revanche, deux Start-Zones ne peuvent pas être agrégées.

Deux situations différentes peuvent conduire au redécoupage d'une Zone en cellules de tailles plus petites :

- Il est possible de trouver une taille de cellule plus petite, telle que la densité moyenne de participant par cellule reste encore supérieure au seuil de redécoupage.
- Les participants ayant les valeurs de satisfaction les plus faibles se trouvent dans cette Zone.

Dans le premier cas, les agents se basent sur la densité de participants dans la Zone pour calculer une taille de cellule plus appropriée, tout en prenant en compte le nombre de groupes multipoints

disponibles. Dans le second cas, les agents observent tout d'abord la répartition des participants insatisfaits dans la Zone. Afin de détecter une concentration de participants insatisfaits dans une partie de cette Zone, les agents commencent par calculer la satisfaction minimale des participants dans chaque Zone-Unit constituant la Zone. La satisfaction minimale de chaque Zone-Unit est ensuite comparée avec la valeur moyenne des satisfactions de toutes les Zone-Units de la Zone. En fonction du résultat de ces comparaisons, les agents décident de conserver la Zone telle quelle ou de la découper en deux nouvelles Zones (chacune des nouvelles Zones étant encore constituée de une ou plusieurs Zone-Units contiguës appartenant à la Zone découpée).

De la même manière, les agents peuvent décider de redécouper une Zone en cellules de plus grande taille. Cette situation ne peut se produire que lorsque la densité de participants par cellule est inférieure au seuil de redécoupage.

## 6.8 Conclusion

Bien que le modèle multipoint courant souffre d'un grand nombre d'imperfections pour la gestion d'applications de type EVGE, nous avons présenté SCORE, une architecture de communication permettant à de telles applications d'être déployées sur l'Internet aujourd'hui. De plus, notre architecture est suffisamment flexible pour être facilement adaptable aux nouvelles approches du multipoints telles que *EXPRESS* [HC99] ou *Simple Multicast* [PLB<sup>+</sup>98], et également bénéficier du filtrage de sources proposé dans IGMPv3 [CDT00]. Toutefois, nous ne prétendons pas que SCORE soit capable de répondre à toutes les problématiques soulevées dans les EVGEs et que n'importe quel EVGE aurait intérêt de s'appuyer sur notre architecture. En effet, notre architecture de communication introduit un coût supplémentaire par la présence des agents et des messages de contrôle et de signalisation échangés. Afin d'évaluer dans quel cas notre architecture peut s'avérer intéressante, il convient de comparer ce coût avec l'amélioration des performances liée à SCORE, pour différents niveaux de dynamicité et d'hétérogénéité dans l'application. Le Chapitre 7 présente une série d'expérimentations et tente de répondre à cette question.

Enfin, une dernière remarque concerne la limitation des groupes multipoints disponibles dans notre architecture. Cette limitation est bien entendu indépendante de SCORE. Cependant, elle peut avoir des conséquences néfastes sur l'application. En effet, cette limitation du nombre de groupes multipoints disponibles entraîne implicitement une limitation du nombre de cellules dans l'EV. Mais lorsque la densité de participants par cellule excède un certain seuil dans l'EV, les agents ne sont plus capables d'améliorer la satisfaction de leurs participants (aucune autre architecture de communication ne pourrait d'ailleurs y prétendre). Toutefois, diverses solutions sont envisageables afin de préserver la qualité perçue par les utilisateurs dans un EVGE. Toutes ces solutions proposent une manière de limiter la densité de participants dans l'EV :

- Avoir un EV "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'EV reste toujours inférieure à un certain seuil.
- Limiter la nombre maximum d'utilisateurs connectés à l'EV, et construire un EV suffisamment grand pour que cette densité ne dépasse pas un seuil maximal. Cette solution permettrait effectivement d'avoir un grand nombre d'utilisateurs connectés, mais on ne peut plus vraiment parler ici de scalabilité.
- Utiliser des protocoles tels que ceux définis dans *MAAA* [KRT<sup>+</sup>98], de manière à permettre une allocation dynamique de groupes multipoints dans SCORE, lorsque la densité de participants excède un seuil maximal. Cependant, cette solution ne résout qu'une partie du problème en permettant aux participants de réduire le débit moyen de trafic superflu entrant. En effet, au delà d'un certain seuil, la densité de participants serait telle, que même en approximant parfaitement sa zone d'intérêt grâce à des très petites cellules, le débit entrant d'un participant dépasserait sa capacité.



## Chapitre 7

# Expérimentations de SCORE

### 7.1 Développement

#### 7.1.1 Introduction et paramètres de lancement

Afin de démontrer à la fois la faisabilité et l'efficacité de notre architecture de communication SCORE, nous avons réalisé une implémentation de SCORE. Cette implémentation est constituée de deux programmes indépendants, respectivement le participant et l'agent. La programmation s'est faite en C sous Linux et Solaris. Le développement de SCORE a été pensé dans l'optique de réaliser des expérimentations intensives avec plusieurs milliers de participants. Cette implémentation représente une véritable plate-forme expérimentale puisque presque tous les mécanismes de SCORE sont directement paramétrables par la ligne de commande au moment du lancement de chacun des exécutable.

Pour le participant, les paramètres sont les suivants :

- la taille de l'environnement virtuel,
- le nombre de groupes multipoints disponibles (i.e. le nombre maximum de cellules),
- le nombre de Start-Zones,
- le nombre de Zone-Units par Start-Zone,
- la valeur maximale de la latence de l'abonnement,



- la vitesse de déplacement,
- la taille de la zone d'intérêt,
- le débit d'émission,
- le mode sans réception ou avec réception des paquets de données.

Ce dernier paramètre permet de lancer des participants en mode sans réception. Dans ce cas, le participant émet normalement son trafic de données sur le groupe multipoint associé à la cellule dans laquelle il se trouve mais ne se place en réception que dans les groupes de signalisation où les informations de mise en correspondance sont émises. Placé dans ce mode, le participant n'a pas à effectuer de traitement particulier sur les paquets de données émis par les autres participants. La charge CPU d'un tel participant en est donc considérablement réduite, ce qui permet de lancer un grand nombre de participants sur une même machine. Nous verrons dans la Section 7.2, les raisons qui nous ont conduit à définir une telle option.

En ce qui concerne l'agent, les paramètres en entrée sont :

- la taille de l'environnement virtuel,
- le nombre de groupes multipoints disponibles (i.e. le nombre maximum de cellules),
- le nombre de Start-Zones,
- le nombre de Zone-Units par Start-Zone,
- la valeur maximale de la latence de l'abonnement,
- la période de calcul et d'émission des informations de mise en correspondance dans les groupes multipoints de signalisation,
- la période de calcul des satisfactions minimales dans chaque Zone-Unit,
- le mode statique ou dynamique.

En mode dynamique, l'agent va se comporter conformément à la description qui en a été faite dans le Chapitre 6. En revanche, en mode statique, les participants vont continuer d'émettre leur trafic de contrôle vers l'agent mais aucune information de mise en correspondance ne sera transmise en retour. Dans ce cas, le découpage de l'EV en cellules est effectué au lancement

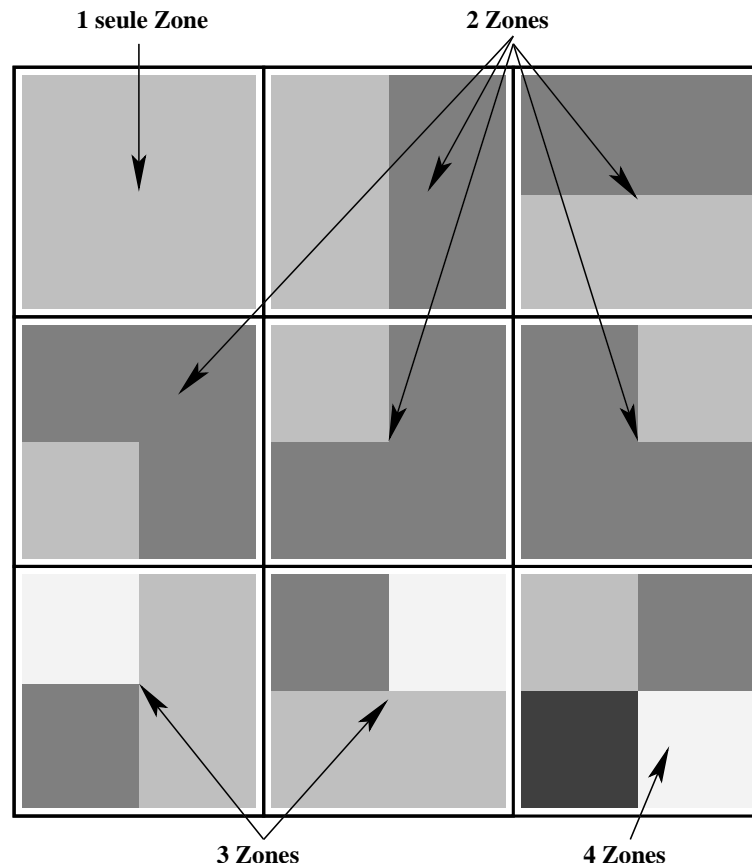


FIG. 7.1 – Exemples de découpage en Zones avec 9 Start-Zones et 4 Zone-Units/Start-Zone

des participants, ainsi que l'affectation des adresses multipoints qui demeure inchangée tout au long de la session. Grâce à cette option, les agents vont pouvoir calculer les satisfactions des participants au cours de la session. Ces résultats seront sauvegardés sous forme de fichiers de traces. Ils nous permettront par la suite d'effectuer une véritable comparaison (au niveau des satisfactions) entre un découpage statique et le découpage dynamique proposé dans SCORE.

### 7.1.2 Présentation du cas étudié

Au cours de nos expérimentations, nous avons préféré fixer certains des paramètres afin d'éviter une trop grande confusion dans l'interprétation des résultats. Etant donné que l'objectif principal de nos expérimentations est d'évaluer SCORE en tant qu'architecture de communica-

tion pour EVGEs, nous avons défini un pseudo monde virtuel, sans pour autant faire intervenir un quelconque moteur de rendu. Afin de faciliter le développement des algorithmes de localisation des participants, nous avons défini un monde virtuel carré de 1200x1200 découpé en 3x3 Start-Zones carrées de taille identique. Nous fixons à 144 le nombre de groupes multipoints disponibles pour le trafic de données échangées entre les participants. Chaque Start-Zone est composée de 2x2 Zone-Units carrées. La taille d'une Zone est donc égale à 1,2,3 ou 4 fois la taille d'une Zone-Unit. Dans le cas où on a une Zone recouvrant 4 Zone-Units, la Zone occupe toute la Start-Zone. Sur le schéma 7.1, nous pouvons observer un découpage du monde virtuel en Zones de différentes tailles possibles. De plus, les cellules sont de forme carrée également, et il ne peut y avoir qu'un nombre entier de cellules par Zone. Si on se place au niveau d'une Start-Zone, on peut distinguer deux types de Zones : les Zones composées de 1 ou 4 Zone-Units, et les Zones composées de 2 ou 3 Zone-Units. Dans le premier cas, les Zones sont carrées et le nombre de cellules dans ces Zones prend sa valeur dans  $\{1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121\}$ . Dans le deuxième cas, le nombre de cellules prend sa valeur dans  $2 * \{1, 4, 9, 16, 25, 36, 49, 64\}$  si la Zone est constituée de 2 Zone-Units et dans  $3 * \{1, 4, 9, 16, 25, 36, 49\}$  si la Zone est constituée de 3 Zone-Units. Dans tous les cas, le nombre total de cellules dans notre monde virtuel n'excède jamais 144.

Afin d'évaluer SCORE dans le cadre d'un EV à grande-échelle, nous avons décidé d'impliquer jusqu'à 1000 participants dans nos expérimentations. En ne lançant qu'un unique agent sur un Pentium Pro 200Mhz, les expérimentations nous ont montré que moins de 10% du CPU était utilisé. Par conséquent, l'ensemble de nos expérimentations a été réalisé avec la présence d'un unique agent. Néanmoins, le fait de n'utiliser qu'un seul agent dans SCORE ne modifie en rien les mécanismes que nous cherchons à évaluer dans ce chapitre.

### 7.1.3 Description du participant

Sans rentrer dans les détails de l'implémentation du participant, nous dressons maintenant l'enchaînement des étapes réalisées successivement par chaque participant. Nous avons développé

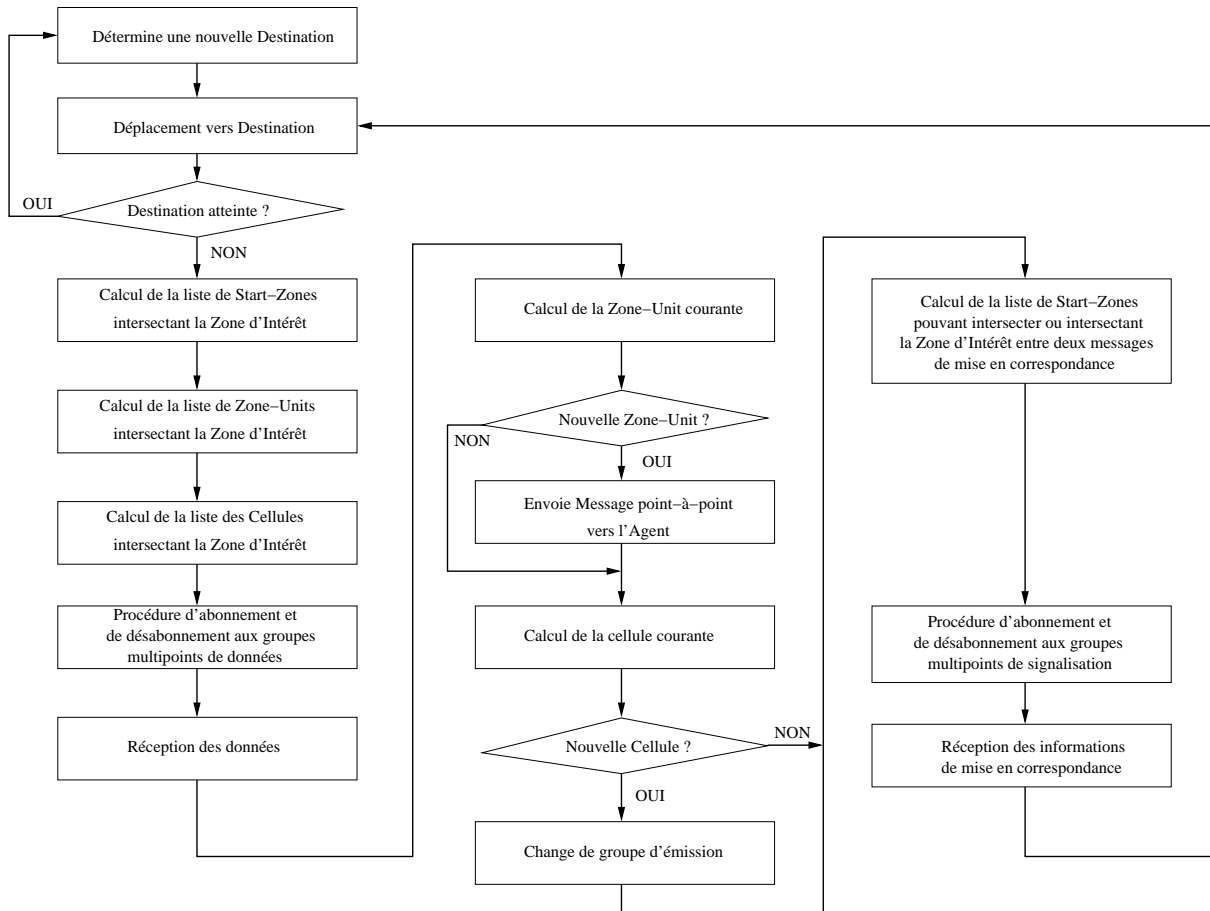


FIG. 7.2 – *Enchaînement des étapes au niveau du participant*

un ordonnanceur spécifique à l'application, afin de régler le débit d'émission de chaque participant sur le paramètre passé en ligne de commande. Entre chacune des étapes de la Figure 7.2, notre ordonnanceur reprend la main et décide de lancer ou non la procédure d'envoi des paquets dans les groupes multipoints de données, en fonction du temps qui s'est écoulé depuis la précédente émission. Pour cette raison, la procédure d'émission n'apparaît pas sur notre Figure.

#### 7.1.4 Hétérogénéité et Dynamicité

Afin d'introduire suffisamment d'hétérogénéité entre les participants au cours de nos expérimentations, la capacité  $C$  de chaque participant est définie aléatoirement au démarrage. Les

participants ayant un débit d'émission identique, nous exprimons cette capacité en nombre de sources par seconde. Ces capacités sont uniformément distribuées entre 20 et 40 sources par seconde dans la plupart de nos expérimentations. Par exemple, nous considérons qu'un participant ayant une capacité  $C = 30$  sources/s est capable de recevoir et de traiter en temps-réel le trafic généré par 30 participants. Si celui-ci reçoit un trafic en provenance d'un nombre de participants supérieur à 30, alors nous considérons que les pertes subies sont équitablement distribuées entre le trafic utile et le trafic superflu entrant. Dans notre précédent exemple, si le trafic émis à destination du participant est égale à 60 sources/s et se composent de 10 sources utiles par seconde (i.e. présentes dans la zone d'intérêt du participant) et de 50 sources inutiles par seconde, alors le trafic utile reçu et traité sera égal à 5 sources/s. En effet, ici le taux de perte est bien égal à 50% mais il ne modifie en rien la proportion de trafic utile. Une autre façon de définir le trafic à destination d'un participant revient à calculer la somme des abonnés aux groupes multipoints auxquels le participant a souscrit.

La dynamicité dans notre environnement virtuel est introduite non seulement par la vitesse des participants mais également par la notion de "Start-Zones chaudes" et de "Start-Zones froides" que nous allons définir un peu plus loin. A son arrivée dans le monde virtuel, le participant est tout d'abord placé aléatoirement selon une loi de distribution uniforme sur l'axe des  $x$  et sur l'axe des  $y$ . Puis, sa destination est déterminée en deux étapes. La Start-Zone de destination est choisie aléatoirement en fonction d'une matrice donnant la probabilité  $P(i, j)$  pour que la Start-Zone de destination ait les coordonnées  $(i, j)$ . Etant donné que notre environnement virtuel est découpé en 3x3 Start-Zone, la matrice est également de 3x3 et la somme de ses éléments est égale à 1, soit  $\sum_{i,j} P(i, j) = 1$ . L'ensemble des éléments de cette matrice est une combinaison des valeurs suivantes :  $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}, \frac{1}{256}\}$ . L'affectation de ces 9 probabilités aux 9 Start-Zones est effectuée périodiquement par l'agent (nous fixons ici cette période à 360s) et de façon aléatoire; les combinaisons étant équiprobables. La nouvelle matrice est annoncée par l'agent à tous les participants simultanément via les groupes multipoints de signalisation.

## 7.2 Environnement expérimental

### 7.2.1 Plate-forme d'expérimentation

Nous avons réalisé l'ensemble de nos expérimentations sur le réseau de test du projet PLANETE/RODEO de l'INRIA Sophia-Antipolis. Notre environnement expérimental se compose de 7 PCs sous linux 2.2, connectés à un réseau ethernet 10Mbits/s. Les 1000 participants sont répartis sur ces 7 PCs de la façon suivante: 996 participants sont lancés en mode sans réception sur 3 PCs (2 Pentium III 500Mhz et 1 Pentium III 600Mhz), soit 332 participants sur chacun des PCs. 4 participants sont lancés en mode réception sur 4 PCs différents (1 Pentium III 500Mhz, 2 Pentium II 450Mhz et 1 Pentium Pro 200Mhz). L'agent est lancé sur le même PC Pentium Pro que l'un des 4 participants en mode réception.

Le noyau linux 2.2 des 3 PCs, sur chacun desquels se trouvent 332 participants en mode sans réception, a dû être légèrement modifié afin de permettre l'ouverture simultanée d'un grand nombre de descripteurs de fichiers. Toutefois, grâce à l'activation du mode sans réception pour chacun des participants, la charge CPU sur ces 3 PCs n'a jamais excédé les 15% . Le rôle de ces 996 participants consiste donc exclusivement en la génération de trafic dans les groupes multi-points de données ainsi que dans l'envoi et la réception de messages de contrôle et de signalisation.

Les paramètres de lancement pour les 1000 participants sont strictement identiques, mise à part bien sûr le mode sans ou avec réception. Toutefois, et afin de faciliter le lancement et la suppression des 996 participants en mode sans réception, une interface graphique (voir Figure 7.3) a été réalisée en Tcl/Tk.

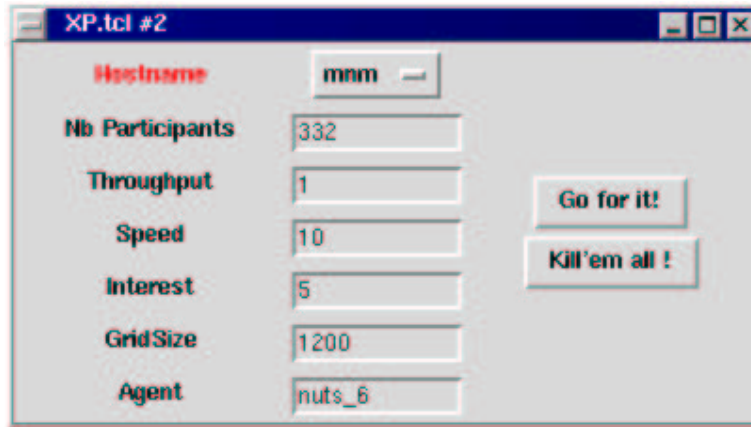


FIG. 7.3 – Interface graphique de lancement des participants en mode sans réception

## 7.2.2 Génération et utilisation des traces

Etant donné le nombre important de paramètres à faire varier et la quantité considérable d'informations pertinentes à recueillir lors d'une session d'expérimentation (lancée avec un ensemble de paramètres donnés), cette étape s'est avérée particulièrement complexe. De plus, la réalisation d'une session d'expérimentations peut demander jusqu'à plusieurs heures avant que l'analyse des traces puissent débuter. Enfin, bien que nous ayons défini par avance les différents scénari que nous désirions étudier, les résultats obtenus nous ont amené à reconsidérer à maintes reprises notre politique d'expérimentation.

Les informations pertinentes ont ainsi pu être recueillies aussi bien au niveau des participants qu'au niveau de l'agent. Les participants lancés en mode réception ainsi que l'agent mémorisent une trace de leur activité durant toute la session. Ces traces sont analysées une fois la session terminée. Voici la liste des informations figurant dans ces traces :

- le débit moyen de trafic de donnée entrant au niveau du participant (en provenance d'autres participants),
- le débit moyen de trafic de signalisation entrant au niveau du participant (information de mise en correspondance en provenance de l'agent),

- la fréquence d’abonnements et de désabonnements aux groupes multipoints de données,
- le débit moyen de trafic de contrôle (changement de Zone-Unit et maintien de connexion) émis vers l’agent,
- l’évolution de la satisfaction des participants en cours de session,
- l’évolution du nombre de groupes multipoints souscrits par participant
- l’évolution de la distribution des participants dans les différents groupes multipoints,
- l’évolution du nombre de groupes multipoints de données (ou de cellules) assignés par l’agent.

L’utilisation de certains types de traces a également nécessité une méthodologie appropriée. Ainsi les quatre derniers éléments de notre liste d’informations recueillies représentent une collection de plusieurs milliers voire centaines de milliers de données générées tout au long d’une session d’expérimentation. Dans le cas de l’évolution de la satisfaction, des fichiers de plusieurs centaines de Mo ont été générés. Afin de réaliser un traitement efficace de ces très gros fichiers de trace, nous nous sommes appuyés sur le gestionnaire de base de données relationnelles **mySQL9.35**. Ce traitement a pu s’effectuer au travers des étapes suivantes :

- Création d’une base de données et de diverses tables dans mySQL,
- Chargement des données à partir des fichiers vers une "table d’entrée" de la base,
- Opérations de traitement sur la table d’entrées ainsi chargée et génération des résultats obtenus vers les autres "tables de résultats",
- Exportation des tables de résultats vers des fichiers textes tabulés.

### 7.3 Les différentes situations étudiées

Afin d’étudier et de comparer les différentes informations pertinentes recueillies au cours de nos expérimentations, nous avons défini plusieurs situations en faisant varier certains paramètres :

- Le mode de découpage, statique ou dynamique, (voir Section 7.1.1) nous permet de faire apparaître l’amélioration des performances liée au découpage dynamique dans SCORE, par



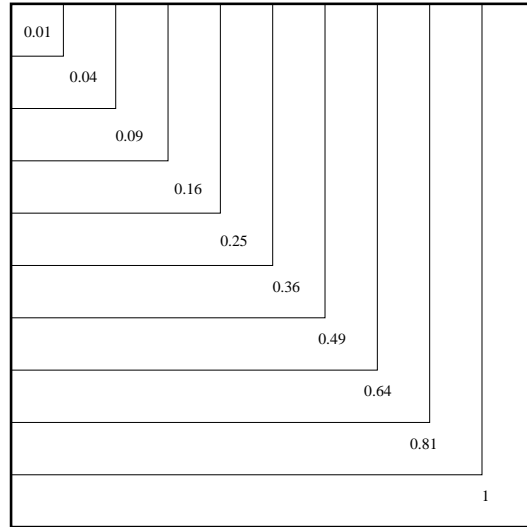


FIG. 7.4 – Les différentes tailles de zone d'intérêt expérimentées

rapport à un découpage statique.

- La surface de la zone d'intérêt ( $IArea$ ) que nous exprimons en fonction de la surface d'une cellule ( $CellArea$ ) dans le cas statique qui est égale au ratio entre la surface de l'EV (ici 1200x1200) et le nombre de groupes multipoints de données disponibles (ici 144). La Figure 7.4 nous montre les différentes zones d'intérêt possibles, à l'intérieur d'une cellule.
- **La période de redécoupage** ( $PR$ ) exprime la durée en secondes, séparant deux découpages de l'EV en cellules au niveau de l'agent. Elle est égale ici à la période séparant deux messages de signalisation émis par l'agent dans chaque groupe de signalisation. Nous avons étudié deux situations avec  $PR = 1s$  et avec  $PR = 6s$ .
- La vitesse  $V$  de déplacement des participants au sein de l'EV nous donne le degré de dynamicité des participants (voir Section 7.1.4). Nous avons comparé deux situations avec  $V = 100$  et  $V = 10$ , sachant que dans le cas d'un découpage statique la surface d'une cellule est de 100x100.
- La distribution des capacités  $C$  reflète le degré d'hétérogénéité entre les participants. Nous nous sommes penchés sur trois situations différentes. Dans les deux premières situations les

capacités sont uniformément distribuées, respectivement entre 20 et 40 sources/s, et entre 10 et 50 sources/s. Dans la troisième situation, 99% des participants ont leur capacités uniformément distribuées entre 30 et 50 sources et 1% des participants entre 10 et 20 sources.

Pour chacune de nos expérimentations, nous avons 1000 participants dans l'EV et 144 groupes multipoints disponibles. Par conséquent, le seuil de redécoupage (voir Section 6.7) est égal à  $\frac{1000}{144} = 6.94$  participants par cellule.

## 7.4 Mise en évidence de l'amélioration de performance liée à SCORE

Sur les figures apparaissant dans cette section, nous pouvons observer les distributions cumulatives des satisfactions (voir Section 6.2) de l'ensemble des participants, calculées à partir des formules développées dans le modèle de la Section 5.3.1. Ces satisfactions ont été sauvegardées tout au long de la session dans des fichiers de trace. Pour une valeur donnée de satisfaction prise sur l'axe des  $x$ , nous pouvons observer, pour chaque courbe (en projetant sur l'axe des  $y$ ), le pourcentage de satisfactions inférieures ou égales à cette valeur. Il est à noter que très souvent, les courbes se recoupent lorsque la satisfaction tend vers 1 (typiquement autour de 0.8). Ce phénomène s'explique par le fait que nous traçons ici les satisfactions **cumulatives** et chaque courbe se rejoint de toute façon en  $(1, 100)$  (100% des satisfactions sont inférieures ou égales à 1 par définition).

### 7.4.1 Comparaison des satisfactions entre le cas statique et le cas dynamique pour différents niveaux d'hétérogénéité entre les participants

Sur les Figures 7.5 et 7.6, nous pouvons comparer les satisfactions pour un découpage statique et pour un découpage dynamique. Deux niveaux d'hétérogénéité sont présentés sur chacune des figures, avec des capacités uniformément distribuées entre 20 et 40 sources/s, et entre 10 et 50

sources/s. A partir de ces 10 figures, correspondant à 10 tailles de zones d'intérêt différentes, nous pouvons dégager les observations suivantes :

Sur toutes les figures, et quelque soit le niveau d'hétérogénéité entre les participants, SCORE permet une amélioration très nette des satisfactions. En effet, pour un niveau d'hétérogénéité donné, la courbe représentant le cas dynamique (SCORE) est toujours en dessous de la courbe représentant le cas statique. Prenons nous dans le cas où  $I\text{Area} = 0.25\text{CellArea}$ . Nous observons ici que dans le cas dynamique, moins de 5% des participants pour  $C \in [20, 40]$ , et moins de 20% des participants pour  $C \in [10, 50]$  ont une satisfaction inférieure à 0.8, alors que dans le cas statique, entre 40% et 50% des participants sont dans ce cas. Lorsque  $I\text{Area} = 0.16\text{CellArea}$  la différence est encore plus visible puisque près de 100% des participants pour  $C \in [20, 40]$  ont une satisfaction supérieure à 0.9 grâce aux mécanismes de SCORE. En revanche, dans le cas statique, 50% des participants ont une satisfaction inférieure à 0.9, et 10% des participants ont une satisfaction inférieure à 0.5. De plus, si on observe les satisfactions minimales relevées en cours de session, on s'aperçoit que SCORE permet également d'améliorer les performances par rapport à un découpage statique. Ainsi sur la figure correspondant à  $I\text{Area} = 0.04\text{CellArea}$ , les satisfactions minimales que l'on peut observer sont respectivement 0.9 ( $C \in [20, 40]$ ) et 0.6 ( $C \in [10, 50]$ ) dans le cas dynamique, et 0.55 ( $C \in [20, 40]$ ) et 0.3 dans le cas statique.

Ces résultats démontrent donc l'amélioration de la scalabilité qu'apporte SCORE par rapport à un découpage statique. Cependant, il convient de limiter ces bons résultats aux constatations suivantes : Lorsque la surface occupée par la zone d'intérêt devient grande, les performances se dégradent quelque soit le mode de découpage en cellules. De plus, cette dégradation semble avoir pour effet de diminuer progressivement l'amélioration des satisfactions résultant d'un découpage dynamique. Ainsi sur la figure correspondant à  $I\text{Area} = \text{CellArea}$ , l'écart entre les courbes dynamiques et statiques est bien moindre que sur la figure où  $I\text{Area} = 0.36\text{CellArea}$ .

De la même façon, lorsque la surface occupée par la zone d'intérêt devient très petite, le trafic reçu diminue et les satisfactions des participants tendent vers 1, dans le cas dynamique

comme dans le cas statique. Dans ce cas, SCORE présente un intérêt moindre puisqu'il introduit un surcoût inutile par rapport à un découpage statique. Toutefois, si le nombre de participants augmente, cela revient à augmenter la taille de la zone d'intérêt, et dans ce cas SCORE permet d'améliorer sensiblement les satisfactions des participants.

Enfin, si nous comparons l'évolution des courbes de satisfactions entre les deux niveaux d'hétérogénéité présentés dans cette étude, un phénomène intéressant apparaît. En effet, pour la totalité des courbes concernant le cas dynamique, le passage de  $C \in [20, 40]$  à  $C \in [10, 50]$  entraîne une diminution des satisfactions. Bien que cela soit le plus souvent le cas lors du découpage statique, la figure correspondant à  $I\text{Area} = 0.36\text{CellArea}$  nous montre le résultat inverse. Cette fois-ci, les satisfactions s'améliorent globalement lorsque  $C \in [10, 50]$  pour le cas statique alors qu'elle se détériorent dans le cas dynamique. Ceci semble s'expliquer par le fait que l'algorithme de découpage de SCORE, diminue la taille des cellules dans les Zones où se trouvent le participant le moins satisfait (i.e. "max-min fairness"). Cette approche vise donc à maximiser la satisfaction minimale plutôt qu'à augmenter la moyenne des satisfactions. Une autre "politique d'équité" pourrait toutefois être appliquée dans SCORE sans modification majeure de notre architecture. La section suivante nous démontre cependant l'intérêt et l'efficacité des mécanismes de gestion de l'hétérogénéité dans SCORE, qui apparaît beaucoup plus clairement dans le cas d'une distribution non uniforme des capacités.

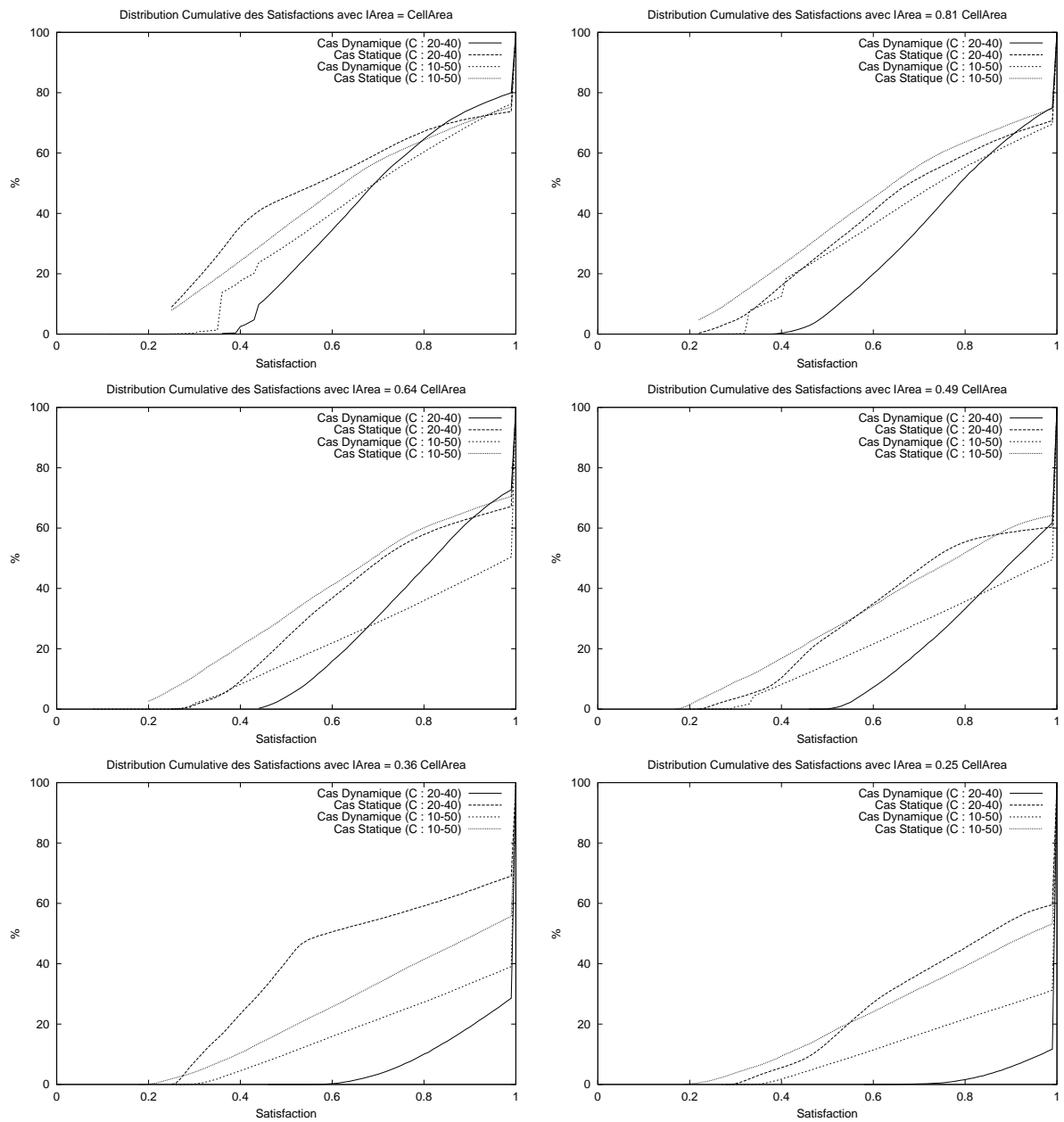


FIG. 7.5 – *Comparison des Satisfactions avec  $V = 100$ ,  $PR = 1s$  et  $IArea \geq 0.25$*

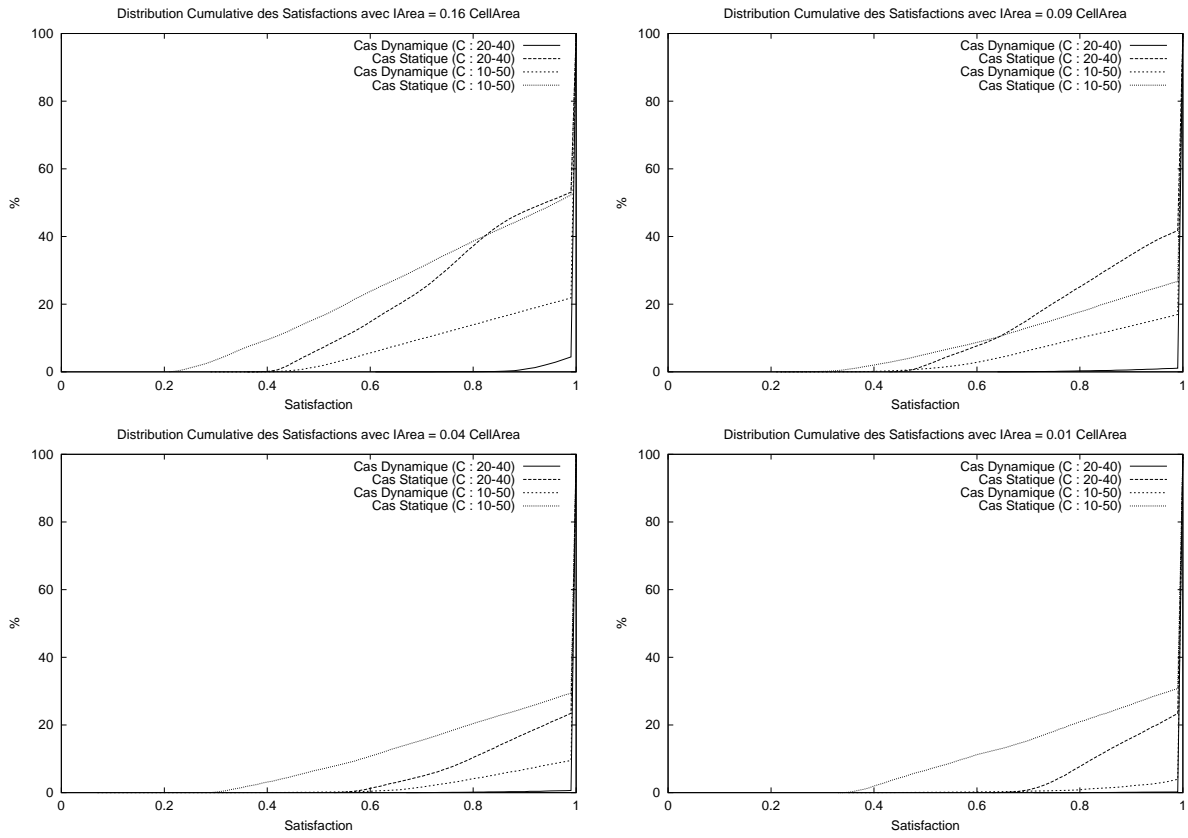


FIG. 7.6 – *Comparaison des Satisfactions avec  $V = 100$ ,  $PR = 1s$  et  $IArea \leq 0.16$*

## 7.4.2 Comparaison des satisfactions pour deux distributions différentes des capacités

Sur les graphes apparaissant sur la Figure 7.7, nous comparons les satisfactions de 10 participants, soit 1% de la population totale de l'EV, pour deux distributions différentes des capacités. Dans une première distribution, ces 10 participants ont leur capacités uniformément distribuées dans  $[10, 20]$ , tandis que les 99% restant ont leur capacités uniformément distribuées dans  $[30, 50]$ . Nous qualifions donc cette distribution globale des capacités de **non uniforme** (les courbes en ligne continue sur les figures). Dans une deuxième distribution, les capacités de l'ensemble des 1000 participants sont toutes uniformément distribuées dans  $[10, 20]$ . Nous sommes donc dans le cas d'une distribution **uniforme** (les courbes en pointillés). A partir de ces 8 figures, correspondant à 8 tailles de zones d'intérêt différentes, nous pouvons dégager les observations suivantes :

Dans le cas où la zone d'intérêt est "grande", c'est à dire lorsque celle-ci se rapproche d'une taille de cellule (on rappelle que cette taille est celle d'une cellule prise après un découpage statique), la distribution cumulative des capacités est sensiblement identique dans les deux distributions de capacité. C'est le cas pour  $I\text{Area} = \text{CellArea}$  et pour  $I\text{Area} = 0.81\text{CellArea}$ , où les deux courbes nous montrent également que le nombre de participants ayant une satisfaction maximale est proche de 0. Dans le cas d'une distribution non uniforme des capacités, certains participants dont la capacité est pourtant supérieure, se trouvent dans des Zones plus denses, et par conséquent leur satisfaction est plus faible. SCORE va donc diminuer en premier la taille des cellules de ces Zones. Toutefois, même si l'ensemble des satisfactions est inférieur à 1, il se peut que SCORE alloue plus de cellules dans les Zones où se trouvent les 10 participants à faible capacité. Mais après cette première allocation de ressources (il s'agit de groupes multipoints), la satisfaction de ces 10 participants va dépasser la satisfaction d'autres participants, et dans ce cas aucune nouvelle allocation n'aura lieu dans leur Zone.

En revanche sur les autres graphes, typiquement pour  $I\text{Area} \leq 0.64\text{CellArea}$ , de plus en plus

de participants ayant une capacité uniformément distribuées dans  $[30, 50]$ , ont une satisfaction maximale. Par conséquent, les deux explications énoncées au précédent paragraphe ne sont plus vraies. Dans ce cas, après un premier calcul des tailles de cellule dans les différentes Zones en fonction de la densité, ces participants auront une satisfaction proche de 1. L'intégralité des groupes multipoints restant sera donc allouée dans les Zones où se trouvent les 10 participants de faible capacité. On s'aperçoit ici que les mécanismes de SCORE prennent donc efficacement en compte l'hétérogénéité des participants. En effet, si on regarde le graphe de la Figure 7.7 correspondant à  $I\text{Area} = 0.49\text{CellArea}$ , on observe que moins de 40% des satisfactions sont inférieures à 0.5 dans le cas d'une distribution non uniforme des capacités, alors que ce pourcentage monte à près de 80% dans le cas d'une distribution uniforme des capacités.



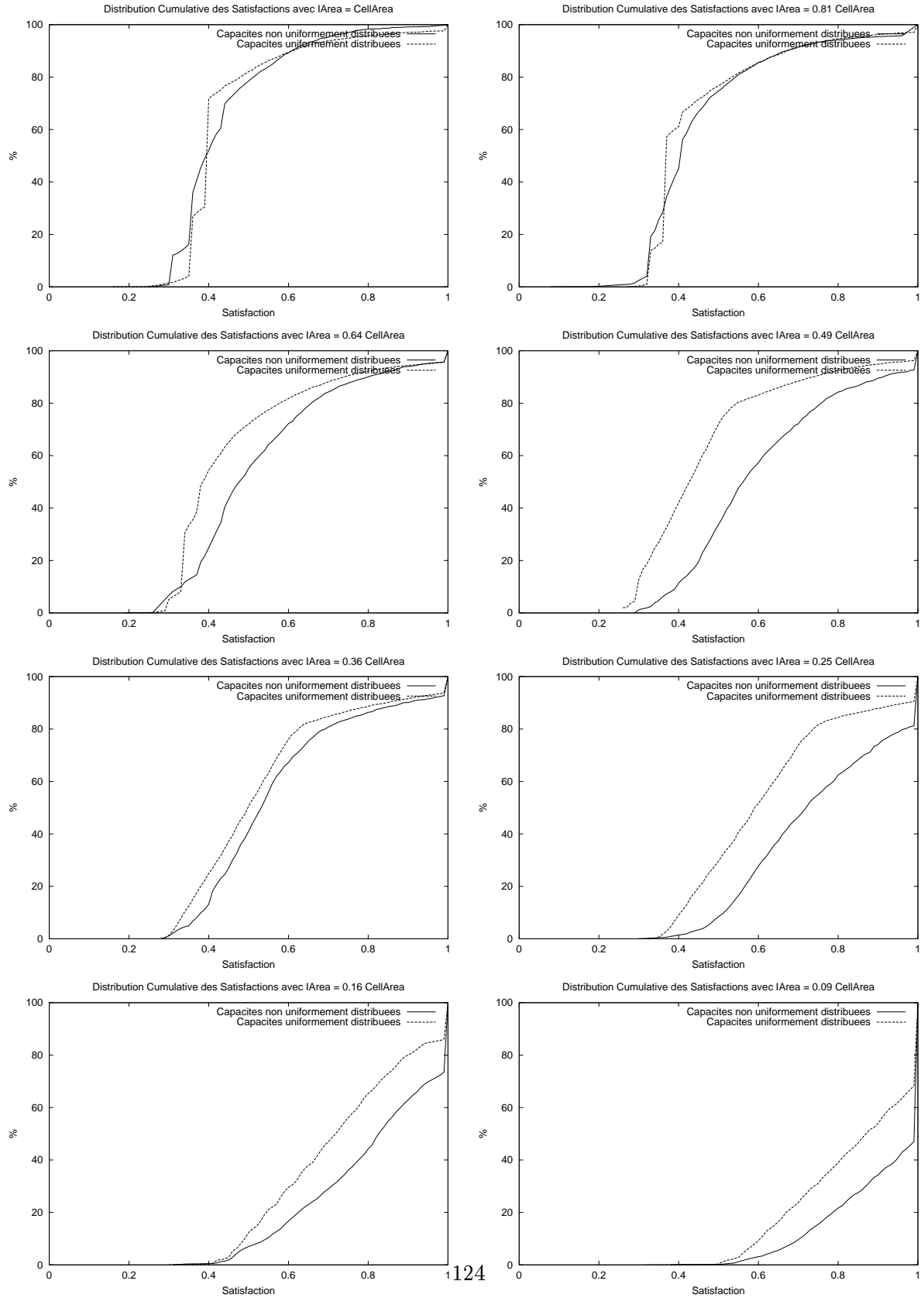


FIG. 7.7 – Comparaison des satisfactions pour deux distributions différentes des capacités

### 7.4.3 Comparaison des satisfactions en fonction de la période de redécoupage et de la dynamicité

Sur les graphes apparaissant sur les Figures 7.8 et 7.9, nous pouvons observer les distributions cumulatives des participants pour deux périodes de découpage différentes, et pour deux vitesses différentes des participants. Ces graphes présentent les résultats obtenus au cours des expérimentations pour 8 tailles différentes de surfaces d'intérêt. A partir de ces 16 figures, nous pouvons dégager les observations suivantes :

Sur la quasi totalité des graphes, le découpage dynamique permet d'obtenir de meilleures satisfactions que dans la cas statique. Ceci est vrai pour les deux niveaux de dynamicité des participants, quelque soit la période de redécoupage. Cependant, la graphe de la Figure 7.8 correspondant à  $V = 100$  et  $I\text{Area} = 0.81\text{CellArea}$  ne vérifie pas (en tout cas pas assez nettement) cette amélioration de performance, lorsque la période de redécoupage est de 6s. En effet, on constate que le pourcentage de satisfactions inférieures à 0.8 est légèrement plus important dans le cas dynamique (pour  $PR = 6s$ ) que dans le cas statique. On explique ce phénomène par la combinaison simultanée d'une vitesse de déplacement élevée et d'une fréquence de redécoupage très faible. Entre deux instants de découpage de l'EV par les agents, les distributions des participants au sein des différentes Zones de l'EV ont pu être suffisamment modifiées pour qu'un découpage statique soit finalement plus efficace "en moyenne" entre ses deux instants qu'un découpage dynamique. Cette explication s'est d'ailleurs confirmée beaucoup plus nettement au cours d'expérimentations où nous avons choisi une période de redécoupage exagérément grande (celles-ci ne sont toutefois pas présentées ici).

Si nous observons plus finement la différence entre les deux courbes correspondant aux cas dynamiques dans l'ensemble des figures, nous constatons que le passage d'une période de redécoupage de  $PR = 6s$  à  $PR = 1s$  améliore sensiblement les satisfactions dans le cas où  $V = 100$  par rapport à  $V = 10$ . Cette constatation est particulièrement remarquable pour une zone

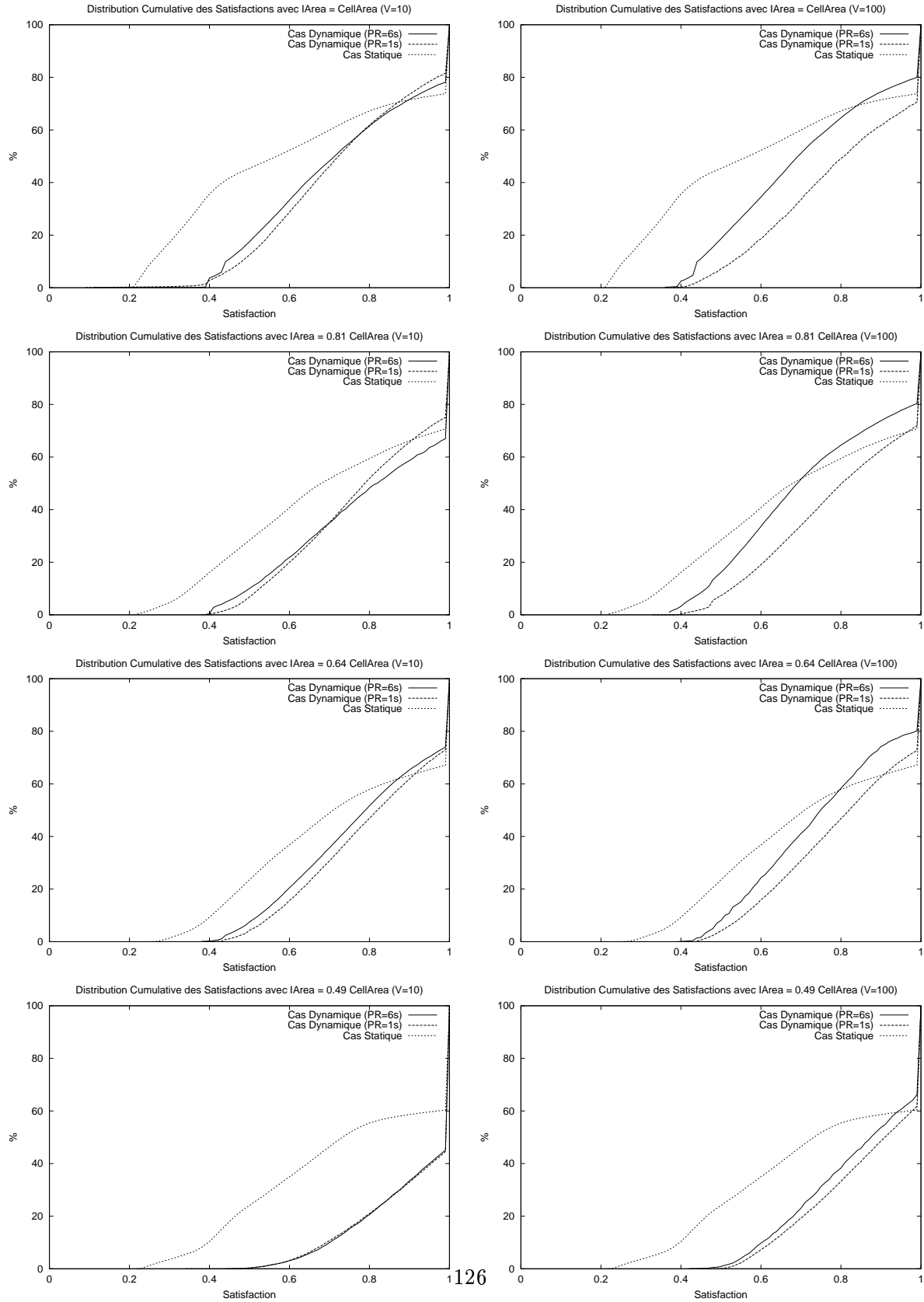


FIG. 7.8 – Comparaison des satisfactions pour  $PR = 1s$  ou  $6s$ ,  $V = 10$  ou  $100$  et avec  $IArea \geq 0.49$

d'intérêt  $I\text{Area} \geq 0.25\text{CellArea}$ . Par exemple, si on se place sur le graphe correspondant à  $I\text{Area} = 0.81\text{CellArea}$ , on constate que 40% des participants ont une satisfaction inférieure à 0.7 dans le cas où  $V = 10$ , que ce soit avec une période découpage de 1s ou de 6s. En revanche, dans le cas où  $V = 100$ , ce pourcentage est également de 40% pour  $PR = 1s$  mais monte à près de 60% lorsque la période de redécoupage passe à 6s. Cependant, lorsque l'on renouvelle l'expérience avec des surfaces de zone d'intérêt plus petite ( $I\text{Area} \leq 0.16\text{CellArea}$ ), l'amélioration des satisfactions liée à l'augmentation de la fréquence de redécoupage devient moins visible voire nulle. Ceci s'explique par le fait que la plupart des satisfactions est déjà maximale avec  $PR = 6s$ , par conséquent l'amélioration des performances ne porte plus que sur un nombre très faible de participants lorsqu'on passe à  $PR = 1s$ .

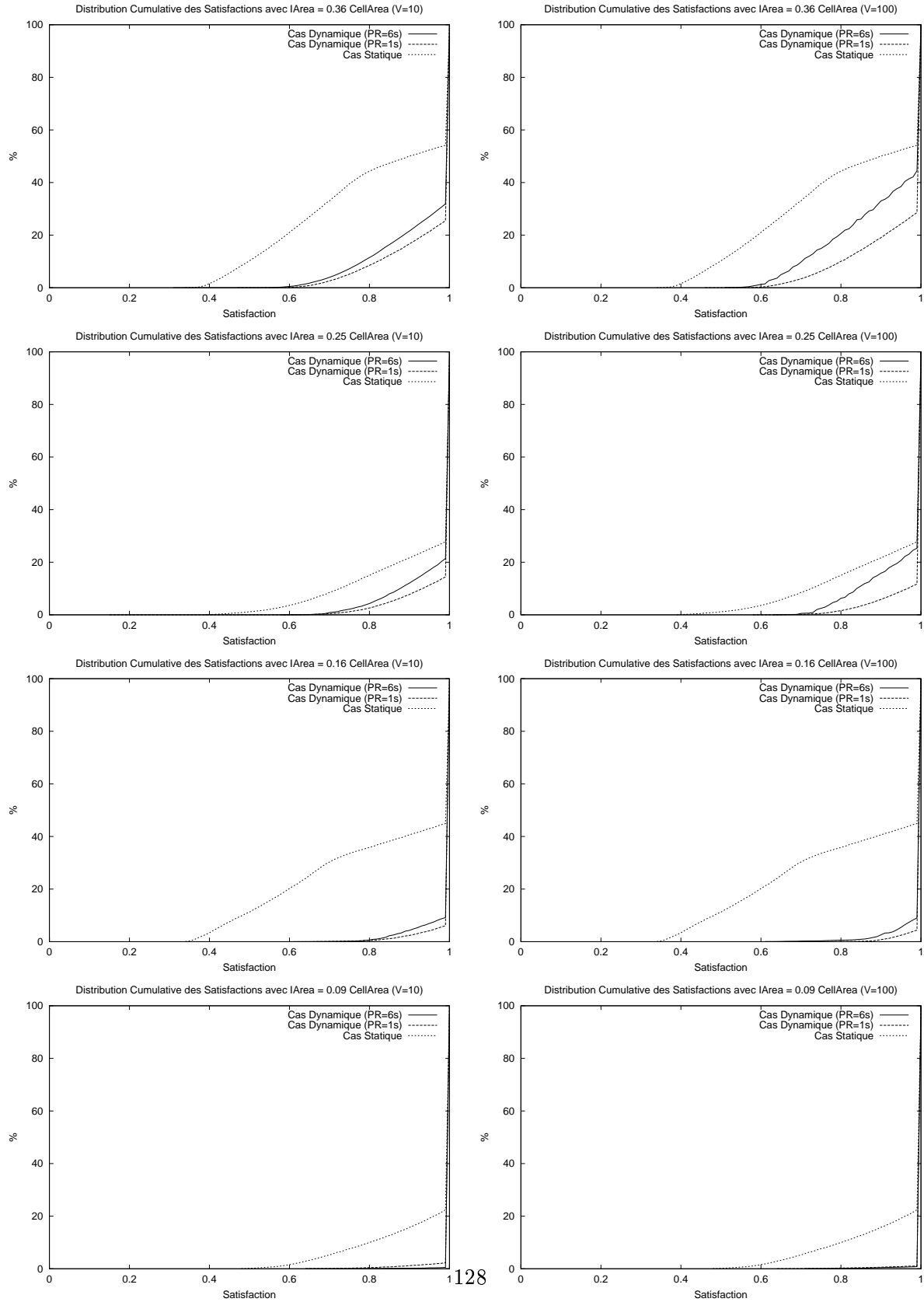


FIG. 7.9 – Comparaison des satisfactions pour  $PR = 1s$  ou  $6s$ ,  $V = 10$  ou  $100$  et avec  $0.09 \leq IArea \leq 0.36$

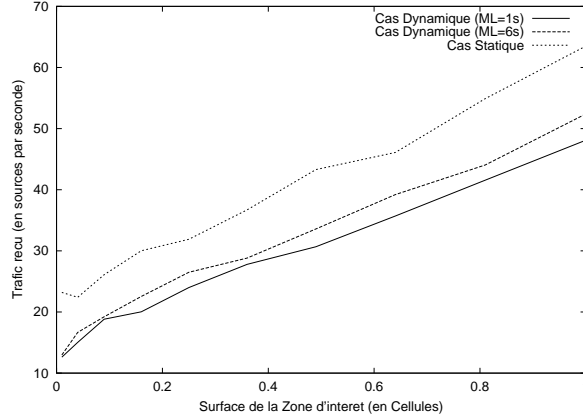


FIG. 7.10 – *Trafic de données entrant (Nb Sources/s) par participant ( $V = 10$ )*

#### 7.4.4 Comparaison du trafic de données reçu avec et sans SCORE

Sur la Figure 7.10, nous observons la moyenne du trafic de données reçu par un participant type. Ce trafic intervient dans le calcul de la satisfaction du participant. Nous avons choisi de présenter cette mesure dans le but de démontrer l'efficacité des mécanismes de filtrage basés sur l'utilisation de plusieurs groupes multipoints avec découpage dynamique, tels que ceux implémentés dans SCORE. Ce trafic est mesuré en sources/s pour deux périodes de redécoupage différentes ainsi que dans le cas statique. Etant donné que nos expérimentations ont été réalisées sur un réseau local, la bande passante est ici largement supérieure à la totalité du trafic de données échangées entre l'ensemble des participants. Par conséquent, nous pouvons considérer ce trafic reçu comme le trafic total entrant pour un participant type, le taux de perte étant négligeable.

Nous pouvons observer que l'écart entre les trois courbes reste approximativement constant lorsque la zone d'intérêt du participant passe de  $I\text{Area} = 0.01\text{CellArea}$  à  $I\text{Area} = 1\text{CellArea}$ . En revanche, l'écart relatif entre les courbes n'est pas le même entre les différentes tailles de zones d'intérêt. Typiquement, pour  $I\text{Area} = 0.16\text{CellArea}$ , ce trafic est égal à 20 sources/s dans le cas dynamique ( $PR = 1s$ ) et de 30 sources/s dans le cas statique, soit une augmentation de

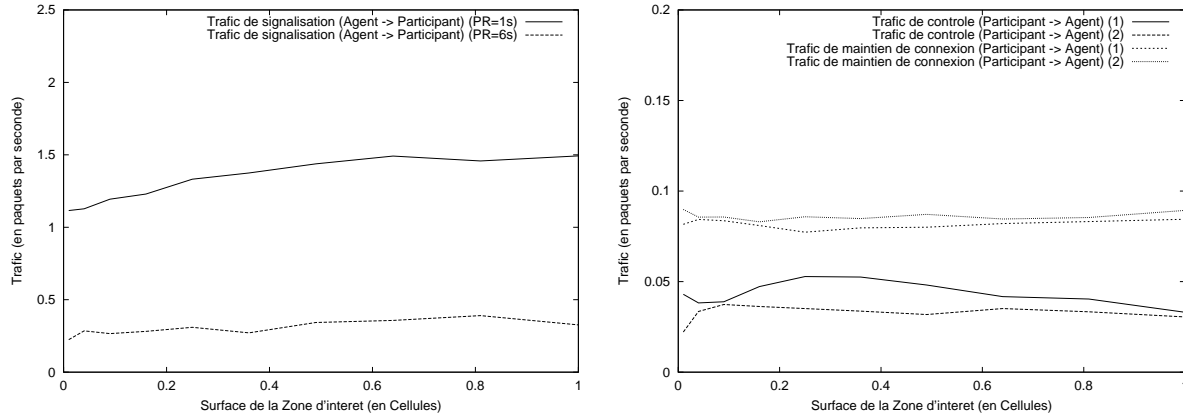


FIG. 7.11 – *Trafic de signalisation reçu et de contrôle émis (en paquets/s) avec  $V = 10$*

50% du trafic entrant. Il faut d'ailleurs considérer ces 50% de trafic supplémentaire comme étant exclusivement du trafic superflu (voir Section 7.1.1). En effet, que ce soit avec ou sans découpage dynamique, le participant s'abonne ici à l'ensemble des groupes multipoints associés aux cellules intersectant sa zone d'intérêt. Par conséquent, la courbe correspondant au découpage dynamique avec  $PR = 1s$  (la courbe la plus basse sur le graphe de la Figure 7.10) nous donne déjà une borne max sur le trafic entrant utile. En revanche pour  $I\text{Area} = \text{CellArea}$ , le trafic entrant passe approximativement pour le découpage dynamique de 50 sources/s à 65 sources/s pour le découpage statique, présentant ainsi une augmentation de 30% du trafic. Les mécanismes de SCORE permettent donc aux participants de mieux approximer leur zone d'intérêt grâce à des cellules plus petites. Plus exactement, cette meilleure approximation de la zone d'intérêt n'est possible que dans les régions de l'EV où la densité de participants est la plus élevée. En effet, dans ces régions, une légère diminution de la surface superflue permet de réduire une grande quantité de trafic superflu entrant.

## 7.5 Mesure et estimation du surcoût lié à SCORE

### 7.5.1 Comparaison du trafic de signalisation reçu et de contrôle émis

Les Figures 7.11 et 7.12 mettent en évidence le très faible surcoût lié au trafic de signalisation et de contrôle circulant entre l'agent et les participants. Nous mesurons ici le trafic en nombre de paquets par seconde. Or, il est important de rappeler que la taille des paquets de signalisation émis par l'agent dans les groupes de signalisation est ici de 8 octets (charge utile) tandis qu'elle est de 16 octets (charge utile) pour les paquets de contrôle émis par les participants en point-à-point vers l'agent. En rajoutant les 20 octets d'en-tête IP et les 4 octets d'en-tête UDP, on obtient respectivement 32 octets pour les paquets de signalisation et 40 octets pour les paquets de contrôle. Sur la Figure 7.11, le trafic de signalisation pour  $PR = 1s$  est au maximum de 1.5 paquets par seconde pour  $I\text{Area} = \text{CellArea}$ . En effet, les participants anticipent l'abonnement aux groupes multipoints de signalisation en tenant compte de leur vitesse, de la taille de leur zone d'intérêt et de la période de redécoupage (voir Section 6.4.3). Cette valeur de 1.5 nous informe donc qu'un participant ayant une vitesse de 10 et une zone d'intérêt de la taille d'une cellule (soit 100, ce qui est la taille calculée dans le cas statique), est abonné en moyenne à 1.5 groupes de signalisation. Il reçoit donc un trafic de signalisation moyen de 48 octets/s soit 0.384 kbits/s si la période de redécoupage est égale à la période d'émission des paquets dans les groupes multipoints de signalisation. Il faut également rappeler qu'au cours des mises à jour des abonnements (voir Section 6.6), la fréquence d'émission des paquets de signalisation augmente. Toutefois, ce trafic restera toujours négligeable par rapport au trafic de données entrant.

Sur la Figure 7.11 nous observons également le trafic de contrôle émis par deux participants pour  $V = 10$ . Nous présentons sur cette courbe deux traces différentes afin de montrer que le trafic de maintien de connexion diminue lorsque le trafic de contrôle augmente, et réciproquement. Ceci se vérifie pour les courbes des deux participants. La Figure 7.12 nous permet de comparer les trafics de contrôle et de signalisation pour deux vitesses différentes dans le cas



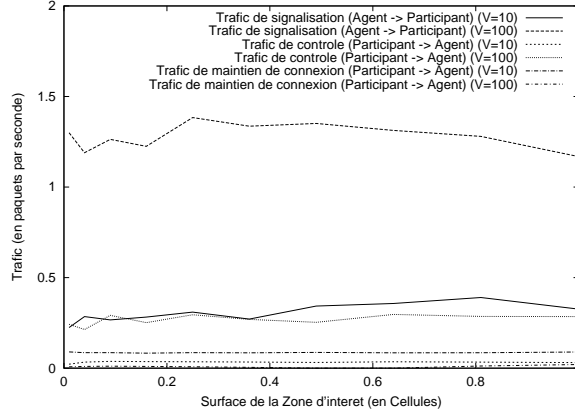


FIG. 7.12 – *Trafic de signalisation reçu et de contrôle émis avec  $PR = 6s$*

où la période de redécoupage est égale à 6s. Il est intéressant de constater deux phénomènes : Contrairement à  $V = 10$ , le trafic de contrôle pour  $V = 100$  est désormais plus important que le trafic de maintien de connexion. En effet, le participant franchit suffisamment fréquemment de Zone-Units pour que le temps moyen séparant l'émission de deux paquets de contrôle devienne inférieur à la période minimale  $P_{min}$  de séparation entre deux messages de contrôle fixée par l'agent (voir Section 6.4.3). Par conséquent, le participant n'ayant pas de connexion à maintenir "artificiellement" avec l'agent, le trafic de maintien de connexion est très proche de 0 paquet/s. Le deuxième phénomène intéressant à constater concerne l'évolution du trafic de signalisation quand la vitesse de déplacement des participants passe de 10 à 100. En effet, ce trafic ne croît pas proportionnellement avec la vitesse bien que le nombre de groupes de signalisation souscrits augmente. Etant donné que nous avons 9 Start-Zones de (400x400), et que la période de redécoupage est de 6s, un participant ayant une vitesse de 100 peut traverser potentiellement un nombre important de Start-Zones entre la réception de deux messages de signalisation. Si les administrateurs d'EVGEs ont une bonne connaissance a-priori de la vitesse des participants, la période d'émission des paquets de signalisation peut être calculée en fonction de la taille moyenne des Start-Zones, de façon à minimiser ce trafic de signalisation.

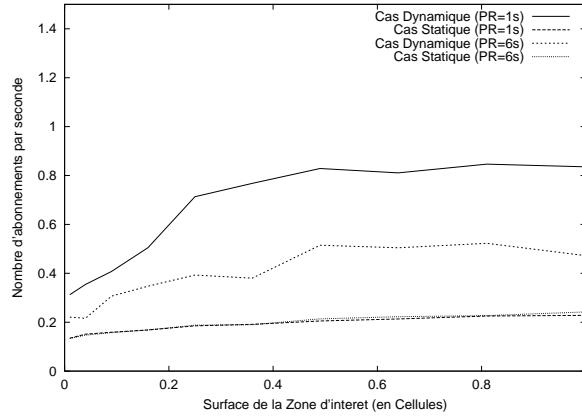


FIG. 7.13 – Fréquence d'abonnements avec  $V = 10$

### 7.5.2 Comparaison de la fréquence d'abonnements en fonction de la période de redécoupage

La Figure 7.13 trace le nombre d'abonnements par seconde (à multiplier par deux pour obtenir le nombre d'abonnements et de désabonnements par seconde) pour deux périodes de redécoupage différentes avec une vitesse  $V = 10$ . On remarque que cette fréquence double dans le cas dynamique lorsque la zone d'intérêt passe de  $0.01CellArea$  à  $0.49CellArea$  puis semble se stabiliser ensuite. Ceci s'explique par le fait que la surface de la zone d'intérêt est multipliée par 50 dans la moitié gauche de la figure et uniquement par 2 dans la moitié droite. En conséquence, le nombre moyen de cellules intersectées par la zone d'intérêt croît également beaucoup plus rapidement dans la première partie. Chaque cellule étant associée à un groupe multipoint, l'évolution du nombre d'abonnements par seconde est donc directement liée à ce comportement. Nous verrons un peu plus loin dans ce chapitre l'évolution du nombre de groupes multipoints souscrits par participant en fonction de la taille de sa zone d'intérêt.

Sur la Figure 7.13, on constate que la fréquence d'abonnements est 2 fois supérieure dans le cas où  $ML = 6s$  et 4 fois supérieure dans le cas où  $ML = 1s$ , en comparaison avec le cas statique. Sur la Figure 7.14, nous pouvons également observer l'évolution du nombre d'abonne-

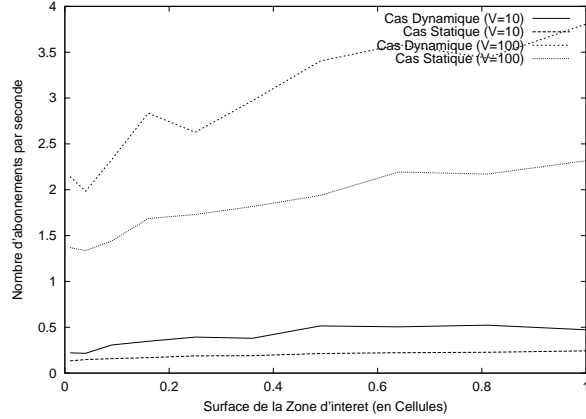


FIG. 7.14 – Fréquence d’abonnement avec  $PR = 6s$

ments par seconde pour une vitesse de déplacement des participants de 100. Si on compare le cas dynamique avec le cas statique pour  $V = 100$ , on se rend compte que cette fois-ci la fréquence d’abonnement avec SCORE est inférieure à 2 fois à la fréquence d’abonnement dans le cas statique. En conséquence, bien que très logiquement la vitesse de déplacement ait un impact direct sur la fréquence d’abonnement, si on compare le cas statique et le cas dynamique pour  $V = 10$  puis pour  $V = 100$ , on s’aperçoit que l’écart (en proportion) entre les fréquences d’abonnements statique et dynamique diminue. Ceci s’explique par le fait que SCORE calcule des tailles de cellule plus petites dans les régions où se trouvent la grande majorité des participants (notre découpage dynamique est basé sur la densité de participants). Avec  $V = 100$  et  $PR = 6s$ , cette affirmation est vraie principalement au moment où un nouveau découpage a lieu. Entre deux découpages, la répartition des participants va changer beaucoup plus nettement que si on avait  $V = 10$ . Nous avons vu dans les analyses portant sur les satisfactions cumulatives que cela avait d’ailleurs un impact sur les satisfactions des participants.

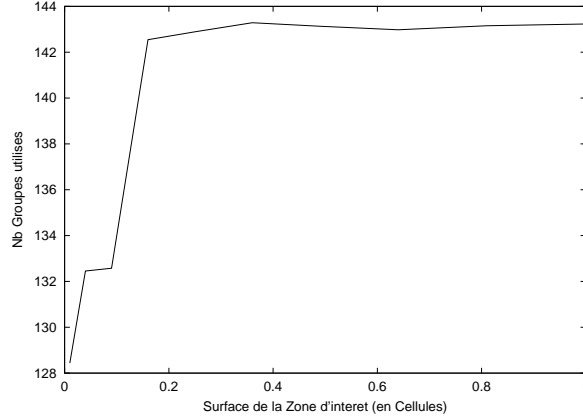


FIG. 7.15 – Nombre de groupes multipoints utilisé par l'agent avec  $C \in [20, 40]$

## 7.6 Utilisation des groupes multipoints dans SCORE

### 7.6.1 Nombre de groupes multipoints utilisés par l'agent

Sur la Figure 7.15, nous pouvons observer le nombre moyen de groupes multipoints de données utilisés par l'agent lors du découpage dynamique de l'EV en cellules. Le nombre de groupes multipoints utilisés correspond donc au nombre total de cellules de l'environnement virtuel. Sur l'axe des  $x$ , nous faisons varier la surface de la zone d'intérêt. La lecture de cette courbe nous permettra également de mieux comprendre certains des résultats des sections suivantes.

Pour pouvoir interpréter les résultats de cette courbe, il est important de se référer à l'algorithme de découpage implémenté dans SCORE (voir Section 6.7). En fait, au cours de l'exécution de cet algorithme, une taille de cellule est calculée dans chaque Zone de l'EV en fonction du résultat de la comparaison entre la densité de participants par cellule dans la Zone, et le seuil de redécoupage. A l'issue de ce premier calcul, et compte tenu du fait que les tailles de cellules prennent des valeurs discrètes (voir Section 7.1.2), il reste toujours un certain nombre de groupes disponibles. Ici, ce nombre est égal à la différence entre 144 et le nombre de cellules calculé dans la première étape de notre algorithme de découpage. Si les participants ont tous une satisfaction maximale, ces groupes multipoints restant ne seront pas inutilement alloués, réduisant ainsi le surcoût lié

à l'établissement d'un groupe multipoint dans le réseau. En revanche, lorsque la zone d'intérêt augmente, certains participants commencent à subir des pertes, liées au fait que leur trafic entrant dépasse leur capacité. Dans ce cas, et afin d'augmenter leur satisfaction, l'agent va réduire la taille de cellule de la Zone dans laquelle ils se trouvent. Bien entendu, cette réduction de taille de cellule va entraîner l'utilisation d'un nombre plus important de groupes multipoints disponibles. Il doit donc rester à l'agent un nombre de groupes multipoints disponible suffisamment important afin d'effectuer cette deuxième étape dans l'algorithme de découpage.

### 7.6.2 Comparaison du nombre de groupes multipoints souscrits par participant

Sur les graphes apparaissant sur les Figures 7.16 et 7.17, nous pouvons observer l'évolution du nombre de groupes multipoints souscrits par un participant tout au long d'une session. Ces graphes présentent les résultats obtenus au cours des expérimentations pour une période de redécoupage  $PR = 6s$ , une vitesse de déplacement  $V = 10$  et pour 10 tailles de zone d'intérêt différentes. Sur chacun des graphes, nous comparons le cas statique avec le cas dynamique tel que nous l'avons implémenté dans SCORE. A partir de ces 10 figures, nous pouvons dégager les observations suivantes : Pour les zones d'intérêt de surface  $IArea \leq 0.49CellArea$ , le nombre de groupes multipoints souscrits dans le cas statique évolue globalement dans le même intervalle  $([1, 4])$  que pour le cas dynamique. Ceci peut s'expliquer pour les raisons suivantes : Premièrement pour une raison évidente liée à la petite taille de la zone d'intérêt qui va intersecter moins de cellules qu'une zone d'intérêt plus "vaste". Deuxièmement, la zone d'intérêt étant petite, un premier découpage dynamique en fonction de la densité permet à la plupart des participants d'obtenir une satisfaction maximum. Par conséquent, l'agent ne va pas calculer une taille de cellule encore plus petite pour une Zone donnée si les satisfactions des participants si trouvant sont déjà maximales (voir l'analyse de la Section 7.6.1). Or, calculer une taille de cellule plus petite revient à affecter plus de groupes multipoints pour cette Zone et donc à augmenter le nombre de groupes souscrits pour les participants se trouvant dans la Zone.

En revanche, lorsque la surface de la zone d'intérêt augmente, de plus en plus de cellules sont intersectées. D'ailleurs, si on observe l'évolution du nombre de groupes multipoints souscrits dans le cas statique, on observe que celui-ci prend de plus en plus souvent la valeur de 4 quand  $I\text{Area}$  tend vers  $Cell\text{Area}$ . Par conséquent, le trafic entrant augmente, et les participants ayant une faible capacité voit leur satisfaction diminuée. L'agent diminue donc la taille de cellule dans les Zones où se trouvent les participants les moins satisfaits. Ceci a pour effet d'augmenter le nombre de cellules intersectées (donc de groupes multipoints souscrits) tout en diminuant la surface de la zone superflue du participant précédemment insatisfait, en lui permettant une meilleure approximation de sa zone d'intérêt. Ce phénomène s'accroît encore sur les graphes correspondant à  $I\text{Area} = 0.81\text{CellArea}$  et  $I\text{Area} = \text{CellArea}$ . En effet, la probabilité pour que le participant le moins satisfait se trouve dans la zone la plus dense, augmente. Par conséquent, notre participant ayant également une forte probabilité de se diriger vers les Zones chaudes (voir Section 7.1.4) et dense en participants, sera plus souvent en présence de petites cellules. Ceci explique les nombres de groupes multipoints souscrits atteignant souvent 14 ou 16.

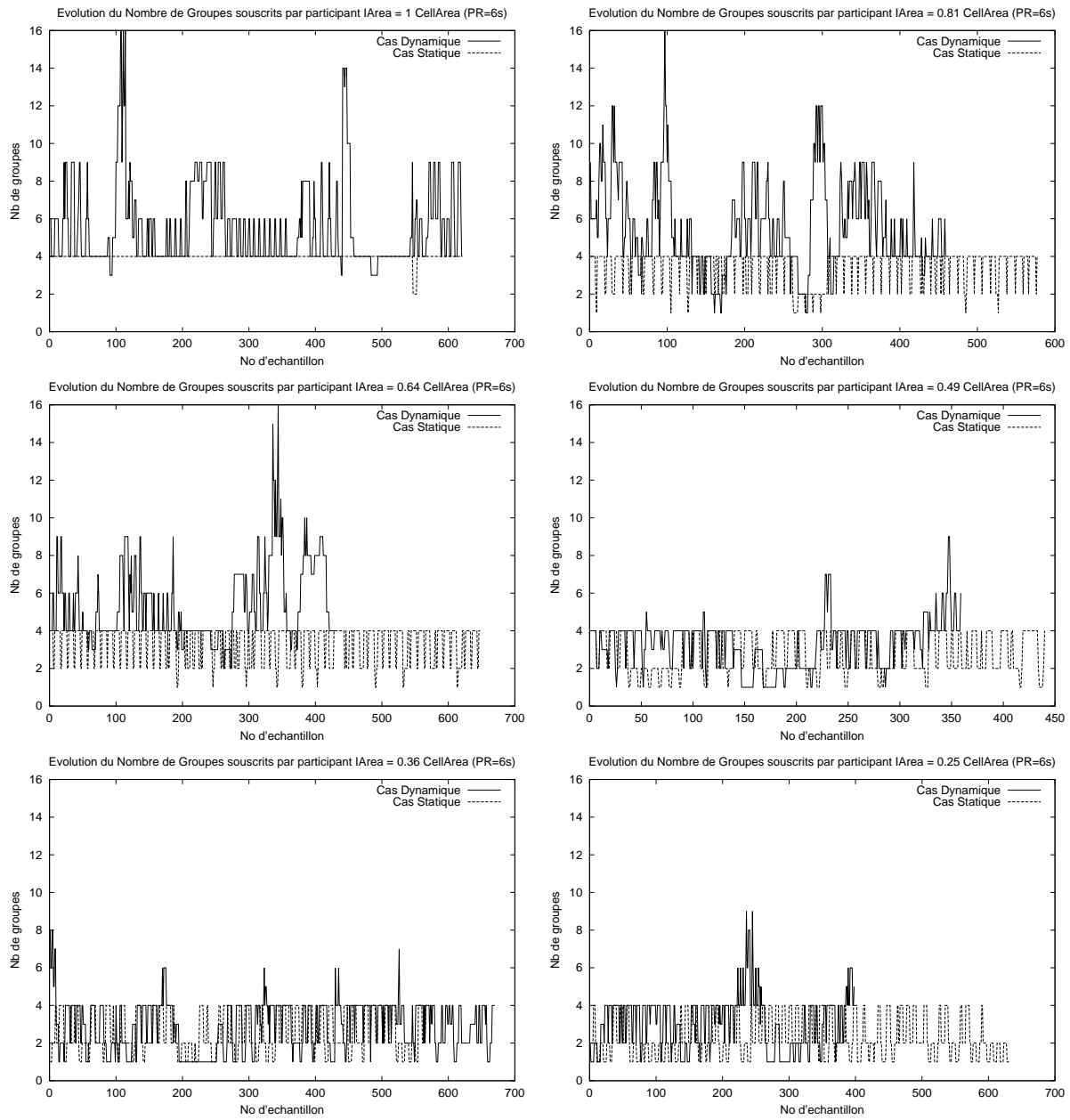


FIG. 7.16 – Evolution du nombre de groupes souscrits par participant avec  $V = 10$  et  $IArea \geq 0.25$

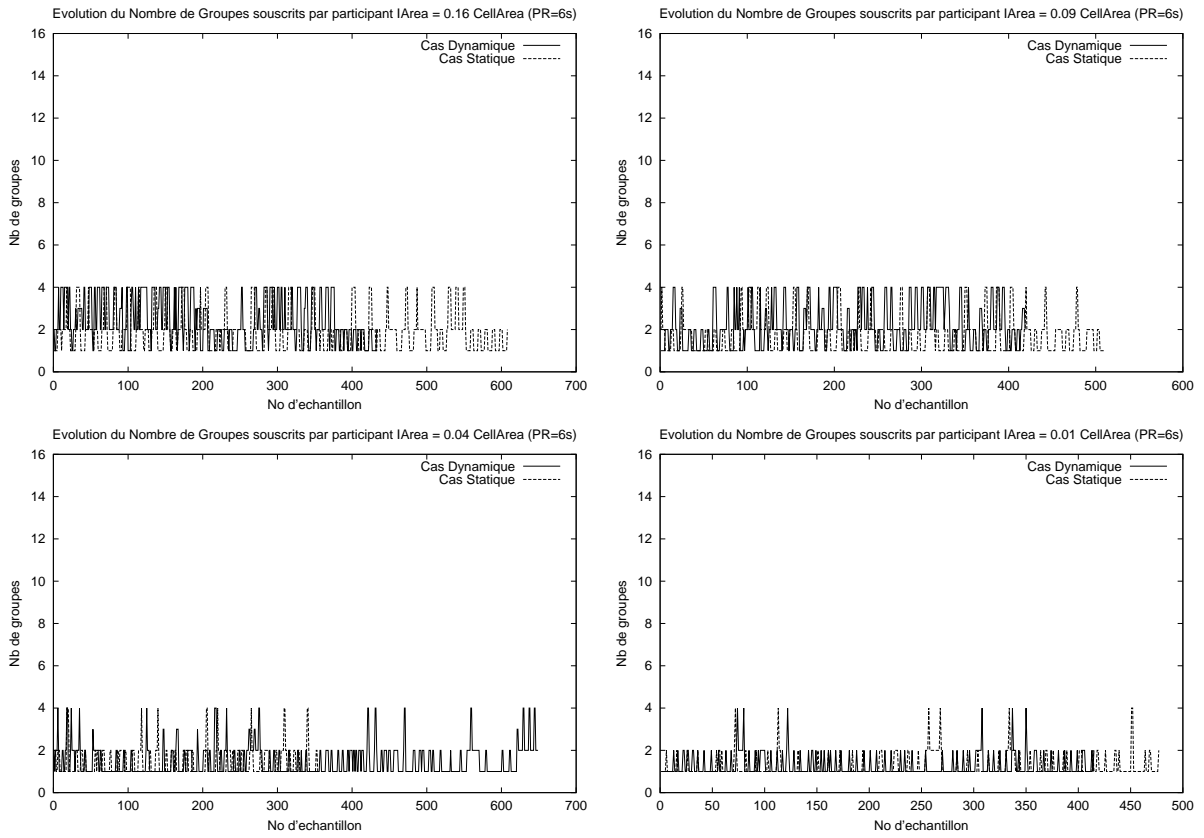


FIG. 7.17 – Evolution du nombre de groupes souscrits par participant avec  $V = 10$  et  $IArea \leq 0.16$



### 7.6.3 Distribution des participants dans les groupes multipoints

Sur les graphes apparaissant sur la Figure 7.18, nous pouvons observer la distribution des participants dans les groupes multipoints. Pour un nombre de participants  $N$  donné pris sur l'axe des  $x$ , nous pouvons observer pour chaque courbe, le pourcentage de groupes multipoints contenant entre  $N$  et  $N - 2$  participants abonnés. Dans le cas du découpage dynamique de SCORE, il s'agit des groupes multipoints de données qui ont été effectivement associés à des cellules de l'environnement virtuel. Les groupes multipoints disponibles restant ne sont pas inclus dans cette distribution (voir Section 7.6.1). A partir de ces 8 figures, correspondant à 8 tailles de zones d'intérêt différentes, nous pouvons dégager les observations suivantes :

Les courbes correspondant au découpage dynamique de SCORE présentent toutes un pic aux alentours de 7 participants. Ceci s'explique par le fait que le seuil de redécoupage dans nos expérimentations est égal au ratio entre les 1000 participants connectés et les 144 groupes disponibles, soit  $\frac{1000}{144} = 6.94$  participants par cellule. Ceci nous démontre très clairement les capacités d'adaptation de SCORE aux distributions non uniformes et dynamiques de participants dans les EVPs. En revanche, sur l'ensemble des graphes, les courbes correspondant au cas statique nous montrent que près de 30% des groupes multipoints ne contiennent que 1 ou 2, voire 0 participants. Par conséquent un très grand nombre de ces groupes multipoints sont "sous-exploités" alors qu'ils représentent ici une ressource de première importance en tant qu'outils de filtrage du trafic de données.

De plus, si on compare le pourcentage de groupes multipoints contenant un nombre important de participants, entre un découpage statique et un découpage dynamique, on s'aperçoit que celui-ci est beaucoup moins important lors d'un découpage dynamique. Par exemple, si on se place au niveau du graphe correspondant à  $I\text{Area} = 0.04\text{CellArea}$ , la différence est très nette. Moins de 1% des groupes multipoints dans le cas statique contiennent un nombre d'abonnés supérieur à 17, tandis que dans le cas statique, il faut attendre d'arriver à 27 participants avant de voir un pourcentage de groupes multipoints inférieur à 1.

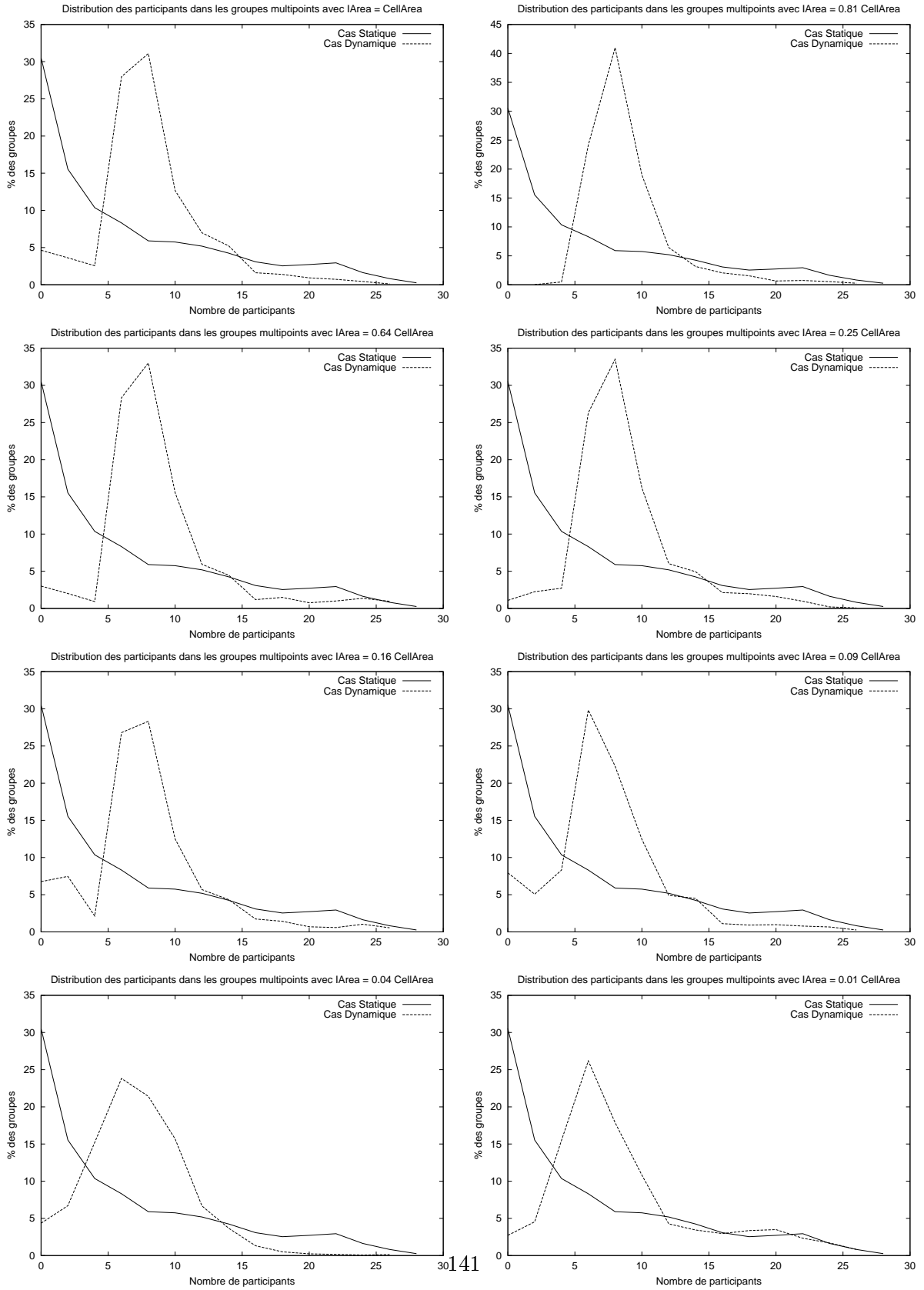


FIG. 7.18 – Distribution des participants dans les groupes multipoints  $V = 10$  et  $PR = 1s$

## 7.7 Conclusion

Nous avons montré dans ce chapitre que l'architecture de communication SCORE permet d'améliorer de manière significative la scalabilité au sein des environnements virtuels distribués à grande-échelle. L'implémentation et les expérimentations de SCORE nous ont permis de démontrer la faisabilité de cette architecture de communication, et le bon fonctionnement de ses mécanismes. De plus, l'analyse des traces générées au cours de nos expérimentations nous a montré qu'il était possible grâce à SCORE, d'améliorer sensiblement les performances, mesurées ici en terme de satisfaction au niveau de l'utilisateur final, pour un coût très faible en terme de trafic de signalisation, de trafic de contrôle, et d'utilisation du CPU au niveau de l'agent (10% du CPU sur un PC Pentium Pro 200Mhz pour 1000 participants). Toutefois, lorsque le nombre de groupes multipoints disponibles est suffisamment grand, que la capacité des participants est importante, ou que la zone d'intérêt des participants est très petite, un découpage statique de l'EV en cellules peut parfois être suffisant. Dans ce cas, SCORE présente un intérêt moindre puisqu'il introduit un surcoût inutile par rapport à un découpage statique. Notons cependant que cette situation est comparable à celle d'un réseau surdimensionné (afin d'éviter les congestions), où dans ce cas, la notion de scalabilité perd un peu de son sens.

# Chapitre 8

## Conclusion

### 8.1 Résumé de nos travaux

Nous avons décrit dans ce mémoire la conception, l'implémentation et l'analyse d'une architecture de communication pour environnements virtuels distribués à grande-échelle sur l'Internet. Nous avons préalablement identifié les besoins et les problèmes de scalabilité liés à ce type d'application. La problématique de mise en réseau à grande échelle d'applications interactives de réalité virtuelle a pu être caractérisée au travers de l'application MiMaze3D, un jeu 3D multi-utilisateurs basé sur l'utilisation d'un unique groupe multipoint. A partir de cette première étude, nous avons proposé une méthode de découpage de l'espace virtuel en cellules de taille identique, prenant en compte la densité de participant ainsi que la taille de leur zone d'intérêt. En nous appuyant sur les résultats obtenus, nous avons défini et implémenté SCORE, une nouvelle architecture de communication permettant d'améliorer la scalabilité et de déployer à grande-échelle des EVPs. Dans cette architecture, nous avons élaboré des mécanismes et des protocoles s'appuyant sur plusieurs agents et sur un ensemble de groupes multipoints. Afin de répondre de façon réaliste aux problèmes de scalabilité identifiés, nous proposons l'utilisation de plusieurs groupes multipoints tout en considérant le coût et les limitations du multipoint. Ces groupes multipoints jouent le rôle d'outils de filtrage de trafic pour les participants.

Sur ce premier niveau d'architecture distribuée, nous avons proposé dans SCORE un second niveau de communication en ajoutant des agents. Leur mission consiste à associer des cellules de différentes tailles aux groupes multipoints en prenant en compte l'évolution de la distribution des participants au sein de l'environnement virtuel. Les informations de mise en correspondance entre les différentes régions de l'environnement virtuel et les groupes multipoints sont diffusées par les agents dans des groupes multipoints de signalisation associés de façon statique à des sous-parties de l'espace virtuel. En s'abonnant ou se désabonnant aux groupes multipoints, les participants peuvent donc non seulement filtrer le trafic de données qu'ils désirent recevoir mais également le trafic de signalisation émis par les agents. En outre, SCORE apporte une réponse aux problèmes de scalabilité dans les EVGEs grâce aux trois mécanismes suivants :

- Son architecture multi-agent doublement distribuée, s'appuyant sur une communication multipoint entre les participants, entre les agents, et entre les agents et les participants.
- Son algorithme de découpage en cellule, prenant en compte non seulement la répartition des participants tout au long de la session, mais également l'hétérogénéité des participants en terme de capacité à recevoir et traiter les informations en temps-réel.
- Sa technique de désignation des cellules, le format de ses informations de mise en correspondance et des messages de contrôle émis, permettant de réduire sensiblement le trafic de signalisation entrant ainsi que le trafic de contrôle émis au niveau des participants.

Nous avons analysé les performances générales de notre architecture de communication, en tenant compte de la limitation du nombre de groupes multipoints disponibles. Nous avons également montré au travers de différentes expérimentations, les cas dans lesquels notre architecture peut s'avérer intéressante ou au contraire introduire un coût supplémentaire non justifié. L'ensemble des analyses montre que les mécanismes proposés dans ce mémoire et implémentés dans SCORE peuvent améliorer de manière significative la scalabilité au sein des environnements virtuels distribués.

## 8.2 Travaux futurs

Les expérimentations de SCORE nous ont permis d'analyser les performances et le comportement des mécanismes présents au sein de notre architecture de communication. Nous avons conduit ces expérimentations en nous plaçant du côté de l'étude de la scalabilité et non de la consistance et de la synchronisation entre les participants. Cependant, SCORE introduit non seulement un coût en terme de trafic de signalisation et de contrôle, que nous avons fait apparaître dans nos mesures, mais a également un impact sur la consistance de l'application. Nous faisons principalement référence à notre mécanisme de mise à jour des abonnements. Les travaux futurs devraient donc porter en premier lieu sur l'étude de ce mécanisme et sur les coûts éventuels qu'il pourrait engendrer. Toutefois, et c'est sans doute l'étape future la plus importante, seule une véritable application de réalité virtuelle déployée sur l'Internet à grande-échelle au dessus de SCORE, permettrait de fournir à cette étude les traces nécessaires à une analyse complète. L'état actuel de déploiement du multipoint, et surtout la réticence de nombreux opérateurs réseaux à laisser circuler le trafic multipoint dans les différents systèmes autonomes de l'Internet rend cette expérimentation à grande-échelle pour l'instant difficile.

En attendant, la réalisation de notre plate-forme d'expérimentation pour SCORE peut également servir de base à d'autres travaux et réflexions. En particulier, l'extension de notre architecture afin de supporter plusieurs flots de données en émission par source, avec des débits variables entre les sources, pourrait constituer la suite logique de nos travaux. Un autre axe de recherche qu'il serait intéressant d'approfondir, porterait sur l'intégration de protocoles de transport multipoint fiable avec SCORE, ainsi que la mise en place de mécanismes de filtrage s'appuyant sur la sélection par source proposée dans IGMPv3[CDT00].



# Bibliographie

- [AA96] Almeroth (Kevin C.) et Ammar (Mostafa H.). – Collecting and modeling the join/leave behavior of multicast group members in the mbone. 1996.
- [AWZ98] Abrams (Howard), Watsen (Kent) et Zida (Michael). – Three-tiered interest management for large-scale virtual environments. *In: Proceedings VRST'98*.
- [BBM<sup>+</sup>94] Blau, B., Moschell (J. M.), et McDonald (B.). – The dis (distributed interactive simulation) protocols and their application to virtual environments. 1994.
- [BBP98] Bagnall (P), Briscoe (R) et Poppitt (A). – Taxonomy of communication requirements for large-scale multicast applications. *Internet-Draft*, November 1998.
- [BCKR98] Bates (T.), Chandra (R.), Katz (D.) et Rekhter (Y.). – Multiprotocol extensions for bgp-4. *RFC 2283*, February 1998.
- [BFC93] Ballardie (Tony), Francis (Paul) et Crowcroft (Jon). – Core based trees (cbt) an architecture for scalable inter-domain multicast routing. *In: Proceedings ACM SIGCOMM'93*.
- [BG87] Bertsekas (D.) et Gallager (R.). – *Data Networks*, chap. 6, pp. 524–529. – Prentice-Hall, 1987.
- [BG97] Benford (Steve) et Greenhalgh (Chris). – *Introducing Third Part Objects into the Spatial Model of Interaction*. – Rapport technique nCGR-97, 1997.
- [BGL97] Benford (Steve), Greenhalgh (Chris) et Lloyd (David). – Crowded collaborative virtual environments. *In: Proceedings ACM CHI'97*.



- [BMR96] Baldoni (R.), Mostefaoui (A.) et Raynal (M.). – Causal delivery of message with real-time data in unreliable networks. *Journal of Real-Time Systems*, vol. 10, 1996, pp. 1–18.
- [BZ99] Baccelli (F.) et Zuyev (S.). – Poisson-voronoi spanning trees with applications to the optimization of communication networks. *to appear in Operations Research*, 1999.
- [CDT00] Cain (Brad), Deering (Steve) et Thyagarajan (Ajit). – Internet group management protocol, version 3. *Internet-Draft*, 2000.
- [CH93] Carlsson et Hagsand. – Dive - a multi user virtual reality system. *In: Proceedings IEEE VRAIS, Seattle, Washington*.
- [CM84] Chang (Jo-Mei) et Maxemchuk (N. F.). – Reliable broadcast protocols. *ACM Transactions on Computer Systems*, vol. 2, n3, August 1984, pp. 251–273.
- [CWSP] Caronniy (Germano), Waldvogelz (Marcel), Sunz (Dan) et Plattnerz (Bernhard). – Efficient security for large and dynamic multicast groups.
- [Dee91] Deering (Steeve). – *Multicast routing in a datagram internetwork*. – Ph.d. dissertation, 1991.
- [DEF<sup>+</sup>96] Deering (Stephen), Estrin (Deborah), Farinacci (Dino), Handley (Mark), Helmy (Ahmed), Jacobson (Van), gung Liu (Ching), Sharma (Puneet) et and (David Thaler Liming Wei). – Protocol independent multicast-sparse mode (pim-sm). *Internet-Draft*, October 1996.
- [Fel71] Feller (William). – *An Introduction to Probability Theory and Its Applications*. – Chapman - Hall, Limited, 1971volume 2.
- [Fel99] Felon (Adrien). – Diffusion, cohérence et contraintes temporelles dans les mondes virtuels 3d - système vreng, September 1999.
- [FJL<sup>+</sup>97] Floyd (Sally), Jacobson (Van), Liu (Ching-Gung), McCanne (Steven), et Zhang (Lixia). – A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, December 1997.

- [FRL<sup>+</sup>00] Farinacci (Dino), Rekhter (Yakov), Lothberg (Peter), Kilmer (Hank) et Hall (Jeremy). – Multicast source discovery protocol (msdp). *Internet-Draft*, February 2000.
- [FT99] Friedman (Timur) et Towsley (Don). – Multicast session membership size estimation. *INFOCOM'99*, 1999.
- [GB95] Greenhalgh (C.) et Benford (S.). – Massive: A distributed virtual reality system incorporating spatial trading. *In: Proceedings 15th International Conference on Distributed Computing Systems (DCS'95)*.
- [GBR99] Greenhalgh (Chris), Benford (Steve) et Reynard (Gail). – A qos architecture for collaborative virtual environments. 1999.
- [GKD99] Gautier (Laurent), Kurose (Jim) et Diot (Christophe). – End-to-end transmission control mechanisms for multiparty interactive applications on the internet. *In: Proceedings IEEE INFOCOM'99*.
- [Gre96a] Greenhalgh (Chris). – *Dynamic, embodied multicast groups in MASSIVE-2*. – Rapport technique nNOTTCS-TR-96-8 1, 1996.
- [Gre96b] Greenhalgh (Chris). – *Spatial Scope and Multicast in Large Virtual Environments*. – Rapport technique nTTCS-TR-96-7 1, September 1996.
- [HC99] Holbrook (Hugh W.) et Cheriton (David R.). – Express multicast making multicast economically viable. *In: Proceedings ACM SIGCOMM'99*.
- [HPS99] Hanna (S.), Patel (B.) et Shah (M.). – Multicast address dynamic client allocation protocol (madcap). *RFC 2730*, December 1999.
- [HRC94] Hook (Daniel J. Van), Rak (Steven J.), et Calvin (James O.). – Approaches to relevance filtering. *In: Proceedings 11th DIS Workshop*.
- [HSC95] Holbrook (Hugh W.), Singhal (Sandeep K.) et Cheriton (David R.). – Log-based receiver-reliable multicast for distributed interactive simulation. *In: Proceedings ACM SIGCOMM'95*, pp. 328–341.
- [JMC99] J.M.Pullen, M.Myjak et C.Bouwens. – Limitations of internet protocol suite for distributed simulation in the large multicast environment. *RFC 2502*, February 1999.

- [KHTK99] Kasera (Sneha K.), Hjalmtysson (Gisli), Towsley (Don) et Kurose (Jim). – Scalable reliable multicast using multiple multicast channels. *Transaction on Networking IEEE/ACM 99*, 1999.
- [Kin93] Kingman (John). – *Poisson processes*. – Clarendon Press, 1993, *Oxford Studies in Probability, No3*.
- [KRT<sup>+</sup>98] Kumary (Satish), Radoslavov (Pavlin), Thaler (David), Alaettinoglu (Cengiz), Estrinz (Deborah) et Handley (Mark). – The masc/bgmp architecture for inter-domain multicast routing. *In: Proceedings ACM SIGCOMM'98*.
- [LGD98] Léty (Emmanuel), Gautier (Laurent) et Diot (Christophe). – Mimaze, a 3d multiplayer game on the internet. *In: Proceedings 4th International Conference on VSMM'98 (Virtual Systems and MultiMedia)*. – Gifu, Japan, November 1998.
- [LGLA97] Levine (B. N.) et Garcia-Luna-Aceves (J. J.). – Improving internet multicast with routing labels. *In: Proceedings ICNP'97*.
- [LHM] Lea (Rodger), Honda (Yasuaki) et Matsuda (Kouichi). – Virtual society: Collaboration in 3d spaces on the internet.
- [LHMM97] Lea (Rodger), Honda (Yasuaki), Matsuda (Kouchi) et Matsuda (Satoru). – Community place: Architecture and performance. *In: Proceedings VRML'97*.
- [LHnJR97] Lea (Rodger), Honda (Yasuaki) et nad Jun Rekimoto (Kouichi Matsuda). – Technical issues in the design of a scalable shared virtual world. 1997.
- [LN] Liu (Chuanhai) et Nonnenmacher (Jorg). – Broadcast audience estimation.
- [Loc95] Locke (John). – *An Introduction to the Internet Networking Environment and SIM-NET/DIS*. – Rapport technique, by John Locke (Internet: jxxl@cs.nps.navy.mil) Computer Science Department,, October 1995.
- [LR91] Lougheed (K.) et Rekhter (Y.). – A border gateway protocol 3 (bgp-3). *RFC 1267*, October 1991.
- [LT99] Léty (Emmanuel) et Turetletti (Thierry). – Issues in designing a communication architecture for large-scale virtual environments. *In: Proceedings the 1st International*

*Workshop on Networked Group Communication, Pise, Italie.*

- [LTB99] L ty (Emmanuel), Turletti (Thierry) et Baccelli (Francois). – *Cell-based Multicast Grouping in Large-Scale Virtual Environments*. – Rapport technique n RR-3729, INRIA, July 1999.
- [Mac95] Macedonia (Michael R.). – *A Network Software Architecture for Large-scale Virtual Environments*. – Ph.d. dissertation, June 1995.
- [Mil92] Mills (David L.). – Network time protocol (version 3). *RFC 1305*, March 1992.
- [MJV96] McCanne (Steven), Jacobson (Van) et Vetterli (Martin). – Receiver-driven layered multicast. *In: Proceedings ACM SIGCOMM'96, Stanford*.
- [ML] Meyer (D.) et Lothberg (P.). – Glop addressing in 233/8. *RFC-2770*.
- [Moy94] Moy (J.). – Multicast extensions to ospf. *RFC 1584*, March 1994.
- [MZ96] Macedonia (Michael R.) et Zyda (Michael J.). – A taxonomy for networked virtual environments. *IEEE Multimedia*, 1996.
- [MZP<sup>+</sup>94] Macedonia (Michael R.), Zyda (Michael J.), Pratt (David R.), Barham (Paul T.) et Zeswitz (S.). – Npsnet : A network software architecture for large scale virtual environments. *MIT Presence 3(4)*, 1994.
- [MZP<sup>+</sup>95] Macedonia (Michael R.), Zyda (Michael J.), Pratt (David R.), Brutzman (Donald P.) et Barham (Paul T.). – Exploiting reality with multicast groups: A network architecture for large-scale virtual environments. *IEEE Computer Graphics and Applications*, vol. 15, n5, September 1995, pp. 38–45.
- [NB97] Nonnenmacher (Jorg) et Biersack (Ernst). – Scalable feedback for satellite broadcasts. *In: ACM/IEEE MobiCom'97 Workshop on Satellite-based Information Services (WOSBIS'97)*, pp. 15–21.
- [PLB<sup>+</sup>98] Perlman (R), Lee (C-Y), Ballardie (A.), Crowcroft (J.), Wang (Z.) et Maufer (T.). – Simple multicast: A design for simple, low-overhead multicast. *Internet-Draft*, November 1998.

- [Pry99] Pryce (Nat). – Group management and quality of service adaptation in distributed virtual environments. 1999.
- [PSLt97] Paul (Sanjoy), Sabnani (Krishan K.), Lin (John C.) et tacharyya (Supratik Bhat). – Reliable multicast transport protocol(rmtp). *IEEE Journal on Selected Areas in Communications*, 1997.
- [Pus98] Pusateri (T.). – Distance vector multicast routing protocol. *Internet-Draft*, August 1998.
- [Ra95] Rekhter (Y.) et and (T. Li). – A border gateway protocol 4 (bgp-4). *RFC 1771*, March 1995.
- [RC98] Rak (Steven J.) et Christensen (Paul J.). – Battlefield communication network modeling. 1998.
- [RH96] Rak (Steven J.) et Hook (Daniel J. Van). – Evaluation of grid-based relevance filtering for multicast group assignment. *In: Proceedings 14th DIS workshop*.
- [Riz98] Rizzo (Luigi). – Fast group management in igmp. *Hipparc*, 1998.
- [RS99] Robinson (John) et Stewart (John). – Rtp payload format for shared multicast virtual worlds (smvw). *Internet-Draft*, 1999.
- [SCFJ98] Schulzrinne, Casner, Frederick et Jacobson. – Rtp: A transport protocol for real-time applications. *Internet-Draft*, November 1998.
- [SDW92] Strayer (W.T.), Dempsey (B.J.) et Weaver (A.C.). – *XTP: The Xpress Transfer Protocol*. – Reading, Mass, Addison-Wesley, 1992.
- [SFL<sup>+</sup>00] Speakman (Tony), Farinacci (Dino), Lin (Steven), Crowcroft (Jon), Leshchiner (Dan), Gemmell (Jim), Luby (Michael), Montgomery (Todd) et Rizzo (Luigi). – Pgm reliable transport protocol. – Internet-draft, April 2000.
- [Soc95] Society (IEEE Computer) (édité par). – *IEEE Standard for Interactive Distributed Simulation*. – 1995.
- [SSM97] Seidensticker (Steve), Smith (W. Garth) et Myjak (Michael). – Scenarios and appropriate protocols for distributed interactive simulation. *Internet-Draft*, 1997.

- [TEM98] Thaler (D.), Estrin (D.) et Meyer (D.). – Border gateway multicast protocol (bgmp): Protocol specification. *Internet-Draft*, November 1998.
- [VRM97] The virtual reality modeling language specification version 2.0. – ISO/IEC CD 14772, 1997.
- [WAB<sup>+</sup>96] Waters (R.), Anderson (D.), Barrus (J.), Brogan (D.), Casey (M.), McKeown (S.), Nitta (T.), Sterns (I.) et Yerazunis (W.). – *Diamond Park and Spline: A Social Virtual Reality System with 3D Animation, Spoken Interaction, and Runtime Modifiability*. – Rapport technique nTR-96-02a, November 1996.
- [WAS97] Waters (Richard C.), Anderson (David B.) et Schwenke (Derek L.). – *The Interactive Sharing Transfer Protocol Version 1.0*. – Rapport technique nTR 97-19, October 1997.
- [WH98] Wray (Mike) et Hawkes (Rycharde). – Distributed virtual environments and vrml: an event-based architecture. *In: Proceedings 7th International WWW Conference, Brisbane, Australia*.
- [WKM00] Wong (Tina), Katz (Randy) et McCanne (Steven). – An evaluation of preference clustering in large-scale multicast applications. *In: Proceedings IEEE INFOCOM'00*.
- [ZAD99] Zou (Li), Ammar (Mostafa H.) et Diot (Christophe). – An evaluation of grouping techniques for state dissemination in networked multi-user games. *In: ICMP'99*.