Mémoire

présenté par

Éric FLEURY

en vue de l'obtention du diplôme

d'habilitation à diriger des recherches de l'INSA de Lyon et de l'Université Claude Bernard – Lyon 1

(Numéro d'ordre HDR 2002 011)

COMMUNICATIONS DE GROUPE – DU PARALLÉLISME AU AD HOC –

Soutenue le 19/12/2002

Après avis de : Serge FDIDA
Olivier FESTOR
Philippe JACQUET

Devant la commission d'examen formée de :

Serge FDIDA (Rapporteur)
Professeur à l'Université Pierre et Marie Curie – Paris VI
Olivier FESTOR (Rapporteur)
Chargé de recherche à l'INRIA Lorraine
Andrzej DUDA (Examinateur)
Professeur à l'ENSIMAG
Philippe JACQUET (Rapporteur)
Directeur de recherche à l'INRIA Rocquencourt
Yves ROBERT (Examinateur)
Professeur à l'Université Claude Bernard – Lyon 1
Stéphane UBÉDA (Examinateur)
Professeur à l'INSA de lyon

pour les travaux effectués au Laboratoire de l'Informatique du Parallélisme de l'ENS-Lyon, au Software Engineering and Network Systems Laboratory de Michigan State University, au Laboratoire lorrain de recherche en informatique et ses applications et au Centre d'Innovations en Télécommunications & Intégration de services de l'INSA de Lyon.

Aux longues siestes méridionales de Lorik et à la présence généreuse et compréhensive de Pitchoune.

Remerciements

Je tiens, en premier, à exprimer ma reconnaissance aux membres du jury : Serge FDIDA, Olivier FESTOR, Andrzej DUDA, Philippe JACQUET, Yves ROBERT et Stéphane UBÉDA.

Les travaux réunis dans ce manuscrit couvrent six années et sont avant tout le fruit d'un travail d'équipe. Je suis fier d'avoir travaillé avec des « *étudiants-chercheurs* ». Merci à Guillaume et Hend de m'avoir supporté, d'accepter mon rangement stratifié auto-organisé intrusif et d'une certaine façon de m'avoir aussi fait confiance. Si ces travaux on pu être menés, si des étudiants ont pu être formés « *à la recherche par la recherche* » c'est grâce au soutien d'un certain nombre d'institutions et de laboratoires. Je remercie donc le LIP de l'ENS-Lyon et j'en profite pour remercier les différentes personnalités hautes en couleur de ce laboratoire qui m'ont fait découvrir le métier passionnant de la recherche[1]. Merci au Professeur Philip MCKINLEY pour son accueil au Communications Research Group (CRG) à Michigan State University. J'associe le LORIA et le projet RESEDAS qui m'ont accueilli en postdoc et en tant que chargé de recherche INRIA. Je tiens à remercier André SCHAFF pour la confiance dont il a fait preuve. Je voudrais aussi rendre hommage à Michel COSNARD[2] pour son amitié, ses conseils et son soutien. Merci au CITI pour m'avoir accueilli à l'INSA de Lyon et croire dans la collaboration INRIA / INSA !

Je voudrais aussi remercier les différents acteurs des communautés TAROT et RHDM qui sont propices aux échanges scientifiques fructueux. Il m'est impossible d'en dresser la liste exhaustive. Je vais certainement en oublier, qu'ils ne m'en veuillent pas : Jean-Claude BERMOND, Christian BONNET, Luc BOUGÉ, Michel DIAZ, Bertrand DUCOURTHIAL, Afonso FERREIRA, Pierre FRAIGNIAUD, Cyril GAVOILLE, Jean-Claude KÖNIG, Michel MORVAN, Thomas NOËL, Jean-Jacques PANSIOT, Guy PUJOLLE, Patrick SENAC, Ahmed SERHROUCHNI, Laurent TOUTAIN, Véronique VÈQUE, Laurent VIENNOT, Sandrine VIAL...

Je n'oublierais pas les chercheurs d'ARES, équipe des plus conviviales, Par ordre alphabétique car le plus « *political correct* »[3] : Stéphane FRÉNOT (dit `root`), Jean-Marie GORCE (wave master), Isabelle GUERIN-LASSOUS (je demande un avocat avant de mettre sous presse le moindre commentaire), Véronique LEGRAND (spécialiste es sécurité), Farid NAÏT-ABDESSELAM (un pro windows ou un anti Linux ?), Stéphane Ubéda (le directeur... joueur et ami), Fabrice VALOIS (la rigueur et le sérieux markovien. Remarque gratuite car je n ai jamais mangé avec Markov).

Finalement, je tiens à remercier tous les amis qui ont su donner à ces six années un parfum de convivialité, à savoir Virginie, Frédéric, David, Thierry, Olivier et les habitués des coins café...

[1] Engagez-vous qu'ils disaient.

[2] mon ex-directeur au Loria, mon ex-directeur au LIP, et mon ex-co-directeur de thèse...

[3] s'offrait l'ordre par âge, géographique...

Table des matières

Chapitre 1

Introduction

Un mystère, c’est la plus profonde chose qu’il y ait pour l’imagination humaine.

Jules Amédée BARBEY D’AUREVILLY

1.1 Parcours

Ce document présente un ensemble de travaux de recherche que j'ai mené depuis l'obtention de mon doctorat en 1996. J'ai rassemblé mes diverses activités autour des *communications de groupe* pour obtenir une certaine cohérence. J'ai délibérément choisi de ne pas présenter mes travaux sur l'algorithmique parallèle ou sur les problématiques liées au métacomputing. La présentation de ces travaux n'aurait abondé ni dans le sens d'une plus grande cohérence ni vers une plus grande clarté du document. Ces activités se sont déroulées au **Laboratoire de l'Informatique du Parallélisme de l'ENS-Lyon** où j'ai été attaché temporaire d'enseignement et de recherche, dans le **Software Engineering and Network Systems (SENS) Laboratory de Michigan State University** où j'ai effectué un postdoctorat, au **Laboratoire lorrain de recherche en informatique et ses applications à Nancy** où j'ai été recruté après un an de postdoctorat au sein du projet RÉSÉDAS[1] en tant que chargé de recherche INRIA et au **Centre d'Innovations en Télécommunications & Intégration de services à l'INSA de Lyon** où j'exerce mes activités de chargé de recherche INRIA au sein de l'équipe ARES[2].

1.2 Contexte de recherche

Mes travaux ont été, bien sûr, très influencés par les chercheurs que j'ai côtoyés depuis ma thèse et par l'évolution des domaines de recherche. Dans cette section, je positionne mes travaux dans cet environnement de travail.

Ma première thématique de recherche était très liée au calcul parallèle comme solution possible aux besoins sans cesse croissants de puissance de calcul. Le leitmotiv était durant ces années la « quête du Tera Flops ». Mes recherches dans ce domaine traitent des communications entre les entités constituant ces machines parallèles. Un processeur peut traiter de manière indépendante les données stockées localement. Malheureusement, les données se trouvant sur un autre nœud ne sont accessibles que par envois de messages au travers du réseau. Les communications apparaissent ainsi comme un point crucial dans le calcul parallèle/distribué : le fait de vouloir résoudre un problème en utilisant un grand nombre de processeurs/nœuds implique inévitablement la distribution des données, l'échange de résultats intermédiaires ou la diffusion de solutions de sous-problèmes. Toutes ces opérations nécessitent des communications qui sont autant de temps gaspillé car les communications contribuent à accroître le temps d'exécution total.

Afin de développer des programmes performants et portables, la notion de bibliothèque de communication est apparue. Ces bibliothèques de communication [Des01, GBD+94, SOHL+96, EsFG00] offrent à l'utilisateur une interface simple pour effectuer des schémas de communication classiques et communs à diverses applications tout en laissant à leur développeur une certaine liberté quant au choix de l'algorithme permettant de les résoudre. Les bibliothèques permettent aussi de cacher à l'utilisateur les caractéristiques de sa machine cible et de laisser au développeur de bibliothèques la lourde tâche d'utiliser au mieux les interactions entre le matériel et le logiciel.

Dans ce contexte, j'ai étudié les problèmes de diffusion partielle (multicast), opération de communication globale qui consiste en l'envoi, par un processeur/nœud source, d'un message à

[1] Projet CNRS, INRIA, Université Nancy 1, Université Nancy 2, INPL

[2] équipe INSA de Lyon – INRIA

un sous-ensemble des processeurs/nœuds du réseau. Cette opération de communication globale intervient dans de nombreuses applications mais aussi dans la mise en œuvre d'autres opérations usuelles de communication globale : une opération de diffusion au niveau applicatif peut se révéler être un schéma de diffusion partielle au niveau matériel dans le cas où seulement un sous-ensemble des processeurs de la machine est alloué à l'utilisateur effectuant la diffusion.

Après l'engouement énorme pour la mise au point des machines de taille toujours plus importante (jusqu'à 9000 Pentiums pour la machine du programme américain ASCI), on tente de nos jours d'agréger la puissance de nombreuses machines de taille moins importante pour former des grappes/clusters/metacomputers. Ces grappes peuvent être reliées par un réseau dédié à très haut débit, par un réseau classique local ou être éparpillées dans le monde et reliées par l'Internet. Le concept de métacomputing évolue en permanence et commence à pénétrer l'opinion publique puisque même le journal *Le Monde* en date du 5 septembre 2000 titrait « *Des bataillons de PC à l'assaut des super-calculateurs* ». L'utilisation de logiciels distribués est de plus en plus courante mais de nombreux problèmes de recherche persistent. Nous sommes encore loin de pouvoir offrir des infrastructures logicielles distribuées permettant l'interconnexion d'un grand nombre de composants et ce, dans un souci de transparence totale pour l'utilisateur qui ne doit pas avoir à se soucier de l'endroit où son application s'exécute réellement.

Dans ce contexte de calcul totalement distribué sur l'Internet, le problème initial des communications efficaces reste un point clé si l'on veut pouvoir mettre en œuvre des applications distribuées performantes : le temps nécessaire pour communiquer entre deux ou plusieurs entités apparaît toujours comme un gaspillage de temps et de ressources. C'est notamment dans ce contexte plus orienté réseau que j'ai fait évoluer ma thématique de recherche tout en gardant en trame de fond l'étude des communications multicast. Cette évolution de l'Internet couplée au développement de nouveaux paradigmes et aux nouvelles façons d'employer ces canaux de communication a permis d'envisager de nouvelles applications et de développer les supports réseaux nécessaires à ces dernières. Le support des communications multicast, *i.e.*, la possibilité d'avoir un échange de flux d'information entre plus de deux intervenants, demeure l'un des sujets toujours très actifs au sein de la communauté et représente un support indispensable à beaucoup d'applications.

Dans les années 60, les communications téléphoniques étaient uniquement transportées sur des fils de cuivre partant d'un concentrateur général jusqu'à l'abonné final (entreprise ou particulier) alors que la télévision était, elle, uniquement diffusée par voie hertzienne. À la fin des années 80, cette situation s'est inversée et les communications téléphoniques étaient de plus en plus transportées par voie hertzienne tandis que la télévision arrivait de plus en plus chez l'abonné par câble (cuivre coaxial ou fibre optique). Aujourd'hui, la situation est encore plus complexe. Les réseaux de téléphones cellulaires sont toujours en pleine croissance mais il existe dans le même temps diverses propositions pour mettre en œuvre des réseaux de transport haut débit sans fil (expérience des réseaux satellitaires en orbite basse), des réseaux mobiles embarqués dans les avions. Certaines compagnies [BVGLA99, Nok01] envisagent aussi de déployer des systèmes de diffusion haut débit sans fil capables de concurrencer les accès filaires sur paire de cuivre comme l'ADSL. Ces réseaux sans fil permettent de couvrir toute une zone sans se soucier a priori du degré des équipements (contrairement à un concentrateur ADSL où le nombre de ports est borné). Il est alors possible de déployer un réseau maillé sans fil avec un coût fixe par équipement tout en ayant la possibilité de l'étendre facilement en fonction de la densité de population. C'est dans

ce contexte du *tout mobile tout IP* que s'inscrivent mes recherches plus récentes. Ces dernières portent sur la définition et la mise en œuvre d'une architecture adaptée aux réseaux ad hoc, à son extension et son intégration dans IPv6. Dans la perspective des réseaux de 4ème génération (en passe un jour de devenir réalité, paraît-il [AA02, AAPV02]) qui se veulent un système universel fonctionnant avec divers standards de transmission et offrant de nombreux « *nouveaux* » services, je présente mes travaux sur les protocoles de découverte de services dans les réseaux ad hoc, petite pierre à l'édifice de la configuration transparente des mobiles au sein de leur environnement ambiant.

1.3 Organisation du document

Ce document a été organisé en utilisant les divers projets de recherche auxquels je me suis intéressé et contient donc trois chapitres principaux.

Dans le premier chapitre, je présente les communications de groupe vue au travers du prisme de la communauté « calcul parallèle » et les divers impacts que ces recherches ont pu avoir, notamment sur le développement de bibliothèques de communication, sorte de « standard de programmation ». Dans le deuxième chapitre, je présente mes travaux autour du déploiement d'arbres multicast dans les réseaux IP. Ce projet, initié durant mon séjour postdoctoral à Michigan State University, avait pour but de mettre en œuvre des mesures efficaces pour pouvoir comparer et évaluer les arbres multicast déployés au sein d'un AS afin d'être en mesure de le redéployer de façon dynamique. Dans le troisième chapitre, je présente mes travaux récents, concernant les réseaux ad hoc. Pour finir, le chapitre 4 conclut et donne des perspectives à mes derniers travaux de recherche. Chaque chapitre comporte également la liste des publications issues des contributions présentées.

Une annexe comporte des articles pour chaque domaine de contributions ainsi qu'un curriculum-vitæ étendu (activités scientifiques, développements logiciels, travaux d'encadrement de jeunes chercheurs, tâches collectives, liste de mes publications).

Bibliographie

[AA02] K. Al Agha, *Évoluer vers la 4e génération*, Habilitation à diriger des recherches, Université de Paris Sud-XI, paris, France, Octobre 2002.

[AAPV02] K. Al Agha, G. Pujolle, and G. Vivier, *Réseaux de mobiles et réseaux sans fil*, Eyrolles, 2002, ISBN : 2-212-11018-9.

[BVGLA99] D. Beyer, M. Vestrich, and J. Garcia-Luna-Aceves, *The first 100 feet : Options for internet and broadband access*, ch. Rooftop Community Network : Free, High-Speed Network Access for Communities, The MIT Press, Cambridge, Massachusetts, 1999, ISBN 0-262-58160-4.

[Des01] F. Desprez, *Contribution à l'algorithmique parallèle – calcul numérique : des bibliothèques aux environnements de métacomputing –*, Habilitation à diriger des recherches, Université Claude Bernard Lyon 1, Lyon, France, Juillet 2001.

[EsFG00] T. Es-squalli, E. Fleury, and J. Guyard, *MPC : a new Message Passing library in Corba*, International Workshop on Metacomputing Systems and Applications (MSA) (Toronto, Canada), IEEE, August 2000.

[GBD+94] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Mancheck, and V. Sunderam, *PVM Parallel Virtual Machine. a users' guide and tutorial for networked parallel computing*, Scientific and Engineering Computation Series, MIT Press, 1994.

[Nok01] *Wireless broadband for residential markets*, `http://www.nwr.nokia.com/`, 2001, White Paper.

[SOHL+96] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI : The complete reference*, MIT Press, 1996.

Chapitre 2

Multicast dans un monde clos : cas des machines à mémoire distribuée

Aujourd'hui on peut faire de la musique avec des ordinateurs, mais l'ordinateur a toujours existé dans la tête des compositeurs...

Milan KUNDERA

2.1 Introduction

Au début des années 90, les réseaux de processeurs étaient carrés, du moins planaires[1] et en forme de grille. Les plus imaginatifs ou ceux ayant des longueurs de câble inutilisées avaient réalisé des grilles toriques et les plus audacieux des hypercubes. La recherche explorait la théorie des graphes qui devenait une des clefs maîtresses des réseaux d'interconnexion des machines massivement parallèles et étudiait les propriétés fondamentales des graphes de DEBRUIJN [BP89] à poil ras ou long, selon l'école à laquelle on avait prêté serment, des *star graph* [BFP96, MJ94] (l'étoile filante prometteuse du moment) et des graphes de CAYLEY [LJD93].

Plus sérieusement, la majorité des travaux que l'on peut qualifier de « théoriques »[2] portent sur la recherche de modèles de communication adaptés aux différentes architectures multi-processeurs afin de pouvoir dériver des bornes sur les temps de communication d'un certain nombre d'opérations classiques comme la diffusion, la multi-diffusion, l'échange total, la multi-distribution. Chaque modèle apporte ses propres contraintes ou variations [Fle96, FL94, HHL86] quant à la possibilité de communiquer sur tous les liens de sortie, de recevoir et/ou d'envoyer en même temps. Pour un modèle donné, certaines classes de graphes se montrent beaucoup plus appropriées que d'autres et permettent d'atteindre les bornes inférieures pour le nombre d'étapes de communication requises ou pour l'espace mémoire nécessaire pour router de façon optimale (typiquement un $\log$ du nombre de processeur [Gav96, Gav00] !).

Mes travaux de thèse s'inscrivaient dans cette thématique générale. Je vais brièvement revenir sur les problèmes d'interblocage qui se posent dans les architectures utilisant un mode de communication *wormhole* afin de donner un aperçu des domaines actuels d'utilisation et d'application de mes travaux. Je profite de cette section pour présenter une conjecture laissée ouverte en fin de thèse sur la possibilité de trouver un algorithme optimal (en temps ou en ressource) pour effectuer une opération de multicast dans une grille en utilisant un mode de communication nommé *path-based*. Comme on le voit, on retrouve la grille ainsi qu'un modèle particulier de communication.

2.2 Routage wormhole et problèmes d'interblocage

Depuis les travaux fondateurs de KERMANI et KLEINROCK sur le « *virtual cut-through* » [KK79] et un peu plus tard ceux de DALLY et SEITZ sur le « *wormhole switching* » [DS86, DS87], il y a eu un nombre de travaux considérable sur les techniques de routage. À l'heure actuelle, on peut distinguer un certain nombre de paradigmes prédominants employés au sein des systèmes d'interconnexion des machines multi-processeurs [DYN02] dont font partie les modes de commutation de type wormhole [DRS01].

2.2.1 Routage wormhole

Le mode de communication wormhole reprend la notion de cut-throught mais supprime la notion de stockage intermédiaire lorsque le canal que doit emprunter le message est occupé. Plus

[1] la terre devait encore être plate

[2] sans préjugé péjoratif d'aucune sorte sur cet adjectif

précisément, dans le mode de communication wormhole, on considère qu'un message est composé de flits et que seul un petit nombre de flits peut être stocké dans chaque nœud.

Définition 2.1 Mode de communication wormhole [DS86, NM93] Un flit (pour *flow control digit*) est la plus petite unité d'information sur laquelle le contrôle de flux est exécuté, c'est-à-dire la plus petite unité d'information que peut accepter ou refuser la file d'attente d'un canal (une file d'attente peut néanmoins contenir un nombre de flits supérieur à 1).

L'en-tête du message (*i.e.*, le flit de tête) contient la destination. Les messages progressent flit par flit dans le réseau. Dès que l'en-tête d'un message arrive dans un routeur, ce dernier le traite et détermine le canal de sortie que doit emprunter le message. Si ce canal est disponible (en accord avec le contrôle de flux), le flit d'en-tête est retransmis directement sur ce dernier, le reste des flits le suivant à la « queue leu leu » (voir la figure 2.1).

Notons qu'il est possible que l'en-tête arrive à destination avant que la totalité du message n'ait été émise par le processeur source. À l'inverse, si le message est suffisamment court, il est possible que la source soit libérée avant que l'en-tête ne soit reçu par le processeur destinataire. La plupart des flits ne contenant aucune information de routage, les flits d'un même message doivent impérativement être sur des canaux contigus dans le réseau (*i.e.*, on ne doit pas « couper » le flot des flits) et les flits de deux messages ne doivent pas s'entremêler.

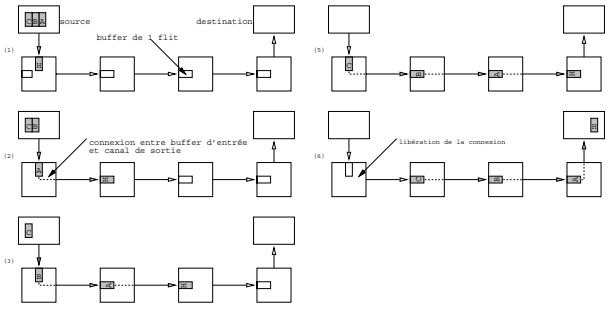


FIG. 2.1 – Mode de commutation wormhole.

Définition 2.2 Un message est l'unité logique d'une communication. C'est la seule entité visible pour la couche service du réseau. Un message peut être découpé en un ou plusieurs paquets. Un paquet est la plus petite unité d'information contenant l'information nécessaire au routage –par exemple, l'adresse de la destination. Cette information est codée dans l'en-tête du paquet qui est composé du ou des flits de tête. Si le message est découpé en plusieurs paquets, un paquet peut devoir contenir l'information nécessaire à la reconstitution du message. Un paquet est lui-même formé de plusieurs flits.

On peut reprendre l'analogie développée par DALLY dans [SB90] pour illustrer la différence entre un paquet et un flit. Les paquets sont comparables à des automobiles. Puisque chaque automobile sait, a priori, où elle va. Elles peuvent être entremêlées les unes aux autres sans que cela ne porte réellement à conséquence. En revanche, un flit est lui comparable à un wagon de

train où seul la locomotive connaît la destination (*i.e.*, le flit de tête), les autres wagons devant impérativement suivre le premier sans être entremêlés avec d'autres wagons d'un autre train.

Une fois qu'un canal a été réservé par l'en-tête d'un paquet, il reste réservé pour la totalité du paquet. Le canal est libéré quand le dernier flit du paquet (fin de paquet) l'a emprunté. Quand l'en-tête d'un paquet est bloqué lorsque le canal de sortie qu'il veut emprunter n'est pas disponible, il reste bloqué jusqu'à ce que le canal se libère. Les autres flits du paquet cessent d'avancer et, par là même, interdisent à tout autre paquet d'emprunter les canaux qu'ils occupent.

En supposant qu'il n'y ait pas de conflit au sein du réseau, le temps de latence d'une communication d'un message de taille L dans le mode commutation wormhole entre deux sommets x et y est modélisé par :

$$T_{x \to y}(L) \quad = \quad \alpha + d(x,y)\delta + L\tau \tag{2.1}$$

où α correspond au temps requis pour la préparation d'un message et pour les recopies mémoire/buffer, τ est l'inverse de la bande passante des liens et δ prend en compte le temps de commutation d'un routeur et le temps mis par un flit pour être transmis sur un canal. On peut décomposer δ en $\delta' + L_f\tau$ où L_f est la taille d'un flit. Si $L_f << L$ et $d(x,y)\delta' << L\tau$, l'effet de la distance $d(x,y)$ sur la latence du réseau devient négligeable. La principale différence entre le mode commutation de paquets et le mode wormhole est la suivante : dans le mode commutation de paquets, le contrôle de flux est effectué au niveau des paquets alors que dans le mode de commutation wormhole, on effectue le contrôle de flux sur une unité beaucoup plus petite, le flit.

L'intérêt du mode de commutation wormhole est donc d'éviter la présence et la gestion complexe de buffers de taille importante dans chaque nœud. La façon dont les paquets acquièrent des canaux dans le mode wormhole met en évidence un deuxième avantage de ce mode par rapport au mode commutation de circuits : le partage des canaux. En effet, dans le mode commutation de circuits, une fois qu'un canal est réservé pour un paquet, il ne peut en aucun cas être utilisé par un autre paquet avant d'avoir été libéré. Or, cela n'est plus forcément vrai en mode wormhole. Le mode wormhole permet à un canal d'être partagé par plusieurs paquets par la mise en œuvre de multiplexage, ce qui équivaut à introduire la notion de canaux virtuels.

2.2.2 Fonction de routage et deadlock

Le mode de commutation wormhole est sujet aux interblocages car les messages sont autorisés à détenir un grand nombre de ressources tout en voulant en requérir d'autres (voir figure 2.2). Il est donc crucial de pouvoir définir des fonctions de routage sans interblocage.

L'approche classique pour éviter les interblocages est bien évidemment de restreindre la fonction de routage de telle sorte que l'apparition de cycles dans le graphe de dépendance des ressources (ici les canaux) soit impossible. Une approche plus efficace, car donnant un degré de liberté plus grand à la fonction de routage, consiste à autoriser la présence de cycles entre certaines des ressources en garantissant qu'il existe toujours une échappatoire qui, elle, est exempte d'interblocage. Une dernière approche consiste à autoriser la présence de deadlock et à retirer du réseau les messages prisonniers [AP95, KLC94, MRLD01], ce qui peut engendrer des chutes de performance surtout quand on atteint le point de saturation du réseau où la probabilité d'interblocage devient alors très importante.

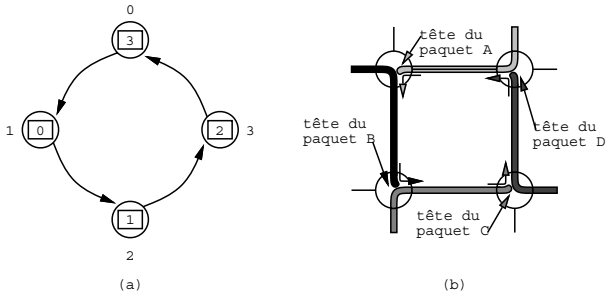


FIG. 2.2 – Situation de deadlock.

Beaucoup de travaux se sont intéressés à ce type de problème et le but n'est pas ici d'en faire la liste exhaustive. Bien que tous traitent du routage wormhole, ils considèrent soit des réseaux spécifiques, soit ils utilisent une notion de fonction de routage différente. Il m'a semblé important de formuler une théorie qui prend en compte une notion de fonction de routage beaucoup plus générale. Cette notion de fonction de routage se définit uniquement à partir de l'en-tête h du message et de l'état γ local du routeur. Cette notion de fonction de routage permet notamment de couvrir les fonctions déterministes, les fonctions adaptatives, les fonctions permettant de faire du multicast...

À partir de cette théorie, nous avons montré une condition nécessaire et suffisante pour qu'une fonction de routage soit sans interblocage [FF98a]. Le but n'est pas ici de présenter ces travaux qui sont pour la majeure partie couverts par ma thèse [Fle96] mais de voir l'importance de cette notion de fonction de routage dans le contexte actuel.

2.3 Travaux liés

La règle en matière d'évolution des architectures des machines multi-processeurs est sans doute que les performances des processeurs progressent chaque année. Une station de travail actuelle atteint sans complexe les performances des super-calculateurs des années 80 ! Si le massivement parallèle à la portée de tous avec les machines SIMD a vite atteint ses limites, les processeurs actuels recèlent des unités fonctionnelles multiples et donc un parallélisme interne au processeur, des pipelines super-scalaires, des hiérarchies mémoire importantes et des tailles de caches internes dignes des mémoires des stations de travail du passé.

L'évolution actuelle des architectures est très orientée « *cluster de SMPs* ». Cette tendance est renforcée par la disparition totale des processeurs spécialisés développés par les constructeurs au profit de processeurs « *on the shelf* » utilisés plutôt dans les PCs puissants ou les stations de travail que dans les machines parallèles. Les dernières versions de ces cartes peuvent abriter jusqu'à 8 processeurs pour donner des puissances jusqu'au Gigaflops.

Ces stations de travail puissantes interconnectées par des commutateurs *off the shelf* construits sur des technologies de commutation, ont créé une brèche dans le marché des multi-processeurs. Le lecteur peut se référer à l'habilitation à diriger les recherches de Frédéric DESPREZ [Des01]

pour une vision plus approfondie des évolutions des architectures[3]. Le grand avantage de ces réseaux de PCs (*NOW* pour *Network of Workstations*) est leur flexibilité en terme de possibilités d'interconnexion et un coût généralement beaucoup moins élevé que les multi-processeurs propriétaires. Ainsi, l'arrivée sur le marché de nombreux commutateurs (*switch*) à bas prix ayant des caractéristiques très impressionnantes font que les NOWs sont devenus très populaires [BCF+95, CV96, Gal97, GW97, Hor95, Inf02, NKN+01, NKN+00, SBB+90, She98].

Le fait que l'on puisse interconnecter des commutateurs entre eux comme on le désire, change radicalement des machines multi-processeurs dont la topologie est pré-définie, figée et peu extensible[4]. L'utilisateur peut créer sa propre topologie et la faire évoluer au gré du temps. Cela implique que les fonctions de routage ne peuvent plus être pré-câblées dans les commutateurs mais doivent pouvoir être configurées et surtout supporter des topologies non régulières dont le degré n'est borné que par le nombre de ports des switches (par opposition aux grilles, aux hypercubes). On peut faire le parallèle entre ces clusters de PCs et un LAN en gardant à l'esprit que les topologies des clusters sont souvent plus complexes car elles contiennent de nombreux cycles entres les switches. Les trois facteurs importants [SB90] dans la mise en place d'un réseau d'interconnexion sont :

- la topologie à proprement parler. On retrouve ici les caractéristiques fondamentales des différentes classes de graphes et les études plus théoriques ;
- le type de commutation mis en œuvre. Le mode de commutation wormhole s'est imposé à l'heure actuelle ;
- la fonction de routage employée.

Les topologies des NOWs sont donc irrégulières et basées sur un mode de commutation de type wormhole (Myrinet [BCF+95] ou Servnet II [GW97]). Il est donc important de pouvoir garantir des fonctions de routage sans interblocage pour une classe de graphes très générale. De plus il est préférable de pouvoir mettre en œuvre une fonction de routage adaptative [VSD01] afin de bénéficier de leurs nombreux avantages par rapport à une fonction purement déterministe [CP01, SD00a].

2.3.1 Fonction de routage pour NOWs

Parmi les différentes fonctions de routage qui ont été proposées (*up*/down** [SBB+90], *adaptive trail* [QN96], *minimal adaptive routing* [SD00b], *smart-routing* [CKR95]) pour les réseaux de stations inter-connectées par des commutateurs *off the shelf* construits sur des technologies de commutation, la fonction de routage *up*/down** proposée au départ pour Autonet reste la plus employée et c'est elle qui est notamment utilisée par Myrinet.

Le principe de la fonction de routage *up*/down** est très simple et repose sur l'orientation des liens de communication. Cette orientation se fait au moyen d'un arbre couvrant construit par un parcours en largeur d'abord (*Bread-First Search (BFS)*). L'orientation des liens est telle que tous les sommets peuvent atteindre la racine de l'arbre couvrant en n'empruntant que des liens *up*, et réciproquement, la racine est en mesure de joindre tous les autres sommets en n'empruntant que

[3] le privilège de l'âge, te donne, Ô grand sage, une plus grande expérience ! Pour preuve, j'ai repris ton style LaTeX. Merci à toi Fred.

[4] on peut difficilement rajouter 2 ou 3 nœuds à une grille ou à un hypercube

des liens *down*. La racine est le sommet ayant le plus petit ID. L'extrémité *up* de chaque lien est définie comme étant :

1. celle qui est la plus proche de la racine dans l'arbre couvrant ;
2. celle qui est attachée au switch ayant le plus petit ID si les deux extrémités d'un lien sont attachées à des switches équidistants de la racine.

On montre facilement que tout cycle du réseau possède au moins un lien *up* et lien *down*. Pour éviter les interblocages tout en permettant d'emprunter tous les liens, on définit un chemin comme étant valide s'il est de la forme : $\{up\}^*\{down\}^*$, *i.e.*, constitué de zéro ou plus liens *up* suivi de zéro ou plus liens *down* (Voir la figure 2.3). Cette règle empêche l'existence de tout cycle dans le graphe de dépendance puisque un paquet n'est jamais en mesure de traverser un lien de type *up* après avoir emprunté un lien de type *down*.

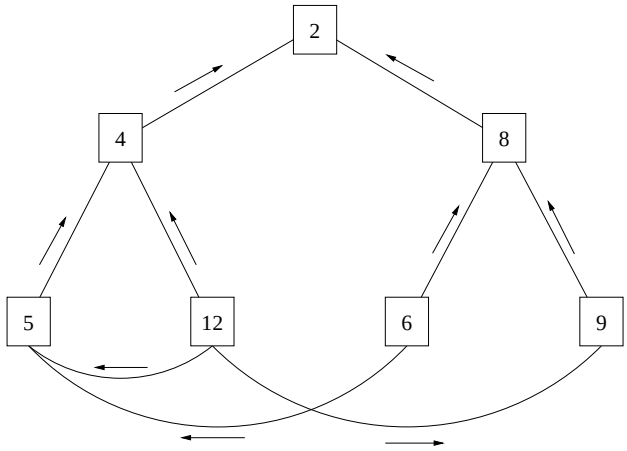


FIG. 2.3 – Fonction de routage *up*/down**.

Cette fonction de routage ne garantit évidemment pas un plus court chemin entre toute paire de sommets du graphe (sur la figure 2.3, le chemin entre le sommet 4 et le sommet 9 n'est pas un plus court chemin). L'autre inconvénient des méthodes basées sur la construction d'un arbre est qu'elles entraînent un déséquilibre de la charge au sein du réseau. En effet, plus le réseau est grand, plus il y a de chance que l'on tombe sur une interdiction de type $down \rightarrow up$, ce qui va concentrer le trafic dans le voisinage de la racine. Diverses propositions ont été faites pour améliorer la fonction de routage *up*/down**.

Une première idée est de « *tricher* » et d'autoriser les transitions $down \rightarrow up$ en divisant le chemin non valide en plusieurs sous chemins $\{up\}^*\{down\}^*$ valides. En d'autres termes, les dépendances $down \rightarrow up$ sont supprimées en stockant le message dans les nœuds intermédiaires (*in-transit buffer (ITB)*) [FLMD02].

Une autre classe d'amélioration consiste à essayer de générer une orientation du réseau qui soit meilleure que celle obtenue par un BFS. Dans [ABC$^+$00], les auteurs montrent que pour de grands graphes, il est très souvent possible de trouver une meilleure orientation du réseau à partir d'algorithmes génétiques. Dans [SRD00], les auteurs proposent une autre méthode d'orientation

qui se fonde sur la construction d'un graphe acyclique au moyen d'un parcours en profondeur d'abord (*Depth First Search (DFS)*), puis sur l'ajout des liens restants mais en renumérotant les sommets du graphe permettant ainsi de casser les cycles possibles. La fonction de routage est similaire à *up*/down** mais se base sur une construction différente du graphe acyclique, ce qui permet une plus grande flexibilité dans la construction de ce graphe acyclique.

Une dernière classe de solutions introduit la notion classique de canaux virtuels, ce qui permet, soit de retrouver des réseaux virtuels disjoints, soit de garantir qu'il existe toujours une échappatoire. Dans [LS01] les auteurs proposent de mettre en œuvre plusieurs racines permettant ainsi de répartir la charge du trafic et de gagner ainsi en débit. La garantie de non interblocage est assurée par l'introduction de plusieurs canaux virtuels (un par racine). Dans [SLT02], les auteurs proposent un routage déterministe qui utilise des plus courts chemins mais la borne supérieure sur le nombre de canaux virtuels est en $\lceil N/2 \rceil$, où N est le nombre de switches de la topologie irrégulière.

D'autres travaux, toujours dans la même lignée, se sont intéressés à l'impact de la reconfiguration de ce type de réseau, tout en préservant la propriété de non interblocage [LD00]. Certains travaux s'intéressent plus à la façon de mettre en place une nouvelle fonction de routage [CBD+01] alors que d'autres approches tentent de réduire l'impact de cette reconfiguration sur la structure déjà existante.

Tout ces travaux reprennent dans les grandes lignes les principes fondamentaux employés dans tous les travaux traitant du routage wormhole et des propriétés de non interblocage. En effet, dans toutes les théories proposées pour les réseaux irréguliers, on retrouve les mêmes notions (graphe de dépendance, hiérarchie de graphes virtuels, suppression des cycles). Pour clôturer ce petit survol, citons l'article de J. DUATO et de T. PINKSTON [DP02] qui proposent une théorie permettant de traiter à la fois le routage wormhole et le routage cut-through (qui nécessite la présence de buffers de stockage (partagé ou non) au sein des routeurs). Bien que ce travail permette d'unifier les diverses théories, il est restreint, du fait de sa définition, à un seul type de fonction de routage et il serait peut-être intéressant d'étendre encore plus cette théorie pour reprendre la notion de fonction de routage très générale [FF98a] que nous avions proposée.

L'apparition d'une technologie prometteuse nommée InfiniBand [Inf02] ouvre un nouveau domaine d'application à toutes ces théories développées initialement pour les réseaux d'interconnexion des machines massivement parallèles. Cette technologie a pour but de devenir le nouveau standard qui doit s'imposer pour tout ce qui est communication entre processeurs et devices d'entrée/sortie (*System Area Network (SANs)*) mais elle permet aussi d'interconnecter des processeurs entre eux pour former des topologies complexes et irrégulières (NOWs). Pour ce, l'architecture InfiniBand (IBA) définit des switches qui permettent de relier des équipements par le biais de liens point-à-point (voir figure 2.4). Il est bien évident que l'on peut de nouveau proposer des fonctions de routage sans interblocage en prenant en compte les spécificités de cette nouvelle technologie [FLS+02]. Dans InfiniBand, les switches n'effectuent leur décision de routage que sur le switch courant et sur la destination du paquet. Une fois encore, la généralisation apportée par notre théorie peut s'avérer utile.

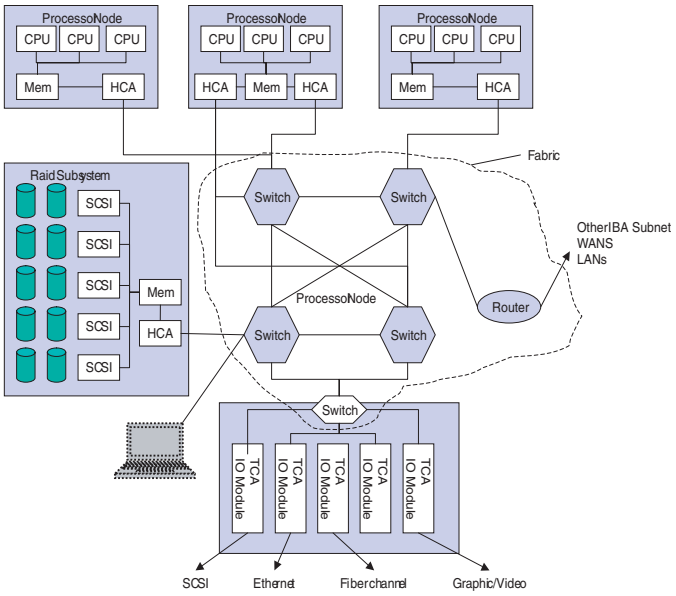


FIG. 2.4 – Réseau d'interconnexion dans l'architecture IBA (schéma emprunté à [Inf02]).

2.4 *Path-based* multicast sans interblocage dans la grille

Nous restons dans le cadre des machines à mémoire distribuée employant un mode de communication de type wormhole (voir définition 2.1). Une opération de multicast peut soit être effectuée en ne se basant que sur des opérations point-à-point (*unicast* [MXEN94]), soit en employant des fonctionnalités matérielles spécifiques [SKPS98] mises en œuvre au sein des routeurs (ou des interfaces [BPDS00]) qui permettent de répliquer les messages à la volée (en wormhole on ne bufferise pas le message avant de le réémettre comme dans le mode *Store-and-Forward*). Deux types de mécanismes ont été proposés : la réplication de message et la réception à la volée.

La réplication de message permet à un routeur de répliquer, à la volée, un message flit par flit, *i.e.*, de les envoyer sur différents canaux de communication en même temps. Ce mode est comparable à la mise en œuvre matérielle d'un arbre de broadcast ou de multicast. Son principal inconvénient est que dès qu'une branche de l'arbre est stoppée, c'est tout le message qui se trouve bloqué (un message est une sorte de pipeline de flits). Des dépendances pouvant intervenir entre les branches constituent une source supplémentaire d'interblocage.

La seconde technique permet à un routeur de copier un message dans sa mémoire (plus exactement dans la mémoire du processeur attaché à ce routeur) en même temps qu'il continue à router le message. Ainsi, un message ayant plusieurs destinations (*multi-destination worm* [PSP94]) peut être émis par la source, sous la forme d'un seul message, routé au travers du réseau et son contenu délivré à l'ensemble de ses destinations (sorte de source routing). Ce mode de communication est nommé *path-based* [LMN94, LN91, CN94].

Les principaux critères employés pour évaluer la mise en œuvre d'une opération de multicast sont le *trafic* ou *bande passante* (nombre de canaux utilisés) et la *latence* (temps écoulé entre l'envoi du premier flit du message par la source et la réception du dernier flit par les destinations).

Suivant les différents modes de communication employés, ces problèmes ont été formulés de différentes façons :

Chemin multicast : trouver un chemin de longueur minimale dont l'une des extrémités est la source et qui passe par toutes les destinations ;

Arbre de STEINER : Trouver un arbre de STEINER de longueur minimum ;

Arbre multicast : Trouver un arbre multicast tel que la distance entre la source et les membres soit minimum ;

Étoile multicast ($\mathcal{MS}$) : Trouver une collection de chemins multicast.

Tous ces problèmes sont $\mathcal{NP}$-complet dans le cas général. Le problème de l'arbre de STEINER et du chemin multicast reste $\mathcal{NP}$-complet pour des topologies simples comme la grille 2D et l'hypercube [GJ79, LN91]. Le problème de l'arbre multicast reste lui aussi $\mathcal{NP}$-complet pour l'hypercube. Trouver une $\mathcal{MS}$ minimisant le nombre total de canaux reste $\mathcal{NP}$-complet pour les grilles et les hypercubes [LN91].

Ce qui nous intéresse ici et ce qui a motivé ce travail fait en collaboration avec V. BOUCHITÉ et J. COHEN est de montrer que lorsque l'on prend en compte le problème d'interblocage (comme il a été défini dans la section 2.2.1), notamment en utilisant une fonction de routage qui impose de traverser les canaux de communication suivant un ordre établi par un chemin hamiltonien [LMN94, LN91], alors nous sommes en mesure d'exhiber des algorithmes polynômiaux calculant des $\mathcal{MS}$ optimales en nombre de canaux utilisés ou en longueur maximale des branches.

2.4.1 Définitions

Dans la grille 2D, le routage le plus communément employé est le routage par dimension (*XY-routing*), on progresse dans un premier temps dans la dimension X puis dans la dimension Y. Ce routage est sans interblocage pour les communications unicast [DS87] mais cette propriété n'est pas conservée si l'on utilise le mode de communication path-based [FF98b]. Pour résoudre ce problème, Lin et Ni [LN91] ont proposé une nouvelle fonction de routage qui impose de traverser les canaux de communication suivant un ordre établi par un chemin hamiltonien [CST02]. Plus précisément, on assigne un label $\mathcal{L}(u)$ donné par la numérotation d'un chemin hamiltonien à chaque sommet u d'une grille $m \times n$. $\mathcal{L}(u)$ est défini par :

$$\text{Si } u = (x, y) \text{ alors } \mathcal{L}(u) = \begin{cases} y \times n + x & \text{si } y \text{ est pair} \\ y \times n + n - x - 1 & \text{si } y \text{ est impair} \end{cases}$$

où x et y sont les coordonnées colonne et ligne du sommet u. Une fois les sommets de la grille étiquetés de 0 à $nm - 1$ suivant ce chemin hamiltonien, la route suivie par un message est donnée par la fonction de routage $\mathrm{R}(u, v) = w$ telle que w soit un voisin de u et telle que :

$$\mathcal{L}(w) = \begin{cases} \max\{cL(z) \,|\, \mathcal{L}(z) \leq \mathcal{L}(v), & z \text{ voisin du sommet } u \text{ si } \mathcal{L}(u) < \mathcal{L}(v)\} \\ \max\{cL(z) \,|\, \mathcal{L}(z) \geq \mathcal{L}(v), & z \text{ voisin du sommet } u \text{ si } \mathcal{L}(u) > \mathcal{L}(v)\} \end{cases}$$

Le but d'un algorithme de multicast en mode path-based est de trouver un ensemble de chemins multicast nommé *étoile multicast* que l'on définit de la façon suivante :

Définition 2.3 Soit un graphe $G(V, E)$, $\mathcal{L}$ un étiquetage des sommets de ce graphe et R une fonction de routage nécessitant que les destinations soient parcourues de façon monotone par

rapport à $\mathcal{L}$. Soit (u_0, K) un ensemble de multicast ($u_0 \in V$ est la source et $K = \{u_1, \ldots, u_k\} \subset V$ est l'ensemble des destinations) et on note Δ le degré de u_0. Une *étoile multicast ($\mathcal{MS}$)* est un ensemble de chemins multicast $\{P_i\}$ définis par l'ensemble des sommets destination $\mathcal{D}_i$ qu'ils doivent atteindre, $1 \leq i \leq \Delta$, tel que :

1. Aucun sommet destination n'appartient à plus d'un chemin multicast, *i.e.*, $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ si $i \neq j$;
2. L'ensemble des chemins multicast couvre tous les sommets destinations, *i.e.*, $\bigcup_{1 \leq i \leq \Delta} \mathcal{D}_i = K$;
3. Les destinations au sein de chaque chemin multicast sont visitées dans un ordre monotone en accord avec l'étiquetage $\mathcal{L}$, *i.e.*, $\mathcal{L}(u_{i_j}) < \mathcal{L}(u_{i_{j+1}})$ si $u_{i_1} > u_0$ ou $\mathcal{L}(u_{i_j}) > \mathcal{L}(u_{i_{j+1}})$ si $u_{i_1} < u_0$ où $u_{i_j} \in \mathcal{D}_i$, pour $1 \leq i < \Delta$ et $1 \leq j < |\mathcal{D}_i|$;
4. Le chemin multicast P_i emprunte le canal de sortie i de la source u_0, $1 \leq i \leq \Delta$.

Si on se base sur le modèle employé pour modéliser les communications wormhole (voir équation 2.1), le temps de communication d'une $\mathcal{MS}$ sera borné par :

$$\alpha + (L-1)\tau + \delta \max_{1 \leq i \leq \Delta} \sum_{j=0}^{|\mathcal{D}_i|-1} d_{\mathrm{R}}(u_{i_j}, u_{i_{j+1}}), \text{ avec } u_{i_0} = u_0,\ 1 \leq i \leq \Delta \tag{2.2}$$

où $d_{\mathrm{R}}(u, v)$ est la longueur du chemin de u à v défini par la fonction de routage R.

Nous pouvons maintenant formuler les deux problèmes de minimisation :

Problème 2.1 [$\mathcal{OCMS}$] Trouver une étoile multicast minimisant le nombre de canaux (*optimal channel multicast star ($\mathcal{OCMS}$)*, *i.e.*, une $\mathcal{MS}$ telle que le nombre de canaux utilisés soit minimum :

$$\min_{\mathcal{MS}} \sum_{i=1}^{\Delta} \sum_{j=0}^{|\mathcal{D}_i|-1} d_{\mathrm{R}}(u_{i_j}, u_{i_{j+1}}) \text{ avec } u_{i_0} = u_0,\ 1 \leq i \leq \Delta \tag{2.3}$$

Problème 2.2 [$\mathcal{OTMS}$] Trouver une étoile multicast minimisant le temps de communication (*optimal time multicast star ($\mathcal{OTMS}$)*, *i.e.*, une $\mathcal{MS}$ tel que le temps donné par l'équation 2.2 soit minimum :

$$\min_{\mathcal{MS}} \max_{1 \leq i \leq \Delta} \sum_{j=0}^{|\mathcal{D}_i|-1} d_{\mathrm{R}}(u_{i_j}, u_{i_{j+1}}) \text{ avec } u_{i_0} = u_0,\ 1 \leq i \leq \Delta \tag{2.4}$$

Nous sommes prêts pour poursuivre et présenter les algorithmes permettant de résoudre ces deux problèmes. On peut supposer, sans perte de généralité, que le label de la source est inférieur au label des sommets destinations ($\mathcal{L}(u_0) < \mathcal{L}(u_i), 1 \leq i \leq k = |K|$). Comme nous travaillons dans une grille, on notera $\{\ell, \ell'\}$ les deux canaux de sortie connectant la source à ses voisins ayant chacun un label supérieur au sien. Si la source ne possède qu'un seul lien de sortie, le problème est trivialement résolu en construisant un seul chemin multicast où les destinations sont ordonnées par ordre croissant.

2.4.2 Algorithme calculant une $\mathcal{OCMS}$ dans une grille

Nous désirons construire une $\mathcal{MS}$ ayant au plus deux chemins minimisant le nombre de canaux utilisés. L'algorithme que nous allons présenter est divisé en trois phases : la première construit un graphe biparti pondéré à partir des données fournies (u_0, K) et des contraintes imposées par $\mathcal{L}$ et R ; la deuxième phase calcule un couplage parfait de poids minimum et la troisième et dernière phase extrait l'étoile multicast optimale de ce couplage.

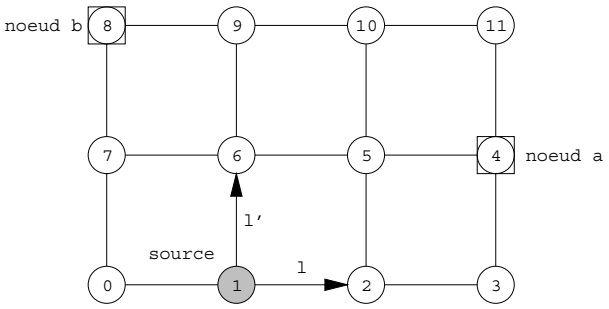


FIG. 2.5 – Contraintes imposées par $\mathcal{L}$ et R.

Avant de poursuivre, nous formalisons la notion de contrainte introduite par la fonction de routage R et l'étiquetage $\mathcal{L}$. En effet, la source joue un rôle prépondérant puisqu'elle possède deux canaux de sortie à partir desquels elle peut construire les deux chemins multicast. La figure 2.5 illustre ce type de contraintes. De plus, la fonction de routage impose aussi qu'un message ne puisse atteindre une destination u_j après avoir atteint la destination u_i seulement si $\mathcal{L}(u_i) < \mathcal{L}(u_j)$. Pour formaliser ces contraintes, nous définissons une fonction de contrainte nommée c : $K \times K \times E^* \rightarrow N$ où $E^* = E \cup \{*\}$. La fonction de contrainte entre deux sommets u et v relativement à un canal de sortie ℓ est définie par :

$$\mathrm{c}(u_i, u_j, \ell) = \begin{cases} d_{\mathrm{R}}(u_i, u_j) & \text{if } \ell = * \text{ and } \mathcal{L}(u_i) < \mathcal{L}(u_j) \\ d_{\mathrm{R}}(u_i, u_j) & \text{if } \ell = (u_i, w), \mathcal{L}(u_i) < \mathcal{L}(u_j) \text{ and } \mathrm{R}(u_i, u_j) = w \\ \infty & \text{otherwise} \end{cases} \tag{2.5}$$

Cette fonction $c(v, w, \ell)$ retourne ∞ s'il est impossible de retransmettre un message de v vers w en empruntant le canal de sortie ℓ. Si c'est faisable, la fonction de contrainte retourne le coût du chemin donné par la fonction de routage. De plus, $\ell = *$ indique que l'on ne précise pas le canal de sortie, ce qui implique que si $c(v, w, *)$ retourne ∞, la fonction de routage n'autorise pas la retransmission d'un message de v à destination de w, et ce, indépendamment du canal de sortie.

À partir de cette fonction de contrainte c, l'idée directrice est de construire un graphe biparti $\mathcal{B}_{(u_0,K)} = (Q, R, F, w)$, où Q et R sont les deux ensembles de sommets, F l'ensemble d'arêtes et w une fonction de poids. Ce graphe biparti va modéliser toutes les contraintes imposées par la fonction de routage R et/ou induites par l'étiquetage $\mathcal{L}$. De façon plus formelle, on note $\Omega(u_0)$ l'ensemble des canaux de sortie d'une source u_0 tel qu'un chemin multicast empruntant l'un de ces canaux peut atteindre au moins un des sommets destination de K (Dans la grille, $|\Omega(u_0)| = 1$ ou 2). On définit le graphe biparti $\mathcal{B}_{(u_0,K)} = (Q, R, F, w)$ par :

– chaque sommet $u_i \in K, 1 \leq i \leq k = |K|$ possède une instance $q_i \in Q$ et une instance $r_i \in R$;

- le sommet source u_0 possède $|\Omega(u_0)|$ instances $\{q^{\ell}_{0,\ell\in\Omega(u_0)}\} \in Q$. Ces $|\Omega(u_0)|$ instances représentent les $|\Omega(u_0)|$ canaux de sortie disponibles pour construire les $|\Omega(u_0)|$ chemins multicast ;
- afin d'avoir le même nombre de sommets dans Q que dans R nous créons artificiellement $|\Omega(u_0)|$ sommets $\{r_{k+i,1\leq i\leq|\Omega(u_0)|}\} \in R$. Ces $|\Omega(u_0)|$ sommets fantômes vont représenter les extrémités terminales des $|\Omega(u_0)|$ chemins multicast ;
- il existe un arc[5] $e = (q^0_l, r_j)$ dans $\mathcal{B}_{(u_0,K)}$ si $c(u_0, u_j, \ell) \neq \infty$ et le poids de l'arc e est $w(e) = c(u_0, u_j, \ell)$. Cela indique que la fonction de routage R permet de transmettre un message de u_0 à u_j en empruntant comme premier canal de sortie le lien ℓ et que la longueur du chemin est donnée par $d_{\mathrm{R}}(u_0, u_j)$. De plus, il existe un arc $e = (q_i, r_i)$ dans $\mathcal{B}_{(u_0,K)}$ si $c(u_i, u_j, *) \neq \infty$ et le poids de l'arc e est $w(e) = c(u_i, u_j, *)$. Cela indique que la fonction de routage R permet de transmettre un message de u_i à u_j ;
- tous les sommets $q \in Q$ sont connectés par un arc de poids nul à tous les sommets de $\{r_{k+i,1\leq i\leq|\Omega(u_0)|}\}$. Cela indique que tout sommet $u_i, 0 \leq i \leq k$ de l'ensemble multicast peut être la destination finale des $|\Omega(u_0)|$ chemins multicast. Notons que cela inclut la source u_0.

Nous pouvons maintenant présenter les propriétés importantes du graphe biparti que l'on vient de définir. Les preuves des lemmes sont données en annexe ([BCF98]). Nous rappelons [LP86, GM95] simplement qu'un *couplage* M de $\mathcal{B}_{(u_0,K)}$ est un sous-ensemble d'arcs $M \subset F$ tel qu'il n'existe pas deux arcs de M qui soient adjacents. Un couplage est dit *parfait* si chaque sommet de $Q \cup R$ est l'une des extrémités des arcs du couplage M.

Lemme 2.1 *$\mathcal{B}_{(u_0,K)}$ contient un couplage parfait.*

Lemme 2.2 *Chaque $\mathcal{MS}$ d'un ensemble multicast (u_0, K) donné est représentée par un couplage parfait du graphe biparti $\mathcal{B}_{(u_0,K)}$ et réciproquement.*

Théorème 2.1 *Chaque $\mathcal{OCMS}$ d'un ensemble multicast (u_0, K) donné est représentée par un couplage parfait de poids minimum du graphe biparti $\mathcal{B}_{(u_0,K)}$ et réciproquement.*

Preuve. Le lemme 2.2 nous assure que chaque $\mathcal{MS}$ d'un ensemble multicast (u_0, K) donné est représentée par un couplage parfait du graphe biparti $\mathcal{B}_{(u_0,K)}$. Il suffit de noter que le nombre de canaux employés par la $\mathcal{MS}$ est égal au poids du couplage parfait associé. Étant donné qu'une $\mathcal{OCMS}$ d'un ensemble multicast (u_0, K) donné emploie un nombre minimum d'arêtes, le théorème est valide. □

Un algorithme qui met en œuvre cette construction en trois phases est donné en annexe. Sa complexité est en $O\left(k^3\right)$. La preuve de l'algorithme est assurée par le théorème 2.1.

2.4.3 Algorithme calculant une $\mathcal{OTMS}$ dans une grille

Nous désirons maintenant construire une $\mathcal{OTMS}$ *i.e.*, une $\mathcal{MS}$ dont le maximum des longueurs des chemins multicast soit minimum. L'algorithme que nous allons présenter se décompose tout comme le précédent en trois phases : une première phase d'initialisation, une deuxième phase

[5] un arc est orienté par définition [Ber83]

de calcul et une troisième et dernière phase d'extraction du résultat. L'idée sous-jacente est de prendre en compte (ce qui est différent d'explorer) tous les choix possibles permettant d'effectuer un multicast de la source u_0 vers l'ensemble de destination $K = \{u_1, \ldots, u_k\}$. De façon plus précise, nous allons nous intéresser à toutes les $\mathcal{MS}$ des ensembles de multicast (u_0, K_i) où $K_i = \{u_1, \ldots, u_i\}, 1 \leq i \leq k = |K|$. Pour ce, nous allons construire un ensemble de sommets Q de la façon suivante :

- la source u_0 possède $|\Omega(u_0)|$ instances $\{u^{\ell}_{0,\ell\in\Omega(u_0)}\}$. Ces $|\Omega(u_0)|$ instances représentent les $|\Omega(u_0)|$ canaux de sortie disponibles pour construire les $|\Omega(u_0)|$ chemins multicast. Pour simplifier les notations dans le cas de la grille, on notera q_0 et q_1 ces deux instances ;
- chaque sommet $u_i \in K, 1 \leq i \leq k = |K|$ possède une instance $q_{i+1} \in Q$.

Nous choisissons alors de représenter une $\mathcal{MS}$ d'un ensemble multicast (u_0, K_i) par le tuple $\mathcal{X} = (i + 1, j, \ell_1, \ell_2)$ tel que :

- $i + 1 > j$;
- un des deux chemins multicast a comme extrémité terminale le sommet q_{i+1} (correspondant au sommet u_i) et la longueur de ce chemin multicast est ℓ_1 ;
- l'autre chemin multicast a comme extrémité terminale le sommet q_j (correspondant au sommet u_{j-1} si $j > 0$) et sa longueur est ℓ_2.

Ce tuple $\mathcal{X}$ permet de modéliser l'ensemble des $\mathcal{MS}$ possibles. Considérons le cas particulier de l'ensemble multicast $(u_0, \emptyset)$. Il existe une seule $\mathcal{MS}$ correspondant à cet ensemble multicast qui est bien définie par le tuple $(1, 0, 0, 0)$. En d'autres termes, la $\mathcal{MS}$ est composée d'un chemin multicast dont l'extrémité est q_1 qui correspond au sommet u^{ℓ}_0 ce qui implique que le chemin multicast emprunte le canal de sortie ℓ. Toute $\mathcal{MS}$ d'un ensemble multicast (u_0, K) est représentable sous la forme d'un tuple $(|K| + 1, j, \ell_1, \ell_2)$. En effet, l'un des deux chemins multicast doit se terminer au sommet ayant le plus grand label et l'autre peut avoir comme extrémité terminale tout autre sommet.

L'idée générale est donc d'utiliser cette représentation sous forme de tuple et de remplir un tableau $T[\mathcal{X}]$ parcourant l'ensemble des tuples « utiles ». Pour ne pas explorer trop de cas et garder un algorithme polynômial en la taille des données, nous allons employer une approche basée sur la programmation dynamique [CLRS01]. Les trois phases de l'algorithme sont les suivantes :

1. Initialisation d'un tableau à quatre dimensions correspondant aux tuples définis ci-dessus. L'ensemble des éléments de ce tableau est initialisé à False sauf l'élément $(1, 0, 0, 0)$ qui est initialisé à $True$. Nous verrons ci-dessous comment borner les indices de ce tableau. Pour simplifier les notations dans la description de l'algorithme on définit la fonction c' : $Q \times Q \rightarrow N$ à partir de la fonction de contrainte c de la façon suivante :

$$c'(q_i, q_j) = \begin{cases} c(q_i, q_j, \ell) & \text{si } q_i = u^{\ell}_0 \text{ où } \ell \in \Omega(u_0) \\ c(q_i, q_j, *) & \text{si } q_i \notin \{u^{\ell}_0, u^{\ell'}_0 \end{cases}$$

   Cette fonction c' exprime la fonction de routage R contrainte par $\mathcal{L}$ appliqué à l'ensemble Q que l'on vient de définir. Par exemple, si $c'(q_i, q_j) = d$ cela exprime simplement qu'il est possible de router un message de u_{i-1} (ou u_i si $i = 0$) à u_{j-1} (ou u_j si $j = 0$).

2. La seconde étape remplit le tableau en employant l'algorithme 2.1.

Algorithme 2.1 Fill(T,u,K)

▷ *Initialise le tableau T*
1 **for** i_1 = 2 to k **do**
2 **for** i_2 = 2 to i_1 **do**
3 **for** k_1, k_2 = 2 to N **do**
 ▷ *N est le nombre de sommets de la grille*
4 **if** T(i_1,i_2,k_1,k_2)= True **then**
5 T(i_1 + 1,i_2,k_1+$c'(q_{i_1}, q_{i_1+1})$,k_2) ← True
6 T(i_1 + 1,i_1,k_2+$c'(q_{i_2}, q_{i_1+1})$,k_1) ← True
7 **end if**

Cet algorithme garantit que la relation suivante est vérifiée :

$$T[\mathcal{X}] = True \Leftrightarrow \text{il existe une } \mathcal{MS} \text{ définie par un tuple } \mathcal{X} ;$$

3. Pour finir, parmi toutes les $\mathcal{MS}$ de l'ensemble de multicast (u_0, K) qui sont représentées par les éléments $(|K|+1, j, \ell_{|K|,\ell_j})$ de T tel que $T[|K|+1, j, \ell_{|K|,\ell_j}] = True$ on calcule les deux chemins multicast qui minimisent le maximum des distances.

Il nous reste à borner les indices du tableau T et à donner la preuve de l'algorithme 2.1. On peut remarquer que la longueur maximum des chemins multicast est bornée par le nombre de sommets du graphe G puisque la fonction de routage impose de visiter les sommets suivant un ordre monotone, ce qui interdit de repasser par un sommet. En conséquence, chaque $\mathcal{MS}$ peut être pleinement définie par un tuple $\mathcal{X}$ dans l'espace $[0, \ldots, k]^2 \times [1, \ldots, N-1]^2$ où $N = nm$ est le nombre de sommets du graphe G. Notons aussi que l'algorithme 2.1 rempli le tableau T en calculant le tuple $\mathcal{X}$ avant le tuple $\mathcal{X}'$ si ce dernier est plus grand que $\mathcal{X}$ dans l'ordre lexicographique. Cela implique qu'un chemin visitera les destinations dans l'ordre monotone de leur étiquette. Pour finir la preuve de l'algorithme, nous montrons le théorème suivant (la preuve est disponible dans l'article [BCF98] qui est inséré en annexe) :

Théorème 2.2 *S'il existe une $\mathcal{MS}$ pour un ensemble de multicast (u_0, K_i) tel que*

1. *l'une des extrémités terminales du chemin multicast P_1 est le sommet u_i et la longueur du chemin P_1 est égale à ℓ_1 ;*
2. *l'une des extrémités terminales du chemin multicast P_2 est le sommet u_{j-1} (si $j > 0$ ou u_0 sinon) et la longueur du chemin P_2 est égale à ℓ_2 ;*
3. $K_i = \{u_1, \ldots, u_i\}, u_j \in K, 1 \leq j \leq i$

alors l'élément $(i+1, j, \ell_1, \ell_2)$ du tableau T est égal à $True$ et réciproquement.

Grâce au théorème 2.2, une fois le tableau T rempli, les tuples $\mathcal{X} = (|K|+1, j, \ell_{|K|+1}, \ell_j)$ tels que $T[\mathcal{X}] = True$ forment l'ensemble des $\mathcal{MS}$ possibles pour l'ensemble de multicast $((u_0, K)$. La dernière tâche à exécuter est de calculer le minimum des longueurs maximum de ces chemins. La complexité de cet algorithme est en $O\left(k^2N^2\right)$. La phase d'initialisation et de calcul du tableau T pouvant s'effectuer en $O\left(k^2N^2\right)$ opérations, l'extraction de la solution revient à extraire une valeur minimum parmi $O\left(kN^2\right)$ valeurs.

2.4.4 Et si on change de graphe ?

Dans les deux sections précédentes 2.4.2 et 2.4.3 le graphe est fixé. On peut se demander si les deux algorithmes que l'on vient de décrire restent valides pour une classe de graphes plus grande.

En fait, pour un graphe G hamiltonien, il est toujours possible de numéroter ses sommets en fonction d'un parcours hamiltonien et de mettre en œuvre une fonction de routage respectant les propriétés énoncées dans la section 2.4.1. Pour généraliser nos deux algorithmes, il faut étendre la notion de fonction de contrainte. On note $d_{\mathrm{R}}(u, v, \ell)$ la distance de u à v du chemin donné par la fonction de routage R en empruntant le canal de sortie ℓ en u. La généralisation de la fonction de contrainte est donnée par :

$$\mathrm{c}(u_i, u_j, \ell) = \begin{cases} \min_{\ell' \in \Omega(u_i)} d_{\mathrm{R}}(u_i, u_j, \ell') & \text{si } \ell = * \\ d_{\mathrm{R}}(u_i, u_j, \ell) & \text{si } \ell = (u_i, w), \text{ et } \mathrm{R}(u_i, u_j) = w \\ \infty & \text{sinon} \end{cases} \tag{2.6}$$

Le calcul d'une $\mathcal{OCMS}$ reste identique. On ne modifie que la construction du graphe biparti : la source u_0 possède $|\Omega(u_0)|$ copies dans Q et R. La complexité est en $O\left((k + |\Omega(u_0)|)^3\right)$. Le calcul d'une $\mathcal{OTMS}$ reste le même. Il suffit de modéliser une $\mathcal{MS}$ par un tuple possédant $2|\Omega(u_0)|$ étant donné que la $\mathcal{MS}$ possède $|\Omega(u_0)|$ chemins multicast. La complexité est en $O\left((kN)^{|\Omega(u_0)|}\right)$ et dépend donc du degré de la source. Elle reste polynômiale pour les graphes ayant des degrés constants.

2.5 Conclusion

Fort de l'expérience acquise dans le domaine de l'interblocage, il faut retenir que de nombreuses théories ont été proposées pour résoudre le problème d'interblocage mais que peu sont assez généralistes dans leurs fondements et hypothèses pour couvrir l'ensemble des cas. Comme nous l'avons noté, même la dernière en date, proposée par J. DUATO et de T. PINKSTON [DP02], qui tente une approche fédératrice puisqu'elle englobe à la fois le mode de commutation wormhole et le mode cut-throught, se trouve limitée dans son domaine d'application car la définition de la notion de fonction de routage n'est pas assez large, contrairement à la définition que nous avions proposée. Ces travaux, qualifiables de théoriques, et développés pour la grande majorité dans la mouvance « machine massivement parallèle » trouvent aussi des domaines d'application. Une nouvelle technologie relance l'attrait d'une recherche des fonctions de routage wormhole sans interblocage. En effet, après des technologies comme myrinet, InfiniBand se propose d'être le nouveau standard qui doit s'imposer pour tout ce qui est communication entre processeurs et devices d'entrée/sortie (*System Area Network (SANs)*) et pour la connexion point-à-point de divers équipements InfiniBand via des switches. Un autre domaine d'application consiste à se pencher, non plus sur le réseau d'interconnexion entre processeurs, mais sur le réseau employé au sein des switches eux-mêmes où l'on peut retrouver une problématique très similaire [TL00].

L'autre leçon qu'il faut retenir pourrait être *il ne faut pas se tromper de problème*. Le corollaire est que sous certaines conditions, il est possible d'exhiber des algorithmes polynômiaux pour résoudre un problème qui est connu comme NP-complet dans le cas général. Ce qui est important ici, c'est la modélisation du système qui permet justement d'introduire des contraintes, notamment sur la fonction de routage. Dans le cas du multicast path-based, ces contraintes ont été bénéfiques et nous ont permis d'exhiber des algorithmes minimisant soit le trafic, soit le temps nécessaire pour réaliser un multicast dans une grille.

De façon plus générale ou plus personnelle, étudier des graphes ou des familles de graphes pour en déduire des caractéristiques, des propriétés, des comportements est fondamental et

nécessaire puisque cela représente la base dès que l'on se pose la question des performances possibles d'un réseau d'interconnexion. Malheureusement, les clusters de NOWs et, de façon plus large encore, l'Internet ne sont pas des graphes réguliers et sont assez difficiles à caractériser. Il devient alors difficile de les étudier en les considérant comme un objet de type graphe classique même si une telle modélisation offre des outils très précieux pour montrer la NP-complétude d'un problème.

De nouvelles perspectives aux retombées potentielles énormes sont apparues avec les nanotechnologies. Le réseau d'interconnexion n'est plus entre des processeurs, ni entre des PC, ni entre des réseaux locaux mais intégré sur un chip grâce à des technologies permettant de faire des portes logiques de l'ordre de 50nm à 100nm. Ce concept de *SoC* (*Systems on Chips*) [BDM01, HMH01] introduit de nombreux défis qui présentent de fortes similitudes avec la problématique des réseaux d'interconnexion des multiprocesseurs :

- Les SoC vont être assemblés en utilisant des composants préexistants (processeurs, contrôleur, mémoire).
- Les SoC devront offrir des opérations fonctionnellement correctes et fiables entre les composants assemblés. La couche d'interconnexion physique sera une contrainte en terme de performance et de consommation d'énergie. Il faut avoir un *dessin* du réseau d'interconnexion et donc de la surface synthétisée qui soit régulier.
- Le but final de la conception des SoCs est d'avoir une certaine QoS (*sic*.) en consommant le moins d'énergie possible.

Ces nouveaux travaux font appel à des connaissances qui transcendent la seule conception de circuit et offrent une réelle opportunité de collaboration entre différentes communautés scientifiques. De par les contraintes et le type de réseau cherché, il semble que les travaux, que j'ai qualifiés en début de chapitre de théoriques redeviennent d'actualité. De plus, ces SoCs qui ne seront pas entièrement synchrones mais vraisemblablement *GALS* (Globally-Asynchronous Locally-Synchronous), font appel à une organisation en couches et (re)définissent la couche physique, la couche liaison, la couche réseau et la couche transport. Pour chaque couche, on retrouve des problématiques inhérentes comme l'accès au médium, le codage, la fiabilité du transport, ce qui n'est pas sans rappeler les architectures déployées dans le monde des réseaux avec en plus de très fortes contraintes énergétiques.

Bibliographie

[ABC+00] F. Alfaro, A. Bermúdez, R. Casado, F. Quiles, J. Sánchez, and J. Duato, *On the performance of up*/down* routing*, Communication, Architecture and Applications for Network-based Parallel Computing (CANPC '00) (Toulouse, France), January 2000.

[AP95] K. Anjan and T. Pinkston, *DISHA : A deadlock recovery scheme for fully adaptive routing*, International Parallel Processing Symposium (IPPS '95), April 1995.

[BCF+95] N. Boden, D. Cohen, R. Felderman, A. Kulawik, C. Seitz, J. Seizovic, and W.-K. Su, *Myrinet – a gigabit-per-second local-area network*, IEEE Micro **15** (1995), no. 1, 29–36.

[BCF98] V. Bouchitté, J. Cohen, and E. Fleury, *Optimal deadlock-free path-based multicast algorithms in meshes*, 5th International Colloquium on Structural Information and Communication Complexity (SIROCCO'98) (Amalfi, Italy), June 1998.

[BDM01] L. Benini and G. De Micheli, *Powering networks on chips*, ISSS'01 (Montréal, Canada), October 2001.

[Ber83] C. Berge, *Graphes*, 3e édition ed., Gauthier-Villars, 1983.

[BFP96] P. Berthomé, A. Ferreira, and S. Perennes, *Optimal information dissemination in star and pancake networks*, IEEE Transactions on Parallel and Distributed Systems **7** (1996), no. 12, 1292–1300.

[BP89] J-C. Bermond and C. Peyrat, *de Bruijn and Kautz networks : a competitor for the hypercube ?*, Hypercube and Distributed Computers (F. Andre and J. P. Verjus, eds.), North-Holland, 1989, pp. 279–294.

[BPDS00] D. Buntinas, D. Panda, J. Duato, and P. Sadayappan, *Broadcast/multicast over myrinet using NIC-assisted multidestination messages*, Communication, Architecture, and Applications for Network-based Parallel Computing (CANPC '00) (Toulouse, France), January 2000, pp. 115–129.

[CBD+01] R. Casado, A. Bermúdez, J. Duato, F. Quiles, and J. Sánchez, *A protocol for deadlock-free dynamic reconfiguration in high-speed local area networks*, IEEE Transactions on Parallel and Distributed Systems **12** (2001), no. 2, 115–132.

[CKR95] L. Cherkasova, V. Kotov, and Rockicki, *Fiber channel fabrics : Evaluation and design*, International Conference on System Sciences, February 1995.

[CLRS01] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms, second edition*, MIT Press, 2001, ISBN : 0262032937.

[CN94] C-M. Chiang and L. M. Ni, *Multi-address encoding for multicast*, Parallel Computer Routing and Communication, First International Workshop, (PCRCW '94) (K. Bolding and L. Snyder, eds.), Lecture Notes in Computer Science, no. 853, Springer-Verlag, May 1994, pp. 146–160.

[CP01] Y. Choi and T. Pinkston, *Evaluation of crossbar architecture for deadlock recovery routers*, Journal of Parallel and Distributed Computing **61** (2001), 49–78.

[CST02] S. Chen, H. Shen, and R. Topor, *An efficient algorithm for constructing Hamiltonian paths in meshes*, Parallel Computing **28** (2002), 1293–1305.

[CV96] J. Carbonaro and F. Verhoorn, *Cavallino : The teraflops router and NIC*, Hot Interconnects IV, 1996, pp. 157–160.

[Des01] F. Desprez, *Contribution à l'algorithmique parallèle – calcul numérique : des bibliothèques aux environnements de métacomputing –*, Habilitation à diriger des recherches, Université Claude Bernard Lyon 1, Lyon, France, Juillet 2001.

[DP02] J. Duato and T. Pinkston, *A general theory for deadlock-free adaptive routing using a mixed set of resources*, IEEE Transactions on Parallel and Distributed Systems **12** (2002), no. 12, 1219–1235.

[DRS01] J. Duato, A. Robles, and F. Silla, *A comparison of router architectures for virtual cut-through and wormhole switching in a now environment*, Journal of Parallel and Distributed Computing **61** (2001), 224–253.

[DS86] W. J. Dally and C. L. Seitz, *The torus routing chip*, Distributed Computing **1** (1986), no. 3, 87–196.

[DS87] ———, *Deadlock-free message routing in multiprocessor interconnection networks*, IEEE Transactions on Computers **C-36** (1987), no. 5, 547–553.

[DYN02] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection networks : An engineering approach*, Morgan Kaufmann, July 2002, ISBN : 1558608524 (Revised edition).

[FF98a] E. Fleury and P. Fraigniaud, *A general theory for deadlock avoidance in wormhole-routed networks*, IEEE Transactions on Parallel and Distributed Systems **9** (1998), no. 7, 626–638.

[FF98b] ———, *Strategies for path-based multicasting in wormhole-routed meshes*, Journal of Parallel and Distributed Computing **53** (1998), no. 1, 26–62.

[FL94] P. Fraigniaud and E. Lazard, *Methods and problems of communication in usual networks*, Discrete Applied Mathematics **53** (1994), 79–133, (special issue on broadcasting).

[Fle96] E. Fleury, *Communications, routage et architectures des machines à mémoire distribuée – autour du routage wormhole*, Ph.D. thesis, École Normale Supérieure de Lyon, Lyon, France, October 1996.

[FLMD02] J. Flich, P. López, M. Malumbres, and J. Duato, *Boosting the performance of myrinet networks*, IEEE Transactions on Parallel and Distributed Systems **13** (2002), no. 7, 693–709.

[FLS+02] J. Flich, P. López, J.C. Sancho, A. Robles, and J. Duato, *Improving infiniband routing through multiple virtual networks*, International Symposium on High Performance Computing (ISHPC IV) (ansai Science City, Japan), LNCS, no. 2327, May 2002, pp. 49–63.

[Gal97] M. Galles, *Spider : A high speed network interconnect*, IEEE Micro (1997), 34–39.

[Gav96] C. Gavoille, *Complexité mémoire du routage dans les réseaux distribués*, Ph.D. thesis, École Normale Supérieure de Lyon, January 1996.

[Gav00] ———, *Structures de données compactes et distribuées*, Habilitation à diriger des recherches, Université Bordeaux I, Bordeaux, France, Décembre 2000.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and intractability : A guide to the theory of NP-completeness*, Computer science / mathematics, W. H. Freeman and Company, 1979.

[GM95] M. Gondran and M. Minoux, *Graphes et algorithmes*, Eyrolles, 1995, ISBN : 2212015712.

[GW97] D. Garcia and W. Watson, *ServerNet II*, Parallel Computer, Routing and Communication Workshop, June 1997.

[HHL86] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, *A survey of gossiping and broadcasting in communication networks*, Networks **18** (1986), 319–349.

[HMH01] R. Ho, K. Mai, and M. Horowitz, *The future of wires*, Proceedings of the IEEE, January 2001.

[Hor95] R. Horst, *Tnet : A reliable system area network*, IEEE Micro **15** (1995), no. 1, 37–45.

[Inf02] *InfiniBand architecture specification volume 1*, InfiniBand Trade Association, June 2002, release 1.0.a, `http://www.infinibandta.com`.

[KK79] P. Kermani and L. Kleinrock, *Virtual cut-through : A new computer communication switching technique*, Computer Networks **3** (1979), 267–286.

[KLC94] J. Kim, Z. Liu, and A. Chien, *Compressionless routing : A framework for adaptative and fault-tolerant routing*, International Symposium Computer Architecture, April 1994.

[LD00] O. Lysne and J. Duato, *Fast dynamic reconfiguration in irregular networks*, International Conference on Parallel Processing (ICPP 2000), 2000, pp. 449–458.

[LJD93] S. Lakshmivarahan, Jung-Sing Jwo, and S. K. Dhall, *Symmetry in interconnection networks based on caley graphs of permutation groups : A survey*, Parallel Computing **19** (1993), 361–407.

[LMN94] X. Lin, P. K. McKinley, and L. M. Ni, *Deadlock-free multicast wormhole routing in 2D-mesh multicomputers*, IEEE Transactions on Parallel and Distributed Systems **5** (1994), no. 8, 793–804.

[LN91] X. Lin and L. M. Ni, *Deadlock-free multicast wormhole routing in multicomputer networks*, 18th Annual International Symposium on Computer Architecture (Toronto), ACM, May 1991, (see also Technical Report MSU-CPS-ACS-29, 1990), pp. 116–125.

[LP86] L. Lovasz and M. Plummer, *Matching theory*, North-Holland, 1986, ASIN : 0444879161.

[LS01] O. Lysne and T. Skeie, *Load balancing of irregular system area networks through multiple roots*, International Conference on Communication in Computing (CIC 2001), CSREA Press, 2001, pp. 165–171.

[MJ94] J. Mišić and Z. Jovanović, *Routing function and deadlock avoidance in a star graph interconnection network*, Journal of Parallel and Distributed Computing **22** (1994), 216–228.

[MRLD01] J. Martínez-Rubio, P. López, and J. Duato, *A cost-effective approach to deadlock handling in wormhole networks*, IEEE Transactions on Parallel and Distributed Systems **12** (2001), no. 7, 716–729.

[MXEN94] P. K. McKinley, H. Xu, A-H. Esfahanian, and L. M. Ni, *Unicast-based multicast communication in wormhole-routed networks*, IEEE Transactions on Parallel and Distributed Systems **5** (1994), no. 12, 1252–1265.

[NKN+00] S. Nishimura, T. Kudoh, H. Nishi, J. Yamamoto, K. Harasawa, N. Matsudaira, and H. Amano, *64-Gbit/s highly reliable network switch using parallel optical interconnection*, IEEE Journal of Lightwave Technology **18** (2000), no. 12, 1620–1627.

[NKN+01] S. Nishimura, T. Kudoh, H. Nishi, J. Yamamoto, K. Harasawa, N. Matsudaira, S. Akutsu, K. Tasho, and H. Amano, *RHiNET-3/SW : an 80-Gbit/s high-speed network switch for distributed parallel computing*, Hot Interconnect 9, 2001, pp. 119–123.

[NM93] L. Ni and P. McKinley, *A survey of wormhole routing techniques in direct networks*, IEEE Computers **26** (1993), no. 2, 62–76.

[PSP94] D. Panda, S. Singal, and P. Prabhakaran, *Multidestination message passing mechanism conforming to base wormhole routing scheme*, Parallel Computer Routing and Communication (PCRCW '94) (Seattle, Washington, USA) (K. Bolding and L. Snyder, eds.), Lecture Notes in Computer Science, no. 853, Springer-Verlag, May 1994, pp. 131–145.

[QN96] W. Qiao and L. Ni, *Adaptive routing in irregular networks using cut-trough switches*, International Conference on Parallel Processing (ICPP '96), August 1996.

[SB90] R. Suaya and G. Birtwistle (eds.), *VLSI and parallel computation*, Morgan Kaufmann, 1990, ISBN 0-934613-99-0.

[SBB+90] M. Schroeder, A. Birrell, M. Burrows, H. Murray, R. Needham, T. Rodeheffer, E. Satterthwaite, and C. Thacker, *Autonet : a high-speed, self-configuring local area network using point-to-point links*, Tech. Report 59, Digital Equipment Corporation, 1990.

[SD00a] F. Silla and J. Duato, *High-performance routing in networks of workstations with irregular topology*, IEEE Transactions on Parallel and Distributed Systems **11** (2000), no. 7, 699–719.

[SD00b] ———, *On the use of virtual channels in networks of workstations with irregular topology*, IEEE Transactions on Parallel and Distributed Systems **11** (2000), no. 8, 813–828.

[She98] R. Sheifert, *Gigabit ethernet*, Addison-Wesley, 1998.

[SKPS98] R. Sivaram, R. Kesavan, D. Panda, and C. Stunkel, *Architectural support for efficient multicasting in irregular networks*, International Conference on Parallel Processing (ICPP '98), IEEE, 1998, (Also appeared as OSU Technical Report OSU-CISRC-10/98-TR41), pp. 452–459.

[SLT02] T. Skeie, O. Lysne, and I. Theiss, *Layered shortest path (LASH) routing in irregular system area networks*, Communication Architectures for Clusters (CAC 2002), 2002.

[SRD00] J.C. Sancho, A. Robles, and J. Duato, *A new methodology to computer deadlock-free routing tables for irregular networks*, Communication, Architecture, and Applications for Network-based Parallel Computing (CANPC '00) (Toulouse, France), January 2000, pp. 45–60.

[TL00] I. Theiss and O. Lysne, *Deadlock avoidance for wormhole based switches*, Euro-Par 2000, LNCS, no. 1900, Springer-Verlag, 2000, pp. 890–899.

[VSD01] A. Vaidya, A. Sivasubramaniam, and C. Das, *Impact of virtual channels and adaptive routing on application performance*, IEEE Transactions on Parallel and Distributed Systems **12** (2001), no. 2, 223–237.

Publications

Livres, chapitre de Livre

[DFM00] F. Desprez, E. Fleury, and J.-F. Méhaut (eds.), *Workshop on metacomputing and applications (msa'2001)*, Valence, Spain, IEEE Computer Society, AUG 2000.

[DFMR00] F. Desprez, E. Fleury, J.-F. Méhaut, and Y. Robert (eds.), *Workshop on metacomputing and applications (msa'2000)*, Toronto, Canada, IEEE Computer Society, AUG 2000.

Journaux, conférences

[BCF98] V. Bouchitté, J. Cohen, and E. Fleury, *Optimal deadlock-free path-based multicast algorithms in meshes*, 5th International Colloquium on Structural Information and Communication Complexity (SIROCCO'98) (Amalfi, Italy), June 1998.

[EsFDG99] T. Es-sqalli, E. Fleury, E. Dillon, and J. Guyard, *Using MeDLey in a CORBA environment*, Euro-par'99 Parallel Processing (Toulouse, France) (P. Amestoy, P. Berger, M. Dayde, I. Duff, V. Fraysse, L. Giraud, and D. Ruiz, eds.), Lecture Notes in Computer Science, vol. 1685, September 1999, pp. 113–116.

[EsFG00a] T. Es-squalli, E. Fleury, and J. Guyard, *A broadcast message passing protocol based on corba event service*, Distributed Objects in Computational Science (DOCS'2000) (Las Vegas, Nevada), CSREA Press, June 2000.

[EsFG00b] ———, *MeDLey : from point-to-point to collective communications*, High Performance Computing in Asia-Pacific Region (HPCA-Asia 2000) (Beijing, China), IEEE, May 2000.

[EsFG00c] ———, *MPC : a new Message Passing library in Corba*, International Workshop on Metacomputing Systems and Applications (MSA) (Toronto, Canada), IEEE, August 2000.

[FF98a] E. Fleury and P. Fraigniaud, *A general theory for deadlock avoidance in wormhole-routed networks*, IEEE Transactions on Parallel and Distributed Systems **9** (1998), no. 7, 626–638.

[FF98b] ———, *Strategies for path-based multicasting in wormhole-routed meshes*, Journal of Parallel and Distributed Computing **53** (1998), no. 1, 26–62.

Chapitre 3

Multicast dans le monde IP

L’amitié se nourrit de communication...

Michel Eyquem DE MONTAIGNE

3.1 Introduction

Dans le courant des années 80, les réseaux étaient totalement dédiés à un service unique et spécifique (réseau téléphonique pour la voix, X.25 pour les données, CATV 1 pour la télévision). Ces réseaux ne permettaient pas, sauf dans des proportions très limitées, de déployer d'autres services que ceux pour lesquels ils étaient conçus et dévolus. Les avancées connues ces vingt dernières années dans le domaine des technologies des communications ont été fulgurantes et déterminantes. L'Internet, qui démarra comme un projet expérimental devant connecter quelques sites militaires et universitaires, a été déployé sur l'ensemble des continents et n'est plus le privilège de quelques petits groupes de chercheurs ou d'universitaires. En effet, pour beaucoup de monde, l'Internet est désormais au cœur de la vie quotidienne pour un usage professionnel, ludique ou culturel [CWSB02]. Dans le même temps, des investissements à long terme dans les infrastructures de télécommunications, souvent nommées *autoroutes de l'information* (téléphone, satellite, télévision par câble) ont été entrepris par beaucoup de pays, ce qui permet d'offrir un large spectre de services jusqu'à l'abonné privé (*i.e.*, jusqu'au particulier chez lui). Ces nouveaux services couvrent la vidéo à la demande, les services de téléphonie multimédia, les jeux télévisuels interactifs, la recherche d'informations.

Notons que ces avancées dans le domaines des technologies des communications ne se sont pas uniquement restreintes à des débits toujours plus élevés avec des taux de perte toujours plus faibles, mais ont aussi donné naissance à de nouveaux paradigmes, de nouvelles façons d'employer ces canaux de communication. Le support des communications multi-parties (*multiparty communication)*, *i.e.*, la possibilité d'avoir une conversation entre plus de deux intervenants, demeure l'un des sujets de recherche toujours très actifs. Dans ce chapitre, nous présentons les grandes classes d'applications nécessitant des communications multi-parties et les protocoles de communication de groupe (*multicast protocols*) qui permettent de mettre en œuvre ce type d'applications. Mes contributions se focalisent sur la façon de construire un « bon » arbre de multicast et principalement sur les mécanismes à mettre en œuvre pour que cet arbre puisse évoluer en harmonie avec les membres du groupe multicast.

Plus précisément, mes travaux sur les protocoles de multicast dans l'Internet s'articulent principalement autour de trois problèmes qui se posent lorsque l'on désire mettre en œuvre un protocole de multicast basé sur un arbre partagé enraciné en un routeur spécifique nommé *core* (ou *RP*), comme c'est le cas pour les protocoles CBT, PIM-SM et YAM. On nomme ce type d'approche *CBF* pour « *Core Based Forwarding* ». Dans ce type d'approche il faut prendre en compte :

Le choix du core au niveau réseau. Si cette tâche importante de sélection de la racine de l'arbre partagé qui est déployé pour mettre en œuvre un protocole de multicast incombe aux hôtes, l'interface multicast entre les hôtes et le réseau va être dépendante du type de protocole multicast mis en œuvre dans le réseau, ce qui n'est pas souhaitable. (*e.g.*, dans les réseaux utilisant une approche de type CBF, une requête d'adhésion devra nécessairement inclure l'adresse du core correspondant au groupe de multicast alors que cette information ne sera pas requise pour d'autres protocoles). Une sélection automatique du core au niveau réseau est préférable à un choix effectué au niveau hôte [HJ98].

La défaillance du core. Un point faible des approches de type CBF réside dans les défaillances potentielles du core. Il est donc nécessaire de mettre en œuvre des mécanismes efficaces pour assigner de nouveaux cores aux groupes ayant subit la perte des leurs ;

La migration du core. Au cours de la durée de vie d'un groupe de multicast, la géométrie des membres est susceptible d'évoluer de façon dynamique (nouvelles adhésions, nouveaux retraits) et les ressources disponibles au sein du réseau sont aussi susceptibles d'évoluer. Le rôle du processus de *migration de core* est d'identifier un nouveau core au sein du réseau qui engendrera de meilleures performances, *i.e.*, dont l'arbre induit par l'état du réseau et la configuration des membres offre un comportement nettement supérieur à l'arbre du core actuel.

Notre point de vue est que la complexité induite par la gestion du core peut être grandement réduite dans certains réseaux si le processus de gestion arrive à tirer parti des informations et des services offerts par le protocole de routage sous-jacent. La section 3.4 présente un protocole nommé *LCM* qui définit une méthode de gestion du core adaptée aux réseaux ayant un protocole de routage à état de lien (*Link State Routing (LSR)*). La section 3.5 présente l'impact de la sélection du core sur les performances du protocole de multicast et propose une heuristique simple et efficace. Finalement, la section 3.6 présente la mise en œuvre d'un protocole de multicast intégrant certaines des fonctionnalités mentionnées ci-dessus au moyen de la technologie active.

3.2 Applications multi-parties

Le terme de *communication multi-parties* (*multiparty communication*), aussi nommée *communication de groupe* (*group communication* [Pow96]), fait référence à un large spectre d'applications incluant les interactions hommes/hommes, les simulations interactives distribuées, la gestion et la supervision d'informations distribuées et la distribution efficace d'informations. Une telle diversité d'applications engendre des besoins et des attentes très différents en ce qui concerne les services fournis par le réseau sous-jacent. Bien que dans ce manuscrit, notre objectif soit plus de nous concentrer sur le support au sein du réseau pour les communications de groupe, nous allons brièvement recenser les applications et les services qui peuvent être mis en œuvre au-dessus d'un tel service réseau.

3.2.1 Interaction hommes/hommes.

Cette classe d'application rassemble les individus pour lesquels il est difficile ou coûteux de se rencontrer de visu mais qui doivent néanmoins effectuer des travaux de façon coopérative ou collaborative. L'exemple type est la vidéo-conférence qui permet à un groupe de personnes de communiquer visuellement et verbalement au travers d'un réseau [Cla92, AE92]. Un type particulier de télé-conférence est nommé *computer telephony*, et emploie des ordinateurs et des réseaux de transport de données à la place des réseaux téléphoniques commutés pour transmettre la voix en temps réel [Ude94]. L'émergence des réseaux large bande a fortement contribué à l'intégration partielle sur un réseau physique des services de base de transfert de voix et de données. Les applications de téléconférence ne nécessitent pas nécessairement des flux multimédia ; des applications de téléconférence basées sur un mode textuel, aussi nommées *chat rooms* connaissent un réel engouement au sein de l'Internet [OR93]. À l'opposé, les environnements de travail collaboratif (*Computer Supported Cooperative Workspace*) permettent à différents acteurs se

trouvant géographiquement éloignés mais possédant chacun des domaines d'expertise variés, de réaliser des tâches complexes et de manipuler des équipements sophistiqués de façon distante [HRC92, RSVW94, LIN01].

On peut essayer de caractériser les exigences en terme de fiabilité du multicast comme support à cette classe d'applications d'interaction hommes/hommes. Les besoins sont en fait relativement faibles étant donné que l'on peut supporter occasionnellement la perte de certaines données. En effet, l'interaction n'intervenant pas entre machines, c'est un être humain qui va recevoir et interpréter les données : un caractère manquant dans un talk/chat en mode texte sera plutôt « interprété » comme une coquille que comme une erreur de transmission. De même, dans les applications dites multimédia, un certain taux de perte (perte de quelques pixels, de quelques frames audio/vidéo) est supportable et la conversation peut continuer alors qu'une gigue ou un délai trop important peut être ressenti comme perturbant [HW99]. Beaucoup d'applications de cette première catégorie emploient un multicast de type *best effort*, qui ne garantit aucune QoS, ni la livraison de toutes les données à tous les récepteurs, ni la réservation d'aucune ressource réseau au préalable.

3.2.2 Simulations interactives distribuées.

Dans une application de type simulation interactive distribuée (SID) (*Distributed Interactive Simulation (DIS)*) un environnement virtuel est simulé de façon collective par un ensemble d'hôtes disséminés au sein d'un réseau [CLD+99]. Comme exemple classique d'environnement virtuel, nous pouvons citer la mise en œuvre d'un centre commercial virtuel, les mondes virtuels pour des jeux à la « Doom » [BL01], un champ de bataille virtuel pour l'exercice de forces armées[1]. Dans ces environnements virtuels, un certain nombre d'objets sont passifs et statiques (arbres ou lacs d'un parc) alors que d'autres sont actifs et réagissent à des stimuli ou sont en mesure de se mouvoir par eux-mêmes (personnes se promenant dans le parc). Certains objets peuvent être calculés, gérés et simulés par un ordinateur (forces ennemies en présence sur le champ de bataille) alors que d'autres sont contrôlés par des êtres humains (pilote d'avion s'entraînant sur le simulateur). En général, les objets présents dans un environnement virtuel doivent interagir entre eux en temps réel. Pour ces raisons, les informations telles que la position actuelle, le mouvement et les actions en cours de chaque objet doivent être diffusées à tout hôte participant de la façon la plus rapide possible. Les réseaux offrant un service de multicast peuvent donc être employés pour augmenter les performances de ce type d'application [GD98].

Une application de type simulation interactive distribuée se caractérise le plus souvent par sa taille (le nombre de participants d'un environnement virtuel pouvant aller d'une petite douzaine à plusieurs milliers) et par le type de réseau employé (d'un LAN à un WAN). La taille et la distribution géographique des participants soulèvent le problème de l'extensibilité suivant le support de communication de groupe disponible. De plus, une application de type simulation interactive distribuée nécessite la mise en œuvre d'un *multicast fiable sélectif* (*selective reliable multicast*) [HSC95]. En effet, considérons qu'un utilisateur X perde la position de son adversaire Y suite à la perte de la séquence des trois derniers paquets indiquant la position de ce dernier. La sémantique conventionnelle de la notion de fiabilité voudrait que X demande la retransmission

[1] certainement l'exemple type où le virtuel est beaucoup moins onéreux que le grandeur nature !

de l'ensemble de la séquence perdue alors que, dans notre cas, X n'est en fait intéressé que par la dernière position connue de Y. Un multicast fiable sélectif doit donc garantir le caractère récent des informations d'état des objets maintenus par les hôtes sans garantir néanmoins une diffusion fiable de tous ces états [HSC95].

3.2.3 Gestion et supervision d'informations distribuées.

Les solutions mono-serveur traditionnelles que l'on trouve dans les domaines de la gestion de l'information (système de fichiers, bases de données) laissent de plus en plus la place à des solutions distribuées pour des raisons d'extensibilité et de tolérance aux pannes (système de fichier CODA [Bra98, SKK+90, BN99]). Ce genre d'application nécessite des opérations de multicast *atomiques*, *i.e.*, soit l'ensemble des destinataires reçoivent le message diffusé soit aucun d'entre eux ne le reçoit.

3.2.4 Distribution efficace d'informations

Cette catégorie englobe les applications qui nécessitent la dissémination d'un ensemble d'informations à une très large audience. Une caractéristique de ce genre d'application est la présence d'une seule source (ou d'un faible nombre) et d'un nombre potentiellement illimité de récepteurs.

Certains processus de diffusion d'information actuels peuvent bénéficier des nouvelles technologies. En effet, le processus classique pour diffuser des logiciels du domaine public se fait par la mise en œuvre d'un site FTP (*File Transfer Protocol*) [PR85]. L'apparition de l'HTTP (*Hyper-Text Transfer Protocol*) [FGF+99] et du web (*World Wide Web*) [BLC95] a largement supplanté l'utilisation de FTP pour la diffusion de logiciels mais le problème de la surcharge des sites demeure et a même empiré du fait du plus grand nombre d'utilisateurs et de la plus grande facilité/convivialité des interfaces des browsers. Une solution plus efficace serait que les distributeurs mettent en place un groupe de communication permettant aux membres de recevoir simultanément par multicast une copie du soft. Le protocole FDP *File Distribution Protocol* est un multicast conçu pour ce besoin précis [CK96] et nécessite l'emploi d'un protocole de multicast fiable [FJM+95, HBC95].

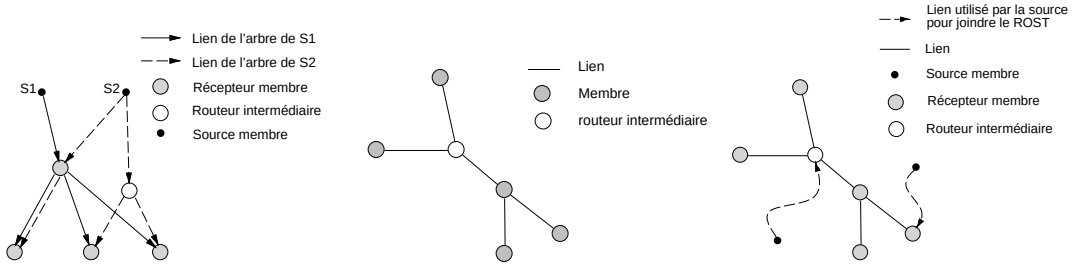
3.3 Communications multicast

Les opérations de multicast, qui délivrent des messages à une ou plusieurs destinations, apparaissent comme cruciales pour la mise en œuvre d'applications de communication multi-parties. Un *protocole de multicast* est donc un protocole mis en œuvre au sein du réseau, qui définit un ensemble de règles et de conventions permettant à un flux de données émis par une source d'être acheminé vers un ensemble de destinations. Cette section donne un aperçu des solutions proposées pour déployer le multicast dans l'Internet, *i.e.*, comme protocole de niveau 3. Nous débutons par un bref rappel sur les différentes topologies de routage.

3.3.1 Topologie pour le routage des communications multicast

Si la grande majorité des protocoles de multicast ne sont concernés que par la construction d'arbres de multicast individuels (un ensemble de liens connectant une source à un ensemble de destinations), nous pensons que l'on peut définir une structure de routage multicast plus générale que l'on nomme *topologie de multicast*. On définit trois principaux types de topologie multicast :

1. *Arbre enraciné à la source (Source Rooted Trees (SRT))*. Ce type de connexion multicast comprend généralement une forêt [Ber83] d'arbres, tous construits pour une source de trafic spécifique (Voir la figure 3.1(a)). Ce type de topologie est adapté aux applications ayant un faible nombre de sources et un nombre potentiellement élevé de récepteurs (*e.g.*, enseignement à distance). Ce type de topologie est assez facile à construire et est supporté par la majorité des protocoles multicast existants. Les topologies de multicast basées sur des arbres enracinés à la source sont néanmoins coûteuses à maintenir : un nouvel arbre devant être mis en œuvre pour chaque nouvelle source et l'arrivée (ou le départ) d'un nouveau récepteur entraînant l'extension et la modification de l'ensemble des arbres déjà existants. Les arbres enracinés à la source sont supportés par les protocoles DVMRP, MOSPF et PIM. Notons que les circuits virtuels permettant de mettre en œuvre du multicast dans les réseaux ATM supportent aussi les arbres enracinés à la source.
2. *Arbre partagé symétrique (Symmetric Shared Trees (SST))*. Un seul arbre couvrant l'ensemble des membres est déployé (Voir la figure 3.1(b)). Ce type de topologie est adapté aux applications pour lesquelles chaque membre est à la fois récepteur et émetteur (*e.g.*, cas d'une téléconférence). Ce type de topologie utilise beaucoup moins de ressources qu'une forêt d'arbres enracinés en chaque source.
3. *Arbre partagé en réception (Receiver-Only Shared Tree (ROST))*. Un seul arbre couvrant l'ensemble des récepteurs est déployé tandis que les émetteurs utilisent un chemin unicast unidirectionnel pour joindre l'un des nœuds de l'arbre (Voir la figure 3.1(c)). Ce type de topologie est adapté aux applications pour lesquelles la distinction entre récepteur et émetteur est importante. (*e.g.*, dans un système de serveurs dupliqués connectés par un arbre partagé en réception, un client ne voit qu'une seule entité et l'ajout d'un nouveau serveur ne perturbe d'aucune manière les communications client/serveur en cours).



(a) : deux SRTs (b) : un SST (c) : un ROST

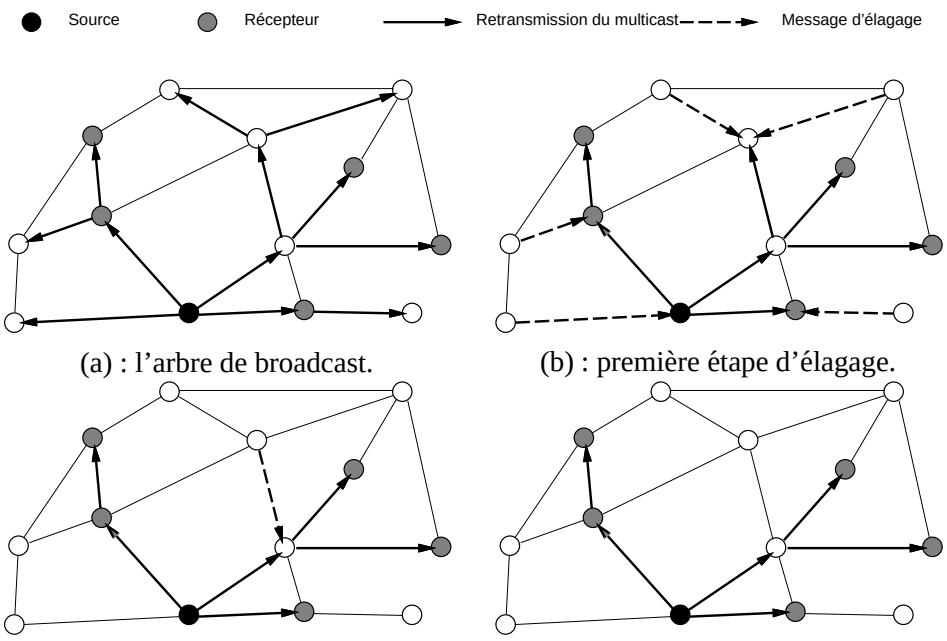
FIG. 3.1 – Trois différents types de topologie multicast.

Une fois le choix du type de topologie multicast effectué, il reste le choix de l'algorithme à mettre en œuvre pour construire la topologie souhaitée. Il existe de nombreux moyens de construire chaque type de topologie cité précédemment. En effet, pour chaque type de topologie multicast, un éventail très varié de critères d'optimisation est disponible : borne sur les délais de transmission, nombre de ressources réseau utilisées, taux de pertes ou tout autre paramètre comme nous le verrons plus en détail dans la section 3.5.2. On peut déjà citer un algorithme présenté par Zhu [ZPGLA95] qui optimise le coût en présence de contraintes sur les délais. Bauer [BV95] a examiné le problème de la construction d'arbres multicast en prenant en compte des contraintes sur le degré des nœuds. Waxam [IW91, Wax93] aborde le problème des arbres multicast dynamiques pour lesquels les mises à jour des appartenances au groupe se font une par une. Cette incrémentalité a aussi été étudiée par A. IRLANDE, J.-C. KONIG et C. LAFOREST [IKL00]. Pour un survey très complet sur les méthodes permettant de construire des arbres de STEINER, le lecteur peut consulter les articles de référence suivants [GM93, Win87].

3.3.2 Protocoles de multicast dans l'Internet

L'Internet n'est pas un réseau orienté connexion. Quand un émetteur E désire envoyer un datagramme à un un destinataire D, il n'a pas besoin a priori de contacter D avant d'effectuer sa transmission. Une fois envoyé, et si E et D ne partagent pas le même médium de communication, le datagramme est routé dans l'Internet en fonction de l'adresse IP du destinataire. L'Internet a étendu ce modèle basique de transmission de datagramme point-à-point au multicast [Dee89, DC90]. Un datagramme ayant comme adresse destination une adresse multicast [Dee89, Dee91] (*i.e.*, 224.0.0.0 à 239.255.255.255 pour IPv4) est appelé datagramme multicast et doit être retransmis et routé vers tous les hôtes qui se sont déclarés intéressés par cette adresse. Pour plus de détails sur la façon dont un hôte se déclare intéressé par une adresse multicast (un groupe multicast est identifié par son adresse multicast) et comment il s'enregistre à ce groupe auprès de son routeur multipoint le plus proche, voir IGMP [Dee89, Fen97]. Pour présenter des arguments à notre discussion sur le multicast IP, nous allons présenter quelques protocoles de routage multicast qui ont été proposés dans la littérature et/ou à l'IETF. Notre objectif n'est ni d'être exhaustif, ni de détailler chaque protocole dans ses moindres spécificités. On se référera soit au RFC de chaque protocole pour plus de détails, soit aux articles [DDC97, RCV$^+$00] pour un survey plus détaillé et plus complet. Dans cette discussion, le terme de groupe de communication ou groupe multicast fait référence à un ensemble d'hôtes qui sont en écoute sur une adresse multicast IP. Suivant cette sémantique, les groupes de multicast au sein de l'Internet sont des groupes de *récepteurs*.

Le protocole Distance Vector Multicast Routing Protocol (DVMRP). Étant donnée une adresse de multicast M, DVMRP construit un SRT (arbre enraciné à la source) pour chaque source présente dans le groupe M. Cette construction s'effectue par un processus d'inondation (*flooding*) dans tout le réseau et par un processus d'élagage (*pruning*). Initialement, le flux multicast est diffusé dans tout le réseau par un processus de diffusion de type *reverse path forwarding* [DM78] : un routeur R recevant un paquet de multicast émis par S et à destination de M vérifie si le paquet P lui est parvenu par le lien ℓ qui constitue le premier saut d'un plus court chemin de R à S ; si tel est le cas, R retransmet le paquet P à tous ses voisins (*i.e.*, sur tous ses liens de sortie) excepté sur le lien ℓ par lequel P est arrivé. Dans le même temps, les routeurs qui ne sont pas intéressés



(a) : l'arbre de broadcast.

(b) : première étape d'élagage.

(c) : seconde étape d'élagage.

(d) : l'arbre multicast résultant.

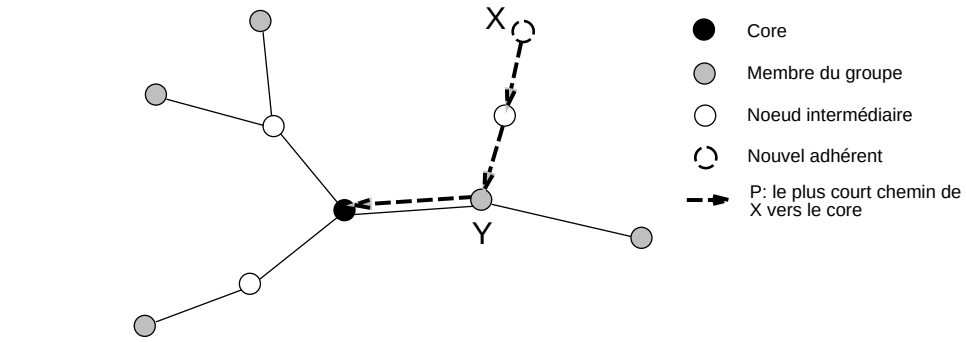
FIG. 3.2 – Opérations successives réalisées dans DVMRP. Sur la figure 3.2(a) la source multicast emploie un arbre de diffusion pour joindre les cinq membres. Sur la figure 3.2(b) les cinq feuilles de l'arbre non membre envoient un message d'élagage représenté en traits pointillés. Sur la figure 3.2(c), un noeud intermédiaire de l'arbre reçoit un message d'élagage de la part de tous ses fils et envoie à son tour un message d'élagage en amont. L'arbre résultant du processus d'élagage est présenté sur la figure 3.2(d).

par le trafic multicast du groupe M envoient des messages d'élagage en amont, *i.e.*, un saut en direction de la source S. Si un router en amont s'aperçoit que tous ses liens le reliant à ses routeurs voisins en aval ont été « élagués », il émet à son tour un message d'élagage en amont sauf s'il est lui-même membre du groupe M ; ce processus d'élagage se répète jusqu'à ce que tous les routeurs impliqués dans le processus de retransmission de S vers M soient membres de M ou aient des voisins en aval qui sont membres de M. In fine, on obtient un SRT enraciné en S qui couvre l'ensemble des routeurs appartenant à M. Ce processus est illustré sur la figure 3.2.

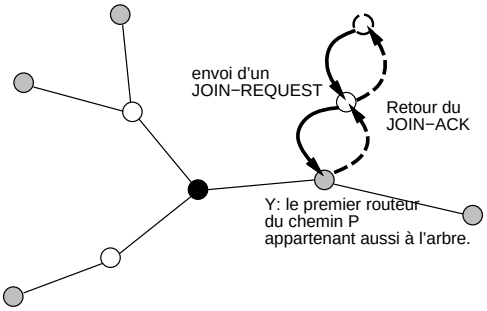
Une des caractéristiques de DVMRP est que l'information sur l'appartenance à un groupe de multicast n'est pas disséminée dans le réseau, mais est découverte lors de la construction de l'arbre par un mécanisme de « *non appartenance* » (message d'élagage). Néanmoins, l'inconvénient majeur de ce mécanisme est que l'on ne peut pas intégrer aisément des changements d'appartenance à un groupe au sein d'un SRT déjà existant. Pour remédier à ce problème, les SRTs doivent périodiquement être régénérés complètement [WPD88, DC90]. Cette approche induit des délais supplémentaires dans la prise en compte d'un nouveau membre : un nouveau membre ne recevant pas de paquet d'un flux multicast avant la fin de la prochaine phase de régénération d'arbre. De plus, ces phases de régénération induisent un trafic superflu durant les phases stables. Notons que l'approche arbre partagé n'est pas prise en compte dans DVMRP.

Le protocole Core-Based Tree (CBT). À l'inverse de DVMRP, le protocole CBT [Bal97] construit un arbre partagé pour chaque groupe de multicast. Un routeur spécifique C, nommé « *core* », est assigné à chaque groupe M. Un membre s'enregistre auprès du groupe en envoyant à destination du core C un message `JOIN-REQUEST` ; ce message sera intercepté par le premier nœud faisant déjà partie de l'arbre (au pire le core). Une branche est alors créée entre l'arbre existant et le nouveau membre par l'envoi d'un message `JOIN-ACK` qui suit le chemin inverse emprunté par le message `JOIN-REQUEST`. Un membre quitte le groupe (*i.e.*, se détache de l'arbre) en envoyant un message `QUIT-REQUEST` vers son père dans l'arbre qui, à son tour, quitte le groupe s'il n'est pas membre du groupe et s'il s'agissait de son dernier fils. La figure 3.3 illustre ce mécanisme de construction.

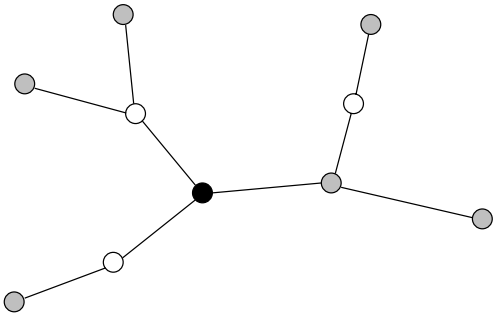
Le protocole CBT prend en compte les évènements hostiles qui peuvent se présenter dans le réseau comme la panne d'un lien ou d'un routeur en envoyant de façon périodique un message `CBT-ECHO-REQUEST` en amont. Si aucun message `CBT-ECHO-REPLY` n'est entendu en retour, le membre se doit de rejoindre le groupe en trouvant un nouveau chemin permettant d'atteindre le core C. Comparé au protocole DVMRP, le protocole CBT gère l'ajout ou la suppression d'un membre de façon orientée événement (*event-driven*) mais a toujours recours à un processus d'envoi périodique de messages pour gérer les changements de topologie, ce qui induit un délai dans la prise en compte des modifications de topologie. Une telle approche hybride peut se révéler efficace pour certaines applications mais inappropriée pour certaines applications critiques nécessitant une totale transparence vis-à-vis des changements de topologie. Un autre inconvénient du protocole CBT est sa rigidité en terme de topologie multicast puisqu'il ne prend pas en compte les topologies de type SRT. De plus, le fait qu'un paquet de multicast soit nécessairement envoyé au core avant d'être retransmis et diffusé au sein de l'arbre engendre des étapes supplémentaires pas toujours justifiées. La figure 3.4 illustre le sur-coût engendré par cette restriction lorsque les membres considérés dans la figure 3.3(c) jouent aussi le rôle de source pour le groupe.



(a) : plus court chemin P du nouveau membre X au core.

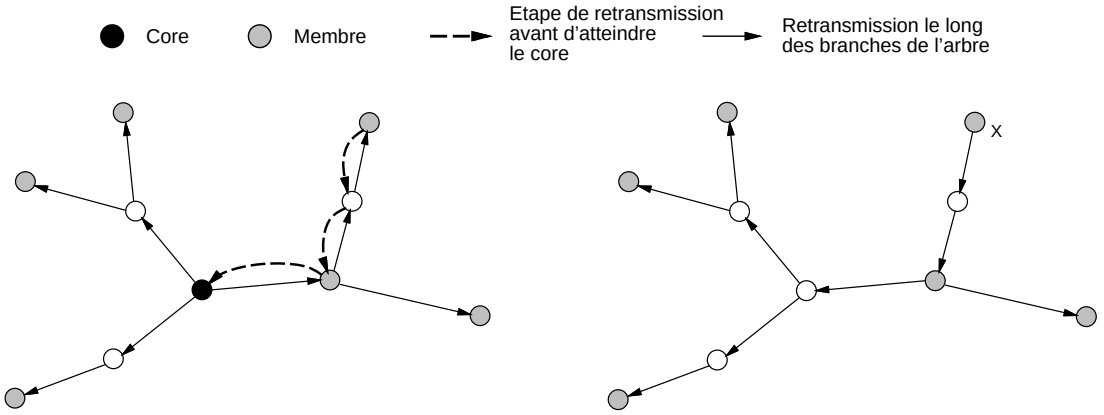


(b) : émission du message `JOIN-ACK`.



(c) : l'arbre résultant.

FIG. 3.3 – Exemple d'adhésion au sein du protocole CBT. La figure 3.3(a) illustre le plus court chemin allant du nouveau membre X au core. C'est le premier routeur Y déjà membre de l'arbre qui va prendre en charge le message `JOIN-ACK` et retourner en réponse un message `JOIN-ACK` comme l'illustre la figure 3.3(b). Le résultat de cette opération d'adhésion est illustré sur la figure 3.3(c).

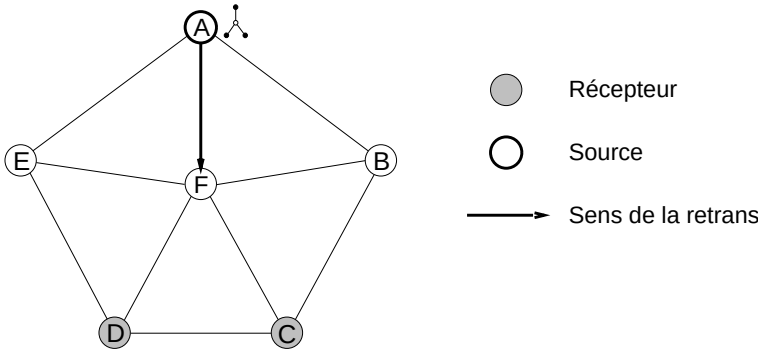


(a) : utilisation du protocle CBT.

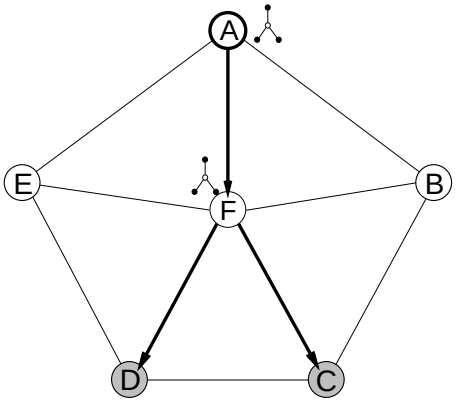
(b) : utilisation d'un SST.

FIG. 3.4 – Comparaison de la diffusion d'un message multicast entre un arbre de type CBT et un SST. La figure 3.4(a) illustre le cas CBT et la figure 3.4(b) illustre le cas SST. Le protocole CBT engendre des étapes de communication suplémentaires illustrées en traits pointillés sur la figure 3.4(a).

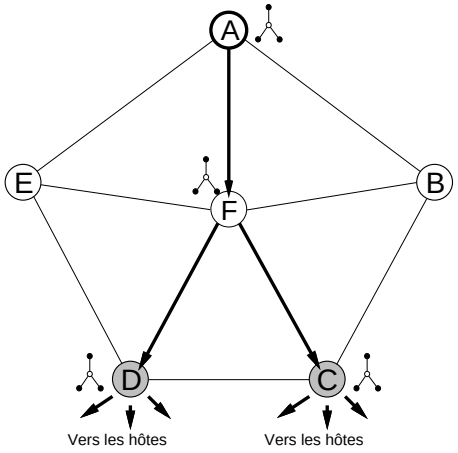
Le protocole MOSPF. Le protocole MOSPF [Moy94a] est une extension d'un protocole de routage IP à état de lien nommé OSPF [Moy94b]. MOSPF diffuse les identités (adresse IP) des membres de chaque groupe multicast au moyen de LSA (*Link State Advertisment*) de type `GROUP-MEMBERSHIP` de telle sorte que tous les routeurs tiennent à jour les listes complètes des membres de chaque groupe de multicast (*i.e.*, de chaque adresse multicast active). La mise en œuvre du canal de distribution des données d'un groupe de multicast est établie lors du premier envoi d'un datagramme à destination de l'adresse multicast de ce groupe. À la réception du premier datagramme par une source S et à destination d'une adresse multicast M, un routeur consulte sa base de données locale pour récupérer la liste des membres du groupe M et il calcule un arbre T de plus court chemin enraciné en S (plus précisément le routeur multicast qui prend en charge S) qui recouvre l'ensemble des membres de M (plus précisément, l'ensemble des routeurs qui prennent en charge les membres de M). À partir de ce calcul d'arbre T, le routeur construit une table de routage multicast lui permettant de savoir sur quels liens il doit retransmettre les paquets en provenance de S qui sont à destination de M. Le fait de retransmettre des paquets va déclencher d'autres calculs et d'autres mises à jour de table de routage multicast au sein des routeurs en aval. La figure 3.5 illustre la construction d'arbres MOSPF. Comme le montre l'exemple précédent, le protocole MOSPF induit des calculs redondants (les mêmes calculs d'arbres sont effectués au sein de chaque routeur qui prend part à l'arbre de multicast). Cet inconvénient est accentué du fait que MOSPF ne supporte que les SRT, ce qui induit que les calculs se font « pour chaque source de chaque groupe » et non pas pour « chaque groupe » uniquement. De plus, à chaque changement de topologie ou de configuration de la liste des membres indiqué par un LSA spécifique, une reconstruction et donc un recalcul de l'arbre doivent être effectués lorsqu'un datagramme arrive.



(a) : calcul de l'arbre et retransmission en A.



(b) : calcul de l'arbre et retransmission en F.



(c) : calcul de l'arbre et retransmission en C et D.

FIG. 3.5 – Étapes du protocole MOSPF. Un hôte qui est pris en charge par le routeur A envoie un datagramme multicast au groupe multicast dont les membres sont des hôtes pris en charge par les routeurs C et D. Comme l'illustre la figure 3.5(a), le routeur A calcule un arbre de plus courts chemins enraciné en A qui couvre C et D. Ce calcul est rendu possible par la connaissance de la toplogie grâce au protocole OSPF sous-jacent de type LSR et par la connaissance des routeurs membres (ici C et D) grâce à MOSPF. L'arbre résultant induit que A doit retransmettre ses datagrammes multicast vers F qui à son tour va calculer le même arbre et en déduire qu'il doit retransmettre aux routeurs C et D comme le montre la figure 3.5(b). Quand C et D recoivent à leur tour des datagrammes multicast, il vont aussi exécuter le même calcul et en déduire qu'ils doivent retransmettre le flux muticast aux hôtes qu'ils prennent en charge.

Le protocole PIM-SM. Les protocoles MOSPF et DVMRP impliquent que l'ensemble des routeurs d'un domaine soit partie prenante d'une session multicast. En effet, nous avons vu que dans le cas de MOSPF, tous les routeurs reçoivent et enregistrent la liste des membres de chaque groupe multicast tandis que pour DVMRP, c'est le flux multicast de chaque source qui est diffusé périodiquement dans l'ensemble du réseau. Le sur-coût de cet engagement à l'échelle de tout un réseau peut se justifier uniquement si une très large majorité des hôtes du réseau est intéressée par le multicast. On parle le plus souvent de « *mode dense* » (*dense mode multicast*) [DEF$^+$96, DEF$^+$99]. À l'inverse, le « *mode épars* » (*sparse mode*) caractérise les cas où seule une faible fraction des hôtes est intéressée, ce qui rend la surcharge engendrée à l'échelle de tout le réseau beaucoup trop coûteuse et injustifiable.

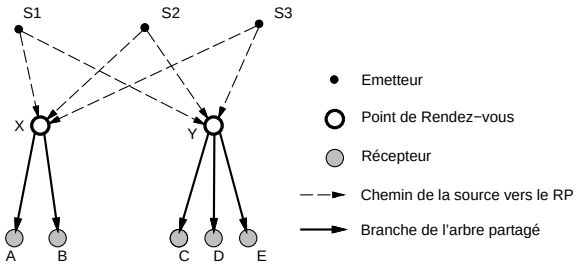


FIG. 3.6 – Arbres partagés construits par le protocole PIM. Deux arbres partagés sont construits pour une adresse multicast qui a trois sources.

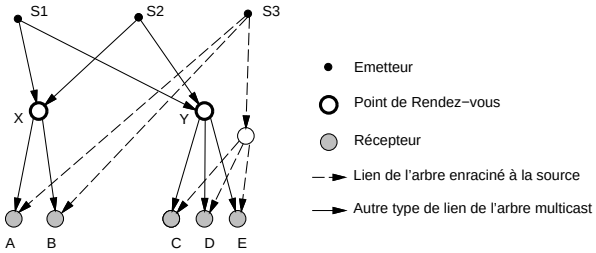


FIG. 3.7 – Résultat du changement de topologie intervenu pour le sommet $S3$.

Le protocole PIM supporte les deux modes (dense et épars) en proposant deux protocoles pour chaque mode. Pour le mode dense, c'est en fait DVMRP qui est employé (MOSPF n'a pas été retenu du fait qu'il dépend trop fortement d'un protocole de routage unicast à état de lien). En mode épars, le protocole PIM construit initialement des arbres partagés en réception (ROST) et la mise en œuvre d'arbres enracinés à la source (SRT) s'effectue de façon sélective et à l'initiative de certaines sources au cours de la session multicast. Une région d'un réseau (LAN, zone de routage, système autonome (AS)) désirant faire partie d'un multicast en mode épars doit configurer l'un de ses routeurs en « *Point de Rendez-vous* » nommé *RP* (pour *Rendez-vous Point*). Ce RP joue un rôle assez similaire à celui du core dans le protocole CBT. Les membres de cette région envoient des messages `RP-JOIN` qui permettent de construire un ROST enraciné au routeur RP. Si N

régions sont intéressées par le groupe multicast, N RP vont être associés à l'adresse multicast correspondante et N arbres partagés vont être déployés. La source d'un datagramme IP destiné à cette adresse multicast doit envoyer une copie à chacun des N RP. La figure 3.6 illustre cette construction.

Le protocole PIM-SM crée des SRT par le biais d'un processus de transition de topologie qui opère dans un mode orienté contrôle de donnée. Lorsque les membres d'un groupe observent un fort trafic en provenance d'une source S, ils déterminent s'il ne serait pas justifié de mettre en œuvre un canal de distribution propre à S et ils émettent vers S un message `SOURCE-JOIN` qui aura comme conséquence de construire un SRT enraciné en S. La figure 3.7 illustre ce principe en reprenant l'exemple de la figure 3.6. L'approche prônée par PIM-SM permettant de supporter différents types de topologie de multicast est élégante.

Le point important qu'il reste à présenter est la façon dont sont sélectionnés les RP. PIM met en œuvre un routeur spécifique nommé *Bootstrap Router (BSR)* qui permet aux routeurs d'une région d'un réseau d'obtenir l'adresse d'un RP. Ce routeur BSR permet de résoudre les problèmes suivants :

- Obtention des informations sur les RPs.
- Élection du BSR.
- Gestion du partitionnement de la région du réseau.
- Gestion de la défaillance d'un RP.

Au sein d'une région, un ensemble de routeurs est configuré comme état des RP potentiels (*RP-set*) et leur identité est diffusée dans les messages du BSR. De même, un ensemble de routeurs est configuré comme étant des BSR candidats et un mécanisme d'élection est employé pour choisir le BSR de la région. Une fonction de hachage est employée pour faire correspondre l'adresse d'un groupe multicast à l'un des RP candidats. Notons que le choix du BSR (effectué par élection entre les BSR candidats) et des RP ne tient pas compte de la topologie du réseau, ce qui peut fortement nuire aux performances, l'arbre de diffusion n'étant pas optimisé. De plus, le choix des BSR candidats et des RP candidats doit se faire manuellement, ce qui empêche son déploiement à grande échelle.

Le protocole YAM. L'un des objectifs du protocole YAM [CC97, CC99] (*Yet Another Multicast*) est de construire des arbres partagés en offrant plusieurs choix aux membres pour s'accrocher à cet arbre. Le protocole YAM reprend la division intra- et inter-domaine du routage point-à-point et définit deux modes de raccordement. La figure 3.8 illustre la création d'une branche intra-domaine.

La figure 3.9 illustre la création d'une branche inter-domaine en utilisant une approche nommée « *spanning join* » qui construit, au moyen d'une diffusion de type *Reverse Path Forwarding* [DM78] (voir la description de DVMRP page 38), un SRT enraciné au nouveau membre vers les autres nœuds de l'arbre. En résumé, YAM utilise un SRT dans le plan de contrôle et un ROST dans le plan des données. Notons que si aucune racine n'existe pour un groupe multicast donné, le routeur contacté par le premier récepteur devient d'office racine de l'arbre partagé.

Les protocoles multicast fiables. Nous avons vu dans la première partie que certaines applications nécessitent la mise en œuvre de protocoles de multicast fiables. Généralement, ces protocoles

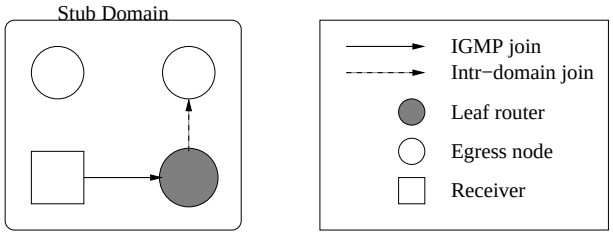


FIG. 3.8 – Création d'une branche intradomaine par le protocole YAM. Un récepteur contacte son routeur feuille via IGMP. Celuici utilise un DNS pour obtenir la correspondance Egress Node (EN)–groupe multipoint. Il contacte alors l'EN par une requête join. Si le récepteur quitte le groupe, alors la branche est coupée. Un émetteur non membre dans le domaine envoie des données à l'adresse du groupe multipoint. Dans ce cas, le routeur feuille recevant ce flux utilise un DNS pour obtenir la correspondance EN–groupe multipoint. Il contacte alors l'EN comme cidessus. Cependant un temporisateur est déclenché suivant sa connexion. Si l'émetteur arrête d'envoyer des données, alors la branche est coupée (schéma emprunté à [Mag02]).

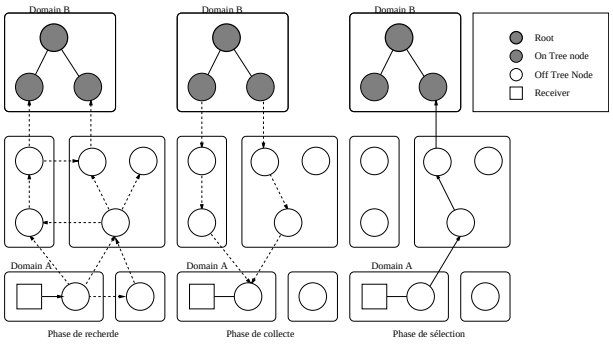


FIG. 3.9 – Création d'une branche interdomaine par le protocole YAM (schéma emprunté à [Mag02]).

mettent en œuvre une hiérarchie de routeurs (nommés « *agent retransmetteur* » dans [Mag02]) qui sont chargés de retransmettre les datagrammes multicast perdus. Nous ne traiterons pas de ces protocoles dans ce manuscrit et le lecteur pourra se référer à la thèse de Damien MAGONI.

3.4 Le protocole LCM

3.4.1 Introduction

Nous montrons que pour une certaine classe de réseaux, la complexité induite par la gestion du core peut être grandement réduite si le processus de gestion arrive à tirer parti des informations et des services offerts par le protocole de routage sous-jacent. Ces travaux ont été réalisés en collaboration avec Y. HUANG et P. MCKINLEY. Plus spécifiquement, nous proposons un protocole réseau de gestion de core employé dans les arbres de type CBF (CBT, PIM, YAM). Ce protocole est adapté aux réseaux utilisant un protocole de routage à état de lien dont la caractéristique primordiale est que la topologie du réseau est maintenue au sein de chaque routeur du réseau. Les protocoles de routage de type LSR sont disponibles et déployés à la fois dans l'Internet avec le protocole *Open Shortest Path First (OSPF)* [Moy94a] et dans les réseaux ATM avec le standard *Private Network-to-Network Interface (PNNI)* [ATM96].

Notre protocole, nommé *LSR-based Core Managment (LCM)* [HFM98] fait usage d'un serveur centralisé nommé *Core-Binding Server (*CBS*)* afin de maintenir les correspondances existantes entre l'ensemble des groupes de multicast actifs et le core qui leur est associé. Afin de traiter les problèmes identifiés et énoncés ci-dessus inhérents à la gestion de core, le protocole LSR s'appuie fortement sur les services offerts par le routage LSR sous-jacent et utilise les informations relatives à l'identité des routeurs, leur statut et à la topologie du réseau. Afin de pallier la défaillance du CBS lui-même, nous avons recours à un processus d'élection robuste permettant de sélectionner de façon dynamique un CBS, ce qui permet de faire face non seulement à des défaillances du réseau mais aussi à un scénario pessimiste de partitionnement.

La contribution majeure de ce travail est de montrer qu'un « *seul* » protocole de gestion de core peut être mis en œuvre de façon efficace et relativement « *simple* » au dessus d'un protocole de routage à état de lien. Nous tenons à préciser que les protocoles de type LSR ne sont pas destinés, a priori, à être déployés directement dans de grands réseaux (l'Internet tout entier) mais dans des domaines de routage (nommé *Autonomous System* dans la terminologie Internet) qui comportent une centaine de routeurs (*e.g.*, pour PIM-SM, un RP est configuré dans chaque domaine qui contient au moins un membre du groupe). La gestion du core sous ces hypothèses est « locale ». Nous ne traitons pas la gestion des protocoles multicast construisant des arbres partagés multicast inter-domaine qui autorisent l'utilisation de tout type de routage multicast au sein de chaque domaine individuel [KRT+98, Tha02]. Un travail reprenant la notion de serveur de gestion [LLS99] et proposant une structure hiérarchique pour améliorer le passage à l'échelle a été proposé. Leur architecture nommée *Core-Manager based Multicast Routing (CMMR)* se base sur un gestionnaire centralisé prédéfini (le centre du réseau) et sur une sélection a priori de l'ensemble des cores candidats. De plus, cette architecture ne tient pas compte du dynamisme du groupe et ne propose pas de migration de core.

3.4.2 Architecture du protocole LCM

Le protocole LCM utilise un serveur centralisé CBS qui maintient la liste des correspondances suivantes :

$$\mathcal{C} \quad = \quad \{\mathrm{Core}(m) \mid m \text{ est une adresse multicast active}\}$$

Quand un hôte désire joindre un groupe multicast m, le routeur local x qui le prend en charge (*designated routeur (DR)*) envoie un message `CORE-MAPPING(M)` au CBS. Si l'association $\mathrm{Core}(m)$ existe dans la liste $\mathcal{C}$ et est égale à l'adresse α, le CBS renvoie un message `CORE-ADRESSE(`α`)` à destination de x. Dans le cas contraire, le CBS choisit un core β pour le groupe m, ajoute l'association $\mathrm{Core}(m) = \beta$ dans la liste $\mathcal{C}$ avant de retourner le message `CORE-ADRESSE(`β`)` à destination de x. À la réception de l'association $\mathrm{Core}(m)$ demandée, le routeur x s'attache à l'arbre multicast du groupe m en suivant la procédure définie par le protocole CBF mis en œuvre.

Élection du CBS**.** L'identité du CBS n'est pas configurée statiquement mais est la résultante d'un protocole d'élection de leader. Chaque routeur x maintient une association $\mathrm{CBS}(x)$ dont la valeur est égale à l'identité id du routeur qui est perçu par x comme étant le CBS. Lorsque le routeur x découvre que la valeur de $\mathrm{CBS}(x)$ est nulle ou qu'il est déconnecté de $\mathrm{CBS}(x)$ du fait d'une défaillance de ce dernier ou d'un partitionnement du réseau, le routeur x déclenche une élection. Le protocole d'élection doit garantir un consensus sur l'identité du CBS. Si le réseau est partitionné en $s > 1$ segments $S_1, S_2, \ldots, S_s$ il doit y avoir un consensus sur chacun des segments S_i et le CBS_i élu au sein du segment S_i aura reçu le scrutin de tous les routeurs de ce segment et de lui seul. Différents protocoles d'élection de leader [ATM96, HM97, CHK$^+$95] satisfont aux critères mentionnés ci-dessus et LCM peut les utiliser indifféremment.

Défaillance du CBS**.** Lors d'une défaillance d'un CBS il ne faut pas seulement réélire un nouveau CBS mais il faut assurer que la liste $\mathcal{C}$ soit collectée et régénérée dans le nouveau CBS. Pour ce, chaque routeur x maintient une liste d'associations des groupes qui le désignent lui-même comme core :

$$\mathcal{C}_x \quad = \quad \{\mathrm{Core}(m) \mid m \text{ est une adresse multicast active et } \mathrm{Core}(m) = x\}$$

Cette liste est incluse dans le suffrage envoyé par x durant le processus d'élection. Puisque le CBS va recevoir l'ensemble des suffrages du réseau/segment, il est alors en mesure de collecter toutes les associations dans $\mathcal{C}$ sauf celles qui désignaient l'ancien CBS comme core et que l'on traite de la façon suivante : tout routeur x qui est membre d'un groupe m dont le core $\mathrm{Core}(m) = \mathrm{CBS}(x)$ doit vider son association $\mathrm{Core}(m)$ chaque fois que la valeur de $\mathrm{CBS}(x)$ change et il doit consulter le nouveau CBS pour obtenir une nouvelle association pour m.

Sélection du core initial. Quand le premier routeur membre d'un groupe de multicast m demande au CBS l'identité du core pour le groupe m, le groupe m devient actif et le CBS doit lui assigner un core. Étant donné qu'aucune information d'adhésion n'est encore disponible, les solutions sont restreintes (par exemple choisir un membre aléatoirement [CZD95]). LCM prône

de mettre en œuvre une variation nommée *first-member* du *random member* qui est la suivante : quand le CBS reçoit un message `CORE-MAPPING(M)` du routeur x et que $\mathrm{Core}(m) \notin \mathcal{C}$ alors il assigne $\mathrm{Core}(m) = x$.

Défaillance d'un core. Le CBS monitore tous les cores listés dans $\mathcal{C}$ au moyen des informations topologiques collectées par le protocole LSR sous-jacent. Quand un CBS perd la connectivité avec le core d'un groupe m, il sélectionne un autre routeur (aléatoirement par défaut) et diffuse cette nouvelle association dans tout le réseau en employant le mécanisme d'inondation utilisé par le routage LSR. Les informations sur la connectivité nécessaire à la détection de la perte d'un core et l'identité des routeurs nécessaires au choix aléatoire d'un nouveau core sont présentes dans le protocole LSR sous-jacent.

Migration d'un core. La migration de core est décrite plus en détail dans la section 3.5. L'initiative d'une migration est prise par l'actuel core α d'un groupe m. Pour ce, le core α envoie un message `CHANGE-CORE(`β`)` au CBS lui indiquant le changement. Ce dernier met à jour la liste $\mathcal{C}$ et diffuse la nouvelle association dans le réseau comme s'il s'agissait d'une défaillance de l'ancien core. Les membres de m initient l'envoi de nouveau `JOIN-REQUEST` vers le nouveau core pour construire un nouvel arbre. Nous verrons dans la section 3.6 différentes stratégies pour la mise en œuvre de la migration au moyen de la technologie active.

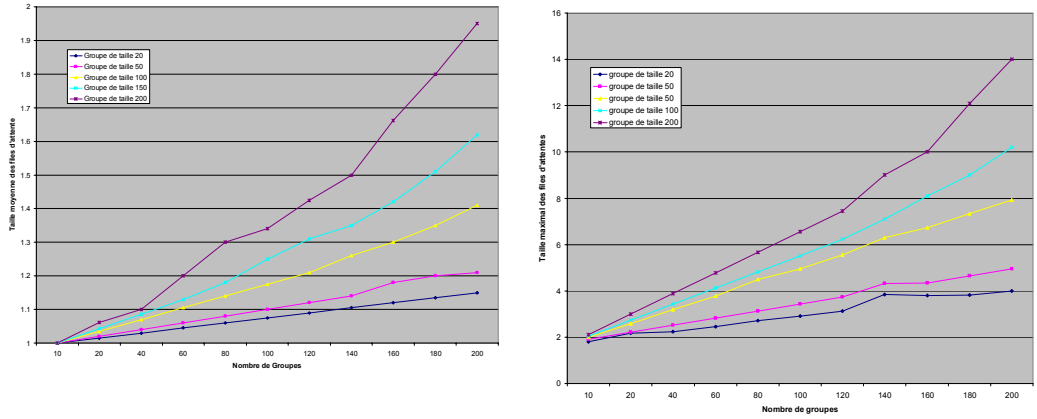
3.4.3 Évaluation de performance

Nous ne présentons ici que les résultats concernant la surcharge occasionnée par la mise en œuvre d'un serveur centralisé de gestion des associations core/groupe multicast. Les résultats spécifiques aux choix d'heuristiques pour la migration d'un core seront présentés dans la section 3.5.

Nos simulations ont pour but de tester les performances du CBS en situation de charge intense. Pour ce, nous avons supposé que K groupes de multicast de S membres sont créés en même temps au temps t_0. Les valeurs de K varient de 10 à 200, et celles de S varient de 20 à 200. Pour un groupe multicast donné, les temps d'arrivée des membres sont distribués selon une loi normale avec une moyenne de 0. Nous avons fixé l'écart type de telle sorte que 99% des temps d'arrivée se trouvent dans un intervalle d'une minute centré en 0 (*i.e.*, [-30 secondes, +30 secondes]). Le pire scénario produit 200 groupes de 200 membres qui sont tous créés en moins d'une minute, ce qui produit 40000 messages `CORE-MAPPING` au cours de ce laps de temps. Nous avons fixé à $700\mu sec$ le temps de traitement requis pour traiter une requête `CORE-MAPPING`, ce qui correspondait au temps de traitement observé sur différentes plate-formes d'un paquet IP/UDP. Notons que nous pouvons négliger le temps mis pour effectuer la recherche dans la liste qui est bien inférieur (en $O(\log S)$) au temps pris pour recevoir et émettre un paquet IP/UDP.

Les résultats de ces simulations sont présentés sur la figure 3.10. Comme nous pouvons le constater sur la figure 3.10(a), la taille moyenne des files d'attente est inférieure à 2, même pour les plus forts taux d'événements. Il est important de noter que la moyenne des files d'attente ne prend en compte que les périodes où le CBS est occupé (la plus petite valeur est donc 1). La figure 3.10(b) présente la taille maximum des files d'attente. Bien que la taille maximum des files d'attente puisse être entre 10 et 14 pour certains scénarios, il est important de noter que sous nos

hypothèses, 20 requêtes sont traitées en 14 milli-secondes, ce qui permet de dire que le CBS est capable de résister à la charge générée par les scénarios les plus « stressants ».



(a) : en moyenne. (b) : maximum.

FIG. 3.10 – Taille des files d'attente dans le CBS.

3.5 Migration de core

3.5.1 Introduction

Nous avons vu l'importance de mettre en œuvre un mécanisme de gestion de core et d'être en mesure de pouvoir choisir un « bon » core de façon dynamique selon un ou plusieurs critères (utilisation des ressources réseaux, délai de bout en bout, temps d'une adhésion, congestion du réseau).

Ce problème de la sélection du core se pose de multiples fois au cours de la vie d'un groupe multicast : à l'initialisation du groupe multicast, en cas de défaillance du core ou en cas de trop mauvaise performance du core actuel. Ceci qui implique de trouver un meilleur core et l'on parle alors de *migration* [DZ96]. La migration du core permet de s'adapter aux changements qui interviennent dans le réseau (éviter une zone trop congestionnée) et à la dynamique des membres du groupe (adhésions/retraits multiples). Une migration partielle ou totale de l'arbre multicast est aussi employée dans l'algorithme de reconfiguration [SMSRM99] qui monitore un indice de qualité de l'arbre pour chaque nouvelle adhésion ou départ d'un membre et déclenche une reconfiguration lorsqu'un seuil est dépassé. Le problème de la sélection du core est logiquement lié au problème de l'arbre de STEINER mais il est important de noter que, dans le cas des protocoles de routage multicast CBF, l'arbre obtenu est le plus souvent l'union des plus courts chemins des membres vers le core et que ces chemins sont fixés par la fonction de routage sous-jacente employée. Ainsi, la topologie de l'arbre est complètement déterminée par la position du core, la fonction de routage (métrique utilisée dans le réseau) et l'ordre des adhésions des membres.

Si la fonction de routage est une fonction qui donne les plus courts chemins et si tout sous-chemin est lui même un plus court chemin valide (au sens de la fonction de routage) alors l'arbre

construit ne dépend pas de l'ordre dans lequel les membres s'abonnent [CFGL02]. Trouver le meilleur arbre qui respecte ces conditions devient polynômial (il suffit de tester les N positions possibles dans le réseau et de prendre la meilleure). Un tel arbre n'est évidemment pas un arbre de Steiner mais il faut noter que l'arbre de Steiner n'est peut être pas accessible par le processus de création considéré. Si l'hypothèse sur les sous-routes est trop forte, on peut même la relâcher. À partir du graphe G et de la fonction de routage, on construit un graphe G' comme étant l'union des chemins des membres à un sommet donné du graphe qui sera le core. Ce graphe G' n'est pas forcément connexe mais la composante connexe à laquelle appartient le core contient tous les membres par définition. Il suffit alors de calculer un arbre de recouvrement de poids minimum sur cette composante et on obtient un arbre multicast de poids minimum qui satisfait les propriétés demandées. Pour trouver le meilleur arbre, il suffit de tester tous les noeuds du réseau comme étant des cores possibles. Encore une fois, cet algorithme polynômial ne donnera pas un arbre de Steiner mais un arbre qui est constructible par le processus décrit (union des chemins des membres au core) ce qui n'est pas forcément le cas de l'arbre de Steiner.

Nous nous intéressons ici à la migration en tant que moyen de choisir un core afin d'améliorer les performances de l'arbre CBF [FHM00]. Nous avons conduit des simulations afin d'étudier le comportement et les performances de diverses heuristiques de sélection de core. Nous avons étudié deux différentes classes d'heuristiques : choix aléatoire parmi un ensemble de candidats donnés ou choix tenant compte de la topologie du réseau et/ou des membres du groupe. Notre contribution majeure est une méthode de sélection de core simple, présentant de très bonnes performances et qui est particulièrement bien adaptée aux protocoles CBF existants. Cette heuristique qui choisit le core parmi les nœuds situés au centre de l'arbre multicast (*tree center*) obtient des performances supérieures aux méthodes purement aléatoires et des performances similaires à des heuristiques qui, tout comme la méthode tree-center, tiennent compte d'informations topologiques mais s'avèrent beaucoup plus complexes/gourmandes.

3.5.2 Métriques pour l'évaluation d'arbres multicast

Pour évaluer les performances d'une heuristique de sélection de core et, par conséquence, la qualité de l'arbre multicast construit, nous avons considéré différentes métriques. Comme nous l'avons mentionné dans la section 3.2, l'importance relative d'une métrique dépend surtout du type d'application multi-parties. Différentes métriques ont été étudiées pour comparer les protocoles PIM et CBT dans [BCFG$^+$97]. Nous en avons retenu certaines et défini de supplémentaires. Les différents critères retenus sont les suivants :

Distance au core (*reach cost*). La distance $d(u, c)$ d'un membre u au core c est la longueur du chemin que doit emprunter un paquet multicast pour atteindre le core. Le maximum des distances des membres au core est un paramètre important pour les applications sensibles au délai (application multimédia temps réel).

Coût de communication (*communication cost*). Le coût de communication d'un arbre multicast est le nombre d'étapes nécessaires entre le moment où le core émet le premier paquet et le moment où le dernier membre reçoit ce paquet. Dans cette étude, nous avons supposé qu'un routeur réémet les paquets multicast dans l'ordre décroissant de la longueur des branches correspondantes à chacun de ses liens de sortie faisant partie de l'arbre multicast.

Si l'on considère des routeurs Δ-port, *i.e.*, capables de réémettre un message sur la totalité de leurs liens de sortie en même temps, le coût de communication d'un arbre multicast est équivalent au maximum des distances des membres au core.

Coût d'adhésion (*join cost*). Le coût d'adhésion d'un nœud u non membre est la distance $d(u, c)$ que doit parcourir le message `JOIN-REQUEST` de u pour atteindre le core c. Si la dynamicité d'un groupe est importante, on peut vouloir optimiser ce paramètre afin de minimiser les surcoûts des changements d'adhésion.

Congestion. La congestion d'un lien d'un arbre de multicast est le nombre de chemins allant d'un membre au core qui empruntent ce lien. Quand chaque membre est aussi une source (application de téléconférence), cette métrique donne une indication sur la concentration de trafic qui s'opère dans l'arbre de multicast.

Nombre d'arêtes de l'arbre multicast (*bandwith*). Ce critère est le plus employé dans la comparaison d'arbres multicast et correspond au nombre de canaux utilisés dans le réseau pour transmettre un paquet multicast à l'ensemble des membres. Dans le cas où plusieurs arbres multicast sont susceptibles de coexister au sein d'un même réseau, vouloir minimiser individuellement le nombre de ressources nécessaires pour chacun des arbres est souhaitable.

3.5.3 Heuristiques de sélection de core

Comme cela a été mentionné précédemment, le choix du core influe directement sur la topologie et le comportement de l'arbre multicast et donc sur les performances obtenues. Chaque heuristique définit une méthode pour identifier l'ensemble $\mathcal{C}$ des routeurs candidats dans lequel sera choisi le core et nous donnons pour chaque heuristique le type d'information qui doit être stocké dans les routeurs.

Notation. On note $M \in V$ l'ensemble des membres d'un groupe multicast au sein d'un réseau $G = (V, E)$ de $N = |V|$ routeurs. Le nombre de membres est $k = |M|$. Ainsi, les différentes complexités en temps des heuristiques seront données en fonction de N et/ou k. On note $e(u) = \max_{v \in V} d(u, v)$ l'excentricité d'un nœud u et on définit le centre d'un graphe par l'ensemble $\{u \in V \mid e(u) = \min_{v \in V} e(v)\}$ des nœuds d'excentricité minimale.

Routeur aléatoire (*random routeur*). Dans cette méthode simple de sélection de core, tous les nœuds du réseau peuvent être employés comme core d'un arbre multicast quelle que soit leur appartenance à un groupe multicast ou non. On a donc $\mathcal{C} = V$. Plusieurs heuristiques peuvent satisfaire cette propriété. Plus précisément, une heuristique est dite aléatoire si les deux conditions suivantes sont remplies :

- Aucune information sur la topologie de G ni sur celle de M n'est employée.
- La correspondance entre l'espace d'adressage $\mathcal{A}$ des groupes multicast et v est distribuée uniformément sur V.

La seconde propriété garantit que différentes heuristiques de choix aléatoire produisent statistiquement des résultats et des comportements similaires bien qu'elles puissent différer dans leur mise en œuvre.

Membre aléatoire (*random member*). Cette méthode restreint le choix du core à l'ensemble des routeurs membres du groupe, *i.e.*, $\mathcal{C} = M$. On peut, tout comme dans la méthode précédente, utiliser une fonction aléatoire ou une fonction de hachage. Nous sommes plus intéressés par un autre choix qui consiste à choisir comme core le premier membre qui s'abonne, ce qui rend cette heuristique applicable à la création du groupe.

Centre topologique de G (*network center*). Cette méthode choisit le core parmi les routeurs qui sont au centre du réseau. Le centre du réseau est facilement calculable si les plus courts chemins entre toute paire de sommets sont connus. Dans les réseaux utilisant un algorithme de routage de type LSR, chaque routeur peut effectuer ce calcul puisque chaque routeur maintien une vue de la topologie du réseau. La complexité en temps est en $O(N^3)$ (algorithme de DIJKSTRA), ce qui pose le problème du passage à l'échelle de cette heuristique. Dans des réseaux n'employant pas de fonction de routage de type LSR, un algorithme distribué doit être mis en œuvre.

Centre topologique de M (*group center*). Cette méthode choisit le core parmi les routeurs $\mathcal{C} = \{u \in V \mid \max_{v \in M} d(u,v) = \min_{x \in V} \max_{y \in M} d(x,y)\}$ qui forment le centre du groupe M. Notons que le centre de M n'appartient pas nécessairement à M. La complexité est similaire au calcul du centre topologique de G. De plus, le core doit garder à jour la liste des membres (via la réception des messages `JOIN-REQUEST`).

Centre topologique restreint de M (*center member*). Cette méthode choisit le core parmi les membres $\mathcal{C} = \{u \in V \mid \max_{v \in M} d(u,v) = \min_{x \in M} \max_{y \in M} d(x,y)\}$ qui forment le centre du groupe M. Dans les réseaux utilisant un algorithme de routage de type LSR, le fait de restreindre le centre à M permet d'avoir une complexité kN^2.

Nœud de l'arbre multicast (*random tree node*). Seuls les routeurs appartenant à l'arbre multicast $T \in V$ sont des candidats possibles, *i.e.*, $\mathcal{C} = T$. Cette heuristique n'est applicable que si un arbre a déjà été construit. La topologie de l'arbre est nécessaire et peut être collectée au moyen des messages `JOIN-REQUEST` si ces derniers sont en mesure de mémoriser le chemin qu'ils ont parcouru (sorte de *record route option* que l'on trouve dans IP).

Centre de l'arbre (*tree center*). Cette méthode choisit le core parmi les membres $\mathcal{C} = \{u \in T \mid \max_{v \in T} d_T(u,v) = \min_{x \in T} \max_{y \in T} d_T(x,y)\}$ qui forment le centre de l'arbre multicast $T \in V$. La distance d_T est la distance qui ne considère que les arêtes de l'arbre multicast T. La complexité est linéaire en le nombre de nœuds de l'arbre [BH90]. Cette heuristique nécessite que le core maintienne la topologie de l'arbre multicast T mais, à l'inverse des autres heuristiques basées sur le calcul d'un centre d'un ensemble, elle ne nécessite pas la connaissance de la topologie complète du graphe G.

Parmi toutes ces heuristiques, seule l'heuristique routeur aléatoire utilisant une fonction de hachage peut employer une méthode de gestion de core *implicite*, *i.e.*, l'identité du core peut être déduite uniquement à partir de l'identité du groupe et n'a pas besoin d'être stockée dans chaque nœud (c'est le cas de la méthode de gestion de PIM-SM [EFH+98] qui emploie une fonction de hachage). Le prix à payer pour utiliser toutes les autres heuristiques est la mise en œuvre d'une méthode de gestion de core *explicite* où l'identité du core de chaque groupe multicast actif doit être stockée dans le réseau (en utilisant, par exemple, le protocole LCM [HFM98] décrit précédemment). Notons que dans [TR97], les auteurs proposent un algorithme distribué pour calculer le centre d'un ensemble de candidats (les nœuds du réseau, les membres, les sources).

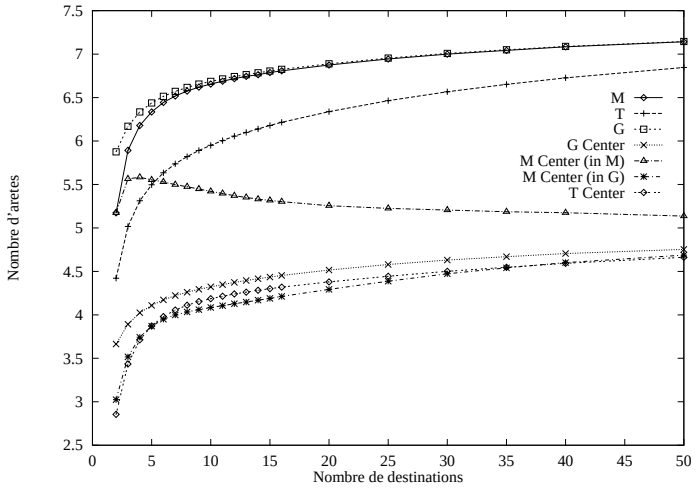


FIG. 3.11 – Comparaisons de la distance au core pour les différentes heuristiques.

3.5.4 Évaluation de performance

Nous avons effectué un jeu de simulations pour comparer les sept heuristiques proposées suivant les cinq critères exposés. Nous ne présentons que quelques résultats dans cette section, l'ensemble des courbes est disponible dans l'article [FHM00] fourni en annexe. Les courbes sont présentées suivant l'ensemble des nœuds qui sont candidats. Les légendes sont : G pour l'heuristique *routeur aléatoire*, M pour l'heuristique *membre aléatoire*, T pour l'heuristique *nœud de l'arbre*, $GCenter$ pour le centre topologique de G, $MCenter(in\,G)$ pour le centre topologique de M, $MCenter(in\,M)$ pour le centre topologique restreint de M et $TCenter$ pour le centre de l'arbre.

Nous avons généré aléatoirement une série de 100 graphes de 144 nœuds, chacun ayant un degré moyen de 4 et un diamètre moyen de 10. Nous avons employé un modèle hiérarchique à trois niveaux, similaire à ceux décrits par [ZCB96, ZCD97]. Ces modèles essayent de recréer la hiérarchie à plusieurs niveaux que l'on peut trouver dans Internet (*e.g.*, Hôtes-Routeurs-Systèmes Autonome). Notons que depuis 1999, la découverte des lois de puissance dans l'Internet par FALOUSTOS et al. [FFF99] a entraîné la création d'un nouveau type de modèle topologique. Ce dernier, que MAGONI nomme dans sa thèse [Mag02] « *modèle topologique des lois de puissance* », tente de modéliser toutes, ou une partie, des lois de puissance découvertes dans Internet. Ainsi la distribution des routeurs en fonction de leur degré obéit à une loi de puissance : la moitié des routeurs a un degré de 1, le quart a un degré de 2 pour atteindre des degrés très élevés en fin de distribution. Cette propriété diffère des graphes aléatoires [Doa96, Wax88, ZCD97] dans lesquels la distribution des nœuds en fonction de leur degré obéit à une loi de Poisson. Néanmoins, ne simulant que des domaines de routage et non pas l'Internet tout entier,

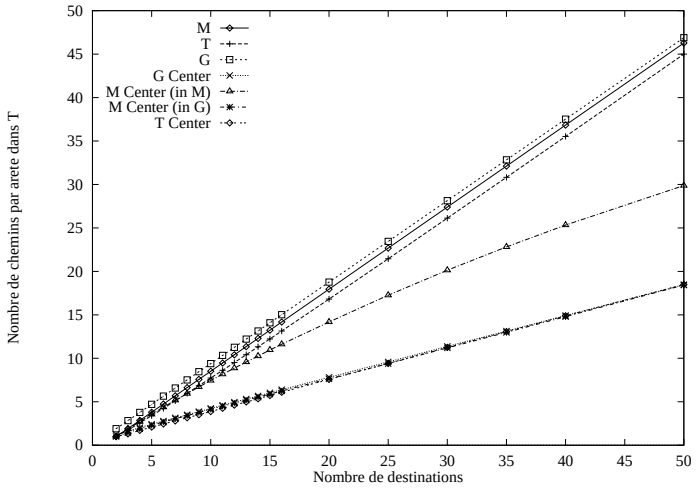


FIG. 3.12 – Comparaisons de l'arête congestion pour les différentes heuristiques.

nos résultats restent valides car les méthodes de génération de graphes aléatoires respectant les lois de puissance ne sont valides que pour un nombre de sommets très élevé (au delà du millier) [ACL00, Mag02, MP01, MIB00, PS00]

Les résultats pour le maximum des distances au core sont présentés sur la figure 3.11. Les résultats concernant la moyenne des distances au core sont similaires. Nous pouvons constater trois classes de comportement. Les meilleurs résultats sont obtenus avec les heuristiques calculant le centre du réseau, le centre de l'arbre et le centre du groupe, les différences entre ces trois heuristiques sont mineures. Les performances les moins bonnes sont obtenues avec les choix aléatoires (routeur aléatoire, membre aléatoire et nœud de l'arbre). La performance de l'heuristique calculant le centre topologique restreint de M se plaçant entre ces deux premières classes.

Les résultats pour le maximum de l'arête congestion sont présentés sur la figure 3.12. Cette congestion représente par définition la densité de concentration de trafic qu'il y a sur les liens de l'arbre multicast quand tous les membres du groupe émettent un trafic multicast. La figure 3.12 montre que le nombre de chemins des membres vers le core croît linéairement avec le nombre de membres. Quand le choix du core est aléatoire, la pente de la courbe atteint la borne supérieure pour cette métrique qui est égale à 1. En utilisant des heuristiques basées sur le calcul d'un centre, la pente est égale à 0.4, ce qui représente une amélioration très significative.

Pour finir, nous présentons sur la courbe 3.13 le nombre d'arêtes des arbres multicast pour chaque heuristique. Pour des groupes relativement petits (une dizaine de membres), l'heuristique consistant à choisir le core aléatoirement au sein du réseau utilise approximativement 10% de ressources en plus que toutes les autres. Pour des groupes de taille supérieure, la différence entre les heuristiques devient très peu significative.

L'analyse de ces résultats nous permet d'observer que l'heuristique calculant le centre de

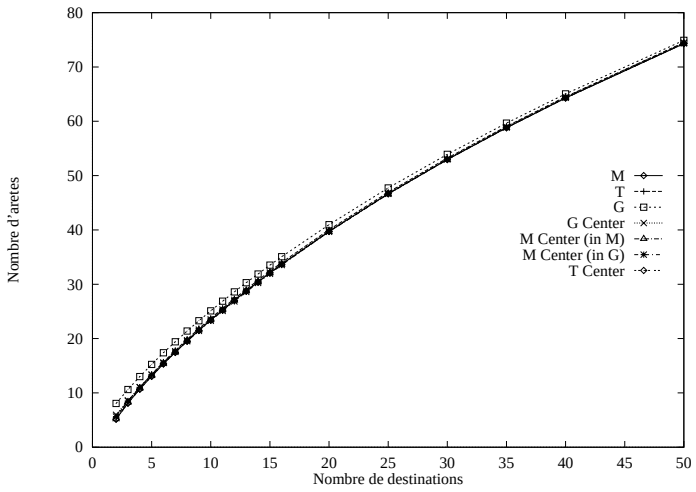


FIG. 3.13 – Comparaisons du nombre d'arêtes des arbres multicast pour les différentes heuristiques.

l'arbre multicast obtient de très bonnes performances sur l'ensemble des 5 métriques définies. Ses performances sont comparables à celles d'heuristiques beaucoup plus complexes en temps de calcul. Il semble donc que cette heuristique offre une solution efficace au problème de la migration de core. Nous avons vu que la complexité de cette heuristique était linéaire en le nombre d'arêtes de l'arbre multicast. Les courbes présentées sur la figure 3.13 semblent indiquer que le nombre d'arêtes de l'arbre multicast suit une croissance régulière. Nous revenons dans la section suivante sur ce facteur important afin de trouver une borne supérieure à notre heuristique.

3.5.5 Quelques bornes sur le nombre d'arêtes d'un arbre multicast

Nous devons quitter les graphes hiérarchiques ou tout autre modèle plus complexe qui suit les lois de puissance de l'Internet pour revenir à un objet plus simple (en tant que modèle) sur lequel nous pouvons espérer obtenir des résultats analytiques. Dans cette section, nous allons prendre comme modèle le graphe aléatoire et tenter de borner le nombre d'arêtes d'un arbre multicast. Bien évidemment, ces résultats ne sont pas valables pour un graphe donné mais pour « *presque tous les graphes* ».

Notation. Soit $G = (V, E)$ fixé, $X \subseteq V$ fixé et T un arbre couvrant de X, on pose :

$$\begin{aligned} m(G, X, T) &= |E(T)|, \text{ le nombre d'arêtes de } T \\ m(G, X) &= \min m(G, X, T), \text{ sur tous les arbres } T \text{ couvrant } X \end{aligned}$$

et pour tout entier $k \geq 1$ on pose :

$$m(G,k) \quad = \quad \frac{1}{\binom{1}{k}} \sum_{X \subseteq V, |X|=k} m(G,X)$$

la moyenne du nombre minimum d'arêtes d'un arbre recouvrant de k sommets d'un graphe G, pour G fixé.

Remarque. Il est préférable de prendre le $\min$ pour $m(G,X)$ car prenons le cas où le graphe G soit un cycle à N sommets et $X = (u,v)$ tel que u et v soient voisins. Il existe alors 2 arbres recouvrants possibles, l'un ayant 1 arête et l'autre possédant $O(n)$ arêtes. Ainsi, si l'on ne prend pas le $\min$, le paramètre $m(G,X)$ est non borné, et ce indépendemment de $|X|$ et du diamètre (il suffit de prendre un sous-ensemble d'un chemin hamiltonien pour s'en convaincre).

Proposition 3.1 *Pour tout entier $k > 0$ et G de diamètre D et de rayon r on a :*

$$k - 1 \ \leq \ m(G,k) \ \leq \ \min\left((k-1)D, kr\right) \tag{3.1}$$

Preuve.

La borne inférieure est triviale puisque si $E(T) < k-1$, alors T ne peut pas être un arbre recouvrant de k sommets.

Pour montrer la borne supérieure, il suffit de construire un arbre T couvrant comme suit :

– Prendre un sommet $u \in X$ comme racine. Construire à partir de u un BFS (*Breath First Search*) jusqu'à couvrir X. On a

$$\begin{aligned} |E(T)| \quad &\leq \quad \sum_{x \in X} d(x,u) \\ &\leq \quad \sum_{x \neq u, x \in X} d(x,u) \\ &\leq \quad (k-1)D \end{aligned}$$

– On choisit comme sommet racine, le sommet u de plus petite excentricité. La même construction que précédemment donnera la borne de kr, u n'étant cette fois-ci pas forcément dans X. □

Théorème 3.1 *Pour presque tout graphe G à n sommets (*i.e.*, tous les graphes sauf une fraction de cardinalité $1/n^3$), et pour tout entier $k > 0$ on a :*

$$m(G,k) \quad \leq \quad 2(k-1) \tag{3.2}$$

Preuve. D'après BOLLOBAS [Bol01], la fraction du nombre de graphes à n sommets ayant un diamètre $D = 2$ est $1 - 1/n^3$. S'il y a N graphes à n sommets au total, alors il y a $(1 - 1/n^3)N$ graphes de diamètre $D \leq 2$, et seulement $N/n^3 = o(N)$ graphes de diamètre supérieur $D > 2$. On dit « *presque tous les graphes* » car $(1 - 1/n^3)N = N - o(N)$, donc le nombre de graphes de diamètre $D = 2$ tend vers le nombre total de graphes à n sommets. Il suffit d'appliquer la formule 3.1 de la proposition 3.1 avec $D = 2$ pour terminer la preuve. □

Théorème 3.2 *Pour presque tout graphe G à n sommets, et pour tout entier $k > 0$ on a :*

$$m(G,k) \quad \leq \quad 1.5(k-1) \tag{3.3}$$

Preuve. Soit $G = (V, E)$ un graphe aléatoire à n sommets, et soit $p = 1/2$ la probabilité d'avoir une arête entre deux sommets quelconques $(x, y) \in V$. On suppose que l'on tire au hasard avec une probabilité uniforme égale à $\frac{1}{\binom{n}{k}}$ un sous-ensemble de k sommets de V : $X \subseteq V, |X| = k$.

Pour chaque ensemble X, on définit un arbre T recouvrant de X de la façon suivante : on choisit un sommet u (au hasard) comme racine et on construit k plus courts chemins de u vers chacun des éléments de X.

On définit la variable aléatoire $Z = m(G, X, T)$. On notera $\tilde{Z}$ l'espérance de cette variable. On a donc clairement $m(G,k) \leq \frac{1}{\binom{n}{k}} \sum_X Z$ et nous voulons montrer que $Z \leq 1.5\,k$.

Montrons que pour tout sommet $x \in V$, la probabilité qu'il soit dans X est $Pr(x \in X) = k/n$. En effet, le nombre d'ensembles à k éléments où x apparaît est $\binom{n-1}{k-1}$ puisque l'on choisit x et $k-1$ autres éléments. Donc on a :

$$\begin{aligned} Pr(x \in X) &= \frac{\binom{n-1}{k-1}}{\binom{n}{k}} \\ &= k/n \end{aligned}$$

Soit u un sommet de V. On note σ le nombre de voisins de u dans X. Alors, le nombre moyen de voisins de u dans X est $\tilde{\sigma} = k/2$. Donc, en moyenne un sommet u de V est connecté à $k/2$ sommets de X, par conséquent, il est connecté aux $k/2$ autres sommets de X par un chemin de longueur 2 puisque le diamètre de G est égal à 2 avec un forte probabilité $1 - o(1)$. Donc l'espérance de Z est :

$$\begin{aligned} \tilde{Z} &\leq k/2 + 2(k/2) \\ &\leq 1.5k \end{aligned}$$

□

Remarque. Ce résultat donnant $m(G,k) = \Theta(k)$ a le mérite d'exister. Néanmoins, il faudrait arriver à montrer que pour tout graphe G, si on choisit un sous-ensemble $X \subseteq V$ aléatoirement, alors $m(G, X) \leq O(|X|)$ ou toute autre borne, en $O(|X| \log n)$ par exemple.

3.6 Apports et mise en œuvre dans une technologie de réseaux actifs

3.6.1 Introduction

Si l'émergence des réseaux large bande a fortement contribué à l'intégration partielle, sur un réseau physique, des services de base de transfert de voix et de données, fait est de constater que le nombre de services offerts reste néanmoins limité en raison de l'impossibilité d'extension de services dans les composants du réseau. En effet, la plupart des équipements intègrent un ensemble

fini de services souvent « codés en dur » et non extensibles. Le réseau intelligent a marqué une évolution supplémentaire dans le domaine des services en définissant, au début des années 90, une infrastructure permettant de coupler le réseau à des bases d'informations afin de traiter de nombreux services qui peuvent être déployés beaucoup plus rapidement que précédemment. Ce type de réseau est cependant totalement orienté vers la téléphonie et les services associés en raison de sa dépendance envers le modèle d'appel utilisé. Celui-ci est bien trop spécifique pour être applicable à d'autres types de réseaux.

Il semble important de nos jours de pouvoir offrir aux opérateurs et fournisseurs de services une réactivité accrue dans les réseaux de transmission de données. Pour cela, il faut être en mesure de proposer des architectures et technologies ouvertes leur permettant de développer, dans des délais records et des conditions technologiques stables, de nouvelles offres de services de communication. Cela leur permettra de mieux répondre aux besoins des usagers. Ce besoin est accru par la multiplication des services applicatifs (communication de groupe, téléphonie sur IP, Web), et par l'explosion du nombre de protocoles de contrôle (MPLS [MPL], RSVP [BZB+97], Differentiated Services [DIF], GSMP [NEH+96], ISUP, PNNI) dont l'émergence trop rapide ne permet pas aux constructeurs d'équipements de les intégrer tous dans une échelle de temps adaptée aux besoins.

Les réseaux programmables offrent une solution à ce problème en ouvrant les équipements et l'ensemble du réseau au travers d'interfaces de programmation et/ou d'interfaces d'accès permettant à des tiers de concevoir, de réaliser et de déployer de nouveaux services sur ces réseaux. Les réseaux actifs et/ou programmables sont véritablement lancés en 1995 par l'organisation de workshops sur la signalisation ouverte et en 1996 par une proposition de TENNENHOUSE et WETHERALL [TW96] ainsi que par un article de LAZAR [Laz97] sur le besoin de programmation des réseaux de télécommunications. Ces approches véhiculent de fabuleux espoirs dans l'ouverture des réseaux sur l'ensemble des plans : supervision, signalisation, données [Fes01, FCF00]. Cette ouverture se matérialise par des infrastructures logicielles qui permettent de déployer dynamiquement de nouveaux services sur tout ou partie d'un réseau.

Nous reprenons ici quelques arguments et définitions donnés dans [FCF00] pour justifier l'utilisation de la technologie active dans la mise en œuvre de services de communication de groupe.

Définition 3.1 Un réseau programmable est un réseau de transmission de données ouvert et extensible disposant d'une infrastructure dédiée à l'intégration et à la mise en œuvre rapide de nouveaux services sur l'ensemble de ses composants.

Cette définition décrit la notion de réseau programmable par opposition aux réseaux dits « traditionnels ». La clef se trouve dans le niveau de programmabilité que l'on veut atteindre et cela induit nécessairement une abstraction des différents composants d'un réseau pour en faire des objets informatiques « classiques ». Cette notion permet de formuler une seconde définition, sorte de corollaire de la première :

Définition 3.2 Un réseau programmable est un réseau dans lequel tout ou partie de l'infrastructure de communication est virtualisée.

3.6.2 Motivation pour l'utilisation de la technologie active

Nous listons brièvement les principaux atouts et apports des réseaux actifs parmi ceux donnés dans [FCF00] qui ont motivé l'emploi de la technologie active pour la mise en œuvre d'un protocole de routage multicast :

- les délais entre la spécification initiale d'un nouveau service, sa normalisation et/ou standardisation et finalement la réalisation et le déploiement à grande échelle sont devenus trop importants. Virtuellement, l'actif ne nécessite plus de standardisation si ce n'est, fait non négligeable, d'avoir une API standardisée ;
- le réseau peut profiter des informations issues d'applications sur la nature et la sémantique de leurs flux. L'actif semble ici tout à fait attrayant pour la mise en œuvre d'un protocole de gestion de core qui peut recourir à des informations beaucoup plus directement liées à l'application [TW96] ;
- les capacités de traitement et de mémoire des composants internes du réseau suivent une croissance exponentielle. Les coûts des processeurs et de la mémoire chutent de manière drastique. L'actif nous permet de déployer au sein des nœuds des fonctions de calcul distribuées permettant une meilleure utilisation de ces capacités de traitement et une meilleure utilisation globale du réseau [LWJ98] ;
- les réseaux actifs offrent aussi une formidable plate-forme d'expérimentation permettant de déployer et de tester des protocoles et des services innovants.

3.6.3 Travaux liés

Nous ne donnons ici que certains travaux ayant trait aux communications multicast. Le lecteur peut consulter l'habilitation à diriger les recherches d'Olivier FESTOR [Fes01] et le rapport de recherche INRIA écrit en collaboration avec Isabelle CHRISMENT et Olivier FESTOR [FCF00] pour de plus amples détails sur les différentes approches.

Divers travaux ont été entrepris sur l'étude de protocoles de multicast dans les réseaux actifs. WETHERALL a présenté dans sa thèse [Wet99] la mise en œuvre d'un protocole inspiré de PIM (Protocol Independant Multicast) [EFH+98] afin de montrer que les réseaux actifs pouvaient être facilement employés pour développer et déployer de réels protocoles et ce, malgré la complexité de ces derniers. Ces travaux ont aussi permis de montrer qu'introduire plusieurs modifications aux protocoles originels est une chose facilement réalisable par l'ajout de quelques capsules et que les modifications apportées restent simples d'utilisation.

Les réseaux actifs sont aussi utilisés pour fiabiliser les protocoles de multicast [LWG98, LGT98] et permettent de résoudre de façon élégante les problèmes dus à l'implosion d'acquittements négatifs (NACK), aux retransmissions inutiles, à la concentration de trafic due aux retransmissions, aux duplications de paquets et au fait que le groupe de membres est fortement dynamique. Dans la solution nommée *Active Reliable Multicast (ARM)* proposée dans [LGT98], les nœuds vont jouer un rôle actif dans le mécanisme de fiabilité :

- ils vont cacher les données transmises afin de pouvoir les réémettre au plus tôt en cas de requête de retransmission ;
- ils vont traiter les NACKS afin d'optimiser les demandes de retransmission et de collecter les informations concernant l'auteur de la demande de retransmission.

Dans [HSZ+02, ST02], les auteurs décrivent la conception et la spécification d'un protocole de communication point-à-multipoint actif, totalement fiable et adapté à des groupes de diffusion de grande taille nommé *MAF*. Les routeurs actifs de l'arbre de diffusion sont programmés pour maintenir une copie des données diffusées afin de pouvoir en réparer les pertes éventuelles. Ce protocole a été déployé sur la plate-forme du projet RNRT Amarrage.

Il semble que les réseaux actifs offrent des améliorations prometteuses dans le domaine des protocoles de multicast fiables car ils font appel à un point clé de ce genre de protocole qui est la possibilité de prendre une décision au sein même du réseau.

Une autre expérimentation a été réalisée dans le projet AMnet [WKZ98, WZ98a, WZ98b]. WITTMANN, KRASNODEMBSKI et ZITTERBART proposent un support pour les communications multi-points (multicast) qui intègre un système de filtres de QoS (Quality of Service) permettant de supprimer de l'information des flux multimédia afin de réduire le débit pour les récepteurs à faible bande passante sans pour autant affecter les récepteurs ayant une large bande passante. L'architecture de leur nœud est active dans le sens où elle comprend des filtres de QoS (MPEG-1) et de la signalisation pour mettre en œuvre la QoS (signalisation basée sur RSVP).

3.6.4 Mise en œuvre

La définition et la construction d'un protocole de gestion de core ne doit pas se limiter à la recherche du « meilleur » core [CZD95, DZ96, HFM98] mais doit aussi s'intéresser au problème de la reconstruction de l'arbre multicast, inhérent à tout processus de migration, qui doit apparaître comme la plus transparente possible aux membres du groupe, *i.e.*, ils doivent continuer à recevoir le trafic multicast sans avoir à subir trop de pertes supplémentaires ou des délais excessifs.

L'architecture active. Notre mise en œuvre d'un protocole similaire à PIM avec migration du core a été effectuée sur la plate-forme ANTS (Active Node Transfer System) [WGT98, Wet99, WGT99, WLG98]. Le principe de cette architecture repose sur la capacité offerte aux applications de déployer dynamiquement dans le réseau les services et protocoles qu'elles utilisent et cela sur tous les routeurs actifs que traversent les flux associés. Pour cela, ANTS offre une architecture de nœud permettant un support de protocoles multiples ainsi qu'un mécanisme de déploiement dynamique de ces protocoles au sein du flot de données de l'application. Les composants principaux de l'architecture ANTS sont :

le protocole : définit de manière conceptuelle le traitement à effectuer sur un flux donné. Un protocole est défini par un identificateur (signature MD5 du code qui l'implante), un ensemble de classes qui en définissent le comportement (groupes de code) ainsi que les unités de distribution de ce code (capsules) ;

la capsule : unité de base de la programmation du réseau. La capsule est utilisée, d'une part pour véhiculer les données d'une application entre les nœuds actifs et, d'autre part, pour transporter le code d'un protocole à déployer sur les nœuds du réseau ;

le nœud actif : environnement d'exécution. Le nœud actif permet la réception, l'exécution et l'envoi de capsules suivant un modèle de traitement prédéfini (protocole sélectionné par l'application ayant émis la capsule). Tout nœud actif dispose en plus d'un routage par défaut.

Remarque. La version d'ANTS que nous avons utilisée fonctionne au dessus d'UDP en IPv4. Dans le cadre d'un projet RNRT nommé AMARRAGE, nous avons retenu IPv6 comme protocole de transport des paquets actifs afin que l'environnement d'exécution se construise directement sur ce protocole. Pour ce, nous avons proposé une extension du protocole ANEP [ABG+97] pour qu'il puisse fonctionner sur IPv6 [DCC+00]. Pour pouvoir émettre et recevoir des paquets actifs depuis ANTS, nous avons étendu et redéveloppé le paquetage `java.net` du jdk 1.2 afin qu'il supporte d'une part IPv6 [CF00] et d'autre part les sockets ANEP.

Mécanisme de reconfiguration de l'arbre multicast. Nous avons mis en œuvre trois approches différentes permettant de faire migrer un core :

Intuitive. Elle consiste à faire la destruction complète de l'ancien arbre de routage, puis à construire le nouveau. Des pertes de trafic peuvent survenir. En effet, pendant la phase correspondant au passage de l'ancien arbre au nouveau, que nous appelons phase transitoire, tout le trafic émis au groupe par une source est perdu. Le core actuel envoie un message `MIGRATION(NEW_CORE)` dans l'arbre multicast. Les membres retransmettent ce message à leurs fils, détruisent les anciennes tables de routage et s'attachent au nouveau core.

La complexité de cette approche peut être estimée par le trafic généré par la destruction et par le trafic dû aux nouveaux messages `JOIN` : $|T_k| + k\tilde{d}$ où $\tilde{d}$ est la distance moyenne dans le réseau. Cette approche, même si elle peut engendrer des pertes de trafic, garantit la simplicité du mécanisme de changement ;

Maintien de deux arbres indépendants. L'idée sous-jacente est d'établir un tunnel entre le nouveau core et l'ancien et de continuer à distribuer le trafic dans les deux arbres afin de minimiser les pertes additionnelles de trafic. Le core actuel établit un tunnel à destination du nouveau core. Il diffuse dans son arbre un message `MIGRATION(NEW_CORE)` dans l'arbre multicast. À l'inverse de l'approche précédente, la destruction va se faire de bas en haut *(bottom-up*) et non plus de haut en bas (*top-bottom*). Il faut néanmoins faire attention à ne pas créer de cycles (voir la figure 3.14). Pour ce, les membres de l'arbre ne rediffusent le trafic que dans l'arbre auquel ils appartiennent tandis que les deux cores vont tunneller le trafic d'un arbre vers un autre (voir la figure 3.15).

La complexité de cette approche peut être estimée par le trafic généré par la destruction (diffusion descendante et montante dans l'arbre) et par le trafic dû aux nouveaux messages `JOIN` : $2|T_k| + k\tilde{d}$. Cette approche ne garantit pas l'absence de pertes puisque l'indépendance des deux arbres est basée sur le fait que la procédure de destruction est effectuée avant que la reconstruction des nouvelles branches ne soit terminée ;

Sans perte de paquet. L'idée sous-jacente est que durant la phase de migration, un membre est attaché, soit à son ancien arbre, soit à l'arbre du nouveau core. Pour garantir cette propriété, la destruction d'une branche de l'ancien arbre ne peut se faire que lorsque tous les membres présents sur cette branche ont effectivement rejoint le nouvel arbre.

La complexité de cette approche peut être estimée par le trafic généré par la destruction et par le trafic dû aux nouveaux messages `JOIN` qui sont acquittés : $2|T_k| + 2k\tilde{d}$

Notons que les complexités font intervenir le nombre d'arêtes de l'arbre multicast, d'où l'importance d'avoir une borne sur ce facteur. Certaines de ces approches se retrouvent dans [CM01].

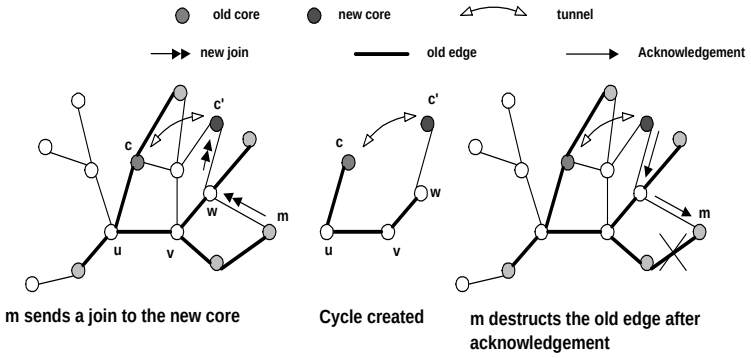


FIG. 3.14 – Mécanisme de reconfiguration d'un arbre multicast créant un cycle.

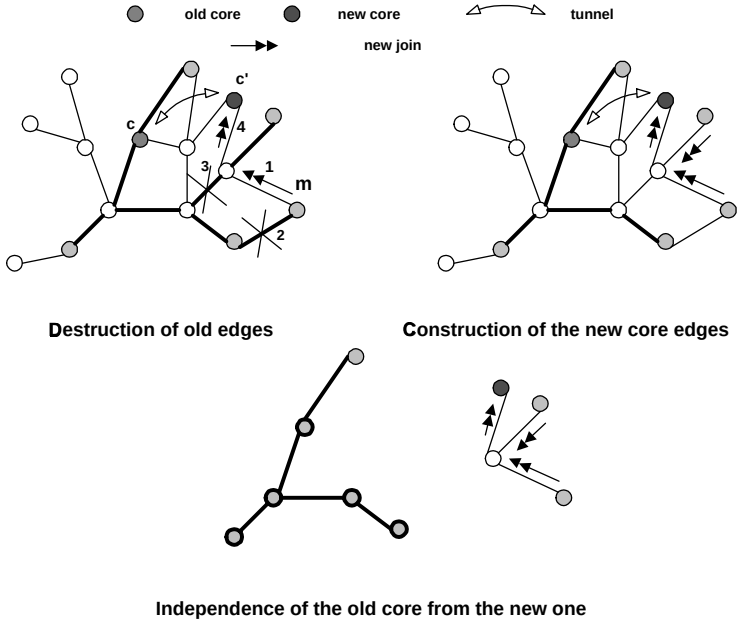


FIG. 3.15 – Mécanisme de reconfiguration d'un arbre multicast sans duplication de trafic.

Les auteurs proposent en effet les approches : *Add First Delete Last (AFDL)* où le nouvel arbre est d'abord créé avant que l'ancien ne soit détruit ; *Delete First Add Last (DFAL)* où l'ancien arbre est détruit avant que le nouveau ne soit construit ; *Interleaved Add Delete (IAD)* qui emploie l'approche AFDL mais les membres n'attendent pas la création du nouvel arbre pour se détacher de l'ancien ; *Interleaved Delete Add (IDA)* qui emploie l'approche DFAL mais les membres s'attachent au nouvel arbre sans attendre que l'ancien arbre soit complètement détruit.

Avantage de la migration Nous présentons quelques résultats mettant en avant les avantages du processus de migration. La figure 3.16 montre la bande passante utilisée par un arbre multicast en fonction de la taille du groupe multicast. La courbe du haut correspond à une version sans migration, la courbe du bas indique la bande passante employée juste après la migration et la courbe intermédiaire est la bande passante moyenne dans la version avec migration. On note un gain substantiel dû à l'utilisation du processus de migration. De plus, ces résultats ont été obtenus avec un réseau de petite taille (40 nœuds) et le gain peut être encore plus important sur des réseaux de plus grande taille.

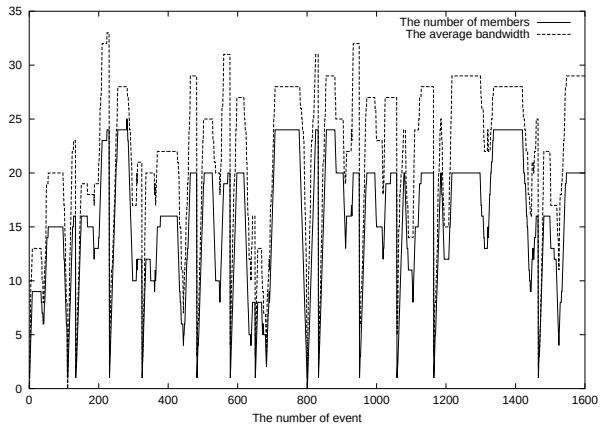


FIG. 3.16 – Bande passante en fonction de la taille du groupe

La figure 3.17 montre sur la courbe en pointillés la bande passante utilisée par un arbre de multicast au cours de la vie d'un groupe et l'autre courbe en continu indique le nombre de membres au sein de ce groupe. Le temps est représenté par les évènements qui interviennent dans le réseau (*join, leave, migration, beacon*). On note que le nombre de membres entre les évènements 940 et 980 reste le même (20). Lors de la migration, la bande passante chute de 32 à 1 avant de remonter progressivement à 27, soit un gain de plus de 15% dans ce cas.

Des résultats de simulation présentés par A. CHAKRABARTI et G. MANIMARAN dans [CM01] illustrent aussi l'importance du processus de migration pour tenir compte de la dynamique du groupe multicast et du réseau. Dans leurs simulations, ils ont considéré deux métriques : l'une, nommée *Service Disruption (SD)*, mesure le taux de discontinuité dans le service multicast comme étant le nombre de paquets perdus au cours du processus de migration et l'autre, nommée *Resource Wastage (RW)*, prend en compte le nombre de ressources superflues non utilisées au

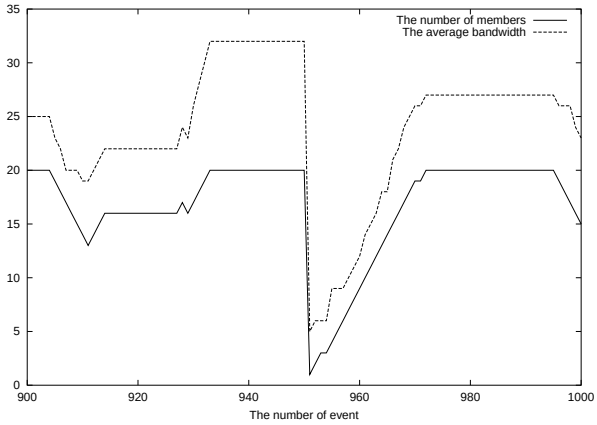


FIG. 3.17 – Exemple de migration.

cours du processus de migration. Leur conclusion est que les approches IAD et IDA offrent des résultats similaires aux approches AFDL and DFAL, ce qui tend à montrer que notre approche maintenant deux arbres indépendants est un bon compromis entre notre première approche intuitive et notre troisième qui nécessite la mise en œuvre d'acquittements.

3.7 Conclusion

Il y a sans doute diverses conclusions possibles dès que l'on parle de multicast dans l'Internet. Une certaine frustration de la communauté des chercheurs est sans doute compréhensible lorsque l'on regarde le nombre de protocoles proposés et la quasi non existence d'un tel service bout-en-bout au sein des réseaux d'opérateurs. Il semble que les opérateurs n'ont trouvé aucun intérêt pécuniaire à déployer un tel service, aucun du moins ne justifiant l'investissement nécessaire pour mettre à jour les routeurs de cœur et pour gérer ce nouveau type de service. Il se peut qu'avoir voulu imposer un service complètement ouvert de bout-en-bout fût une erreur, car cela n'a pas permis aux opérateurs de capitaliser sur un service qu'ils auraient pu déployer uniquement pour les applications qu'eux seuls proposaient. Cet argument *d'intégration verticale* est avancé par D. CLARK, J. WROCLAWSKI, K. SOLLINS et R. BRADEN dans [CWSB02].

L'Internet n'est plus un monde clos (*i.e.*, simple curiosité de recherche du monde académique) au comportement prévisible mais fait intégralement partie de notre société, et à ce titre, on y retrouve un large éventail d'acteurs aux objectifs et aux intérêts souvent conflictuels. Pour réagir au fait que le multicast de niveau 3 ne soit pas largement répandu, on assiste à une multitude de propositions pour la mise en œuvre d'un multicast dit *applicatif*. C'est une sorte de retour de balancier pour contrer les acteurs trop *conservateurs* de l'Internet. Ces propositions sont souvent liées à la mise en œuvre de systèmes peer-to-peer [CDKR02, RD01, GKM01]. Ces mises en œuvre offrent de nombreux avantages [CRSZ02, JGJ+00, Mat02] et c'est ce type de solution que nous avons notamment proposé dans le projet européen IST PROXiTV [CFGL02]. Le projet PROXiTV travaille à l'établissement et à la proposition d'une solution Internet et télévision haut

débit en exploitant les boucles locales haut débit. Nos travaux ont permis de mettre en œuvre une solution pour diffuser et répliquer le contenu mis à disposition par les producteurs (chaînes de télévision, fournisseurs de contenus) depuis un point d'entrée du système ou depuis les sites des producteurs vers les serveurs localisés en tête de chaque boucle locale (ADSL, câble) haut débit.

Les réseaux programmables ont certainement un bel avenir devant eux car la standardisation de différentes interfaces de programmation est en cours (le projet P1520 de l'IEEE poursuit son activité de normalisation des interfaces de programmation). L'introduction de la technologie active, notamment dans le plan des données et de la signalisation, pose de réels problèmes, difficiles et loin d'être résolus (cela explique peut-être le fléchissement actuel dans l'intérêt que la communauté porte à l'actif). Dans le cas du déploiement d'un protocole de multicast, la technologie active nous a été bénéfique. En effet, le concept d'actif permet à l'application de déployer elle-même sa propre fonction de coût qui va servir à évaluer les performances de l'arbre multicast. Une application peut, par exemple, chercher à minimiser le maximum des délais entre les membres, le délai moyen ou les ressources utilisées dans le réseau. Ce travail a permis de montrer que l'on pouvait facilement et rapidement développer et déployer un réel protocole de multicast complexe (PIM + migration). De plus, ce protocole tient compte des besoins des applications qui fournissent elles-mêmes la fonction de coût à appliquer à l'arbre de multicast ; fait qui est totalement spécifique aux services offerts par les réseaux actifs. Il faut néanmoins nuancer cette facilité de déploiement par la difficulté de dimensionnement d'une application. En effet, l'expérience acquise dans le projet RNRT VTHD/VTHD++ montre la complexité à identifier le type de trafic d'une application distribuée. Devoir spécifier quels sont les critères à mettre en œuvre pour le déploiement des opérations de multicast risque de complexifier encore plus cette phase de dimensionnement. On risque, soit de prendre un choix par défaut, soit de vouloir tout optimiser à la fois. On se retrouve dans le cas où l'abondance des critères possibles annule toute chance de les utiliser à bon escient.

L'actif permet non seulement de tirer parti des informations contenues dans les nœuds actifs (routage) mais aussi des ressources disponibles dans ces nœuds permettant de mettre en œuvre de véritables algorithmes distribués (par exemple pour la recherche du meilleur core). Il est aussi possible de tirer parti de ces ressources pour agréger les messages `JOIN` durant la phase transitoire et ainsi minimiser le trafic dans le réseau. Ce type d'opération d'agrégation au sein d'un arbre multicast est encore assez mal traité. Les arbres multicast déployés servent exclusivement à faire de la diffusion en mode 1-vers-N mais rarement à faire du *gather* de N-vers-1, ce qui est surprenant car la structure nécessaire à ce type d'opération existe (l'arbre) et les ressources nécessaires aussi. Nous verrons que nous avons rencontré une problématique assez similaire dans le contexte de la découverte de service dans les réseaux ad hoc (voir section 4.7).

Bibliographie

[ABG+97] D.S. Alexander, B. Braden, C.A. Gunter, A.W. Jackson, A.D. Keromytis, G. Minden, and D.J. Wetherall, *Active Network Encapsulation Protocol (ANEP)*, July 1997, `http://www.cis.upenn.edu/~switchware/ANEP/docs/ANEP.txt`.

[ACL00] W. Aiello, F. Chung, and L. Lu, *A random graph model for massive graphs*, ACM STOC'00, 2000, pp. 171–180.

[AE92] S. R. Ahuja and J. R. Esnor, *Co-ordination and control of multimedia conferencing*, IEEE Communications Magazine (1992).

[ATM96] ATM Forum, *Private network-network interface specification version 1.0*, ATM Forum technical specification af-pnni-0055.0000, March 1996.

[Bal97] A. Ballardie, *Core based trees (cbt version 2) multicast routing – protocol specification –*, Request For Comments 2189, Internet Engineering Task Force, September 1997, (Status : EXPERIMENTAL).

[BCFG+97] T. Billhartz, B. Cain, E. Farrey-Goudreau, D. Fieg, and S. Gordon Batsell, *Performance and ressource cost comparisons for the CBT and PIM multicast routing protocols*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 3, 304–315.

[Ber83] C. Berge, *Graphes*, 3e édition ed., Gauthier-Villars, 1983.

[BH90] F. Buckley and F. Harary, *Distance in graphs*, Addison Wesley, Redwood City, CA, 1990, ISBN : 0201095912.

[BL01] Nathaniel E. Baughman and Brian Neil Levine, *Cheat-proof playout for centralized and distributed online games*, INFOCOM, 2001, pp. 104–113.

[BLC95] T. Berners-Lee and D. Connolly, *Hypertext markup language 2.0*, Request for comments, Internet Engineering Task Force, November 1995.

[BN99] P.J. Braam and P.A. Nelson, *Removing bottlenecks in distributed filesystems : Coda & InterMezzo as examples*, Linux Expo 1999, May 1999.

[Bol01] B. Bollobas, *Random graphs*, Cambridge University Press, (January 2001, ISBN : 0521797225.

[Bra98] P. J. Braam, *The coda distributed file system*, Linux Journal (1998), 46–51.

[BV95] F. Bauer and A. Varma, *Degree-constrained multicasting in point-to-point networks*, Proceedings of the IEEE INFOCOM '95, 1995, pp. 369–376.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP), version 1 functional speciffication*, Request For Comments 2205, Internet Engineering Task Force, September 1997.

[CC97] K. Carlberg and J. Crowcroft, *Building shared trees using a onetomany joining mechanism*, ACM Computer Communicatin Review **27** (1997), no. 1, 5–11.

[CC99] ———, *Examining the construction of shared trees using different metrics*, IEEE Real-Time Technology and Applications Symposium (RTAS'99) Workshop, IEEE, June 1999.

[CDKR02] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, *Scribe : A large-scale and decentralised application-level multicast infrastructure*, IEEE Journal on Selected Areas in Communications, 2002.

[CF00] G. Chelius and E. Fleury, *An IP Next Generation compliant* Java™ Virtual Machine, International Workshop on Java™ for Parallel and Distributed Computing (Cancun, Mexico), Lecture Notes in Computer Science, IEEE, Springer Verlag, Mai 2000.

[CFGL02] J. Cohen, E. Fleury, and I. Guerrin-Lassous, *Route and transfer optimization*, Tech. Report D11, PROXiTV-IST-1999-20352, Mars 2002, (PROXiTV Consortium Restricted).

[CHK+95] I. Cidon, T. Hsiao, A. Khamisy, A. Parekh, R. Rom, and M. Sidi, *The OpeNet architecture*, Tech. Report 95-37, Sun Microsystems, December 1995.

[CK96] Jeremy R. Cooperstock and Steve Kotsopoulos, *Why use a fishing line when you have a net ? an adaptive multicast data distribution protocol*, Proceedings of USENIX Technical Conference '96, 1996.

[Cla92] W. J. Clark, *Multipoint multimedia conferencing*, IEEE Communications Magazine (1992).

[CLD+99] C. Chassot, A. Lozes, M. Diaz, L. Dairaine, and L. Rojas, *Qos requise par une application de DIS distribuée dans un environnement réseau grande distance*, 7ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'99) (Nancy, France), Avril 1999.

[CM01] A. Chakrabarti and G. Manimaran, *A case for scalable multicast tree migration*, Global Telecommunications Conference (GLOBECOM 01) (San Antonio, TX, USA), IEEE, November 2001, pp. 2026–2030.

[CRSZ02] Y. Chu, S. Rao, S. Seshan, and H. Zhang, *Enabling conferencing applications on the internet using an overlay multicast architecture*, SIGCOMM (San Diego, CA, USA), August 2002.

[CWSB02] D. Clark, J. Wroclawski, K. Sollins, and R. Braden, *Tussle in cyberspace : defining tomorrow's internet*, SIGCOMM'02 (Pittsburg, Pennsylvania, USA), ACM, August 2002.

[CZD95] K. Calvert, E. Zegura, and M. Donahoo, *Core selection methods for multicast routing*, International Conference on Computer Communications and Network (ICCCN '95) (Las Vegas, Nevada, USA), September 1995, pp. 638–642.

[DC90] S. Deering and D. Cheriton, *Multicast routing in datagram internetworks and extended LANs*, ACM Transactions on Computer Systems **8** (1990), no. 2, 85–110.

[DCC+00] S. D'Alu, G. Chelius, I. Christment, O. Festor, and E. Fleury, *Intégration du support IPv6 dans l'environnement de supervision de réseaux actifs ANAIS*, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2000) (Toulouse, France), Hermès, Octobre 2000.

[DDC97] C. Diot, W. Dabbous, and J. Crowcroft, *Multipoint communication : A survey of protocols, functions, and mechanisms*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 3, 277–290.

[Dee89] S. Deering, *Host extensions for IP multicasting*, Request For Comments 1112, Internet Engineering Task Force, August 1989.

[Dee91] ———, *Multicast routing in datagram internetwork*, Ph.D. thesis, Stanford University, California, USA, December 1991.

[DEF+96] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, *The pim architecture for wide-area multicast routing*, IEEE / ACM Transactions on Networking **4** (1996), no. 2, 1153–162.

[DEF+99] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer, and L. Wei, *Protocol Independent Multicast version 2 Dense Mode specification (PIM-DM)*, Internet Draft draft-ietf-pim-v2-dm-03.txt, Internet Engineering Task Force, June 1999.

[DIF] *Differentiated services*, `http://www.ietf.org/html.charters/diffserv-charter.html`.

[DM78] Y. Dalal and R. Metcalfe, *Reverse path forwarding of broadcast packets*, Communications of the ACM **21** (1978), no. 12, 1040–1048.

[Doa96] M. Doar, *A better model for generating test networks*, Globecom'96, IEEE, November 1996.

[DZ96] M. J. Donahoo and E. W. Zegura, *Core migration for dynamic multicast routing*, International Conference on Computer Communications and Network (ICCN '96) (Washington DC, USA), IEEE, October 1996, pp. 92–98.

[EFH+98] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C.-G. Liu, P. Sharma, and L. Wei, *Protocol Independent MulticastSparse Mode (PIM-SM)*, Request For Comments 2362, Internet Engineering Task Force, June 1998.

[FCF00] O. Festor, I. Chrisment, and E. Fleury, *Les réseaux programmables 1.0*, Rapport de Recherche 3913, INRIA, Mars 2000, `http://www.inria.fr/rrrt/rr-3913.html`.

[Fen97] W. Fenner, *Internet group management protocol, version 2*, Request For Comments 2236, Internet Engineering Task Force, November 1997.

[Fes01] O. Festor, *Ingénierie de la gestion de réseaux et de services : du modèle osi à la technologie active*, Habilitation à diriger des recherches, Université Henri Poincaré, Nancy, France, Décembre 2001.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On power-law relation-ships of the Internet topology*, ACM SigComM'99 (Cambridge, USA) (ACM, ed.), September 1999.

[FGF+99] R. Fielding, J. Gettys, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, Request For Comments 2616, Internet Engineering Task Force, June 1999.

[FHM00] E. Fleury, Y. Huang, and P. K. McKinley, *On the performance and feasibility of multicast core selection heuristics*, Networks **35** (2000), no. 2, 145–56.

[FJM+95] Sally Floyd, Van Jacobson, Steven McCanne, Ching-Gung Liu, and Lixia Zhang, *A reliable multicast framework for light-weight sessions and applications level framing*, Proceedings of SIGCOMM '95 (Cambridge, MA USA), 1995, pp. 342–356.

[GD98] L. Gautier and C. Diot, *Design and evaluation of mimaze, a multi-player game on the internet*, IEEE Multimedia Systems Conference (Austin, USA), July 1998.

[GKM01] A. Ganesh, A.-M. Kermarrec, and L. Massoulie, *SCAMP : Peer-to-peer lightweight membership service for large-scale group communication*, 3rd International workshop on Networked Group Communication, (London, UK), November 2001.

[GM93] M. X. Goemans and Y.-s. Myung, *A catalog of Steiner tree formulations.*, Networks **23** (1993), no. 01, 19–28.

[HBC95] Markus Hofmann, Torsten Braun, and Georg Carle, *Multicast communication in large scale networks*, Proceedings of Third IEEE Workshop on High Performance Communication Subsystems (HPCS) (Mystic, Connecticut USA), August 1995.

[HFM98] Y. Huang, E. Fleury, and P. K. McKinley, *LCM : A multicast core management protocol for link-state routing networks*, International Conference on Communications (ICC' 98) (Atlanta, Georgia), IEEE, June 1998.

[HJ98] M. Handley and V. Jacobson, *Session description protocol*, Request For Comments 2327, Internet Engineering Task Force, April 1998.

[HM97] Y. Huang and P. K. McKinley, *Group leader election under link-state routing*, International Conference on Network Protocols (ICNP '97), IEEE, October 1997.

[HRC92] M. Harrick, P. Venkat Rangan, and M.S. Chen, *System support for computer mediated multimedia collaborations*, Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work (CSCW '92), November 1992, pp. 203–209.

[HSC95] Hugh W. Holbrook, Sandeep K. Singhal, and David R. Cheriton, *Log-based receiver-reliable multicast for distributed interactive simulation*, Proceedings of SIGCOMM '95 (Cambridge, MA USA), 1995, pp. 328–341.

[HSZ+02] R. Hammi, P. Spathis, D. Zebiane, K. Chen, A. Serhrouchni, and K. Thai, *Deployment and experimentation of an active network at a large scale : Amarrage*, NETCON'02 (Paris, France), IFIP/IEEE, October 2002.

[HW99] D. Hands and M. Wilkins, *A study of the impact of network loss and burst size on video streaming quality and acceptability*, Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'99) (Toulouse, France) (M. Diaz, P. Owezarski, and P. Senac, eds.), LNCS, vol. 1718, October 1999.

[IKL00] A. Irlande, J.-C. Konig, and C. Laforest, *Incrémentalité pour l'arbre de Steiner*, AlgoTel 2000 (La Rochelle), INRIA, May 2000.

[IW91] M. Imase and B. Waxman, *Dynamic Steiner tree problem*, SIAM Journal on Discrete Mathematics **4** (1991), no. 3, 369–384.

[JGJ+00] J. Jannotti, D. Gifford, K. Johnson, M. Frans Kaashoek, and J. O'Toole, *Overcast : Reliable multicasting with an overlay network*, Operating Systems Design and Implementation (OSDI) (San Diego, CA, USA), 2000.

[KRT+98] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley, *The MASC/BGMP architecture for inter-domain multicast routing*, SIGCOMM'98 (Vancouver, Canada), ACM, September 1998, pp. 93–104.

[Laz97] L. Lazar, *Programming Telecommunication Networks*, IEEE Network Magazine (1997), 8–18.

[LGT98] L-W. H. Lehman, J. Garland, and D.L. Tennenhouse, *Active Reliable Multicast*, Proc. IEEE INFOCOM'98, 1998.

[LIN01] *Operation lindbergh. a world first in telesurgery : The surgical act crosses the atlantic !*, Press Conference, September 2001.

[LLS99] C. Liu, M. Lee, and T. Saadawi, *Core-manager based scalable multicast routing*, Advanced Telecommunication & Information Distribution Research Program (ATIRP) Conference, February 1999, pp. 2–5.

[LWG98] U. Legedza, D.J. Wetherall, and J. Gunter, *Improving The Performance of Distributed Applications Using Active Networks*, Proc. IEEE INFOCOM'98 (San Francisco, CA.), April 1998.

[LWJ98] U. Legedza, D. Wetherall, and Gunter J., *Improving the performance of distributed applications using active networks*, INFOCOM'98 (San Francisco, CA.), IEEE, April 1998.

[Mag02] D. Magoni, *Service de recherche d'agent par diffusion multipoint orientée*, Ph.D. thesis, Université Starsbourg I - Louis Pasteur, Janvier 2002.

[Mat02] L. Mathy, *Le multicast applicatif*, RHDM'02 (Autrans, France), 2002, (Cours de RHDM'02).

[MIB00] A. Medina, Matta I., and J. Byers, *On the origin of power laws in internet topologies*, ACM Computer Communication Review **30** (2000), no. 2.

[Moy94a] J Moy, *Multicast extensions to OSPF*, Request For Comments 1584, Internet Engineering Task Force, March 1994.

[Moy94b] _____, *OSPF version 2*, Request For Comments 1583, Internet Engineering Task Force, March 1994.

[MP01] D. Magoni and J.-J. Pansiot, *Internet topology analysis and modeling*, Computer Communications Workshop (Charlottes-ville, Virginia, USA), IEEE, October 2001.

[MPL] *Multi protocol label switching (MPLS)*, `http://www.ietf.org/html.charters/mpls-charter.html`.

[NEH+96] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall, *Ipsilon's general switch management protocol specification version 1.1*, Request For Comments 1987, Internet Engineering Task Force, August 1996.

[OR93] J. Oikarinen and D. Reed, *Internet relay chat protocol*, Request For Comments 1459, Internet Engineering Task Force, May 1993.

[Pow96] D. Powell, *Group communication*, Communications of the ACM **39** (1996), no. 4, 50–97, (special section Group Communication).

[PR85] J. Postel and J. Reynolds, *File transfer protocol (FTP)*, Request For Comments 959, Internet Engineering Task Force, October 1985.

[PS00] C. Palmer and G. Steffan, *Generating network topologies that obey power laws*, Globecom'00, IEEE, 2000.

[RCV+00] V. Roca, L. Costa, R. Vida, A. Dracinschi, and S. Fdida, *A survey of multicast technologies*, Tech. Report RP-LIP6-2000-09-05, LIP6, September 2000.

[RD01] A. Rowstron and P. Druschel, *Pastry : Scalable, distributed object location and routing for large-scale peer-to-peer systems*, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware) (Heidelberg, Germany), 2001, pp. 329–350.

[RSVW94] W. Reinhard, J. Schweitzer, G. Vlksen, and M. Weber, *CSCW tools : Concepts and architectures*, IEEE-Computer (1994).

[SKK+90] M. Satyanarayanan, J. Kistler, P. Kumar, M. Okasaki, E. Siegel, and D. Steere, *Coda : A highly available file system for a distributed workstation environment*, IEEE Transactions on Computers **39** (1990), no. 4.

[SMSRM99] R. Sriram, G. Manimaran, and C. Siva Ram Murthy, *A rearrangeable algorithm for the construction of delay-constrained dynamic multicast trees*, IEEE / ACM Transactions on Networking **7** (1999), no. 4, 514–529.

[ST02] P. Spathis and K. Thai, *MAF : un protocole de multicast fiable*, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2002) (Montréal, CANADA), Hermès, May 2002.

[Tha02] D. Thaler, *Border gateway multicast protocol (bgmp)*, Internet Draft draft-ietf-bgmp-spec-03.txt, Internet Engineering Task Force, June 2002.

[TR97] D. Thaler and C. Ravishankar, *Distributed center-location algorithms*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 3, 291–303.

[TW96] D.L. Tennenhouse and D.J. Wetherall, *Towards an Active Network Architecture*, Computer Communication Review **26** (1996), no. 2.

[Ude94] Jon Udell, *Computer telephony*, Byte **19** (1994), no. 07, 80–99.

[Wax88] B Waxman, *Routing of multipoint connections*, IEEE Journal on Selected Areas in Communications (1988), 1617–1622.

[Wax93] B. Waxman, *Performance evaluation of multipoint routing algorithms*, Proceedings of INFOCOM' 93, 1993.

[Wet99] D.J. Wetherall, *Service Introduction in an Active Network*, Ph.D. thesis, Massachusetts Institute of Technology, February 1999.

[WGT98] D.J. Wetherall, J. Guttag, and D. Tennenhouse, *ANTS : A toolkit for building and dynamically deploying network protocols*, IEEE OPENARCH'98 (SF, CA), April 1998, `http ://www.sds.lcs.mit.edu/activeware/ants`.

[WGT99] D. Wetherall, J. Guttag, and D. Tennenhouse, *ANTS : Network services without the red tape*, IEEE Computer (1999).

[Win87] P. Winter, *Steiner problem in networks : a survey*, Networks (1987), 129–167.

[WKZ98] R. Wittmann, K. Krasnodembski, and M. Zitterbart, *Heterogeneous multicasting based on RSVP and QoS filters*, International Symposium on Broadband Networks (SYBEN'98) (Zürich, Switzerland), May 1998.

[WLG98] D.J. Wetherall, U. Legedza, and J.V. Guttag, *Introducing New Internet Services : Why and How*, IEEE Network. Special Issue on Active and Programmable Networks (1998).

[WPD88] D. Waitzman, C. Partridge, and S. Deering, *Distance vector multicast routing protocol*, Request For Comments 1075, Internet Engineering Task Force, November 1988.

[WZ98a] R. Wittmann and M. Zitterbart, *Active multicasting for heterogeneous groups*, 4th International Conference on Broadband Communications '98 (Stuttgart, Germany), IFIP, April 1998.

[WZ98b] ———, *AMnet : Active multicasting network*, International Conference on Communications (ICC'98) (Atlanta, GA, USA), June 1998.

[ZCB96] E. Zegura, K. Calvert, and S. Bhattacharjee, *How to model an inetrnetwork*, Infocom '96 (San Francisco, USA), IEEE, March 1996.

[ZCD97] E. Zegura, K. Calvert, and M. Donahoo, *A quantitative comparison of graph-based models for internetworks*, IEEE / ACM Transactions on Networking **5** (1997), no. 6, 770–783.

[ZPGLA95] Q. Zhu, M. Parsa, and J. Garcia-Luna-Aceves, *A source-based algorithm for delay-constrained minimum-cost multicasting*, Proceedings of the IEEE INFOCOM '95, 1995, pp. 377–385.

Publications

Livres, chapitre de Livre

[ACFF01] L. Andrey, I. Chrisment, O. Festor, and E. Fleury, *Sytèmes multimédia communicants*, IC2, ch. Infrastructures pour le multimédia : ALF et les réseaux actifs, Hermes, 2001.

[Fle00] E. Fleury (ed.), *Algotel 2000 : 2^{es} rencontres francophones sur les aspects algorithmiques des télécommunications*, INRIA, may 2000, ISBN 2-7261-1157-2.

Journaux, conférences

[ACFF99] L. Andrey, I. Chrisment, O. Festor, and E. Fleury, *Supervision et contrôle dans les réseaux actifs : une nécessité à la mise en œuvre et au déploiement dans les réseaux de télécommunication*, GRES'99 (Montréal, Canada), June 1999.

[CF00] G. Chelius and E. Fleury, *An IP Next Generation compliant* Java™ Virtual Machine, International Workshop on Java™ for Parallel and Distributed Computing (Cancun, Mexico), Lecture Notes in Computer Science, IEEE, Springer Verlag, Mai 2000.

[DCC+00] S. D'Alu, G. Chelius, I. Christment, O. Festor, and E. Fleury, *Intégration du support IPv6 dans l'environnement de supervision de réseaux actifs ANAIS*, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2000) (Toulouse, France), Hermès, Octobre 2000.

[FCF99] O. Festor, I. Chrisment, and E. Fleury, *Les réseaux programmables*, Tutoriel de l'École d'été RHDM'99, 1999, Pointe du Diable, ENST-Bretagne.

[FCF02] ———, *Les réseaux programmables*, Tutoriel des Rencontres Francophones sur l'Algorithmique des Télécommunications (ALGOTEL 2001), mai 2002, Saint-Jean de Luz.

[FHM98] E. Fleury, Y. Huang, and P. K. McKinley, *On the performance and feasibility of multicast core selection heuristics*, Proceedings of the Seventh International Conference on Computer Communications and Networks (IC3N' 98) (Lafayette, Louisiana), October 1998.

[FHM00] ———, *On the performance and feasibility of multicast core selection heuristics*, Networks **35** (2000), no. 2, 145–56.

[HFM98] Y. Huang, E. Fleury, and P. K. McKinley, *LCM : A multicast core management protocol for link-state routing networks*, International Conference on Communications (ICC' 98) (Atlanta, Georgia), IEEE, June 1998.

[KF00a] H. Koubaa and E. Fleury, *Active multicasting*, International Conference on Networks (ICON 2000) (Singapore), IEEE, 2000.

[KF00b] ———, *Algorithmes de reconfiguration des tables de routage multipoints dans un arbre partagé avec migration du core*, AlgoTel 2000 (La Rochelle), INRIA, May 2000, pp. 179–184.

[TCFF02] S. Thibault, X. Cavin, O. Festor, and E. Fleury, *Unreliable transport protocol for commodity based opengl distributed visualization*, Workshop on Commodity-Based Visualization Clusters (Boston, Massachusetts), IEEE, October 2002, (in conjunction with IEEE Visualization 2002).

Rapports de recherche

[CF00a] G. Chelius and E. Fleury, *Implementing IPv4 multicast with IPv6 sockets*, Tr, INRIA, may 2000.

[CF00b] _____, *An IP next generation compliant java*[tm] *virtual machine*, RR RR-3936, INRIA, mai 2000.

[CFGL02] J. Cohen, E. Fleury, and I. Guerrin-Lassous, *Route and transfer optimization*, Tech. Report D11, PROXiTV-IST-1999-20352, Mars 2002, (PROXiTV Consortium Restricted).

[FCF00] O. Festor, I. Chrisment, and E. Fleury, *Les réseaux programmables 1.0*, Rapport de Recherche 3913, INRIA, Mars 2000, `http://www.inria.fr/rrrt/rr-3913.html`.

[Fle00a] E. Fleury, *État de l'art et des contraintes pour la vidéo sur internet*, Tech. Report D1.1, PRIAM - Service et Programme pour l'Internet Haut Débit (SPIHD), Octobre 2000, (Copyright 2000 Groupe SPIHD).

[Fle00b] _____, *Étude sur les technologies de multicasting*, Tech. Report D2.3, PRIAM - Service et Programme pour l'Internet Haut Débit (SPIHD), Octobre 2000, (Copyright 2000 Groupe SPIHD).

Travaux liés

[Kou99] H. Koubaa, *Les communications multipoints dans les réseaux actifs*, Dea, Université Henri Poincaré, Nancy, France, Juillet 1999.

Chapitre 4

Multicast dans les réseaux ad hoc

Cayenne c'est fini

J. HIGELIN

ad hoc Loc. adj. (lat.) invar. Qui convient à un usage déterminé, à une situation précise. *Servez-vous, pour cette manipulation, du dispositif ad hoc.*

4.1 Introduction

Un réseau ad hoc est une collection de mobiles, chacun équipé d'une ou plusieurs interfaces de communication sans fil. Ils sont aussi communément nommés *MANet* pour *Mobile Ad Hoc Network*. Ces réseaux se caractérisent par le fait que les communications entre les entités du réseau ne bénéficient d'aucune infrastructure préexistante ou d'appoint (balise, station centrale, borne, relais). Un réseau ad hoc doit être facilement déployable, les nœuds pouvant joindre et quitter le réseau de façon totalement dynamique sans devoir en informer le réseau et si possible sans effet de bord sur les communications des autres membres. De ce point de vue, il y a une certaine similitude entre la notion d'appartenance d'un nœud à un réseau ad hoc et la notion d'appartenance d'un hôte à un groupe multicast. La notion, ou plus exactement la terminologie de réseau « *ad hoc* » est assez récente. Néanmoins, les premiers fondements de réseaux de paquets transmis de proche en proche par voie hertzienne ont été initiés dans les années 70 aux États-Unis par la DARPA (Defense Advanced Research Projects Agency). Citons le projet ALOHA [AK73, Abr85] mis en œuvre par l'Université d'Hawaii qui a démontré la possibilité d'utiliser un médium radio pour envoyer des paquets à un saut radio. Le projet ALOHA a ensuite donné naissance au projet PRNet [KGBK78] qui traitait les communications multi-sauts, puis au projet SURAN [Bey90]. Pour un historique plus complet, le lecteur est invité à se référer aux ouvrages suivants [HT98, HDL+02, Per01, Toh02].

Un réseau ad hoc est un réseau peer-to-peer, *i.e.*, il permet à deux nœuds qui sont chacun à portée radio l'un de l'autre (conditions appropriées de propagation radio) de rentrer en communication directement. Notons que deux couples d'émetteurs/récepteurs qui sont suffisamment éloignés ont la possibilité d'émettre simultanément sur la même fréquence sans engendrer de collision. C'est ce qu'on appelle la *réutilisation spatiale* [Jac98]. Comme nous l'avons noté ci-dessus, si les conditions de propagation radio ne permettent pas d'établir un lien direct entre deux nœuds MANet (éloignement trop important entre l'émetteur et le récepteur), la mise en œuvre d'un routage multi-sauts est nécessaire afin d'acheminer les paquets de données jusqu'à leur destination finale. Les défis majeurs rencontrés dans ce type de réseau sont le calcul des routes et la mise en œuvre des algorithmes de routage de façon totalement distribuée du fait qu'il n'y a aucune entité centralisée au sein de cet environnement dynamique. On ne peut donc pas compter, comme dans les réseaux cellulaires classiques, sur la présence des nombreux *points fixes* que sont les BS (*Base Station*), HLR (*Home Location Register*), VLR (*Visitor Location Register*). Cette absence de tout point de coordination implique la mise en œuvre d'algorithmes distribués robustes.

Si les termes mobile, sans fil, ambiant sont devenus très à la mode, on peut (avant de présenter nos travaux ayant trait aux réseaux ad hoc) s'interroger sur les facteurs qui font que la mobilité au sens général diffère fondamentalement de l'informatique « *classique* ». Une première trivialité est de remarquer que les ordinateurs/hôtes sont plus petits/portables et que l'information émise utilise un support sans fil (radio, infra rouge) et non plus un support filaire (cuivre, fibre optique). Cette seule différence justifie-t-elle un tel écart entre le monde filaire et le nouveau monde du sans fil ? On peut tenter de caractériser ces différences, notamment en énonçant les contraintes imposées par la mobilité [HDL+02, Jac00, Sat96a, Sat96b] :

– Pour un coût et un niveau de technologie donnés, un hôte mobile sera toujours moins puissant (*i.e.*, plus pauvre en ressource de calcul, mémoire, bande passante) qu'un élément fixe. La bande passante disponible en sans fil est limitée et donc, c'est une ressource qu'il faut

économiser[1] en tentant de limiter la part octroyée à la gestion du réseau. De plus, le taux d'erreurs de transmission est plus important sur un lien radio que sur un lien filaire.
- La mobilité introduit de fait un caractère imprévisible et offre une plus grande vulnérabilité : les liens ne sont pas isolés créant des zones d'interférences étendues.
- La connectivité est intermittente, sporadique, variable en performance et en fiabilité, les propagations sont versatiles.
- Les éléments mobiles fonctionnent sur ressources propres et leur énergie est limitée. La portée de chaque élément est donc limitée par sa puissance d'émission (et par la réglementation des pays) qui influe directement sur la consommation d'énergie du mobile.

Il faut donc non seulement prendre en compte le fait que le médium de communication est sans fil mais que l'environnement est susceptible de se modifier constamment, notamment à cause de la mobilité. Cela implique la mise en œuvre de communications sans fil *adaptatives*, *i.e.*, ces systèmes doivent être conscients de leur environnement pour se configurer dynamiquement de façon distribuée et autonome. Le but ultime est peut-être de définir la station de base universelle capable d'opérer différents protocoles (TDMA, CDMA, OFDM/MC-CDMA), de supporter différentes technologies (smart antenna, Multiple User Detection), de faire de la gestion de puissance et de ressources et d'offrir des services réseaux avancés (ad hoc, auto-organisation, cellulaire, hybride). Cette adaptation forcenée soulève de nombreux défis car elle remet en question certains principes ou modèles établis et demande des innovations majeures sur l'allocation et l'utilisation du spectre radio disponible, sur les communications radio et leur mise en œuvre, sur le plan de l'architecture des réseaux, sur les applications et les services. Le modèle en couches hermétiques (Hardware/Device/Electronique faible consommation/Systèmes Radio/Communications/Réseau/Application Multi Média) apparaît comme obsolète dans une telle perspective et le maître mot devient l'intégration entre des domaines contigus. On imagine vite que tenter d'optimiser toutes les interactions possibles entre toutes ces couches est irréaliste et que l'un des problèmes difficiles à résoudre va être d'évaluer les compromis pertinents.

Le forum WWRF (Wireless World Research Forum), notamment au travers de la publication du *Book of Vision* [WWR01], donne de façon étendue les tendances du marché, de la technologie, des applications et des services. Notre but n'est pas de faire un tour d'horizon de l'ensemble des technologies impliquées dans cette marche vers « Le » réseau sans fil de demain[2], ni de prédire ce qui sera ou ce qui n'arrivera pas dans le monde du sans fil mais, plus humblement, de présenter nos réflexions et notre contribution sur les architectures de réseaux ad hoc et nos divergences avec l'approche parfois sectaire prônée par MANet, nos travaux sur les communications de groupe dans ce type de réseau et leur utilité dans la problématique de la découverte de service. Avant de présenter ces diverses contributions, je reviendrai brièvement dans la section 4.3 sur la classification des algorithmes de routage dans les réseaux ad hoc mais j'épargnerai au lecteur la présentation détaillée des $N > 37$ propositions existantes et recensées. Dans la section 4.4, je tenterai de donner un aperçu un peu plus détaillé des $M < 12$ propositions de protocoles de routage multicast.

[1] car ce qui est rare est cher !

[2] demain, est-ce après le 3G, le 4G, ou le NextG ?

4.2 Modèle pour les réseaux ad hoc

Usuellement, un réseau de communication est représenté dans sa forme la plus générale par un graphe orienté. Les sommets représentent les routeurs et les arcs modélisent la possibilité d'établir une communication directe entre deux sommets. S'il existe un arc du sommet u vers le sommet v, on dit que le sommet u est un *voisin* de v tandis que v est *accessible* depuis u. Le médium radio étant diffusant par nature, lorsqu'un nœud u diffuse un paquet (on emploie aussi le terme de broadcast ou de broadcast local) tous les nœuds qui sont accessibles depuis u reçoivent une copie de ce message. Les transmissions radio interférant fortement, un nœud v ne peut recevoir un message que si exactement un seul de ses voisins émet. S'il s'avère que plusieurs voisins émettent en même temps, il y a une collision et les messages arrivent brouillés et sont inexploitables par le nœud v.

On classe généralement les réseaux radio en quatre catégories. Si un nœud est en mesure de faire la distinction entre le bruit de fond et le bruit résultant d'une interférence on parle alors de réseau avec détection de collision, sinon on parle de réseau sans détection de collision. Si le graphe orienté est un graphe complet bidirectionnel, on parle de réseau à 1 saut sinon on parle de réseau multi-sauts. Les réseaux ad hoc, tels que nous les avons présentés sont donc des réseaux multi-sauts sans détection de collision. De plus, dans un réseau ad hoc, nous supposons que chaque nœud ne connaît que son propre identifiant : il ne connaît pas a priori les identifiants de ses voisins, ni le nombre de nœuds dans le réseau.

Notons que le modèle de communication n'intègre pas directement la mobilité des nœuds. Ce modèle doit être respecté à tout instant t, *i.e.*, à chaque étape de communication. Les contraintes que l'on vient d'énoncer sont proches des modèles de communication proposés pour les réseaux de communication [FL94, HHL86]. On est en face d'un modèle half-duplex 1-port en réception et Δ-port en émission. Ce parallèle doit permettre de réutiliser les travaux et les méthodes employés pour construire des bornes supérieures et inférieures sur le nombre d'étapes nécessaires pour effectuer certaines opérations de communication globale comme la diffusion.

Néanmoins, modéliser un réseau ad hoc par un graphe simple orienté pose, non pas un réel problème de modélisation car on vient d'exposer les contraintes qu'il faut respecter à tout instant mais plus un problème de représentation. La contrainte Δ-port implique que pour chaque envoi de message, l'ensemble des sommets accessibles reçoit une copie du message. Il est donc beaucoup plus judicieux d'employer les hyper-graphes [Ber73] à la place des graphes simples. Quand un sommet émet, l'ensemble des sommets appartenant à toutes ses hyper-arêtes reçoit le message. Cette formulation est équivalente à la précédente, j'en conviens parfaitement pour ce qui est des contraintes d'émission, mais en partant d'un hyper-graphe, on est beaucoup moins tenté de construire des arbres de multicast « classiques ». La notion de lien ne fait plus de sens dans un réseau sans fil : par exemple, que veut dire construire un arbre avec le moins de liens possible ? Si la ressource à minimiser est la bande passante, alors il ne faut plus raisonner en terme de liens mais bien en terme d'hyper-arêtes. Nous reviendrons dans la section 4.6 sur les conséquences de cette modélisation et sur la définition de nouveaux critères pour évaluer et comparer des arbres de multicast.

4.3 Classification des protocoles de routage unicast

Nous avons vu qu'il existe deux grandes classes de réseaux sans fil : les réseaux dits cellulaires par opposition aux réseaux multi-sauts sans infrastructure. Dans les réseaux cellulaires [LGT00] les utilisateurs communiquent avec une station de base qui est toujours à un saut radio de leur mobile, les stations étant elles-mêmes interconnectées au moyen d'un réseau filaire. À l'inverse, dans les réseaux multi-sauts sans infrastructure [CE95, Joh94, JT87, MW97, Per01], la notion de station de base n'existe pas et un message est susceptible d'effectuer plusieurs sauts, d'un mobile à un autre, pour atteindre sa destination finale.

L'ajout du routage entre les nœuds d'un réseau est une chose en soi bénéfique car cela augmente la fiabilité du réseau. C'est la mobilité qui l'est moins ! Prenons un modèle très simple où la probabilité de pouvoir effectuer une communication entre deux entités est simplement donnée par la probabilité p du lien radio. Si l'on suppose que le réseau est un réseau cellulaire, la probabilité de couverture d'un nouvel arrivant est simplement $P_c = p$, *i.e.*, la probabilité du lien entre la base et lui. Si maintenant on place ce nouvel arrivant dans un réseau ad hoc (du moins qui offre du routage de proche en proche) de n entités, la probabilité de couverture est $P_c = 1 - (1-p)^n$. Dans [Jac98], l'auteur définit la fiabilité d'un réseau comme la probabilité d'avoir la totalité des couples d'utilisateurs capables de communiquer entre eux. Pour un réseau cellulaire, il faut qu'il existe un lien entre les n nœuds et la station de base, soit une fiabilité de $(1-p)^n$. Dans un réseau ad hoc, la fiabilité est égale à la probabilité que le réseau ne contienne qu'une seule composante connexe. Cette probabilité est très élevée (> 0.99) même pour un p relativement faible (0.1).

Pour se convaincre, exprimons la probabilité $P_d(h)$ d'être à h sauts d'un sommet v donné. La probabilité d'être à distance h d'un sommet v donné est égale à la probabilité de ne pas être à une distance inférieure à h de v multipliée par la probabilité d'être couvert par un ou plusieurs des nœuds qui sont à une distance inférieure à h de v. La probabilité $P_d(h)$, d'être à une distance h d'un sommet v donné est :

$$P_d(h) = \frac{n - N_{h-1}}{n}\left(1 - (1-p)^{N_{h-1}}\right)$$

où N_h est le nombre de sommets qui sont à une distance inférieure à h de v (v compris). N_h s'obtient au moyen de la formule de récurrence suivante :

$$\begin{aligned} N_h &= \sum_{i=1}^{h}(n - N_{i-1})(1 - (1-p)^{N_{i-1}}) \\ N_0 &= 1 \end{aligned}$$

La figure 4.1 trace la probabilité $P_h(h)$.

Historiquement, ou traditionnellement, les protocoles de routage dans les réseaux ad hoc se divisaient en deux grandes familles : les proactifs et les réactifs[3]. À l'heure actuelle, il est un peu plus difficile de conserver ce découpage manichéen et une troisième classe est apparue, celle des protocoles hybrides ou hiérarchiques. On peut même envisager de définir une quatrième classe, celle des protocoles géographiques. Il semble a priori normal d'avoir différents types de protocole

[3] à l'image des Petits-Boutistes et des Gros-Boutistes dans les voyages de Gulliver.

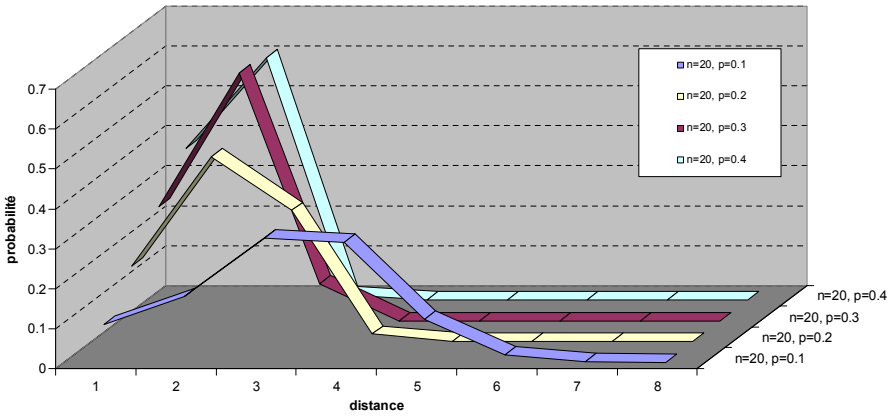


FIG. 4.1 – Probabilité $P_d(h)$ en fonction de n et p.

étant donné que les scénarios d'usage des réseaux ad hoc sont aussi très variés, allant du *Wireless Wellness Monitor* [Pär01, PvGT+00][4] aux réseaux autoroutiers à grande échelle [MJK+00].

4.3.1 Les protocoles proactifs

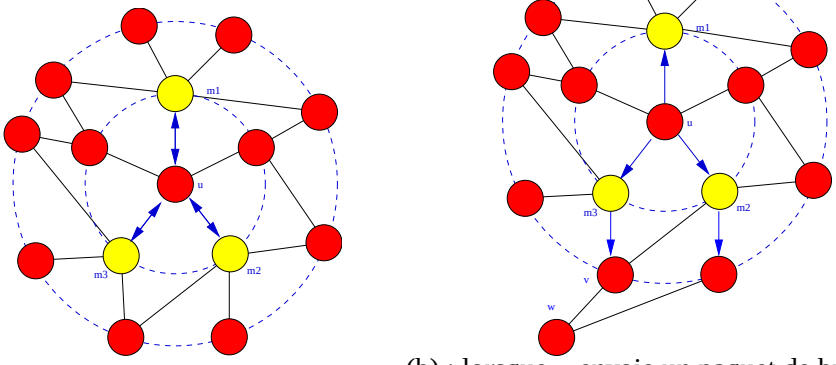
La caractéristique majeure des protocoles proactifs[5] est qu'ils tiennent continuellement à jour une vue de la topologie afin d'avoir à tout instant une route disponible pour toute paire de nœuds du réseau. Cette connaissance de la topologie provient des échanges, entre les nœuds du réseau, de messages de contrôle contenant les informations requises de topologie.

Les premiers protocoles proposés [CRKGLA89, GLA93, PB94] pour les réseaux ad hoc furent des algorithmes proactifs basés sur une technique de type *vecteur de distance*, version distribuée de l'algorithme de BELLMAN-FORD où des modifications furent apportées pour traiter le problème de la convergence de l'algorithme et le problème lié au fort taux de trafic de contrôle.

L'alternative pour résoudre les problèmes de convergence des algorithmes de type vecteur de distance est de mettre en œuvre un algorithme à *état de lien*. Le protocole OLSR (*Optimized Link State Routing*) [Lao02, JMQ+01, JMQ+02] est l'un des protocoles en lice pour la standardisation au sein du groupe de travail MANet de l'IETF. Nous décrivons un peu plus en détail le protocole OLSR du fait que le projet HIPERCOM/INRIA a contribué avec succès à sa définition et à sa défense. De plus, il recèle des mécanismes optimisés qui peuvent être employés hors du contexte OLSR, et servir à d'autres approches, même réactives ! Afin de justifier ses lettres de noblesses et de mériter le « *O* » de son nom, un nœud OLSR ne diffuse pas l'ensemble de son voisinage comme dans un protocole LSR classique mais seulement un sous-ensemble de ce voisinage. Cet ensemble, nommé *ensemble de MPR* (*MultiPoints Relais*) est choisi de telle sorte qu'il couvre l'ensemble des sommets accessibles à distance 2 (voir figure 4.2(a)). Ces MPR permettent de minimiser le trafic de contrôle, d'effectuer des diffusions optimisées des messages de contrôle (voir figure 4.2(b)) et de construire des plus courts chemins. Pour calculer ses MPR, chaque nœud doit connaître son 2-

[4] ou comment recevoir son programme de TBC (Total Body Conditionning *dixit* Isabelle) chaque fois que l'on ouvre le frigo.

[5] CGSR, DBF, DSDV, DTDV, HSLS, LCA, OLSR, STAR, TBRPF, WRP.



(a) : u sélectionne ses MPR $\{m1, m2, m3\}$ parmi ses voisins de telle sorte que les sommets de son 2-voisinage soient couverts.

(b) : lorsque u envoie un paquet de broadcast, uniquement ses MPR vont relayer le paquet. Si v reçoit le paquet en premier de $m3$ il le retransmet. Par contre si v reçoit le paquet de $m2$ en premier il ne le retransmet pas. Néanmoins, aucun sommet ne sera oublié. Tout voisin w de v est à distance 2 de $m2$ et donc l'un de ses MPR le couvrira.

FIG. 4.2 – Multipoint relais dans OLSR (dessins empruntés à L. VIENNOT).

voisinage. Pour ce, chaque nœud OLSR échange avec ses voisins des paquets `HELLO` contenant la liste des ses voisins. Les MPR de chaque nœud sont diffusés dans tout le réseau par des messages `TC` (*Topology Control*). Cette information permet à chaque sommet d'avoir une vue du réseau (l'ensemble des nœuds et un sous-ensemble des liens) lui permettant de calculer une table de routage des plus courts chemins.

Une autre approche employée par TBRPF (*Topology Broadcast based on Reverse-Path Forwarding*) [BOT02] est l'utilisation de la notion de reverse path forwarding pour diffuser les changements de topologie dans le sens inverse des arbres de diffusion maintenus par l'ensemble des nœuds vers le nœud source de la mise à jour. Pour ce, les nœuds TBRPF s'échangent leur voisinage et diffusent leur arbre de topologie. Il existe d'autres propositions alliant parfois certains aspects des protocoles de type vecteur de distance et des protocoles à état de lien [HDL+02, MGLA96].

L'inconvénient des protocoles proactifs est le coût induit par la mise à jour continuelle des informations de topologie et ce, même si la topologie est faiblement dynamique, ou si l'activité au sein du réseau est réduite. Il se peut que l'investissement réalisé en terme de maintien et de calcul des tables de routage soit perdu si *in fine*, les informations sur la topologie ne sont jamais utilisées. Ce type de protocole utilise continuellement une certaine partie de la bande passante du réseau.

4.3.2 Les protocoles réactifs

Les protocoles réactifs[6] abordent le problème d'une façon très différente et se basent sur un dialogue de type « *question/réponse* » initié à la demande d'un nœud cherchant à joindre une destination. Les autres nœuds connaissant une route vers la destination recherchée lui envoient leurs réponses. Ces protocoles ne cherchent pas à maintenir une vue à jour de la topologie du réseau mais ils ne gardent et établissent une route que lorsqu'une communication doit être établie. A priori, sans aucune connaissance de la topologie, le seul moyen de découvrir où se trouve son correspondant est d'inonder le réseau avec un message de type ROUTE QUERY. Les protocoles employant ce type de technique sont AODV (*Ad hoc On Demand Distance Vector routing protocol*) [PRD02, PR99], DSR(*Dynamic Source Routing protocol*) [JM96] et TORA (*Temporally-Ordered Routing Algorithm routing protocol*) [PC97, PC01].

AODV et DSR sont très similaires et renvoient les messages ROUTE REPLY en unicast en utilisant le chemin inverse du message ROUTE QUERY. Dans DSR, un chemin (pas nécessairement le chemin inverse) est mémorisé dans le message ROUTE REPLY car DSR utilise un routage par la source, ce qui permet d'éviter les boucles. AODV se fonde lui sur une technique de type *vecteur de distance*. L'information sur la route trouvée est mémorisée sous forme de *next-hop*. AODV traite le problème des boucles en introduisant un numéro de séquence dans les paquets de contrôle. Dans TORA, le message ROUTE REPLY est diffusé afin de disséminer l'information de routage sous la forme d'un DAG (*Directed Acyclic Graph*) enraciné en la destination.

Si les approches réactives engendrent a priori un trafic de contrôle moindre, notamment durant les périodes de faible activité du réseau, elles présentent néanmoins divers inconvénients. Le processus d'inondation perturbe tout le réseau et est gourmand en bande passante. Si une route casse, des mécanismes supplémentaires de maintenance sont introduits et génèrent à leur tour un surcroît de trafic pour chaque initialisation ou réparation de route. De plus, dans ces approches, il est nécessaire de bufferiser les données dans l'attente d'une éventuelle route, ce qui engendre des délais plus important à chaque défaut de route.

4.3.3 Les protocoles hybrides

Les deux familles précédentes, proactives et réactives, ont chacune leurs partisans et leurs détracteurs au sein du groupe de travail MANet. À l'heure actuelle, il est hasardeux de pronostiquer quel sera le propre point de convergence du groupe de travail MANet. Peut-être verrons nous deux standards, un proactif et un réactif, laissant ainsi le champ libre à d'autres propositions d'intéropabilité entre les deux modes. Une alternative à ces deux solutions extrêmes réside peut-être dans les solutions hybrides ou hiérarchiques[7]. Ces solutions sont qualifiées d'hybrides car elles incorporent, pour la plupart, des aspects réactifs et proactifs. La majorité emploie un protocole proactif dans une zone formant un voisinage plus ou moins étendu et un protocole réactif pour découvrir une route vers une destination qui se trouve dans une autre zone. Ainsi, si la destination se trouve dans la zone de l'émetteur, une route est disponible immédiatement. Le découpage en zones permet aussi d'optimiser le processus de recherche réactif puisqu'un nœud qui reçoit un message ROUTE QUERY est en mesure de répondre immédiatement si la destina-

[6] ABR, AODV, BSR, DSR, DSRFLOW, FORP, LMR, LUNAR, RDMAR, SSR, TORA.
[7] BRP, CBRP, CEDAR, FSR, GSR, HARP, HSR, IARP, IERP, LANMAR, ZRP.

tion se trouve dans sa zone ou non. Le protocole ZRP [PH99] illustre très bien ce compromis recherché. FSR [GHP01, PGC00] n'est pas réellement un protocole hybride et pourrait être catalogué dans la famille des protocoles proactifs. Il tire partie du *Fisheye* (littéralement *œil de poisson* [KS71]), où les informations sont d'autant plus précises que l'on se rapproche du point focal, *i.e.*, le centre de l'œil. Ainsi, dans FSR, l'hypothèse faite est que les changements de topologie lointains ont une moindre influence sur le calcul des routes locales, les informations sur la topologie ne sont pas relayées systématiquement. Tous les protocoles nommés ci-dessus ont une vue à plat du réseau, ce qui n'est pas le cas de LANMAR [GHMP01, PGH00] qui suppose que le réseau est une collection de sous-réseaux prédéfinis, chacun ayant son *landmark* (chef de groupe). Cette notion avait été proposée pour les grands réseaux filaires dans [Tsu88]. Les routes vers les différents landmarks (donc vers chacun des sous-réseaux) sont maintenues de façon proactive par un algorithme de type vecteur de distance et au sein d'un sous-réseau, chaque nœud maintient une table de routage au moyen d'un protocole proactif et a donc une vue précise de sa zone. LANMAR emploie donc une hiérarchie de protocoles proactifs, ce qui permet de le cataloguer proactif et hiérarchique.

4.3.4 Les protocoles géographiques

Cette famille de protocoles géographiques[8] se fonde sur des informations géographiques externes (par exemple obtenues par GPS) et/ou sur la position des nœuds pour trouver des routes. Ce type d'approche peut présenter des inconvénients en environnement *indoor* ou en milieu très urbain où la proximité géographique n'induit pas la proximité au sens radio *i.e.*, la possibilité de communiquer. Néanmoins, couplées à des systèmes de localisation locaux (entre différents amers), ces techniques peuvent être intéressantes.

4.4 Les protocoles de routage multicast

La grande majorité des algorithmes de multicast au sein des réseaux filaires met en œuvre une structure d'arbre (partagé ou non). L'arbre est le moyen le plus efficace en terme de ressources permettant de connecter n nœuds et il garantit la non duplication des données. De plus, les décisions de routage sont très simples et se limitent à retransmettre les données sur les autres interfaces, exceptée celle par laquelle le message est arrivé. Vouloir transposer directement ces principes aux réseaux sans fil peut se révéler très inefficace. Il ne faut pas perdre de vue que l'emploi d'un protocole de multicast pour envoyer une donnée à un ensemble de destinataires doit permettre de réduire le nombre de ressources réseau employées. De plus, la mise en œuvre d'un protocole de multicast peut s'avérer utile car elle offre aussi un moyen robuste pour joindre des destinataires dont l'adresse n'est pas connue a priori ou qui change régulièrement. Comme nous l'avons souligné plus haut, il est important de réduire le nombre de transmissions (et la consommation d'énergie au sein des mobiles) dans un réseau sans fil car la bande passante est limitée. Le multicast doit permettre d'optimiser la gestion du médium radio en évitant les retransmissions superflues de messages et en tirant parti de la caractéristique de diffusion inhérente au médium radio.

[8] DREAM, GLS(Grid), LAR, ZHLS.

Comme il est toujours difficile de faire table rase du passé, les premières propositions de protocole de multicast pour les réseaux ad hoc furent des adaptations de techniques déjà présentes dans le filaire. Dans [GCZ98], les auteurs proposent une adaptation du protocole DVMRP pour construire un arbre enraciné en chaque source et dans [CGZ97] on trouve une adaptation des principes de PIM-SM pour construire un arbre partagé. Si une classification binaire des algorithmes de routage dans les réseaux filaires peut se faire selon qu'ils utilisent un arbre enraciné à la source ou un arbre partagé, elle devient plus complexe dans les réseaux sans fil. Les autres classificateurs possibles sont : approche réactive ou proactive ; utilisation d'une structure d'arbre ou non ; extension directe d'un protocole de routage unicast ou non. Le problème de ce trop grand nombre de critères de classification est que chaque proposition peut, en tant que singleton, être sa propre classe.

Nous allons essayer de présenter les différentes propositions[9] et de les regrouper, un peu arbitrairement, par type. Contrairement au routage multicast filaire, notre premier critère sera le type de structure employée. En effet, certains protocoles ne cherchent pas à construire un arbre mais un maillage au sein du réseau ou tentent de mettre en œuvre un mécanisme d'inondation approprié.

4.4.1 Multicast employant une structure d'arbre

Les protocoles ABAM, DDM, DVMRP, MOLSR et MZR sont des protocoles de routage multicast employant des arbres spécifiques par source. Les protocoles AMRIS, AMRoute et MAODV sont des protocoles de routage multicast mettant en œuvre des arbres partagés. Notons que MOLSR, MAODV et MZR (dans une certaine mesure), sont des extensions de protocoles unicast existants. AMRIS se passe de routage unicast. DVMRP, et AMRoute sont indépendants du routage unicast sous-jacent.

DVMRP

Dans [GCZ98], les auteurs soulèvent les problèmes spécifiques à la mise en œuvre de DVMRP au sein d'un réseau ad hoc. DVMRP (voir la description donnée page 38 et dans [WPD88, DC90]) repose sur un processus d'inondation (*flooding*) dans tout le réseau suivi d'un processus d'élagage (*pruning*). L'un des problèmes qui se pose dans un réseau ad hoc est de détecter les feuilles de l'arbre qui doivent initier l'élagage. En effet, dans un réseau filaire, un nœud peut très facilement se rendre compte s'il est une feuille puisqu'il connaît exactement son nombre d'interfaces. Cette information n'est plus valide dans les réseaux sans fil où tous les nœuds sont des routeurs et, il n'existe pas d'information précise pour chaque « *lien* » qui n'est plus isolé : tous les voisins sont connectés à la même interface. L'autre problème est la dynamique des flooding. Dans DVMRP, ces flooding périodiques sont nécessaires pour prendre en compte les changements de topologie et l'arrivée de nouveaux membres. Pour limiter ces inondations, un mécanisme de greffe (*graft)* explicite est mis en œuvre pour se reconnecter à l'arbre. L'un des inconvénients de DVMRP est qu'il inonde régulièrement le réseau avec des données (et non pas que des paquets de contrôle) ce qui peut être gourmand en ressource radio.

[9] ABAM, ADMR, AMRIS, AMRoute, CAMP, DDM, DSR-MB, DVMRP, LAM, MAODV, MCEDAR, MOLSR, MZR, NSMP, ODMRP, SRMP, XMMAN.

Differential Destination Multicast (DDM)

DDM [JC00] est un protocole de multicast qui construit un arbre spécifique par source. Ce protocole emploie une approche très différente des autres protocoles proposés. Premièrement, au lieu de distribuer la gestion des membres au sein du réseau, DDM centralise les adhésions à la source, lui donnant ainsi accès à la liste des membres. Deuxièmement, au lieu de vouloir maintenir un arbre par des états dans les routeurs, DDM emploie une technique s'apparentant à du source routing. En effet, un entête de longueur variable, encodant la liste des destinations est introduit dans chaque paquet de données (du moins dans le premier paquet et uniquement un différentiel dans les suivants).

Multicast Zone Routing (MZR)

MZR [Dev00] reprend le concept de structure hiérarchique employée par ZRP [PH99] et renferme deux parties : une approche proactive est employée au niveau de chaque zone de routage et une approche réactive entre les zones de routage. La création d'un arbre multicast s'opère en deux étapes. La source commence par informer tous les membres de sa zone en leur envoyant en unicast un TREE CREATE et les nœuds intéressés acquittent ce message. Cette première étape construit classiquement un arbre à partir des chemins inverses. Une fois cette étape finie, la source émet un message TREE PROPAGATE à destination de ses nœuds frontières qui vont à leur tour initier la création d'un arbre multicast dans leur propre zone. À la réception d'un acquittement, un nœud frontière envoie lui aussi un message d'acquittement à la source, permettant d'établir un lien entre la source et lui-même. Ce processus de création locale/propagation se répète jusqu'à ce que tous les nœuds de l'arbre aient reçu l'annonce de création. L'envoi périodique de messages TREE REFRESH par la source permet de maintenir l'arbre. En cas de rupture d'un lien, un nœud peut effectuer une recherche dans sa zone pour se raccrocher et si cette tentative échoue, il demande à ses nœuds frontières de prendre en charge son rattachement à l'arbre.

MZR introduit une très forte latence à la création de l'arbre. De plus, MZR ne profite que partiellement de la structure en zone car lors d'une perte de lien, si un nœud n'est pas en mesure de se rattacher au sein de sa zone, une inondation de tout le réseau va être effectuée. Cette situation est peut-être acceptable pour les réparations mais le problème est que ce processus est aussi employé pour les demandes d'adhésion des nouveaux membres.

Multicast Optimized Link State Routing (MOLSR)

MOLSR [JLV$^+$01, Lao02] est une extension basée sur OLSR et propose une approche proactive du multicast. Contrairement à M-AODV, les branches de l'arbre de diffusion ne sont pas construites par un mécanisme de découverte de route mais en utilisant la connaissance du réseau possédée par chaque nœud. MOLSR laisse la possibilité à chaque nœud de s'impliquer, ou non, dans le multicast. Les nœuds implantant le multicast s'identifient en inondant le réseau avec un message de type « Je suis nœud multicast ». Comme pour le routage unicast, chaque nœud calcule en local la liste de ses Multicast MPR (MMPR), c'est-à-dire un ensemble de ses voisins permettant d'atteindre tous les nœuds à distance deux. Dans ces calculs, ne sont pris en compte que les nœuds supportant le multicast. Ensuite, un algorithme de plus courts chemins (création de routes optimales vis-à-vis du sous-réseau multicast) permet à chaque nœud de calculer le prochain relais

vers tous les nœuds pouvant potentiellement émettre des données. Ces mécanismes correspondent à ceux mis en œuvre pour le routage unicast sauf que seuls les nœuds multicast sont considérés. L'arbre de diffusion est créé de manière inversée. Lorsqu'une source désire envoyer des données à destination d'une adresse multicast, elle diffuse un message de type `SOURCE CLAIM`. Ce message est reçu par tous les nœuds du réseau mais n'est pris en compte que par les membres du groupe. Ces derniers se rattachent à l'arbre. Pour cela, un nœud sélectionne parmi ses MMPR celui qui lui permet de joindre la source. Ce MMPR devient son père dans l'arbre de diffusion. Pour indiquer la création de cette branche, le nœud émet un message de type `CONFIRM PARENT` vers son père. Ce dernier se greffe ensuite à l'arbre suivant le même processus.

Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)

AMRIS [WTT98, WTT99] est un protocole à la demande qui construit un arbre partagé. Chaque nœud participant à la session multicast possède son propre identificateur de session `MSM-ID`. Cet identificateur croît au fur et à mesure que l'on s'éloigne d'un nœud central nommé SID. L'initialisation de l'arbre est effectué par le SID qui diffuse son propre identifiant permettant aux autres sommets de calculer le leur avant de retransmettre le message dans lequel ils placent leur propre `MSM-ID`. AMRIS est indépendant de tout protocole de routage et utilise ses propres messages d'avertissement périodiques afin de maintenir sa propre table de voisinage. En cas de perte d'un lien, la responsabilité de se raccrocher à l'arbre incombe au fils. Si ce dernier possède un parent dans son voisinage, *i.e.*, un nœud ayant un `MSM-ID` plus petit, il lui envoie une demande d'adhésion et ce dernier doit relayer cette demande. S'il n'y a aucun membre potentiel, il diffuse un message de `JOIN` en utilisant une technique de recherche par anneau croissant dont la portée du premier message (le TTL) est limitée à r sauts. Les sommets qui reçoivent ce message et qui sont dans l'arbre doivent l'acquitter pour informer le sommet initiateur de cette recherche.

Ad hoc Multicast Routing (AMRoute)

AMRoute [BLMT98] est un protocole de routage multicast qui se veut robuste de par l'utilisation d'arbres multicast applicatifs et la mise en œuvre de cores logiques. Ce protocole crée un arbre partagé bidirectionnel, servant à la diffusion des données, uniquement entre les émetteurs et les récepteurs du groupe en mettant en place des tunnels unicast qui servent de lien entre les nœuds de l'arbre multicast applicatif. La structure de l'arbre peut rester identique même en cas de changement de topologie. Certains nœuds de l'arbre jouent le rôle de *core logique* et sont responsables de la mise en œuvre et de la gestion de la signalisation d'AMRoute comme la détection de nouveaux membres et la mise en place des liens (tunnel) de l'arbre. À l'inverse des protocoles de type CBF (CBT, PIM-SM), ces cores logiques ne sont pas des points fixes et ne représentent pas un point faible de l'architecture. De même, pour augmenter la robustesse de la construction de l'arbre, AMRoute utilise un flooding périodique, tout comme DVMRP, mais en ne diffusant que des paquets de contrôles et non pas des données.

Multicast Ad hoc On-Demand Distance Vector routing (MAODV)

Comme son nom l'indique, MAODV [RP99] est une extension du protocole de routage unicast AODV [PRD02, PR99] et à ce titre, il va employer les mécanismes de découverte et d'activation

de route utilisés par AODV en unicast. MAODV maintient un arbre de diffusion bidirectionnel. Les branches de cet arbre sont créées dynamiquement lorsqu'un nœud s'inscrit au groupe en émettant un message `ROUTE QUERY` à destination du groupe multicast. Ce message correspond à une découverte de route unicast et va inonder le réseau. Seuls les nœuds déjà membres de l'arbre multicast sont autorisés à répondre à cette requête par un message `ROUTE REPLY`. Contrairement à l'unicast où une route n'est entretenue que si elle est activée, les changements de topologie au sein du réseau nécessitent une maintenance active de l'arbre. Lorsqu'un lien de l'arbre se brise, il incombe au nœud en aval de l'arête de relier les deux parties de l'arbre grâce au protocole de découverte de route. Plusieurs mécanismes permettent d'éviter la création de boucles lors de ces phases. Si le réseau est déconnecté, la fusion des deux sous-arbres est impossible et le nœud en aval devient alors le leader du groupe au sein de son sous-réseau. Le leader du groupe a aussi la charge de maintenir le numéro de séquence du groupe multicast. Ce numéro est diffusé périodiquement dans le réseau via un paquet `HELLO GROUP` permettant d'identifier les situations où deux réseaux fusionnent à nouveau.

Bien que ce protocole soit *on demand* et qu'il soit basé sur un protocole de routage réactif, il introduit plusieurs mécanismes que l'on attribue d'ordinaire plus volontiers aux protocoles proactifs : présence de messages `HELLO` diffusés régulièrement, maintien des branches de l'arbre. Vouloir mettre en place une structure d'arbre multicast réactive apparaît un peu antinomique car elle a besoin d'acquérir des informations de façon proactive pour maintenir la structure initiale.

Associativity-Based Ad Hoc Multicast (ABAM)

ABAM [Toh02, TB01] est un protocole de routage multicast utilisant un arbre spécifique à la source. La construction de l'arbre s'effectue en trois phases. Dans la première phase, la source informe tous les autres sommets de sa présence en diffusant un message `QUERY MULTICAST`. L'ensemble des nœuds intéressés par ce groupe va dans une deuxième étape répondre à la source en lui envoyant un message `QUERY REPLY`. À partir des informations reçues, la source va appliquer un algorithme de sélection pour construire son arbre à partir de critères de stabilité, de minimisation... Une fois cette sélection effectuée, la source va envoyer un message `SETUP` en utilisant une technique de source routing à tous ses récepteurs. Ce message (qui est composé de sommets forwarding/branching/receiving) va configurer les tables de routage multicast de tous les nœuds listés dans le message. Pour adhérer à l'arbre, un nouveau nœud diffuse son message `JOIN` en utilisant une méthode d'anneau croissant par exemple. Les sommets présents dans l'arbre doivent acquitter ce message et le nouveau nœud choisit le chemin qu'il préfère en confirmant son choix par un message de type `SETUP`. Un mécanisme de reconstruction de route est mis en place suivant que la source, les nœuds de l'arbre ou les récepteurs bougent.

4.4.2 Multicast employant un maillage

Afin d'éviter les inconvénients inhérents aux arbres comme la fragilité de la structure due à sa 1-connexité, la nécessité de les reconfigurer régulièrement dans un environnement fortement mobile, certains travaux ont proposé de maintenir une structure maillée (*mesh*) qui est plus robuste car redondante. Le protocole CAMP est dépendant du protocole unicast sous-jacent car il repose sur l'utilisation de certaines informations comme la validité d'un plus court chemin.

Core-Assisted Mesh Protocol (CAMP)

CAMP [GLAM98] est un protocole de routage multicast utilisant un maillage partagé pour chaque groupe de multicast. Un ou plusieurs sommets jouent le rôle de core pour prendre en charge les opérations d'adhésion, supprimant ainsi les opérations d'inondation. Ces sommets ne sont pas forcément des membres du groupe. Par contre, chaque core s'enregistre auprès des autres cores pour établir un maillage entre eux. Un nœud qui veut adhérer vérifie dans un premier temps si parmi ses voisins certains sont déjà membres du groupe ; si tel est le cas, il les notifie sinon, il cherche à joindre l'un des cores. Le chemin pour joindre le core va alors être incorporé entièrement au maillage. Si le ou les cores ne sont pas joignables, un nœud désirant adhérer au groupe a toujours la possibilité de le faire par un processus de recherche par anneau croissant. Chaque sommet du maillage tient à jour la liste des sommets dont il est responsable (*i.e.*, s'il est le prochain hop entre le sommet et une des sources du groupe) en se basant sur la table unicast. CAMP utilise un mécanisme de *heart beat* pour s'assurer que tous les chemins inverses entre les sources et les récepteurs sont bien inclus dans le maillage.

On-Demand Multicast Routing Protocol (ODMRP)

ODMRP [GLAM98] est un protocole de routage multicast utilisant une technique d'inondation. Cependant, contrairement à DVMRP, les données ne sont pas transmises par inondation mais elles sont relayées par un sous-groupe de sommets nommé *forwarding group* qui eux sont maintenus par une inondation périodique de messages de contrôle. Cette notion de *forwarding group* (FG), introduite dans FGMP (Forwarding Group Multicast Protocol) [CGZ98] est un sous-ensemble de nœuds choisi pour relayer les données multicast à destination d'un groupe multicast donné. Les nœuds d'un FG doivent garantir l'existence d'au moins un chemin entre chaque source du groupe multicast et chaque récepteur du groupe. Dans cette approche, toutes les sources d'un même groupe participent à la création de la même structure de multicast. Lorsqu'une source souhaite envoyer des données à destination d'un groupe multicast, elle diffuse périodiquement un message `JOIN QUERY`. Chaque sommet qui reçoit ce message mémorise dans sa table de routage unicast le nœud en amont. Cette information permet de connaître le chemin unicast inverse vers la source. Quand un récepteur reçoit le message `JOIN QUERY`, ses voisins qui sont sur un chemin inverse vers l'une des sources du groupe multicast deviennent des FG. Ce processus de remontée vers la source se poursuit. Cette méthode construit l'union de tous les arbres spécifiques à chaque source. Cependant, les nœuds intermédiaires dans cette structure sauvegardent uniquement l'information indiquant qu'ils sont des Forwarding Group pour un groupe de multicast donné. Cette information ne dépend pas de la source. Afin de minimiser le trafic, un nœud du maillage ne retransmet pas un paquet multicast dupliqué.

Multicasting Core-Extraction Distributed Ad Hoc Routing (MCEDAR)

MCEDAR [SSB99] est un protocole de routage multicast basé sur une extension du protocole SPINE [SDB98]. Le mécanisme de *Core-Extraction* présent dans SPINE est un algorithme distribué qui calcule un ensemble dominant des nœuds du réseau, ce qui permet d'auto-configurer un réseau dorsal au sein du réseau ad hoc. Chaque nœud dominant (*i.e.*, qui est dans le réseau dorsal) connaît le nœud dominant le plus proche et les nœuds qu'il domine. SPINE emploie un

broadcast basé sur ce réseau dorsal plutôt qu'une inondation du réseau pour découvrir les routes. L'infrastructure mise en œuvre pour réaliser du multicast repose entièrement sur le même principe. Chaque groupe de multicast extrait un sous-graphe du réseau dorsal et la diffusion de données est exécutée sur ce sous-graphe en employant le même mécanisme que pour le broadcast. La diffusion d'informations se fait donc sur la base d'un arbre qui, lui, est calculé sur un maillage.

4.4.3 Limitation des protocoles actuels

Les arbres de diffusion construits par une approche réactive, comme dans MAODV, peuvent avoir de nombreux défauts. Les routes construites par ce mécanisme, que ce soit en unicast ou en multicast, ne sont pas optimales. Cela peut entraîner une sur-utilisation du médium importante. L'étude dans un modèle de graphe aléatoire unidimensionnel et bidimensionnel proposée par [JV02] montre que le rapport de la longueur des routes créées sur celle des routes optimales est en moyenne de $4/3$ pour le cas unidimensionnel et de $5/3$ pour le cas bidimensionnel. Dans le cas d'un arbre de diffusion où chaque branche est créée par ce mécanisme, le surcoût peut se révéler important.

L'utilisation d'une approche proactive permet de résoudre certains problèmes. Par exemple, MOLSR assure que les routes construites sont minimales en terme de distance dans le sous-graphe des nœuds supportant le multicast. Ainsi, les branches de l'arbre de diffusion sont des routes de longueur minimale entre la feuille et la source. Cependant, MOLSR ne se demande pas s'il vaut mieux s'accrocher au plus tôt à la source ou, au plus tôt à l'arbre. La politique appliquée par MOLSR peut entraîner la création d'un arbre avec deux branches parallèles (guirlandes) très longues. Ce genre de phénomène n'est pas souhaitable dans les réseaux ad hoc où le médium de communication est précieux. Ce phénomène est aussi aggravé du fait que MOLSR a une vision limitée du réseau. Cette vision est composée des adjacences entre les nœuds et leurs MMPR. Cela signifie que l'arbre ne peut être construit qu'à partir des arêtes entre MMPR, seules arêtes visibles. Ce genre de problème n'est pas propre à MOLSR, il se pose dès que la vision du réseau proposée par le protocole de routage est incomplète.

S'il est reconnu que l'optimisation de la bande passante est un problème important dans les réseaux sans fil, il semble que les protocoles actuels tentent surtout d'optimiser le surcoût du trafic de contrôle induit par la mise en œuvre d'un arbre/maillage pour réaliser une opération de multicast, sans réellement résoudre le problème de l'optimisation de la bande passante consommée par les (re)diffusions des données. On peut néanmoins supposer que la diffusion des données est plus consommatrice que le trafic de contrôle. Il faut donc apporter une attention toute particulière à ce point, sans pour autant négliger le surcoût dû au trafic de contrôle.

4.5 Architecture de réseaux ad hoc

Nos premiers travaux ayant trait aux réseaux ad hoc furent centrés autour de la conception d'un protocole de multicast dans le cadre d'une ARC INRIA (Action de Recherche Coopérative) nommée COMPAS. Nos premières réflexions sur la façon de construire un arbre de multicast, notamment au dessus du protocole OLSR, nous poussèrent à mieux prendre en compte l'une des caractéristiques intrinsèques des réseaux sans fil, qui est que les transmissions ne peuvent pas être sélectives (du moins pas avec des antennes omnidirectionnelles). Cette première évidence nous

a conforté dans le fait que la modélisation au moyen de graphes simples n'est pas la bonne, que vouloir construire des arbres et les optimiser de façon à avoir des branches arêtes disjointes ne fait pas réellement sens mais qu'une représentation par des hyper-arêtes est plus adaptée au modèle même de diffusion locale. Cette nouvelle vision et modélisation a donné naissance à un protocole de routage unicast proactif nommé JUMBO [CFG00, Che01].

L'idée sous-jacente était d'arriver à prendre en compte et à répercuter cette notion d'hyper-arête. Dans les faits, une hyper-arête n'est rien d'autre qu'un ensemble de sommets qui sont tous à portée les uns des autres. En terme de graphe, il nous fallait décomposer le graphe des voisinages, *i.e*, un graphe simple, en une famille $\mathcal{C} = \mathcal{C}_1, \ldots, \mathcal{C}_\ell$ de cliques. C'est cette décomposition en cliques qu'il faut utiliser pour calculer des routes. Un chemin entre deux sommets du graphe n'est plus composé d'arêtes comme il est d'usage dans les réseaux filaires, mais il est décrit en terme de cliques. Le choix du sommet responsable de la retransmission au sein de chacune des cliques permet d'obtenir un degré de liberté supplémentaire. On peut appliquer le même genre de raisonnement pour la construction d'un arbre de multicast. Ce qui nous intéresse ce n'est plus un arbre recouvrant tous les membres du groupe mais bien un arbre recouvrant l'ensemble des cliques où se trouvent les membres du groupe. Nous imposons que la décomposition en cliques du réseau respecte certaines contraintes :

Propriété 4.1 *L'union des cliques de la décomposition est égale au graphe d'adjacence G.*

Cette première propriété permet de garantir que toute arête du graphe d'adjacence G est couverte par au moins un élément de la décomposition, et inversement, un élément de la décomposition ne doit pas couvrir des arêtes n'étant pas dans le graphe d'adjacence G.

Propriété 4.2 *La décomposition $\mathcal{C} = \mathcal{C}_1, \ldots, \mathcal{C}_\ell$ du graphe G doit vérifier $\ell \leq m$ où $m = |E|$ correspond au nombre d'arêtes de G.*

Cette dernière propriété permet d'éviter une gestion trop coûteuse du routage, en raison d'un nombre trop élevé de cliques. Notons que la solution qui consisterait à construire une décomposition du graphe minimale en nombre d'éléments est elle aussi très coûteuse (le problème MINIMUM CLIQUE COVER est NP-complet [GJ79]). La propriété 4.2 permet de garantir un bon compromis entre les deux extrêmes.

En se basant sur une approche proactive (similaire à OLSR) où chaque sommet connaît, grâce à l'envoi de paquets `HELLO`, son voisinage à 2 sauts, nous avons proposé dans [CFG00, Che01] un algorithme distribué pour construire une décomposition du graphe en sous-graphes complets qui vérifie les propriétés 4.1 et 4.2. La diffusion des cliques dans tout le réseau (et non plus des MPR) permet d'avoir en chaque nœud la connaissance complète du graphe. L'intérêt des cliques est de permettre la représentation de 2^k arêtes avec une information de taille k, *i.e.*, l'ensemble des nœuds de la clique. Une mise en œuvre de ce protocole a été effectuée et développée sur les bases du démon de routage `olsrd`.

Nous avons comparé le volume d'information de contrôle (MPR pour OLSR et cliques pour JUMBO) échangé dans le réseau. Les performances de notre approche basée sur la diffusion de cliques et non plus des MPR afin de pouvoir reconstruire l'ensemble de la topologie en chaque nœud du réseau sont bonnes si le graphe n'est pas un graphe très dense. Dans un modèle de graphe de type *grande salle* (graphe aléatoire avec une probabilité d'arête proche de 0.9), OLSR obtient

des résultats bien meilleurs. Ce résultat s'explique par le fait que dans un graphe presque complet, il suffit d'un petit nombre de MPR pour couvrir tous les sommets. Cette hiérarchie est inversée si on prend des graphes plus épars modélisés par des graphes géométriques aléatoires (rayon de 0.5 dans un carré de 1×1). Ces graphes étant moins denses, ils offrent d'avantage de chemins longs entre les nœuds et le nombre de MPR nécessaires à un nœud est plus élevé que pour les graphes aléatoires.

Cette première expérience de la mise en œuvre d'un protocole de routage nous a permis de mieux cerner les points épineux qui surviennent dans la mise en œuvre réelle d'un protocole qui sur le papier paraît simple et efficace ! Cette implémentation nous a ouvert d'autres perspectives et a soulevé des interrogations, notamment sur les diverses incohérences qu'il peut y avoir à vouloir tout développer au niveau IP sans toucher au noyau, *i.e.,* sans modifier la sacro-sainte pile IP. Nous allons revenir dans la section suivante sur ces problèmes architecturaux.

La mise en œuvre d'un algorithme de routage ad hoc et son déploiement sur une plate-forme de test pour qu'il soit utilisé en tant que support de base aux communications des hôtes (laptop, PDA) soulèvent diverses interrogations techniques ou de conception qui ne sont pas aussi anodines qu'elles peuvent paraître de prime abord. Le but de cette section n'est pas de définir un antépénultième nouveau protocole de routage pour les réseaux ad hoc, mais bien de donner une architecture permettant de mettre en œuvre les protocoles existants et de les déployer. Cette architecture se fonde conjointement sur l'utilisation de protocoles de routage ad hoc classiques (OLSR [JMQ$^+$01, JMQ$^+$02], AODV [PRD02, PR99]) et sur la mise en œuvre d'une interface virtuelle ad hoc. Notre architecture, nommée ANANAS (*A New Ad hoc Network Architectural Scheme*), offre un support total pour IP au-dessus du réseau ad hoc ce qui inclut les mécanismes comme DHCP ou l'auto-configuration présente dans IPv6. Cela permet aussi la connectivité avec les services présents dans l'Internet et les protocoles de type multicast.

4.5.1 Architecture ad hoc idéale

Nous allons tout d'abord définir « *l'architecture ad hoc idéale* » en terme de services *indispensables*. Le premier service fondamental est de permettre la communication entre tous les hôtes mobiles du réseau, c'est-à-dire, permettre un routage point-à-point dynamique au sein d'un domaine sans fil [CM99].

Connectivité intranet. Le paradigme du routage est à la base de toute conception de réseau et nous avons déjà démontré toute l'importance que revêt ce service, largement étudié ces dernières années. Il est important de noter que tout hôte d'un réseau ad hoc utilisant les technologies de communication sous-jacentes A et B doit être en mesure de joindre tout autre nœud utilisant une technologie A ou B. Cela implique que le routage unicast doit être conçu sur un graphe de connexion multiple composé de divers graphes de connexion physique. Le second service important qui doit être offert est le broadcast (ou flooding). En effet, ce processus est largement employé par tous les protocoles de routage unicast qu'ils soient proactifs ou réactifs. Les services d'anycast et de multicast doivent également être mis en œuvre (ils sont devenus obligatoires dans IPv6) ;

Support complet pour TCP/IP. Une fois une connectivité assurée, il est impératif d'offrir TCP/IP qui est devenu de facto le standard sur lequel repose l'Internet. Tout hôte doit être

en mesure de se comporter comme s'il appartenait à l'Internet, *i.e.*, « *in an interoperable inter-networking capability over a heterogeneous networking infrastructure* ». De plus, tout support approximatif ou partiel d'IP est à proscrire, ce qui implique trivialement quelques conséquences. IP définit un système d'adressage ainsi que des règles de routage en liaison directe avec cet ensemble. Par exemple, la notion de réseau IP et de sous-réseau IP est basée sur la structure de l'adressage, les règles de broadcast aussi. Un paquet émis à destination de l'adresse `255.255.255.255` est destiné en fait à l'ensemble des nœuds présents sur le lien local de la source du paquet et les routeurs ne sont pas sensés le retransmettre. De plus, divers mécanismes d'auto-configuration ont été développés en association avec IP. En IPv4, DHCP permet à un hôte de récupérer entre autre son adresse IP. En IPv6, les nœuds sont aussi capables de s'auto-configurer et de générer leur propre adresse IP en se basant sur les préfixes émis par les routeurs. Ces divers mécanismes revêtent une importance encore plus grande dans le cas des réseaux spontanés ;

Connectivité avec Internet. Même si l'on imagine que les réseaux ad hoc ne vont être que des « *stub* » et donc que tout le trafic présent au sein d'un tel réseau est soit originaire soit destiné à ce réseau, il apparaît important d'offrir une connectivité globale. Donc, si l'auto-configuration est disponible, un nœud mobile au sens d'IP doit être en mesure de se déplacer vers un réseau ad hoc et de gagner une adresse IP temporaire lui permettant de garder actives toutes ses sessions comme s'il avait migré vers un réseau fixe classique. Cependant, la notion de connectivité globale est plus large que le simple « *care of address* » d'IP mobile. Il s'agit d'offrir un continuum de services entre la partie filaire classique et le réseau ad hoc : recevoir un flux multicast en provenance d'un serveur présent dans l'Internet et donc, être en mesure de relayer des protocoles comme PIM/CBT/YAM vers les protocoles spécifiques aux réseaux ad hoc (MOLSR, MAODV).

4.5.2 Petit plaidoyer contre la philosophie de mise en œuvre de MANet

Le groupe MANet de l'IETF – *Internet Engineering Task Force* – propose une architecture dans laquelle l'élément de base est le *MANet node* : « *a MANet node principally consists of a router, which may be physically attached to multiple IP hosts (or IP-addressable devices), which has potentially *multiple* wireless interfaces–each interface using a *different* wireless technology* ». Les paquets traversent le réseau de nœud en nœud en suivant les règles de routage dictées par IP basées sur les adresses IP des interfaces des nœuds MANet.

Pour répondre à tous les défis énoncés précédemment et listés dans le RFC [CM99], il nous semble inévitable de devoir s'éloigner des solutions disponibles à l'heure actuelle. En effet, la philosophie actuelle prônée au sein du groupe MANet est de mettre en œuvre les protocoles de routage, et plus particulièrement l'unicast au niveau IP. Cela est en fait une solution de facilité à court terme, posant plus de problèmes qu'elle n'en résout vraiment puisqu'elle engendre certaines inconsistances avec IP. Pour s'en convaincre, il suffit de lire les discussions sans fin sur la sémantique du broadcast et/ou du flooding dans un réseau IP ou sur le fait qu'un réseau ad hoc doit supporter ou non des sous-réseaux IP. Bref, vouloir tout mettre au niveau IP, revient à vouloir faire du routage de niveau 2 en trichant, ce qui est acceptable si cela n'engendrait pas in fine un support seulement partiel des fonctionnalités intrinsèques d'IP que nous avons rappelées précédemment.

4.5.3 Notre proposition : ANANAS

À partir des remarques précédentes, il apparaît important de définir une nouvelle architecture offrant une granularité plus fine en introduisant une couche supplémentaire nécessaire à la prise en compte du « *niveau ad hoc* ». Nous pouvons situer cette couche virtuelle au niveau 2.5, *i.e.*, entre le niveau 2 (MAC) et le niveau 3 (IP). Plus précisément, le routage au niveau IP ne permet pas de disposer d'une diffusion atomique sur un sous-réseau. Diffuser un paquet IP afin d'atteindre tous les nœuds d'un réseau ad hoc (au sens IP du terme, *i.e.*, au sein d'un même sous-réseau IP), ne doit en rien interférer avec l'en-tête IP et donc, le TTL (*Time To Live*) d'un paquet ne doit pas être décrémenté tandis que le paquet est routé au sein du sous-réseau IP. De même, mettre en œuvre le routage ad hoc au niveau 2 (HIPERLAN) [Bar97, HIP95] présente quelques inconvénients car cela rend assez difficile la gestion des interfaces multiples et donc, le support de la micro-mobilité. Le niveau ad hoc offre, comme avantage, l'accès à des paramètres fins de gestion des interfaces (*e.g.*, niveau de puissance) qui sont quasi inaccessibles au niveau IP.

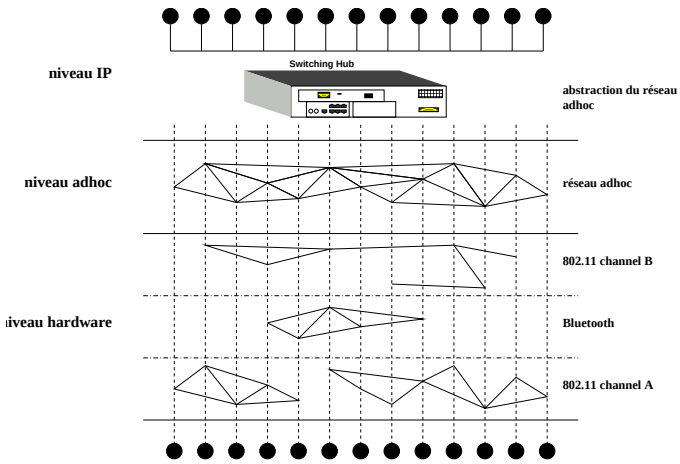


FIG. 4.3 – Les différentes abstractions d'un réseau ad hoc.

Notre proposition d'architecture découpe un réseau ad hoc en 3 niveaux d'abstraction : le niveau hardware, le niveau ad hoc à proprement parler et le niveau IP. La première abstraction repose sur une réalité, la compatibilité physique des interfaces, les deux autres étant plus d'avantage de l'esprit. L'élément de base de la première couche est l'interface physique tandis que les deux autres reposent sur la notion de nœud ad hoc tel qu'il est défini par MANet [CM99].

niveau hardware : ensemble de tous les réseaux physiques. Un réseau physique étant défini par l'ensemble de toutes les interfaces capables de communiquer les unes avec les autres. À ce niveau, la notion de capacité à communiquer est directement liée à la compatibilité des interfaces physiques. Aucun routage n'est mis en œuvre et chaque interface est identifiée par son adresse MAC ;

niveau ad hoc : réunion de tous les réseaux physiques. L'interface physique n'est plus visible, seul le nœud ad hoc est présent. Un nœud ad hoc est représenté par une adresse ad hoc et des communications multi-sauts sont rendues possibles par la commutation de paquets au

sein des nœuds. La façon dont sont commutés/routés les paquets dépend du protocole de routage mis en œuvre et celui-ci devient indépendant de l'architecture ;

niveau IP : monde tel qu'il est vu par la couche 3 du modèle OSI. À ce niveau, le réseau ad hoc est vu comme un bus Ethernet, ou plus précisément comme un switch Ethernet. Le nœud ad hoc ne présente à IP qu'une seule interface tout comme une carte Ethernet classique, *i.e.*, plusieurs interfaces physiques d'un nœud ne sont vues par IP que comme une seule interface ad hoc.

Cette architecture permet une compatibilité totale avec IP. Un paquet émis à destination de l'adresse `255.255.255.255` atteindra tous les nœuds du réseau ad hoc sans subir de modification/traitement au niveau IP. DHCP peut être employé puisque les nœuds du réseau ad hoc sont joignables même si IP n'est pas configuré. De façon générale, IP va réellement se comporter comme il le ferait avec un lien Ethernet classique.

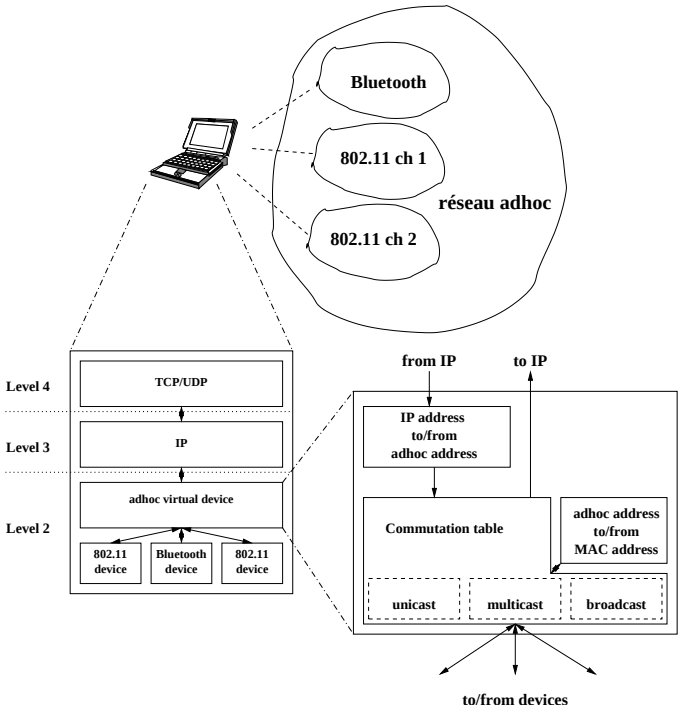


FIG. 4.4 – L'interface virtuelle.

L'architecture interne d'un nœud ad hoc est basée sur la notion d'interface virtuelle. Le rôle de l'interface virtuelle ad hoc est de masquer l'ensemble des interfaces physiques et de donner l'illusion d'un seul réseau virtuel. Au niveau ad hoc, il s'agit d'un réseau sans fil multi-sauts, au niveau IP, il s'agit d'un lien Ethernet switché. Pour les couches supérieures, l'interface virtuelle agit de façon classique en se déclarant comme une interface Ethernet à IP. IP émet ses paquets à destination des nœuds ad hoc via cette interface. Pour la couche basse, l'interface apparaît comme un protocole de couche supérieure. Dès qu'un paquet en transit dans le réseau ad hoc arrive, il est transmis à l'interface virtuelle. Cette architecture ne nécessite aucune modification ni au niveau des drivers spécifiques à chaque type de carte, ni au niveau de la pile de protocole IP.

Notre architecture soulève le problème de l'identification des nœuds ad hoc. Pour transmettre un paquet d'un nœud ad hoc à un autre nœud ad hoc, il est nécessaire d'adresser les interfaces virtuelles et un nouvel espace de nommage doit être créé dans ce plan virtuel. Nous avons décidé d'adresser chaque interface virtuelle par une adresse ad hoc, composée d'un champ sous-réseau, d'un champ spécifique au type de hardware utilisé et d'un champ identificateur de nœud :

```
0        8      16              48
+--------+-------+--------------+
| Net Id | Hw Id |    Node Id   |
+--------+-------+--------------+
```

La concaténation du type `Hw ID` et du `Node ID` (initialisé avec la valeur d'une des adresses MAC des interfaces physiques) doit assurer l'unicité de cette adresse ad hoc.

Le rôle de l'interface virtuelle est de commuter les paquets entre les différentes interfaces physiques et les protocoles du niveau supérieur. À la réception d'un paquet, l'interface décide si elle doit réémettre le paquet, au travers de quelle interface et à destination de quel nœud, ou/et si elle doit le transmettre au niveau supérieur. Nous préférons le terme de commutation car celui de routage a une forte connotation IP. Afin de pouvoir effectuer la commutation, nous avons besoin de deux mécanismes de translation : ATP (adresse ad hoc/adresse MAC) et ARP (adresse ad hoc/adresse IP). Ces deux mécanismes tiennent à jour des tables de correspondance et sont étroitement liés à la façon dont l'algorithme de routage est mis en œuvre. Prenons le cas d'un protocole réactif. Quand un nœud A tente de joindre un nœud B, un processus de recherche de route est lancé. La requête dans notre architecture va être similaire à celle de MANet : « *Je cherche une route vers le nœud* IP_B ». La réponse va être légèrement modifiée pour nous permettre d'apprendre en plus l'adresse ad hoc du nœud B en question : « *Voici une route permettant de joindre le nœud* AH_B *ayant comme adresse* IP_B ». Le même genre de modification est effectué pour les protocoles proactifs. La translation ATP n'est utile qu'en local (*i.e.*, pour l'ensemble des voisins) et cette information est facilement récupérable depuis les paquets de contrôle des protocoles de routage et s'apparente à l'option *Automatique Address Resolution* [BOT02] du protocole de routage TBRPF. Une interface virtuelle est susceptible de supporter plusieurs protocoles et pour chacun d'eux, le réseau ad hoc apparaît comme un lien Ethernet. Une phase de démultiplexage (voir figure 4.5) effectuée dans l'interface virtuelle aiguille le paquet. De plus, ce mécanisme permet d'utiliser une interface physique à la fois en mode ad hoc et en mode classique, ce qui peut s'avérer utile dans des réseaux hybrides. Pour IP, il s'agit de deux interfaces distinctes.

Nous avons mis en œuvre l'ensemble de l'architecture ANANAS [CF02a] et le code est disponible sur sourceforge[10]. Ce développement a été effectué sous linux et se présente sous la forme d'un module que l'on peut charger et configurer de façon dynamique : ajouter/supprimer une interface virtuelle, lui adjoindre une ou plusieurs interfaces physiques, configurer les divers champs de l'interface virtuelle. De plus, nous avons déjà porté OLSR sur notre architecture, ce qui permet d'avoir un processus de commutation de type proactif. Un portage sous Windows, en collaboration avec le Microsoft Research Lab de Cambridge, est en cours et il se fonde sur l'architecture des NDIS (disponible sous Windows 2000, XP et CE) (Voir figure 4.6).

[10] `http://www.sourceforge.net/projects/ananas/`

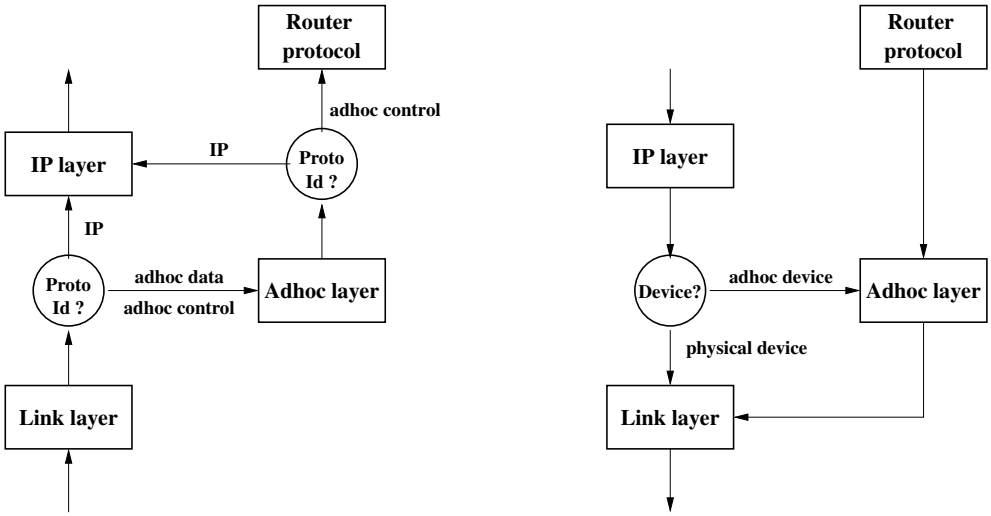


FIG. 4.5 – Processus de multiplexage et démultiplexage dans le nœud ad hoc.

4.5.4 Réseaux ad hoc et IPv6

L'architecture ANANAS se conforme aux prérequis listés dans la section 4.5.1 : support IPv4, IPv6, auto-configuration, support des sous-réseaux IP. L'un des reproches que l'on peut faire à cette architecture est que cette solution va se substituer au rôle du protocole IP qui lui-même doit exécuter cette tâche. Nous avons déjà répondu en partie aux motivations qui poussent à proposer une telle architecture. Il est évidemment possible de modifier IP pour lui intégrer de nouvelles caractéristiques et fonctionnalités mais le legacy dans ce domaine est très lourd et si l'on désire promouvoir ce nouveau type de connexion « *ad hoc* », il est préférable d'offrir un simple module/driver à charger plutôt que de mettre en garde l'utilisateur sur le fait qu'il doit recompiler son kernel[11], désintaller son driver classique pour le remplacer par un driver de routage ad hoc de niveau 2 et ainsi ne plus être en mesure d'utiliser sa carte 802.11 en mode base ! Une des conclusions est que pour mettre en œuvre une solution de routage ad hoc dans le monde IP sans toucher à IP et sans faire passer toutes les données par des démons utilisateurs, il est nécessaire d'ajouter une couche supplémentaire. La question est alors de savoir si cela est nécessaire pour chaque nouveau concept ?

Pour IPv4, nous laisserons la question en suspens et nous allons brièvement présenter notre solution pour IPv6. Bien que ANANAS reste fonctionnel avec IPv6, il apparaît plus judicieux de proposer une solution qui soit complètement IP puisque IPv6 est encore un protocole jeune et plus ouvert. Cette solution [CF02b] reprend la notion d'interface virtuelle qui permet de mettre en œuvre le *node identifier* défini dans [CM99] mais cette fois-ci, nous mettons en place ce mécanisme au sein d'IPv6 en définissant la notion de *connector* qui va aussi permette de rassembler plusieurs interfaces ad hoc. Un hôte peut avoir plusieurs connecteurs et une interface physique peut être liée à plusieurs connecteurs ad hoc.

Les deux schémas d'adressage local proposés par IPv6 ne conviennent pas très bien aux

[11] Je te promets Isabelle que ton portable marchera comme avant !

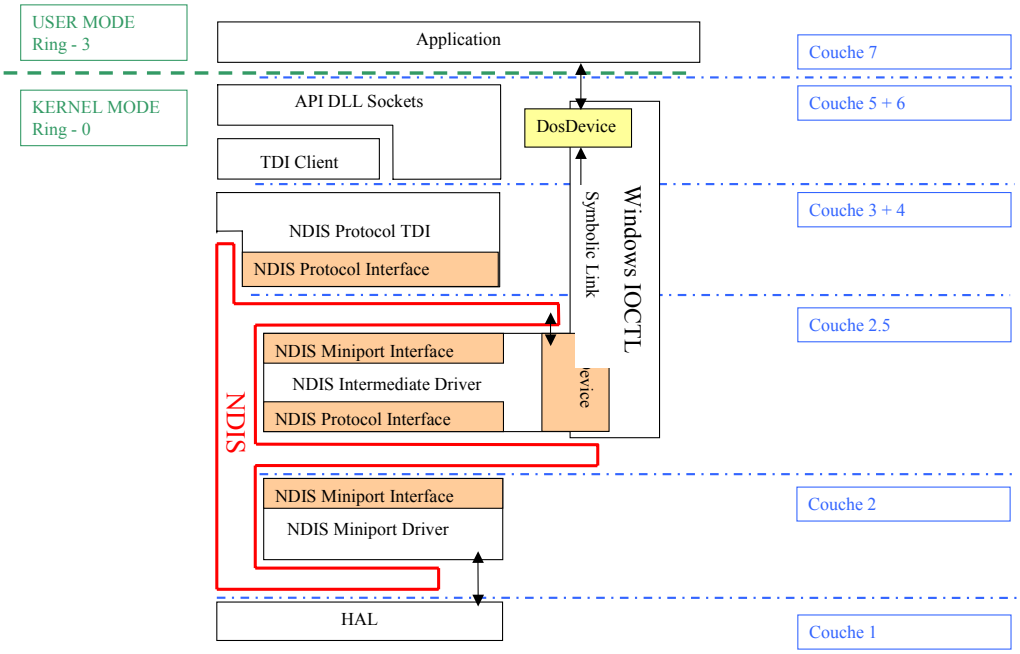


FIG. 4.6 – Mise en œuvre sous windows.

réseaux ad hoc. En effet, les adresses *link-local* unicast et multicast ne sont pas adaptées. Une adresse link-local est liée à une seule interface et sa validité est limitée au lien local, donc elle ne peut pas être routée. Comme un réseau ad hoc est susceptible d'être inclus dans un réseau plus large ou d'être étendu sur différents sites, les adresses *site-local* sont aussi inappropriées. Pour ces raisons, nous proposons d'introduire un troisième type d'adresse IPv6 local : les adresses *ad hoc-local* unicast et multicast. Leur validité est restreinte à un réseau ad hoc. Elles offrent un support minimal pour l'identification des nœuds ad hoc. Une adresse ad hoc local est de la forme :

```
| 10 bits  |    54 bits     |          64 bits      |
+----------+----------------+-----------------------+
|1111111001|        0       | ad hoc connector ID |
+----------+----------------+-----------------------+
```

Dans un schéma d'adressage IPv6, un réseau ad hoc peut être à la fois un *multi-link subnet* et un *multi-link multi-subnet*. Si l'on considère tout le réseau ad hoc comme étant un multi-link subnet, il suffit de mettre en correspondance l'étendue (*scope*) du sous-réseau avec celle du réseau ad hoc. Par contre, pour avoir une vision en multi-link multi-subnet, nous introduisons la notion de zone (*channel* dans [CF02b]). Une zone est un ensemble connexe de connecteurs ad hoc partageant une même valeur de zone. Cette notion de zone permet de restreindre la portée des opérations de multicast et donc d'avoir un support pour la définition et la gestion de sous-réseaux.

L'introduction et l'utilisation brute d'IPv6 soulève d'autres interrogations qui ne sont pas encore résolues. Ces problèmes concernent principalement la gestion des annonces des routeurs que l'on doit mettre en œuvre. Penser que l'auto-configuration sans état d'IPv6 résout tous les problèmes est, au mieux, se cacher la réalité et commettre une erreur de jugement. Peu de travaux existent sur la comparaison entre une approche que l'on peut qualifier de proactive pour laquelle les routeurs envoient des annonces périodiquement et une approche réactive où ils réagissent à des

sollicitations des hôtes. Nous reviendrons dans la section 5.1 sur les réseaux hybrides, alliant un réseau d'accès relativement stable et des réseaux multi-sauts ad hoc très mobiles (Voir figure 4.7).

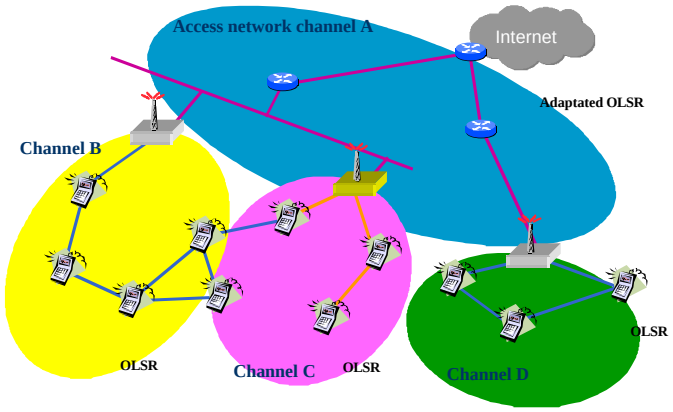


FIG. 4.7 – Exemple d'architecture hybride devant gérer à la fois la micro-mobilité (Cellular IP) et les handoff ad hoc.

L'un des problèmes que l'on rencontre aussi dans la mise en œuvre de réseaux ad hoc en IPv6 est l'auto-configuration des routeurs. L'auto-configuration des hôtes a été en partie résolue et est basée sur les préfixes annoncés par les routeurs. Actuellement, le préfixe de chaque lien d'un routeur doit être configuré manuellement. C'est ce préfixe qui est diffusé par les paquets d'annonce du routeur `ROUTEUR ADVERTISEMENT`. Si cela ne pose pas trop de problèmes pour les grands réseaux où une politique de gestion est clairement définie (AS, grosses compagnies), cela peut être pénalisant pour de plus petites entreprises ou pour des applications de type domotique, réseaux domestiques. Si l'on conçoit un réseau ad hoc comme pouvant servir à configurer automatiquement un réseau sans pour autant avoir nécessairement une très grande mobilité (mesh routing, ricochet, rooftop, domotique), devoir configurer les SLA (*Site Level Agregator*) manuellement n'est pas envisageable. Les topologies de ces réseaux de petite taille peuvent néanmoins se révéler complexes du fait des diverses technologies employées (HomePNA, IEEE 802.11, Bluetooth). L'auto-configuration de l'ensemble du réseau peut difficilement être déléguée à un provider car il peut y avoir plusieurs routeurs présents dans le réseau. Le cas où un seul routeur de bordure est connecté au provider et distribue les paquets vers les autres équipements composant le réseau est un cas très simple. Ce modèle est très restrictif car il interdit les configurations où un réseau sans fil n'est pas directement connecté au routeur de bordure mais est uniquement accessible par un des équipements déployés. De plus, un réseau domestique (*home network*) peut être connecté à plusieurs ISP par différents accès possibles (ADSL, CABLE, UMTS) ou avoir une configuration dynamique du fait que certains équipements sont mobiles (les connexions bluetooth pouvant apparaître et disparaître à tout moment).

Pour résoudre le problème de la configuration automatique du champ SLA des adresses IPv6 des routeurs, nous avons proposé de définir trois nouveaux LSA (*Link State Advertisement*) au sein du protocole OSPFv3. Les routeurs de bordure peuvent, soit être configurés manuellement, soit acquérir leur TLA (*Top Level Aggregator)* auprès du provider. Un TLA désignant un site local (`FEC0 : :/48`) peut toujours être utilisé mais doit être diffusé si aucun TLA global n'est

disponible. Notre algorithme [CFT02a, CFT02b] garantit un consensus et une forte stabilité du SLA choisi par chaque lien en plus de l'unicité de l'association TLA-SLA au sein d'un même site. Ces travaux ne s'appliquent pas directement aux réseaux ad hoc à forte mobilité puisque OSPF n'est pas adapté à ce genre de réseau mais notre algorithme d'auto-configuration des SLA est indépendant d'OSPF en ce sens qu'il n'est pas intégré à la partie calcul des routes. Il utilise les fonctionnalités et facilités offertes par les LSA et peut être intégré au sein d'un protocole de routage ad hoc proactif (OLSR).

4.6 Amélioration des protocoles de routage multicast ad hoc

4.6.1 Critères d'évaluation

Un facteur limitant dans le développement des algorithmes de multicast dans les réseaux ad hoc est la non existence de méthodes d'évaluation des arbres construits. À notre connaissance, aucune étude complète comparant les différents algorithmes de création de structures de diffusion n'a été menée. De telles études sont néanmoins nécessaires. Pour le routage unicast, il est en partie possible d'évaluer le coût des protocoles en utilisant par exemple la longueur moyenne des routes [JV02]. Dans le cas des algorithmes de multicast, les critères classiques employés dans les réseaux filaires [BCFG$^+$97, FHM00] sont mal adaptés. Il peuvent donner un aperçu des performances de la structure mise en place (latence, bande passante) mais ne permettent pas d'avoir une estimation des collisions ou de l'occupation radio. Ils ne donnent aucune information sur le nombre de nœuds sollicités pour router le flux multicast. Or, dans un environnement coopératif de type réseau ad hoc, minimiser le nombre de nœuds non intéressés par le trafic multicast est souhaitable. Nous proposons six critères pour évaluer les performances d'un protocole de multicast :

Récepteur collatéral : nombre de nœuds non membres du groupe multicast qui reçoivent le flux multicast.

Récepteur actif : nombre de nœuds membres du groupe multicast qui reçoivent le flux multicast.

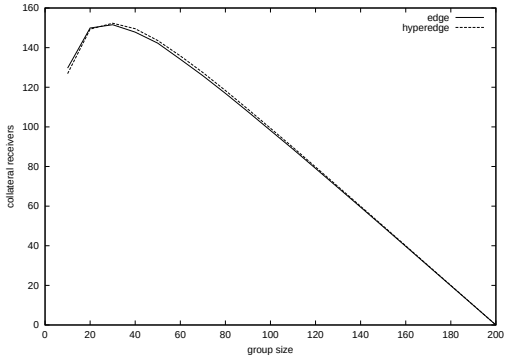
Émetteur collatéral : nombre de nœuds non membres du groupe multicast qui retransmettent des paquets.

Émetteur actif : nombre de nœuds membres du groupe multicast qui retransmettent des paquets.

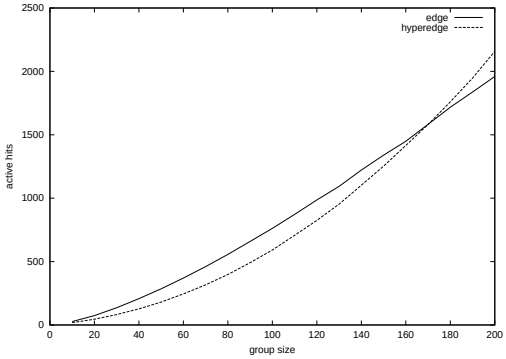
Réception collatérale : nombre de fois qu'un paquet multicast atteint un nœud qui n'est pas membre du groupe.

Réception active nombre de fois qu'un paquet multicast atteint un nœud qui est membre du groupe.

À partir de ces critères nous avons réalisé diverses séries de simulations [CF01, CF02c] effectuées sur des graphes géométriques aléatoires [DPPS01]. Les premiers résultats présentés mettent en valeur le fait qu'il est important de prendre en compte la caractéristique diffusante du médium radio. Nous comparons deux algorithmes qui tous deux construisent des arbres. Le premier algorithme nommé *Algo-edge*, utilise une approche *reverse path* classique. Le second algorithme, nommé *algo-hyper-edge* regarde dans un premier temps si l'un des ses voisins est déjà raccroché à l'arbre, si oui il le sélectionne comme père sinon il exécute l'algorithme précédent. Les



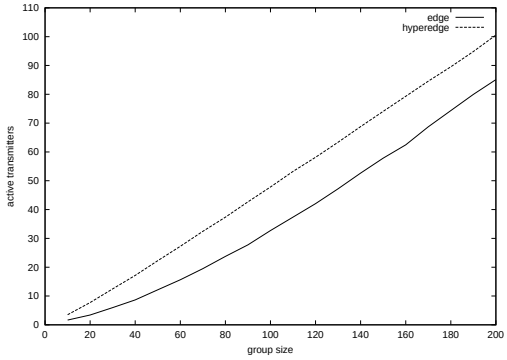
(a) : Nombre de récepteurs collatéraux.



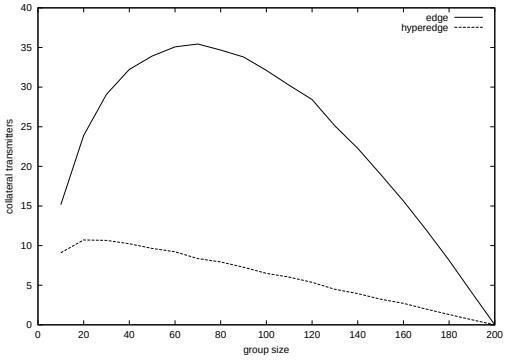
(b) : Nombre de récepteurs actifs.

FIG. 4.8 – Nombre de récepteurs actifs et collatéraux en fonction de la taille des groupes.

simulations montrent que ces deux algorithmes engendrent un nombre identique de récepteurs collatéraux (figure 4.8(a)) et un nombre comparable de réceptions actives (figure 4.8(b)). Par contre l'approche hyper-edge favorise l'utilisation des nœuds membres du groupe, *i.e.* plus d'émetteurs actifs et moins d'émetteurs collatéraux (figure 4.9(a) et 4.9(b)). Pour finir, l'approche algo-hyper-edge induit beaucoup moins de perturbations que l'approche algo-edge car les réceptions collatérales sont beaucoup moins nombreuses (figure 4.9).



(a) : Nombre d'émetteurs actifs.



(b) : Nombre d'émetteurs collatéraux.

FIG. 4.9 – Nombre d'émetteurs actifs et collatéraux en fonction de la taille des groupes.

Nous avons effectué une deuxième série de simulations pour tenter de mesurer l'impact que peut avoir la connaissance totale ou partielle du réseau. L'algorithme M-OLSR ne possède qu'une vision partielle de la topologie (seules les connexions entre MPR sont disponibles). Nous avons donc comparé deux variantes de l'algorithme *algo-hyper-edge* précédent. L'une se fonde sur une

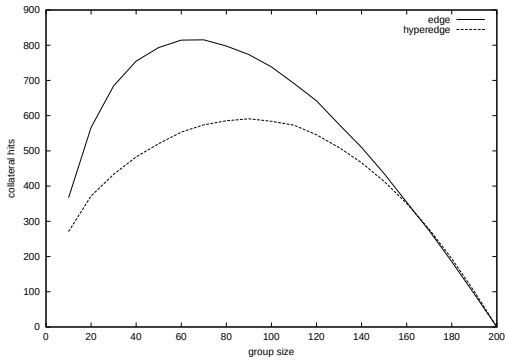
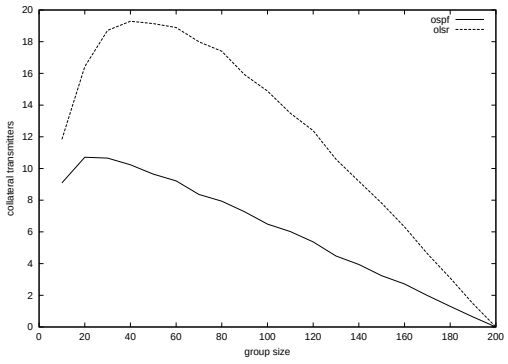
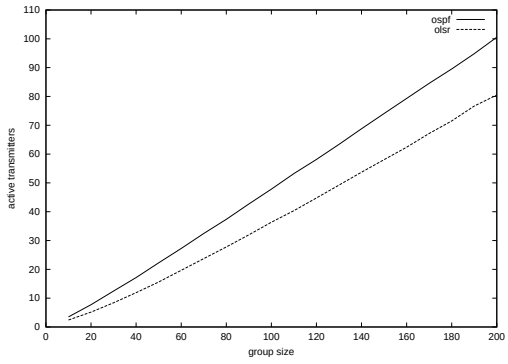


FIG. 4.10 – Nombre de réceptions collatérales en fonction de la taille des groupes.

connaissance partielle (MOLSR) et l'autre détient une connaissance totale du réseau (MOSPF). Les deux approches ont des comportements tout à fait similaires en terme de récepteurs collatéraux, de réceptions actives et collatérales. La seule différence notable est que la connaissance totale engendre deux fois moins d'émetteurs collatéraux (figure 4.11(a)) et que la charge du routage est supportée principalement par les membres du groupe (figure 4.11(b)). En additionnant les courbes 4.11(a) et 4.11(b), on peut remarquer que MOLSR possède moins de nœuds internes dans ses arbres multicast. Ce point positif s'explique par le fait qu'en réduisant le nombre de nœuds potentiels (seuls les MMPR sont susceptibles de participer au routage), MOLSR force la fusion des branches alors que MOSPF va avoir tendance à plus les éparpiller.



(a) : Nombre d'émetteurs collatéraux.

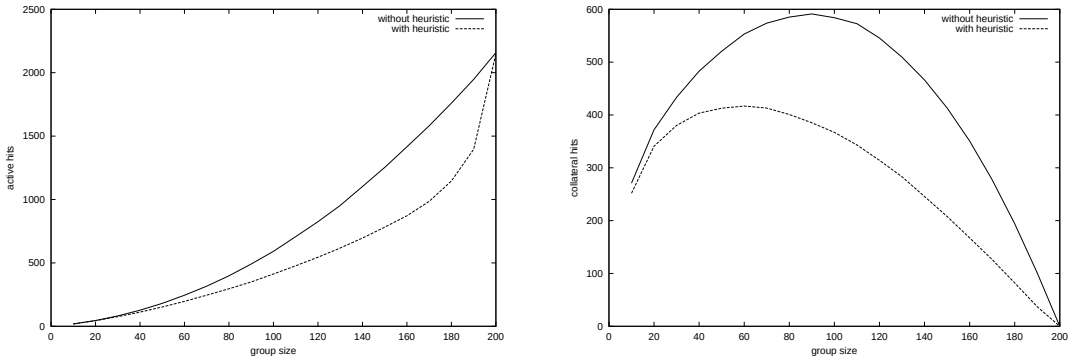


(b) : Nombre d'émetteurs actifs.

FIG. 4.11 – Nombre d'émetteurs actifs et collatéraux en fonction de la taille des groupes.

Pour finir, nous nous sommes intéressés à la façon dont un nœud choisit son père parmi un ensemble de candidats répondant tous au premier critère de sélection des algorithmes présentés

ci-dessus. Notre première heuristique consiste à éttendre son voisinage à plusieurs sauts avant de choisir son père. Quand un nœud doit choisir quel voisin doit être son père, il choisit celui qui est le plus proche de l'arbre en examinant son k-voisinage. Les simulations effectuées ne montrent aucune amélioration notable par rapport à MOSPF même en choisissant un 4-voisinage. Notre seconde heuristique tente de réduire le nombre de réceptions collatérales et de récepteurs collatéraux. Quand un nœud choisit son père, il prend celui qui a le plus petit nombre de voisins qui ne sont pas des membres du groupe. Cette heuristique est employée chaque fois que l'on choisit un sommet de l'arbre et présente de très bons résultats en permettant de réduire de façon très significative le nombre de réceptions collatérales (figure 4.12(a)) et de récepteurs collatéraux (figure 4.12(b)).



(a) : Nombre de réceptions actives. (b) : Nombre de réceptions collatérales.

FIG. 4.12 – Nombre de réceptions actives et collatérales en fonction de la taille des groupes.

4.6.2 Modification de MOLSR

Sur la base des critères définis précédemment, nos premiers résultats de simulation mettent en évidence l'importance de la notion d'hyper-arête : la connaissance de l'appartenance ou non des nœuds de son voisinage à un groupe multicast permet de construire des arbres plus efficaces. Pour mettre en œuvre des heuristiques encore plus efficaces, la seule connaissance du 2-voisinage est nécessaire. Fait surprenant, connaître son k-voisinage n'apporte pas d'amélioration significative.

Il est assez simple de modifier MOLSR [JLV+01, Lao02] pour prendre en compte ces premiers résultats. Nous proposons de remplacer les paquets `CONFIRM_PARENT` émis par un nœud par des paquets `MULTICAST_HELLO` diffusés périodiquement dans son voisinage. Un paquet `MULTICAST_HELLO` émis par un nœud u contient la liste des groupes pour lesquels u est un nœud interne de l'arbre, la liste des groupes auxquels u est abonné (mais n'est pas un nœud interne de l'arbre) et la liste de ses pères pour chacun des arbres des groupes auxquels il est abonné. La réception d'un paquet `MULTICAST_HELLO` est gérée de la même façon que celle d'un `CONFIRM_PARENT` mais permet de garder à jour la connaissance de l'appartenance à un groupe multicast des nœuds de son voisinage.

4.6.3 Protocole adaptatif de routage multicast ad hoc

Quand on étudie un protocole de routage, il est nécessaire de prendre en compte une notion de robustesse. Celle-ci n'est pas quantifiée mais peut être caractérisée (*e.g.*, le comportement de l'algorithme dépend faiblement du facteur mobilité). La robustesse est liée en premier lieu à la structure de communication engendrée par le protocole. Un arbre est un exemple de structure de communication non robuste. Étant donné que c'est un graphe de connexité minimale, la moindre rupture d'un lien entraîne la non-connexité de tout l'ensemble, ce qui entraîne la nécessité de réparer l'arbre. Ainsi tout le trafic de multicast est concentré sur un ensemble de chemins critiques.

L'autre observation, issue des résultats de simulation présentés dans la section 4.6, est que vouloir construire et maintenir un arbre de multicast le plus optimal possible peut être inutile du fait, encore un fois, de la nature diffusante du médium radio : le « *meilleur* » arbre peut quand même toucher l'ensemble des nœuds du réseau (voir figure 4.8). En d'autres termes, pourquoi maintenir une structure complexe et fragile, censée optimiser le nombre de ressources radio quand de toute façon, quel que soit l'arbre considéré, il est impossible de ne pas toucher tout le monde, *i.e.*, d'effectuer, in fine, une opération de broadcast. Ainsi, dans le cas où les membres d'un groupe sont répartis localement de manière très dense, il est tout aussi efficace de réaliser un broadcast de rayon réduit dans cette zone que de mettre en place une structure de diffusion compliquée. De même, un broadcast global optimisé (*e.g.*, MPR) est généralement aussi efficace qu'un arbre lorsque quasiment tous les nœuds du réseau sont intéressés par un groupe.

Pour mettre en œuvre un protocole adaptatif, *i.e.*, qui soit en mesure d'évaluer la configuration, il est nécessaire de caractériser et d'identifier ce que nous appelons les zones denses.

Définition 4.1 Soit $G = (V, E)$ un graphe non orienté. On colorie en rouge l'ensemble des nœuds qui se sont volontairement abonnés à un groupe de multicast donné et en noir les nœuds qui ne font que retransmettre le flux (nœuds relais). On note $\Gamma_k(x)$ le k-voisinage d'un sommet x. $Z = (V', E')$ est une zone dense si et seulement si :

1. Z est un sous-graphe partiel de G ;
2. Z est connexe ;
3. V' comporte au moins un sommet rouge ;
4. Pour tout sommet noir $s \in V'$:
   (a) $\Gamma_1(s) \cap V'$ ne comporte aucun sommet noir. En d'autres termes, dans toute chaîne incluse dans la zone dense, on ne peut jamais avoir deux sommets noirs successifs, ce qui permet de respecter la règle suivante : un nœud qui a pour unique rôle de relayer ne le fait qu'entre deux nœuds membres du groupe de multicast ;
   (b) $k = |\Gamma_1(s) \cap V'| \leq 2$. Si $k = 2$ alors les deux sommets de $\Gamma_1(s) \cap V'$ ne sont pas adjacents. Cette condition signifie qu'un sommet noir ne sera pas maintenu s'il n'est pas essentiel (il n'effectue pas de retransmission, ou est redondant) ;
   (c) Pour tout sommet $u \in \Gamma_2(s) \cap V'$, on a $\Gamma_1(s) \cap V' \not\subseteq \Gamma_1(u) \cap V'$. Cette dernière condition est également une condition de non-redondance des sommets noirs. Si on considère un sommet s et un sommet t à distance 2 de s, dont le 1-voisinage contient celui de s, on crée des chemins redondants.

Pour calculer une zone dense de façon totalement distribuée à partir de la définition précédente, il est nécessaire de transformer la condition 4 précédente :

Lemme 4.1 *Soit s un sommet noir dans une zone dense d'un graphe de connexion $G = (V, E)$. Alors les conditions suivantes sont équivalentes :*

1. $\forall u \in \Gamma_2(s), \Gamma_1(s) \not\subseteq \Gamma_1(u)$;
2. $\bigcap_{t \in \Gamma_1(s)} \Gamma_1(t) \cap \Gamma_2(s) = \emptyset$

Une fois ces zones denses créées (*i.e.*, chaque nœud du réseau sait s'il est membre abonné ou retransmetteur d'une zone dense), le but est de relier ces différentes zones entre elles. Nous avons une alternative centralisée/distribuée pour mettre en place un algorithme prenant en compte cette notion de zone.

Approche centralisée. Le premier type de solution centralisée consiste à faire remonter les paquets `JOIN` en unicast vers la source/core. Cette dernière calcule les zones denses du réseau, choisit un représentant au sein de chaque zone (*e.g.*, l'un des sommets le plus proche d'elle) et construit un arbre reliant ces zones en se basant sur sa connaissance du réseau. On va ensuite employer une méthode [Che01, CF01] à mi-chemin entre le source routing (*e.g.*, DDM) et un routage par table (*e.g.*, MAODV [RP99] ou MOLSR [JLV$^+$01]). Les paquets de données sont routés par table et les paquets de contrôle permettant de diffuser la structure sont routés par source, permettant d'améliorer la diffusion des directives et des consignes de routage. De plus, cette diffusion rafraîchit automatiquement la structure à mettre en œuvre. Pour construire l'arbre, la source (ou le core dans une version structure partagée), diffuse un paquet `TREE CREATE` qui contient l'ensemble des nœuds relais appartenant à l'arbre de diffusion, feuilles exclues. Lorsqu'un de ces nœuds reçoit ce paquet, il ajoute une entrée dans sa table de routage pour l'adresse multicast correspondante, puis réémet le paquet. Un nœud ne se trouvant pas dans le paquet n'effectue pas de réémission et annule la consigne de routage précédente pour le groupe en question. Une donnée multicast est retransmise le long de l'arbre normalement sauf quand elle atteint un nœud d'une zone dense qui va diffuser la donnée dans sa zone dense, *i.e.*, tous les nœuds de la zone dense vont relayer cette diffusion. Pour effectuer ce flooding restreint, il est possible d'employer un mécanisme optimisé basé sur les MPR.

Approche distribuée. L'approche complètement distribuée est plus complexe à mettre en œuvre, du moins dans sa version optimisée. Le problème majeur est l'élection d'un représentant au sein d'une zone dense, si possible sur la frontière la plus proche de la source/core. Une solution triviale consiste à exécuter un processus d'élection dans toute la zone dense. Cette solution est robuste mais génère périodiquement un trafic de contrôle dans toute la zone. Pour éviter cet inconvénient, il semble logique de vouloir effectuer l'élection uniquement sur la frontière. La difficulté réside alors dans la définition d'une frontière. Tentons d'en donner une définition.

Définition 4.2 Soit $Z = (V', E')$ une zone dense d'un graphe $G = (V, E)$. La frontière de la zone dense est $F = \{x \in V' | \Gamma_1(x) \not\subset V'\}$, l'ensemble des nœuds de la zone dense qui possèdent au moins un voisin qui n'appartient pas à la zone dense. On définit de même la frontière supérieure de la zone dense, $\lceil F \rceil = \{x \in V' | \Gamma_1(x) \not\subset V'$ et $\Gamma_1(x) \cap P(x, s) \neq \emptyset\}$ par l'ensemble des

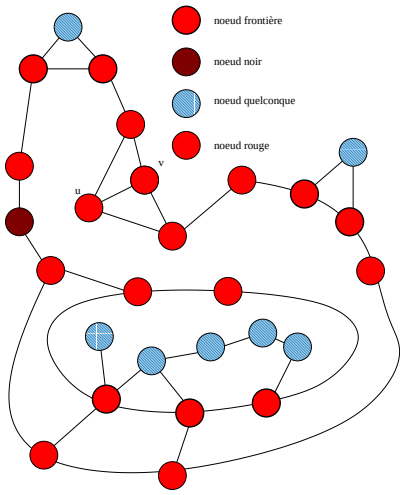
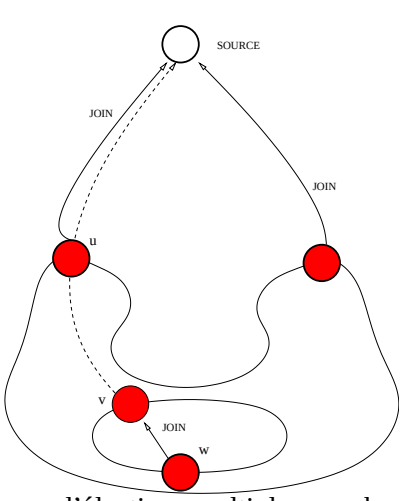


FIG. 4.13 – Frontière d'une zone dense.

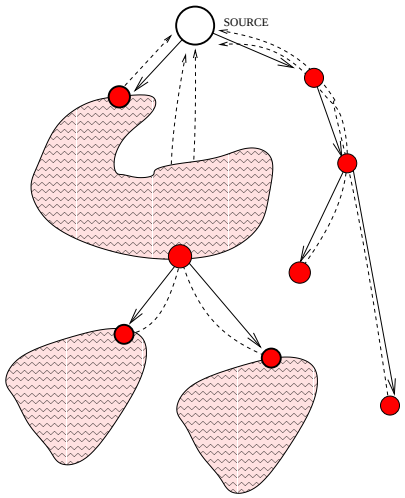
nœuds qui possèdent au moins un voisin appartenant à un plus court chemin vers la source et qui n'appartient pas à la zone dense.

Avec cette définition, une zone dense peut avoir plusieurs frontières supérieures et sa frontière n'est pas connexe (voir figure 4.13). Sans information supplémentaire, il est impossible de construire la frontière d'une zone dense car un sommet au milieu de la zone est similaire à un sommet en bordure mais n'ayant aucun voisin autre que ceux de la zone dense (sur la figure 4.13, les sommets u et v ont un voisinage équivalent et il est impossible, sans information supplémentaire, de les distinguer). Une solution est de déclencher des élections sur chaque segment de frontière supérieure. Une zone dense sera alors raccrochée à la source par plusieurs chemins. Les messages `JOIN` des sommets élus sur les fausses frontières supérieures sont eux supprimés dès qu'ils arrivent sur un nœud interne à la zone dense (voir figure 4.14). Pour pouvoir construire des frontières connexes, il faut obtenir des informations sur la position des sommets. Une telle approche a été proposée dans [Arn02] et consiste à :

1. déterminer un système autonome de coordonnées dans le réseau. En se basant sur [HCH01], on peut élaborer un système de coordonnées autonome, *i.e.*, sans l'appui extérieur d'une technique comme le GPS. Le principe général est le suivant. Chaque nœud i construit son repère local, dont il est l'origine. Les coordonnées de chaque 1-voisin de i sont calculées relativement à i. Par la suite, chaque repère local subit une correction (rotation et/ou symétrie) de manière à ce que tous les repères aient une direction commune. En choisissant un nœud du réseau comme origine, les coordonnées de tous les autres nœuds se déduisent par des translations, ce qui permet d'établir le repère global propre au réseau ;
2. calculer un graphe planaire recouvrant dans une zone dense. À partir d'un graphe géométrique G donné, on peut extraire un graphe partiel planaire G' recouvrant [BMSU01, BFNO02] à partir du graphe de GABRIEL. Le coût de l'algorithme en chaque sommet $v \in V$ est $O(d \log d)$ où d est le degré de v ;
3. calculer la frontière du graphe planaire. En chaque nœud s du graphe planaire G', on initie



(a) : cas d'élections multiples sur chaque segment des frontières.



(b) : cas idéal où l'on sait déterminer les frontières.

FIG. 4.14 – Protocole multicast adaptatif.

un parcours *en direction* de chacun de ses 1-voisins, et en chaque nœud, on tourne le plus à gauche. Si ces parcours sont effectués deux fois, chaque nœud connaît l'ensemble des faces auxquelles il appartient, chaque face étant représentée par une liste ordonnée de sommets. À ce stade, on ne peut rien dire d'autre, *i.e.*, on ne peut pas distinguer la frontière. On affine un peu notre notion de frontière en considérant uniquement la portion la plus extérieure (au sens géométrique) du graphe planaire, l'idée étant que l'ensemble des nœuds appartenant à la *frontière* délimitent la face de périmètre ou d'aire maximale. En chaque nœud s, on calcule le périmètre ou l'aire des faces auxquelles s appartient et localement, on détermine la face d'aire maximale. Ensuite, au niveau de tous les nœuds du graphe planaire G', on détermine le maximum global de manière distribuée. Chaque nœud connaît alors la liste des sommets délimitant la face d'aire maximale et on peut enfin effectuer l'élection parmi les nœuds de cette frontière.

4.7 Découverte de services

Ayant quelque peu péché par prolixité dans les sections précédentes, je vais finir la présentation de mes contributions dans le monde du sans fil par une brève description de mes recherches sur la découverte et le déploiement de services dans les réseaux ad hoc.

Pour utiliser un service présent sur un réseau, un utilisateur doit, à l'heure actuelle, connaître l'adresse réseau du service en question. Par « *service* » nous désignons de façon très générique toute application possédant une interface clairement définie capable d'effectuer des traitements ou des actions au profit et au nom d'un utilisateur, souvent nommé « *client* » dans ce contexte. Les évolutions actuelles des réseaux couvrent un large spectre et permettent d'aller d'une infrastructure offrant un ensemble de services minimum (par exemple des imprimantes) vers des ar-

chitectures beaucoup plus sophistiquées et programmables permettant d'offrir un ensemble complexe de services (jeux de réalité virtuelle, encodage temps réel vidéo). Il semble indubitable que le nombre de services offerts va aller croissant mais il est aussi à parier que la demande de services va elle aussi suivre la même croissance. Dans une telle perspective d'évolution, une configuration manuelle telle que nous la connaissons actuellement d'un grand nombre de services diversifiés, va très vite devenir impraticable. Afin de répondre en partie à ce problème, les protocoles de découverte de services (SDS [CZH+99, Nin], SLP [GPVD99], JINI [JIN], Salutation [Con, Pas99], UPnP [BO99, PP]) permettent à l'utilisateur de ne plus avoir à se soucier de l'adresse réseau du service désiré. La seule chose qui incombe à l'utilisateur est de donner le type et les attributs du service requis. Le rôle d'un protocole de découverte de services va alors être de se charger de résoudre et de trouver l'adresse de la machine sur laquelle se trouve le service désiré.

Cependant, tous les protocoles de services actuellement proposés concernent des réseaux locaux filaires. Or, il semble important de pouvoir continuer à offrir tous ces types de services y compris dans des réseaux ad hoc.

4.7.1 Fonctionnalités d'un protocole de découverte de services

Si l'on étudie les approches proposées [BR00, McG00] (notamment SLP, SDS, Jini), on retrouve à chaque fois la notion de *client* (entités cherchant des services), *provider* (entités possédant des *services*) et *server* (entité mettant en relation les *clients* et les *providers*). En détaillant ces trois approches, on peut mettre en valeur six fonctions principales qui caractérisent en quelque sorte la notion de « *protocole de découverte de services* » :

Localisation du serveur. La notion de serveur fait le lien entre les services/providers et les clients. Localiser un serveur est donc un point clef pour tout protocole de découverte de services. SDS et SLP proposent deux types d'approches similaires permettant aussi bien à un client qu'à un provider de localiser un serveur. Dans la première approche, le serveur s'annonce périodiquement sur un canal multicast attribué au protocole. Dans la seconde approche, c'est le client ou le provider qui initie la localisation du service.

Déclaration de services. Le deuxième aspect crucial est la façon dont les providers doivent déclarer leurs services. Dans SDS, les services doivent être annoncés périodiquement au niveau du serveur afin d'assurer le rafraîchissement des informations sur les déclarations de services. Dans SLP et JINI, la déclaration d'un service se fait par enregistrement et le provider spécifie la durée de validité de cet enregistrement.

Envoi des requêtes dans le réseau local. Une fois le serveur localisé et les services enregistrés, il reste à régler la façon dont les clients vont formuler et envoyer leur requête. Dans SDS, JINI et SLP et de façon tout à fait prévisible, un client envoie sa requête au serveur qu'il a découvert.

Passage à l'échelle et routage des requêtes entre serveurs. Dans l'hypothèse où une requête ne peut pas être satisfaite localement, *i.e.*, auprès du serveur local, il semble opportun de pouvoir la rediriger vers d'autres serveurs et permettre de trouver une éventuelle réponse positive. Ce passage à grande échelle engendre de nombreux problèmes concernant bien

évidement le routage des requêtes d'un serveur à l'autre mais aussi la façon dont les serveurs ont conscience de leur interconnexion, selon quelle topologie et quelles doivent être les informations échangées entre serveurs. De par l'utilisation de DHCP ou de recherche par anneaux croissants, SLP n'est pas très extensible de nature. Seul SDS traite du passage à grande échelle de la façon la plus poussée via l'utilisation de deux techniques : la mise en œuvre d'une hiérarchie entre les serveurs et l'agrégation des informations concernant les services au sein de cette hiérarchie.

Robustesse. La robustesse d'un protocole de déploiement et de découverte de services concerne principalement la défaillance d'un serveur et/ou d'un provider. La panne d'un service est gérée dans SDS par les annonces périodiques du provider. Notons que JINI propose un mécanisme intéressant permettant à un client de s'abonner aux différents évènements qui peuvent survenir au niveau d'un service obligeant ce dernier à en notifier les clients.

La sécurité. Découvrir et accéder à des services sur un réseau soulève inévitablement la question de la sécurité et de la confidentialité. Ces aspects ont été pris en compte dans SDS. En effet, les auteurs ont traité le contrôle d'accès des clients aux services non publics et ont utilisé aussi bien l'authentification que le cryptage pour assurer les communications entre les composants du protocole de déploiement et de découverte de services. Dans SLP, la sécurité n'est pas aussi performante que celle présentée dans SDS. En effet, l'intégrité des informations sur les services est garantie, mais les clients ne s'authentifient pas.

4.7.2 Protocole de localisation de services

Considérons le cas des réseaux ad hoc. A priori, il n'y a pas lieu de changer la sémantique ni des clients ni des providers. Le rôle du serveur apparaît plus problématique car il est difficile de faire reposer une architecture protocolaire sur une seule entité quand cette dernière est susceptible de quitter le réseau à tout moment ! Si l'on décide de ne pas mettre en œuvre de serveur, il reste l'alternative classique proactif/réactif : soit chaque provider diffuse périodiquement la totalité des services qu'il offre et, dans ce cas, chaque client est tenu de mettre à jour la liste des services qu'il écoute ; soit les services ne sont pas diffusés et c'est le client qui diffuse sa requête et dans ce cas, chaque provider étant en mesure de satisfaire une requête se doit d'y répondre ou du moins de prévenir le client de sa présence. L'inconvénient de la première approche est qu'elle génère une charge de trafic inutile due à la diffusion périodique dans tout le réseau des services. Un nœud n'est pas forcément intéressé par les déclarations de services en provenance d'une partie du réseau éloignée s'il est en mesure de trouver le même service dans son propre voisinage. De même, la seconde approche nécessite la diffusion des requêtes dans l'ensemble du réseau et ce chaque fois qu'un nœud cherche un service. Ces deux approches nous semblent mal adaptées car elles font toutes deux une utilisation de la diffusion sans chercher à optimiser les ressources réseaux et chacune d'elles laisse le choix du service au client, ce qui ne permet pas de mettre en œuvre des politiques d'optimisation globale des ressources (service le plus proche) ou de répartition de charge [GWBC99, FGC+97]. Afin de pallier les dégradations de performance induites par des inondations [FK02], il nous semble important d'introduire un troisième composant, jouant le rôle de serveur, que nous avons choisi de nommer médiateur [FK01b, FK01a] (pour se démarquer de l'approche réseau fixe).

Notre idée est de déployer un backbone mobile de fournisseurs de services composé de médiateurs. Ces derniers doivent couvrir les fournisseurs du réseau. La couverture d'un fournisseur signifie (1) contenir l'information sur son adresse et les services qu'il offre et (2) pouvoir utiliser cette information pour répondre aux requêtes des clients. Notre protocole de localisation de service se fonde sur un mécanisme d'élection de médiateurs qui garantit les propriétés suivantes :

1. Si un nœud u du réseau est un fournisseur alors il existe un médiateur m dans le 1-voisinage de u qui le couvre. De plus, chaque fournisseur connaît le ou les médiateurs qui le prennent en charge.
2. Si un médiateur du réseau n'écoute plus aucun service alors il ne peut plus être médiateur.
3. Chaque fournisseur possède un lien bidirectionnel avec au moins un des médiateurs qui le couvre.
4. Chaque médiateur qui couvre un fournisseur ne possède pas forcément un lien bidirectionnel avec lui (ce lien peut être unidirectionnel).
5. Chaque fournisseur peut être couvert par plus d'un médiateur.

Un fois ces médiateurs mis en place, il reste à définir la façon dont un client va procéder pour accéder à un médiateur puisque dans notre approche il représente le « *Sésame, ouvre toi* » vers l'ensemble des services offerts. L'idée sous-jacente est de créer une structure à laquelle vont participer tous les médiateurs afin de mettre en commun tout ou partie de leurs connaissances. L'un des buts étant d'optimiser le nombre de ressources du réseau, nous nous sommes naturellement acheminés vers l'utilisation d'une adresse multicast. Cette structure devra pouvoir servir aux clients pour pouvoir accéder à tous les services du réseau ad hoc sans pour autant perturber les parties du réseau où ne se trouvent ni médiateur ni client. Le problème qui se pose alors est la gestion de ce groupe de multicast : doit-on l'utiliser à des fins d'échanges d'information entre les médiateurs et/ou plus simplement afin de propager les requêtes des clients ? Un facteur décisif est la prise en compte des réponses aux requêtes et notre volonté de pouvoir effectuer de l'agrégation [MSFC02, IEGH01] sur les réponses, afin de minimiser le trafic dans le réseau mais aussi de permettre à notre application de gérer au mieux ses ressources. Pour ces différentes raisons, nous avons opté pour un arbre servant à la propagation des requêtes si le client n'est pas en mesure de trouver le service dans son 1-voisinage.

La première approche que nous proposons est une approche de multicast proactif dans laquelle des messages de signalisation sont périodiquement échangés afin d'offrir une infrastructure de routage pour les requêtes de type arbre partagé ou d'un maillage partagé. La deuxième approche est une approche réactive dans laquelle l'infrastructure multicast est créée à la demande du client. Cette structure peut être soit un arbre soit un maillage regroupant les différents arbres de requêtes déployés. Un argument en faveur du déploiement à la demande d'un arbre par le client est que le calcul d'un arbre multicast entretenu régulièrement n'est pas nécessaire car une requête, contrairement à une vidéo, réside le plus souvent dans un seul paquet IP ! Le trafic de contrôle nécessaire à la maintenance risque alors d'être plus important que le seul trafic utile dans l'arbre. L'autre argument en faveur de la mise en œuvre d'un multicast à la demande applicatif est la possibilité d'appliquer les différents principes vus précédemment sur les réseaux actifs. En effet, l'agrégation des réponses va reposer entièrement sur la structure d'arbre multicast. Cette agrégation dépend

fortement du contexte, du type de service. Chercher un service FTP n'entraîne pas a posteriori les mêmes coûts d'utilisation que la recherche d'un DNS. Il apparaît donc important de pouvoir déclarer différentes classes de services en fonction de l'impact qu'elles engendrent sur le réseau lors de leur utilisation. Nous avons effectué un grand nombre de simulations permettant de tester les différentes approches énoncées : proactif/réactif, arbre/maillage, agrégation/directe. Une mise en œuvre partielle de ce protocole a aussi été réalisée en adaptant SLP. Nous avons principalement rajouté deux types de message permettant l'annonce des services en local et la déclaration en local des providers qui sont pris en charge par un médiateur. L'approche mise en œuvre dans notre implémentation de SLPadhoc est de déployer un arbre multicast à la demande.

Comme pour le multicast, nous nous sommes également intéressés à la robustesse de ce type de protocole de découverte de services. L'approche médiateur tente simplement de réaliser une domination des providers par les médiateurs. Nous avons étudié ce qu'apporte la construction d'une double domination. Étonnamment, une 2-domination n'apporte pas beaucoup plus de robustesse car dans ce type de réseau, une 1-domination est déjà très redondante. Par contre, le fait d'obliger chaque provider à être couvert par 2 médiateurs augmente considérablement le nombre de médiateurs nécessaires, ce qui n'est pas souhaitable.

4.8 Conclusion

Les applications potentielles et les domaines d'application orientés grand public des réseaux ad hoc doivent prendre en compte les enjeux soulevés par la mise en œuvre et le déploiement à grande échelle de réseaux ad hoc. Le cas des réseaux hybrides révèle des intérêts contradictoires et parfois concurrents entre les différents protagonistes présents. D'un côté, le point de vue *scientifique/recherche* qui trouve là un nouveau moyen de liberté, d'étendre l'Internet de façon encore plus libre en lui apportant « *plus d'ubiquité* » et de l'autre côté, les opérateurs qui ne voient pas toujours d'un bon œil le fait que des réseaux spontanés, sans contrôle, puissent se créer, dans une bande de fréquence libre, permettant à plusieurs communautés de partager un accès câble modem ou ADSL. Le même raisonnement peut être tenu pour les fabricants de stations de base qui, si du jour au lendemain, la recherche prône que les stations de base sont inutiles, risquent de faire grise mine devant le manque à gagner que représente le marché du sans fil et l'équipement des infrastructures de ces réseaux sans fil.

Un moyen de concilier le plus grand nombre est d'appliquer la devise *diviser pour régner*. Dans ce cas précis, il s'agirait plutôt de *morceler* les différentes problématiques et solutions apportées pour ne pas laisser une communauté s'emparer entièrement d'une solution technique, monopole vite jugé non acceptable. Il faut peut-être effectivement accorder plus d'importance aux enjeux que peuvent avoir les différents modèles proposés et donc les différents scénarios d'utilisation des réseaux ad hoc et de leur implication. Si l'on reste sur le terrain militaire, tout devient beaucoup plus simple car les protagonistes ont l'entier monopole du marché par définition. Par contre dans le monde mouvementé des télécommunications, des réseaux et de la convergence attendue et redoutée de ces deux mondes, les réseaux ad hoc peuvent indubitablement apporter des solutions efficaces au déploiement d'infrastructures de réseaux domestiques, de réseaux d'accès haut débit, d'extension de couvertures existantes. Les réseaux ad hoc représentent un creuset potentiel de nouveaux services et de nouvelles applications mais là, on tombe dans le domaine des

prédictions, paysage nébuleux où il est toujours hasardeux de s’aventurer.

Bibliographie

[Abr85] N. Abramson, *Development of the ALOHANET*, IEEE Transaction on Information Theory **31** (1985), 119–123.

[AK73] N. Abramson and F. Kuo (eds.), *Computer-communication networks*, ch. The Aloha system, Prentice Hall PTR, 1973.

[Arn02] A. Arnaud, *Caractérisation de zones denses dans les réseaux mobiles ad hoc – application au multicast –*, Dea, ENS-Lyon, Lyon, France, Juin 2002.

[Bar97] S. Barton (ed.), *Wireless personal communication : Special issue on the high performance radio local area network (HIPERLAN)*, vol. 4, Kluwer Academic Publishers, January 1997.

[BCFG+97] T. Billhartz, B. Cain, E. Farrey-Goudreau, D. Fieg, and S. Gordon Batsell, *Performance and ressource cost comparisons for the CBT and PIM multicast routing protocols*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 3, 304–315.

[Ber73] C. Berge, *Graphes et hypergraphes*, Dunod, 1973, 2eme ed.

[Bey90] D. Beyer, *Accomplishments of the DARPA survivable adaptive networks (SURAN) program*, MILCOM Conference (Monterey, California, USA), 1990.

[BFNO02] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny, *Robust position-based routing in wireless ad hoc networks with irregular transmission ranges*, Wireless Communications And Mobile Computing journal (2002), (to appear).

[BLMT98] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, *AMRoute : Ad hoc multicast routing protocol*, Internet Draft draft-talpade-manet-amroute-00.txt, Internet Engineering Task Force, August 1998.

[BMSU01] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, *Routing with guaranteed delivery in ad hoc wireless networks*, Wireless Networks **7** (2001), no. 6, 609–616.

[BO99] C. Bengt and A. Olof, *Universal plug and play connects smart devices*, Windows Hardware Engineering Conference (WinHEC 99), 1999.

[BOT02] B. Bellur, R. Ogier, and F. Templin, *Topology broadcast based on reverse-path forwarding (TBRPF)*, Internet Draft draft-ietf-manet-tbrpf-05.txt, Internet Engineering Task Force, March 2002.

[BR00] C. Bettstetter and C. Renner, *A comparison of service discovery protocols and implementation of the service location protocol*, Sixth EUNICE Open European Summer School (Twente, Netherlands), 2000.

[CE95] S. Corson and A. Ephremides, *A distributed routing algorithm for mobile wireless networks*, ACM/Baltzer Journal of Wireless Networks **1** (1995), no. 1, 61–81.

[CF01] G. Chelius and E. Fleury, *Routage multicast dans les réseaux ad hoc : l'approche jumbo*, Mobiles-services et réseaux mobiles de 3ème Génération – Des architectures aux Services – (MS3G) (Lyon, France), Décembre 2001, (Dans le cadre des XIVèmes entretiens Jacques Cartier).

[CF02a] ———, *Ananas : A new ad hoc network architectural scheme*, RR RR-4354, INRIA, Janvier 2002.

[CF02b] ———, *Ipv6 addressing architecture support for ad hoc*, Internet Draft draft-chelius-adhoc-ipv6-archi-00.txt, Internet Engineering Task Force, August 2002.

[CF02c] ———, *Performance evaluation of multicast trees in ad hoc networks*, RR 4416, INRIA, Mars 2002.

[CFG00] J. Cohen, E. Fleury, and J. Gustedt, *JUMBO : protocole de routage unicast dans les réseaux ad hoc sans fil*, AlgoTel 2000 (La Rochelle), INRIA, May 2000, pp. 31–34.

[CFT02a] G. Chelius, E. Fleury, and L. Toutain, *Configuration automatique de réseaux IPv6*, Conférence IPv6 (Paris), Octobre 2002.

[CFT02b] ———, *Using ospfv3 for ipv6 router autoconfiguration*, Internet Draft draft-chelius-router-autoconf-00.txt, Internet Engineering Task Force, June 2002.

[CGZ97] C.-C. Chiang, M. Gerla, and L. Zhang, *Shared tree wireless network multicast*, International Conference on Computer Communications and Networks (ICCCN'97), IEEE, 1997.

[CGZ98] C.-C. Chiang, M. Gerla, and L. Zhang, *Forwarding group multicast protocol (FGMP) for multihop wireless networks*, ACM/Balzter Journal of Cluster Computing **1** (1998).

[Che01] G. Chelius, *Routage multicast dans les réseaux ad hoc*, Dea, ENS-Lyon, Lyon, France, Juin 2001.

[CM99] S. Corson and J. Macker, *Mobile ad hoc networking (MANET) : Routing protocol performance issues and evaluation considerations*, Request For Comments 2501, Internet Engineering Task Force, January, 1999.

[Con] Salutation Consortium, `http://www.salutation.org`.

[CRKGLA89] C. Cheng, R. Reley, S. Kumar, and J. Garcia-Luna-Aceves, *Loop-free extended bellman-ford routing protocol without bouncing effect*, ACM Computer Communications Review **19** (1989), no. 4, 224–236.

[CZH+99] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, and R. Katz, *An architecture for a secure service discovery service*, Mobicom'99 (Seattle, Washington, USA), 1999.

[DC90] S. Deering and D. Cheriton, *Multicast routing in datagram internetworks and extended LANs*, ACM Transactions on Computer Systems **8** (1990), no. 2, 85–110.

[Dev00] V. Devarapalli, *MZR : A multicast protocol for mobile ad hoc networks*, Internet Draft draft-vijay-manet-mzr-00.txt, Internet Engineering Task Force, November 2000.

[DPPS01] J. Díaz, M. Penrose, J. Petit, and M. Serna, *Approximating layout problems on random geometric graphs*, Journal of Algorithms **39** (2001), no. 1, 78–116.

[FGC+97] A. Fox, S. Gribble, Y. Chawathe, E. Brewer, and P. Gauthier, *Cluster-based scalable network services*, 16th Symposium on Operating Systems Principles (SOSP-97) (New York), vol. 31, Operating Systems Review, no. 5, ACM Press, October 5–8 1997, pp. 78–91.

[FHM00] E. Fleury, Y. Huang, and P. K. McKinley, *On the performance and feasibility of multicast core selection heuristics*, Networks **35** (2000), no. 2, 145–56.

[FK01a] E. Fleury and H. Koubaa, *A fully distributed mediator based service location protocol in ad hoc networks*, Globecom 2001 (San Antonio, Texas), 2001.

[FK01b] _____, *A performance study of a service covering protocol in ad hoc networks*, International Conference on Networks (ICON) 2001 (Bangkok, Thailand), IEEE, October 2001.

[FK02] _____, *Service location protocol overhead in the random graph model for ad hoc networks*, Symposium on Computers and Communications (ISCC'02) (Taormina, Italy), IEEE, July 2002.

[FL94] P. Fraigniaud and E. Lazard, *Methods and problems of communication in usual networks*, Discrete Applied Mathematics **53** (1994), 79–133, (special issue on broadcasting).

[GCZ98] M. Gerla, C.-C. Chiang, and L. Zhang, *Tree multicast strategies in mobile multihop wireless networks*, ACM/Balzter Mobile Networks and Applications Journal (1998).

[GHMP01] M. Gerla, X. Hong, L. Ma, and G. Pei, *Landmark routing protocol (LANMAR) for large scale ad hoc networks*, Internet Draft draft-ietf-manet-lanmar-03.txt, Internet Engineering Task Force, December 2001.

[GHP01] M. Gerla, X. Hong, and G. Pei, *Fisheye state routing protocol (FSR) for ad hoc networks*, Internet Draft draft-ietf-manet-fsr-02.txt, Internet Engineering Task Force, December 2001.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and intractability : A guide to the theory of NP-completeness*, Computer science / mathematics, W. H. Freeman and Company, 1979.

[GLA93] J. Garcia-Luna-Aceves, *Loop-free routing using diffusing computations*, IEEE / ACM Transactions on Networking **1** (1993), no. 1, 130–141.

[GLAM98] J. Garcia-Luna-Aceves and E. Madruga, *The core-assisted mesh protocol*, IEEE Journal on Selected Areas in Communications **17** (1998), no. 8, (Special Issue on Ad Hoc Networks).

[GPVD99] E. Guttman, C. Perkins, J. Veizades, and M. Day, *Service location protocol, version 2*, Request For Comments 2608, Internet Engineering Task Force, June 1999.

[GWBC99] S. Gribble, M. Welsh, E. Brewer, and D. Culler, *The MultiSpace : an Evolutionary Platform for Infrastructural Services,*, Usenix Annual Technical Conference, June 1999.

[HCH01] M. Hamdi, S. Capkun, and J.-P. Hubaux, *GPS-free positioning in mobile ad hoc net-works*, Hawaii International Conference on System Sciences (HICSS-34) (Hawaii), IEEE, January 2001.

[HDL+02] Z. Haas, J. Deng, B. Liang, P. Papadimitratos, and S. Sajama, *Wireless ad hoc networks*, Encyclopedia of Telecommunications (John Proakis, ed.), John Wiley, 2002.

[HHL86] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, *A survey of gossiping and broadcasting in communication networks*, Networks **18** (1986), 319–349.

[HIP95] *Radio equipement and system (res) ; High PErformance Radio Local Area Network (HIPERLAN), type 1, functional specification*, Technical Report ETS 300 652, European Telecommunication Standards Institute, December 1995.

[HT98] Z. Haas and S Tabrizi, *On some challenges and design choices in ad hoc communications*, Military Communications Conference (MILCOM) (Bedford, MA, USA), IEEE, October 1998, pp. 187–192.

[IEGH01] C. Intanagonwiwat, D. Estrin, R. Govindan, and J Heidemann, *Impact of network density on data aggregation in wireless sensor networks*, ICDCS, November 2001.

[Jac98] P. Jacquet, *éléments de théorie analytique de l'information, modélisation et évaluation de performances*, Tech. Report 3505, INRIA, 1998, ISSN 0249-6399.

[Jac00] _____, *Les réseaux mobiles ad hoc*, AlgoTel 2000 (La Rochelle), INRIA, May 2000.

[JC00] L. Ji and M. Corson, *Differential destination multicast (DDM) specification*, Tech. Report draft-ietf-manet-ddm-00.txt, Internet Engineering Task Force, 2000.

[JIN] *Jini technologie specifications*, `www.sun.com/jini/specs`.

[JLV+01] P. Jacquet, A. Laouiti, L. Viennot, T. Clausen, and P. Minet, *Optimized link state routing protocol extensions*, Internet Draft draft-ietf-manet-olsr-extensions-00.txt, Internet Engineering Task Force, 2001.

[JM96] D. Johnson and D. Maltz, *Mobile computing*, ch. Dynamic Source Routing in Ad Hoc Wireless Networks, pp. 153–181, Kluwer Academic Publishers, 1996, edited by T. Imielinski and H. Korth.

[JMQ+01] P. Jacquet, P. Muhletaler, A. Qayyum, A. Laouiti, T. Clausen, and L. Viennot, *Optimized link state routing protocol*, INMIC (Pakistan), IEEE, December 2001.

[JMQ+02] P. Jacquet, P. Muhletaler, A. Qayyum, A. Laouiti, L. Viennot, T. Clausen, and P. Minet, *Optimized link state routing protocol*, Internet Draft draft-ietf-manet-olsr-06.txt, Internet Engineering Task Force, March 2002.

[Joh94] D. Johnson, *routing in ad hoc networks of mobile hosts*, Workshop on Mobile Computing and Applications, 1994.

[JT87] J. Jubin and J. Tornow, *The DARPA Packet Radio NETwork protocols (PRNET)*, Proceedings of the IEEE, vol. 75, IEEE, January 1987, pp. 21–32.

[JV02] P. Jacquet and L. Viennot, *Overhead in mobile ad hoc network protocols*, Tech. Report 3965, INRIA, June 20002.

[KGBK78] S. Kahn, A. Gronemeyer, J. Burchfiel, and R. Kunzelman, *Advances in packet radio technology*, Proceedings of the IEEE, vol. 66, November 1978.

[KS71] L. Kleinrock and K. Stevens, *Fisheye : A lenslike computer display transformation*, Tech. report, UCLA Computer Science Department, 1971.

[Lao02] A. Laouiti, *Unicast et multicast dans les réseaux ad hoc sans fil*, Ph.D. thesis, Université de Versailles Saint-Quentin-en-Yvelines, Juillet 2002.

[LGT00] X. Lagrange, P. Godlewski, and S. Tabbane, *Réseaux gsm-dcs, 5e édition revue et augmentée*, Hermès, Juillet 2000, ISBN : 2746201534.

[McG00] E. McGrath, *Discovery and its discontents : Discovery protocols for ubiquitous computing*, Tech. report, National Center for Supercomputing Applications, 2000.

[MGLA96] S. Murthy and J. Garcia-Luna-Aceves, *An efficient routing protocol for wireless networks*, ACM Mobile Networks and Applications Journal (1996), (Special Issue on Routing in Mobile Communication Networks).

[MJK+00] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. De Couto, *Carnet : A scalable ad hoc wireless network system*, 9th ACM SIGOPS European workshop : Beyond the PC : New Challenges for the Operating System (Kolding, Denmark), ACM, September 2000.

[MSFC02] S. Madden, R. Szewczyk, M. Franklin, and D. Culler, *Supporting aggregate queries over ad hoc wireless sensor networks*, Workshop on Mobile Computing Systems & Applications (Callicoon, NY, USA), IEEE, June 2002.

[MW97] J. McQuillan and D. Walden, *The DARPA network design decision*, Computer Networks **1** (1997), 243–289.

[Nin] *The ninja project*, `http://ninja.cs.berkeley.edu`.

[Pär01] J. Pärkkä, *Wireless wellness monitor*, ERCIM News (2001), no. 46, (Special Theme : Human Computer Interaction).

[Pas99] B. Pascoe, *Salutation architectures and the newly defined service discovery protocols from Microsoft and Sun*, White paper, Salutation Consortium, June 1999, `http://www.salutation.org/whitepaper/Jini-UPnP.PDF`.

[PB94] C. Perkins and P. Bhagwat, *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*, ACM Computer Communications Review **24** (1994), no. 4, 234–244.

[PC97] V. Park and M. Corson, *A highly adaptive distributed routing algorithm for mobile wireless networks*, INFOCOM (Kobe, Japan), IEEE, 1997.

[PC01] _____, *Temporally-ordered routing algorithm (TORA)*, Internet Draft draft-ietf-manet-tora-spec-04.txt, Internet Engineering Task Force, July 2001.

[Per01] C. Perkins (ed.), *Ad hoc networking*, Addison Wesley Longman, 2001, ISBN 0-201-30976-9.

[PGC00] G. Pei, M. Gerla, and T. Chen, *Fisheye state routing : A routing scheme for ad hoc wireless networks*, International Conference on Communications (ICC) (New Orleans, LA, USA), IEEE, June 2000, pp. 70–74.

[PGH00] G. Pei, M. Gerla, and X. Hong, *LANMAR : Landmark routing for large scale wireless ad hoc networks with group mobility*, First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC) (San Francisco, CA, USA), ACM, November 2000, (in conjonction with GLOBECOM 2000).

[PH99] M. Pearlman and Z. Haas, *Determining the optimal configuration for the zone routing protocol*, IEEE Journal on Selected Areas in Communications **17** (1999), no. 8, 1395–1414, (Special Issue on Wireless Ad Hoc Networks).

[PP] Universal Plug and Play, `http://www.upnporg`.

[PR99] C. Perkins and E. Royer, *Ad hoc on-demand distance vector routing*, Workshop on Mobile Computing Systems and Applications (New Orleans, LA, USA), IEEE, February 1999, pp. 90–100.

[PRD02] C. Perkins, E. Royer, and S. Das, *Ad hoc on-demand distance vector (AODV) routing*, Internet Draft draft-ietf-manet-aodv-10.txt, Internet Engineering Task Force, January 2002.

[PvGT+00] J. Pärkkä, M. van Gils, T. Tuomisto, R. Lappalainen, and I. Korhonen, *A wireless wellness monitor for personal weight management*, Information Technology Applications in Biomedicine (ITAB-ITIS 2000) (Arlington, Virginia, USA), IEEE, November 2000, pp. 83–88.

[RP99] E. Royer and C. Perkins, *Multicast operation of the ad hoc on-demand distance vector routing protocol*, 5 th International Conference on Mobile Computing and Networking (MobiCom99) (Seattle, WA, USA), August 1999.

[Sat96a] M. Satyanarayanan, *Fundamental challenges in mobile computing*, Symposium on Principles of Distributed Computing (Philadelphia, PA, USA), ACM, May 1996.

[Sat96b] _____, *Mobile information access*, IEEE Personal Communications **3** (1996), no. 1.

[SDB98] R. Sivakumar, B. Das, and V. Bharghavan, *SPINE routing in ad hoc networks*, ACM/Balzter Journal of Cluster Computing (1998).

[SSB99] P. Sinha, R. Sivakumar, and V. Bharghavan, *Mcedar : Multicast core-extraction distributed ad hoc routing*, Wireless Communications and Networking Conference, IEEE, 1999, pp. 1313–1318.

[TB01] C. Toh and S. Bunchua, *Ad hoc mobile multicast routing using the concept of long-lived routes*, Journal of Wireless Communications and Mobile Computing **1** (2001), no. 3.

[Toh02] C.-K. Toh, *Ad hoc mobile wireless networks : Protocols and systems*, Prentice Hall PTR, 2002, ISBN 0-13-007817-4.

[Tsu88] P. Tsuchiya, *The landmark hierarchy : a new hierarchy for routing in very large networks*, Computer Communication Review **18** (1988), no. 4, 35–42.

[WPD88] D. Waitzman, C. Partridge, and S. Deering, *Distance vector multicast routing protocol*, Request For Comments 1075, Internet Engineering Task Force, November 1988.

[WTT98] C. Wu, Y. Tay, and C.-K. Toh, *Ad hoc multicast routing protocol utilizing increasing id-numbers (amris)*, Internet Draft draft-ietf-manet-amris-spec-00.txt, Internet Engineering Task Force, November 1998.

[WTT99] ———, *AMRIS : A multicast protocol for ad hoc wireless networks*, Military communications conference (MILCOM 99) (Atlantic City, NJ, USA), IEEE, November 1999, pp. 25–29.

[WWR01] *Book of visions 2001 - version 1.0*, Wireless World Research Forum, December 2001, `http://www.wireless-world-research.org`.

Publications

Livres, chapitre de Livre

[Fle00] E. Fleury (ed.), *Algotel 2000 : 2es rencontres francophones sur les aspects algorithmiques des télécommunications*, INRIA, may 2000, ISBN 2-7261-1157-2.

[FM02] E. Fleury and M. Marathe (eds.), *International workshop on discrete algorithms and methods for mobile computing and communications (dialm '2002)*, Atlanta, Georgia, USA, ACM Press, September 2002.

Journaux, conférences

[CF01] G. Chelius and E. Fleury, *Routage multicast dans les réseaux ad hoc : l'approche jumbo*, Mobiles-services et réseaux mobiles de 3ème Génération – Des architectures aux Services – (MS3G) (Lyon, France), Décembre 2001, (Dans le cadre des XIVèmes entretiens Jacques Cartier).

[CF02a] _____, *Ananas : A local area ad hoc network architectural scheme*, Mobile and Wireless Communications Networks (MWCN 2002) (Stockholm, Sweden), IEEE, Sept 2002.

[CF02b] _____, *Ananas : une architecture de réseau ad hoc*, AlgoTel 2002 (Mèze), INRIA, May 2002.

[CFG00] J. Cohen, E. Fleury, and J. Gustedt, *JUMBO : protocole de routage unicast dans les réseaux ad hoc sans fil*, AlgoTel 2000 (La Rochelle), INRIA, May 2000, pp. 31–34.

[CFT02] G. Chelius, E. Fleury, and L. Toutain, *Configuration automatique de réseaux IPv6*, Conférence IPv6 (Paris), Octobre 2002.

[CFU02] G. Chelius, E. Fleury, and S. Ubéda, *Merging ad hoc environment with wireless access : on overview*, Mediterranean Ad Hoc Networking Workshop (Med-hoc-Net 2002) (Sardegna, Italy), Sept 2002.

[FHM00] E. Fleury, Y. Huang, and P. K. McKinley, *On the performance and feasibility of multicast core selection heuristics*, Networks **35** (2000), no. 2, 145–56.

[FK01a] E. Fleury and H. Koubaa, *A fully distributed mediator based service location protocol in ad hoc networks*, Globecom 2001 (San Antonio, Texas), 2001.

[FK01b] _____, *A performance study of a service covering protocol in ad hoc networks*, International Conference on Networks (ICON) 2001 (Bangkok, Thailand), IEEE, October 2001.

[FK02a] _____, *Reflections on ad hoc cooperative teams*, Workshop on Mobile Ad Hoc Collaboration (Minnesota, USA), April 2002.

[FK02b] _____, *Service location protocol overhead in the random graph model for ad hoc networks*, Symposium on Computers and Communications (ISCC'02) (Taormina, Italy), IEEE, July 2002.

[KF00] H. Koubaa and E. Fleury, *Déclaration et découverte de services dans les réseaux ad hoc*, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2000) (Toulouse, France), Hermès, Octobre 2000.

Rapports de recherche, drafts IETF

[CF02a] G. Chelius and E. Fleury, *Ananas : A new ad hoc network architectural scheme*, RR RR-4354, INRIA, Janvier 2002.

[CF02b] _____, *Ipv6 addressing architecture support for ad hoc*, Internet Draft draft-chelius-adhoc-ipv6-archi-00.txt, Internet Engineering Task Force, August 2002.

[CF02c] _____, *Performance evaluation of multicast trees in ad hoc networks*, RR 4416, INRIA, Mars 2002.

[CFT02] G. Chelius, E. Fleury, and L. Toutain, *Using ospfv3 for ipv6 router autoconfiguration*, Internet Draft draft-chelius-router-autoconf-00.txt, Internet Engineering Task Force, June 2002.

[CFU02] G. Chelius, E. Fleury, and S. Ubéda, *Merging ad hoc environment with wireless access : an overview*, RR (en cours), INRIA, October 2002.

[TAI+02] France Télécom, ALCATEL, INRIA, LIP6, LRI, LSIIT, LSR-IMAG, SNCF, and TELECOM PARIS, *Safari, services ad hoc/filaires : Développement d'une architecture de réseau intégré*, Proposition de projet pré-compétitif, RNRT, Septembre 2002.

Logiciels

[CF02] G. Chelius and E. Fleury, ANANAS, 2002, `http://www.sourceforge.net/projects/ananas/`.

Travaux liés

[Arn02] A. Arnaud, *Caractérisation de zones denses dans les réseaux mobiles ad hoc – application au multicast –*, Dea, ENS-Lyon, Lyon, France, Juin 2002.

[Che01] G. Chelius, *Routage multicast dans les réseaux ad hoc*, Dea, ENS-Lyon, Lyon, France, Juin 2001.

[Kou03] H. Koubaa, *Localisation des services dans les réseaux ad hoc*, Ph.D. thesis, Université Henri Poincaré, Nancy, Janvier 2003.

Chapitre 5

Conclusion et perspectives

C’est surtout ce qu’on ne comprend pas qu’on explique.

Jules Amédée BARBEY D’AUREVILLY

5.1 Projet de recherche

À la sempiternelle interrogation sur le bien-fondé d'un projet de recherche afin de discerner s'il abonde dans le sens de ce but noble qui est de « *faire avancer la science* », je m'en remets au jugement de mes pairs. Mon seul élément de réponse égoïste est d'une part que la rédaction fut, je pense, bénéfique et surtout que je reste motivé par *cette recherche*. La convergence des services et le support multi-technologies représentent des enjeux majeurs pour les architectures des futures générations de réseaux. C'est dans ce contexte que je vais détailler mon projet de recherche et les défis associés sans occulter les diverses perspectives déjà mentionnées au cours de ce manuscrit.

5.1.1 Les défis

L'exploitation de supports multiples permet d'envisager l'ubiquité des services. Elle est à la base des réseaux dits de 4ème génération. Le modèle ad hoc permet d'étendre la couverture des réseaux d'accès sans fil (802.11b/a) en mode base et l'utilisation conjointe de ces modes permet, outre l'extension de la connectivité, la création de services novateurs dont les domaines d'application vont des réseaux embarqués aux réseaux domestiques en passant par les espaces intelligents. En raison de leur potentiel important, les technologies ad hoc deviennent candidates à l'intégration dans le réseau de services que devra être l'Internet du futur. Dans cette évolution, le protocole IPv6 joue également un rôle majeur, d'une part en raison de ses capacités d'adressage quasi infinies mais surtout par la prise en compte et la disponibilité de composants pour la qualité de service et la mobilité. Finalement, la configuration, la sécurité des services, le contrôle et la supervision de la qualité des services délivrés aux usagers dans cet espace de communication universel deviennent vitaux pour le succès des services qui seront offerts sur ces infrastructures. L'aboutissement de cette convergence requiert aujourd'hui des innovations technologiques dans plusieurs couches protocolaires qui interviennent dans la livraison des services (réseaux, multicast, sécurité, passerelle applicative, supervision). Cependant, seule une approche transversale faisant intervenir la coopération entre ces différents niveaux permettra de lever les verrous qui restent sur le chemin d'une offre de services fondée sur l'intégration ad hoc/filaire [TAI+02].

Les défis concernant les réseaux sans fil ad hoc proviennent en majeure partie des spécificités mêmes de ces réseaux : rayon de transmission limité, médium diffusant, taux d'erreur important dans les transmissions hertziennes, introduction de la mobilité qui n'est plus occasionnelle ou exceptionnelle mais normale, contraintes énergétiques fortes et pour couronner le tout, les aspects de sécurité exacerbés dans un tel environnement. Cette liste impressionnante de problèmes potentiels explique peut-être l'engouement du monde de la recherche pour ce *nouveau* domaine. Je pense que l'on peut aussi y ajouter l'équation suivante :

$$\begin{array}{l}\text{hétérogénéité}\\ \text{des}\\ \text{équipements}\end{array} \times \begin{array}{l}\text{foisonnement}\\ \text{des critères}\\ \text{de performance}\end{array} + \begin{array}{l}\text{augmentation}\\ \text{des sources}\\ \text{de financement}\end{array} \Rightarrow \begin{array}{l}\text{augmentation sensible des}\\ \text{activités de recherche}\end{array}$$

On peut s'appuyer sur le modèle en couche pour mettre en lumière les problèmes difficiles [Vai01] présents à chacun des étages mais comme nous l'avons déjà évoqué, les défis se trouvent aussi dans les interactions nécessaires entre les couches. Un défi va être d'identifier les compromis judicieux qu'il est nécessaire d'étudier entre les couches : une smart antenna et un

protocole de routage peuvent être optimisés et améliorés si tous deux partagent des informations. Aura-t-on un jour un Ad hoc Level Framing (ALF) ?

Couche physique. Traditionnellement, il existe peu ou pas d'interaction entre la couche physique et les couches supérieures. Pour pouvoir tirer parti des futures fonctionnalités *intelligentes* de la couche physique, il va falloir changer cet état de fait. C'est la couche physique qui est à même de choisir le type de modulation en fonction des conditions de propagation d'un canal. Néanmoins, si ce changement est totalement transparent pour la couche supérieure, la couche MAC n'est plus en mesure de connaître la durée d'un transfert. Il en va de même pour tout ce qui touche au contrôle de puissance qui permet de limiter la consommation électrique [Eph02], de réduire les interférences et donc permet une meilleure réutilisation spatiale. Néanmoins, un contrôle de puissance sans interaction avec les couches supérieures (MAC, routage) entraîne d'autres problèmes. Essayer d'ajuster au mieux la puissance d'émission génère de nombreuses situations de « *nœud caché* ». L'autre domaine où une interaction forte est requise entre la couche physique, les couches MAC et le routage est la mise en œuvre d'antennes intelligentes (*smart antenna*). Les problèmes qui se posent concernent la mise en œuvre de protocoles de routage unicast, broadcast et multicast qui prennent en compte les fonctionnalités offertes par ces antennes. Le dernier point que je mentionnerai est la nécessité d'avoir, dans les outils de simulation employés, des modèles plus réalistes pour la propagation, notamment pour tout ce qui est simulation indoor. Un lien radio n'est pas binaire (on/off) et ne dépend pas uniquement de la distance [GU01b, GU01a]. Le modèle a des incidences sur la simulation et l'étude des protocoles supérieurs [TBLG99].

Couche liaison. Les points clefs que doit traiter cette couche dans le contexte des réseaux ad hoc est l'accès au médium, la mise en œuvre de mécanismes de retransmission et de processus d'ordonnancement des émissions. Il est souhaitable que cette couche intègre des mécanismes de QoS qui soient complètement distribués et fonctionnent en liaison avec la couche de routage [CGL02].

Couche réseau. Les approches réactives et proactives résolvent en partie les problèmes de routage mais pour le moment, ils ne tiennent pas compte des interactions possibles ni avec la couche MAC, ni avec la couche physique, dans un souci de complète indépendance alors que ces informations peuvent se révéler précieuses pour tenir compte des contraintes de QoS [RLP00]. Il existe peu de propositions de protocole de routage ad hoc adaptatif alors qu'il apparaît clairement que le choix de tel ou tel type de protocole va dépendre indéniablement de la mobilité des nœuds, de la dynamique de la matrice de trafic. Le problème majeur qui se pose est de savoir quantifier le moment où une approche réactive devient meilleure qu'une approche proactive.

Couche transport. On aborde les contrées de TCP, roi de la couche transport. Pourtant cette suprématie est mise à mal dans le monde du sans fil car de nombreuses études montrent que ses performances se dégradent vite en cas de rupture de route. Le principal problème est que TCP réagit aux pertes de paquets dus à des collisions ou à des modifications de la topologie en administrant un remède conçu pour traiter la congestion du réseau, en réduisant la taille des fenêtres de transmission. Il s'ensuit une dégradation du débit qui était inutile. Un autre problème est que les caractéristiques d'une route peuvent changer du tout au tout. Com-

ment TCP doit-il prendre en compte ces changements pour les répercuter sur ses propres paramètres (timeout, taille des fenêtres de congestion) ?

Architectures hybrides. Nous allons revenir plus en détail sur ce point qui est plus général et qui couvre divers sous-thèmes. Le problème majeur est d'utiliser une infrastructure hybride alliant à la fois un réseau fixe composé d'une partie filaire et/ou de zones de couverture assurées par des bases et connexion de type ad hoc permettant d'étendre la couverture des bases. Ce type d'architecture pose le problème de l'intéropabilité entre différents modèles et protocoles de mobilité. Le but est d'assurer la continuité des applications en passant d'un système de mobile à un autre (*e.g.*, station de base vers un mobile ad hoc) et ainsi exécuter un handoff vertical. Dans la mise en œuvre de telles architectures, on peut se poser la question de la sécurité qui, j'en conviens tout à fait, peut constituer à elle toute seule un domaine de recherche : sécurité du routage, authentification.

Algorithmique. Pour finir, citons le domaine de l'algorithmique distribuée dont l'importance devient capitale pour les réseaux ad hoc. Les algorithmes distribués conçus dans le cadre des réseaux filaires (RIP, DVMRP) restent corrects au sein d'un réseau ad hoc mais leurs performances risquent d'être mauvaises du fait des hypothèses initiales. En effet, ces algorithmes supposent que la perte d'un lien est un évènement aléatoire et peu probable alors que dans les réseaux ad hoc la mobilité, processus souhaité et donc normal, entraîne des corrélations entre les pertes de liens. Prendre en compte cette corrélation au sein des algorithmes distribués est un problème ardu. De même, les complexités des algorithmes sont classiquement exprimées en fonction de N et ρ (nombre de sommets et probabilité d'erreur sur un lien). Il semble important de pouvoir mesurer la complexité des algorithmes distribués conçus pour les réseaux mobiles en intégrant ce paramètre de mobilité, mais il faut sans doute tenter de définir d'autres mesures, comme la robustesse.

5.1.2 Perspectives

Au vu des différents axes généraux mentionnés ci-dessus, il me semble que la communauté est trop focalisée sur le routage. Je ne prétends pas classer le dossier routage ! Le routage reste un problème difficile et il représente le premier service à mettre en œuvre au sein d'un réseau ad hoc. Néanmoins, les approches réactives et proactives actuelles sont efficaces et fonctionnent. Le routage est indispensable mais il ne sera pas la clef de tous les problèmes et les compromis nécessaires entre les couches et/ou le développement d'une couche physique intelligente sont aussi des domaines qui doivent être considérés et explorés avec attention. Cette section présente plus précisément les thèmes que je souhaite poursuivre ou entreprendre. Ces thèmes s'inscrivent dans les axes généraux que nous venons d'évoquer.

Architectures hybrides

Les réseaux hybrides (voir figure 4.7 page 101), alliant un réseau avec infrastructure (filaire et/ou cellulaire) et des réseaux ad hoc permettant d'étendre les zones de couverture cellulaire classique sont le cadre général dans lequel vont s'inscrire mes recherches futures.

Le raccordement de réseaux ad hoc à des réseaux plus conventionnels se fera au travers d'un équipement particulier que nous désignons sous le terme de passerelle. Cette passerelle de-

vra acheminer le trafic unicast entre les réseaux filaires et les réseaux ad hoc, mais elle devra également acheminer le trafic multicast. Outre l'architecture des nœuds mobiles et de la passerelle, il faut définir et mettre en œuvre l'ensemble des fonctionnalités minimales requises pour découvrir une passerelle et être en mesure de la joindre. S'il existe différentes passerelles, chacune étant sur des sous-réseaux IP différents, il va être nécessaire de mettre en œuvre un couplage « Mobile IP/Manet ». Ce couplage représente une nouvelle thématique de recherche que je compte mener. On peut faire une analogie entre ce modèle de réseau et le concept cellulaire : le roaming est géré par Mobile IP et le handover est pris en charge par le routage ad hoc. C'est en fait plus complexe car, en plus de Mobile IP, il faut aussi gérer les approches où la micro-mobilité (Cellular IP) est présente. La prise en charge du handoff vertical se fait donc entre trois protocoles de gestion de mobilité : macro-mobilité, micro-mobilité et ad hoc.

Couche radio idéale

La plupart des fonctionnalités réseau avancées (multicast, qualité de service, autoconfiguration) sont difficiles à mettre en place sur un réseau sans fil utilisant la technologie actuelle (la plupart des cartes existantes sont basées sur la norme IEEE 802.11). Une couche radio plus complète permettrait de développer plus facilement ce type de fonctionnalités. Par exemple, un mécanisme de priorité d'accès au médium comme celui proposé dans HIPERLAN 1 permet de réaliser une différentiation des services. Citons comme autre mécanisme souhaitable : la synchronisation et la possibilité de faire des sauts de fréquence courants. Cette API radio est aussi un bon moyen pour cerner les interactions nécessaires entre les couches physiques, MAC et réseau.

Une étude intéressante consiste à rassembler l'ensemble des mécanismes dont on voudrait doter une carte radio idéale et tenter ensuite d'en extraire une API minimale permettant de les piloter. L'idée est tout d'abord de se libérer des contraintes fortes imposées par les normes radio existantes pour pouvoir concevoir les protocoles les plus judicieux, indépendamment de la technologie. Bien sûr, seules les solutions envisageables technologiquement doivent être considérées. L'étape suivante consiste à coller cette API sur les couches radio existantes. Cela permettrait d'identifier ce qu'on peut faire ou pas avec telle ou telle interface radio. Outre déterminer ce qu'il est possible d'implémenter avec la norme IEEE 802.11, une attention particulière doit être portée à la récente spécification de Bluetooth. Cette spécification récente est très contraignante dans son fonctionnement mais offre néanmoins de nombreux mécanismes évolués comme la synchronisation et le saut de fréquence.

Consommation d'énergie

En lien avec la problématique précédente, il me semble important d'aborder la problématique de la consommation d'énergie des opérations de diffusion dans les réseaux ad hoc. Le but n'est pas de redémontrer la NP-complétude, cela a été fait dans le cadre de l'algorithmique théorique par A. CLEMENTI, P. PENNA et R. SILVESTRI dans STACS 2000 [CPS00, CPS99][1]. Ce qui me

[1] Je cite ces travaux car ce résultat a été redémontré dans un article paru dans GlobeCom 2002, ce qui révèle que la couche STACS et la couche GlobeCom restent assez hermétiques ! J'insiste sur le fait que l'article en question ne porte pas que sur cette preuve de NP-complétude mais étudie des heuristiques pour la mise en œuvre de flooding. Le propos de cette remarque n'est pas de remettre en cause les fondements scientifiques de cet article. Je désire juste noter l'importance à faire notre possible pour rester ouvert aux autres domaines.

paraît important est que le modèle employé dans les divers travaux publiés[2] ne prend en compte que l'émission, *i.e.*, le modèle sous-jacent est un graphe simple. Il faut encore une fois considérer la caractéristique intrinsèque du médium radio diffusant. Un nœud qui reçoit un paquet parce qu'il se trouve à portée du nœud émetteur « doit le consommer », ce qui entraîne une consommation d'énergie. Un premier modèle simpliste est de considérer qu'envoyer un message de taille L à distance d coûte en terme d'énergie $\alpha d^2 L$ (alpha est une constante) et que recevoir un message coûte βL. Le coût total d'une opération de diffusion va alors être $E = \alpha L \sum_{i=1}^{n} d_i^2 + \beta L \sum_{i=1}^{n} \delta(d_i)$ où $\delta(d_i)$ est le nombre de nœuds accessibles par le nœud i lorsque ce dernier émet à une distance/puissance d_i.

Il est donc important de prendre en compte ces deux facteurs car si un schéma de diffusion est optimal pour les émissions, il peut se révéler catastrophique pour les réceptions. Il est facile de dériver des bornes triviales sur E : $E \geq N\beta L$ (tous les nœuds doivent recevoir le message au moins une fois) et $E \leq \alpha D^2$ où D est le diamètre du réseau. Si un nœud émet avec une puissance maximale (D) permettant de couvrir tout le réseau, il effectue une opération de diffusion en une étape et chaque nœud reçoit le message une seule fois. Notons que ce n'est pas toujours possible dans la réalité (limitation de la puissance des nœuds) et que cela ne constitue qu'une borne supérieure et pas un minimum. En effet, en fonction des valeurs des deux constantes α et β, on peut trouver un optimum sur les différentes puissances nécessaires à chaque nœud relais.

Si l'on ne fait aucune autre hypothèse, ce problème semble assez hermétique. On peut par contre l'étudier dans un contexte plus simple où l'on espère pouvoir en déduire des bornes analytiques. On peut par exemple supposer que les mobiles sont distribués uniformément dans l'espace avec une densité donnée [JV02]. Le problème s'apparente à la couverture totale d'un espace par des boules de telle sorte que la somme de toutes les intersections entre ces boules soit minimum. Un problème lié est la couverture d'une sphère par des boules (par exemple pour la confection de balles de golf !) et ce problème trouve des solutions grâce à la géométrie algébrique [Mig97]. L'autre hypothèse est de discrétiser l'espace et de supposer que les mobiles sont répartis sur une grille régulière. Dans le cas d'un espace à deux dimensions, les boules de diamètre d sont alors les $2d(d+1)$ ou les $4d(d+1)$ voisinages (extension à d sauts des 4 et 8 voisinages classiques en traitement d'image). Arriver à trouver des bornes sur les puissances nécessaires pour diffuser peut ensuite permettre de rajouter un critère de consommation d'énergie dans la sélection des MPR.

Algorithmique distribuée

Je ne vais pas (re)développer ici mon point de vue sur la nécessité d'avoir des modèles parfois un peu trop théoriques et trop éloignés des trames qui passent sur un non lien radio ! À plus long terme et en liaison avec la prolifération actuelle des propositions de protocoles de routage unicast ad hoc, je pense qu'il serait bon d'avoir un modèle permettant de comparer ces divers protocoles, d'avoir des critères et des mesures fiables. Le ou les modèles restent à inventer pour les graphes dynamiques afin qu'ils intègrent la mobilité de façon intrinsèque. Il serait bon de pouvoir caractériser une topologie ad hoc. Dans [KE02], des études ont été faites sur le trafic dans un réseau sans fil classique (infrastructure de points d'accès) mais malheureusement aucun

[2] Du moins ceux que je connais car étant donné le nombre de conférences, workshops, écoles, il est difficile de tout connaître et ce, paradoxalement, malgré la quantité astronomique d'informations contenues et disponibles dans la toile !

modèle n'a été dégagé permettant de reproduire des topologies et des scénarios d'utilisation de ce type de réseau. Peut-on retrouver certaines lois de puissance ? Sous quelles hypothèses ? Cette prospective est à plus long terme car elle demande l'investissement de plusieurs communautés de recherche.

Dans les réseaux ad hoc, nous sommes en présence de groupes très dynamiques et il convient de se poser la question des paramètres pertinents que doivent optimiser les protocoles. La notion de connexité est très différente lorsque les nœuds ne sont actifs que sur certaines périodes (pour optimiser la consommation d'énergie par exemple). De plus, les interactions que réaliseront ces équipements correspondent à des recherches de services suivies de l'utilisation de ces services. On peut donc imaginer un modèle basé sur la probabilité de présence à faible « distance » d'un nœud du service dont il a besoin. Une notion de nœud critique paraît également importante : quels sont les nœuds qui ne doivent pas s'endormir (où limiter l'endormissement) pour maintenir un certain niveau de critère ? Peut-on les détecter efficacement ? Dès lors, on arrive à un problème d'équité : aucun nœud ne doit voir ses ressources trop fortement diminuées parce qu'il travaille pour la maintenance du réseau, au détriment de ses propres besoins. Pour réussir, nous devons étudier des processus d'interactions locales extrêmement simples qui permettront de tendre vers des propriétés globales. Les algorithmes/protocoles doivent impérativement être totalement distribués, c'est-à-dire ne devant requérir que très peu de connaissances sur le réseau (on peut imaginer qu'un nœud ne connaît que son k-voisinage de façon précise, k étant typiquement égal à 2 comme dans OLSR) et les décisions que doivent prendre les nœuds doivent se faire de façon totalement autonome et locale.

L'analyse de ces processus devra dans un premier temps être menée par simulation (l'étude analytique de ce genre de processus étant au-delà de mes connaissances). Le but de ces études et de nos travaux doit être de mettre en évidence le « bon comportement » de nos algorithmes. Il est important que les algorithmes proposés n'aient pas de comportement chaotique, qu'ils ne dépendent pas de l'état initial, qu'ils soient très peu perturbés par de faibles changements locaux. Toutes ces bonnes propriétés globales sont difficiles à vérifier et même à évaluer dans certains cas. L'ensemble des ces pré-requis peut constituer une définition de robustesse des protocoles mais elle s'avère extrêmement complexe à quantifier dans de tels réseaux.

Ces travaux de recherche à long terme représentent une opportunité de collaboration avec la communauté scientifique qui possède les modèles appropriés à notre cadre d'étude car, comme je l'ai noté précédemment, ces « *bons* » modèles sont en dehors du cercle de mes connaissances. Ces modèles que nous nous proposons d'utiliser pour tenter de comprendre et de maîtriser les phénomènes mis en jeu, sont des modèles dynamiques discrets, qui ont initialement été introduits pour la description de systèmes physiques ou biologiques : ce sont notamment des modèles d'interaction à base de graphes comme les « *tas de sable* », les « *Chip firing games* » ou certaines de leurs généralisations. Dans ces modèles, le système est représenté par un graphe valué et par des règles d'évolution locale ; au cours du temps (le plus souvent discret), les règles s'appliquent et transforment le graphe, changeant les valuations ou modifiant la structure. On s'intéresse alors à l'ensemble des états accessibles à partir d'un état donné en fonction de l'ensemble de règles. Dans le contexte qui nous intéresse, il s'agira de mettre en évidence les propriétés que l'on souhaite préserver et par la suite les règles (protocoles) à utiliser.

IPv6

J'ai hésité à mettre IPv6 comme thématique de recherche à part entière. Si j'affiche ce point, c'est pour mettre en exergue le fait que le protocole IPv6 joue un rôle majeur, notamment en raison de ses capacités d'adressage quasi infinies, du fait qu'il offre des composants pour la qualité de service et la mobilité, et étant donné qu'IPv6 est encore *jeune*, tout n'est pas encore figé. De plus, comme je l'ai noté dans le chapitre 4, l'utilisation d'IPv6 ne résout pas tous les problèmes. Prenons le cas du mécanisme d'adressage et d'allocation d'adresse. A priori, l'auto-configuration présente dans la pile protocolaire IPv6 permet de résoudre en partie le problème : si aucun routeur n'envoie d'annonce, chaque terminal va s'auto-configurer avec l'adresse de type lien local. À l'inverse, si un terminal reçoit des messages d'annonce de routeur, il peut configurer son préfixe en utilisant l'option « information sur le préfixe » des annonces de routeur. Ce mécanisme peut néanmoins présenter des inconvénients si le nombre de routeurs générant des annonces est important (une élection au niveau ad hoc, permettant de sélectionner le routeur le plus approprié suivant des critères à définir, est nécessaire) ou si les routeurs sont eux-mêmes mobiles. Il est important de mener des travaux à court terme sur les modifications et le comportement induits par IPv6 sur un nœud ad hoc et sur la comparaison entre une approche que l'on peut qualifier de proactive pour laquelle les routeurs envoient des annonces périodiquement et une approche réactive où ils réagissent aux sollicitations des hôtes.

5.2 Conclusion

Je tiens à remercier mes lecteurs d'avoir atteint l'ultime chapitre de ce manuscrit. Je ne compte pas résumer ici chaque chapitre puisque cette figure imposée est en partie incluse à la fin de chacun d'entre eux. J'aimerais apporter un point de vue plus personnel sur ce monde des réseaux et des télécommunications, avant de passer à une partie plus prospective. La convergence télécommunication/réseau est sans aucun doute là, inévitable et souhaitable, IP étant devenu « *le* » grand unificateur. On commence à discerner les applications et perspectives offertes. Si l'on se replace dans le monde alors clos du calcul parallèle, ce dernier se trouve propulsé au rang de métacomputing au dessus de la grille ! En analysant cette évolution, on voit apparaître des fissures dans le modèle en couche hermétique : la communauté des applications gourmandes en calcul et mémoire commence à s'intéresser au réseau car c'est lui qui relie et compose *La grille*. Cette même communauté ose afficher des velléités de sécurité et de QoS. La prise en compte du facteur humain (avec toutes les susceptibilités et difficultés attachées à cet adjectif) pour faire coopérer diverses communautés de recherche contiguës, même si elles apparaissent comme étant dans des univers éloignés, est un paramètre difficilement appréhendable. Malheureusement, cette difficulté paradoxale de communication se retrouve à l'identique dans les couches basses comme si chaque couche était un univers clos. Si l'application s'intéresse au réseau, l'inverse devrait être vrai. Dire que tout est déjà résolu ou sera résolu uniquement au niveau réseau serait faux et l'expérience de VTHD/VTHD++ le confirme. Le potentiel est présent mais il reste un effort non négligeable à faire pour l'exploiter au mieux, pour définir ces nouveaux services promis, pour s'assurer de la compréhension mutuelle et être sûr d'appréhender les mêmes problèmes afin, le cas échéant, de ne pas redéfinir un concept connu et maîtrisé à l'étage supérieur ou inférieur. Il me semble important de ne pas rester cantonné dans un seul domaine et il faut faire son possible pour pro-

fiter de l'expérience de domaines contigus (Grid/Network, Adhoc/Smart antena, Graph/Network On Chips). Les recherches menées dans le domaine des Sciences et Technologies de l'Information et de la Communication doivent s'orienter vers de nouveaux horizons qui ne doivent pas seulement étendre la portée, les fonctionnalités et l'efficacité des applications et des services. Elle doivent aussi tenter de les rendre disponibles de la façon la plus naturelle et la plus sure à tout citoyen. Ce nouveau concept de disponibilité à tout moment en tout endroit, que l'on nomme souvent « *Ambient Intelligence* », doit mettre la personne au centre du développement des STIC afin de « concevoir des technologies pour la personne et non pas faire que la personne s'adapte aux technologies ». Cela doit être la base de technologies *invisibles, i.e.*, qui se fondent dans notre environnement quotidien et qui sont présentes au moment où nous en avons besoin, tout en offrant une interaction simple et conviviale.

Bibliographie

[CGL02] C. Chaudet and I. Guérin Lassous, *BRuIT : Bandwidth reservation under interferences*, European Wireless (EW), 2002, pp. 466–472.

[CPS99] A. Clementi, P. Penna, and R. Silvestri, *Hardness results for the power range assignement problem in radio networks*, International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (RANDOM/APPROX'99), Lecture Notes in Computer Science, vol. 1671, Springer-Verlag, 1999, pp. 197–6208.

[CPS00] _____, *The power range assignement problem in radio networks on the plane*, XVII Symposium on Theoretical Aspects of Computer Science (STACS'00), Lecture Notes in Computer Science, vol. 1770, Springer-Verlag, 2000, pp. 651–660.

[Eph02] A. Ephremides, *Energy concerns in wireless networks*, IEEE Wireless Communications (2002), 48–59.

[GU01a] J.-M. Gorce and S. Ubéda, *Algorithme multi-résolution dans le domaine de fourier pour la planification radio par la méthode des flux partiels (parflow)*, Algotel'2001 (St Jean de Luz , France.), INRIA, may 2001, pp. 169–176.

[GU01b] J.M. Gorce and S. Ubéda, *Propagation simulation with the parflow method : fast computation using a multi-resolution scheme.*, IEEE 54th Vehicular Technology Conference VTC Fall2001 (Atlantic City, NJ, USA), IEEE VTS, october 2001.

[JV02] P. Jacquet and L. Viennot, *Overhead in mobile ad hoc network protocols*, Tech. Report 3965, INRIA, June 20002.

[KE02] D. Kotz and K. Essien, *Analysis of a campus-wide wireless network*, International Conference on Mobile Computing and Networking (MobiCom'02) (Atlanta, Georgia, USA), ACM, ACM press, September 2002, pp. 107–118.

[Mig97] T. Mignon, *Systèmes linéaires de courbes planes*, Ph.D. thesis, Université de Nice Sophia-Antipolis, Nice, France, Novembre 1997.

[RLP00] E. Royer, S.-J. Lee, and C. Perkins, *The effects of mac protocols on ad hoc network communication*, Wireless Communications and Networking Conference (WCNC), IEEE, September 2000, pp. 543–548.

[TAI+02] France Télécom, ALCATEL, INRIA, LIP6, LRI, LSIIT, LSR-IMAG, SNCF, and TELECOM PARIS, *Safari, services ad hoc/filaires : Développement d'une architecture de réseau intégré*, Proposition de projet pré-compétitif, RNRT, Septembre 2002.

[TBLG99] M. Takai, R. Bagrodia, A. Lee, and M. Gerla, *Impact of channel models on simulation of large scale wireless networks*, International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (Seattle, Washington, USA), ACM, ACM Press, August 1999, (In conjunction with MobiCom'99).

[Vai01] N. Vaidya, *"open" problems in mobile ad hoc networking*, Workshop on Wireless Local Networks (in conjunction with 26th Conference on Local Computer Networks) (Tampa, Florida, USA), November 2001, Keynote talk.

Glossaire

A

ABAM : Associativity-Based Ad Hoc Multicast, 90

ABR : Associativity Based Routing protcol, 85

ACK : Acknowledgment, 60

ADMR : Adaptive Demand-Driven Multicast Routing protocol, 87

AFDL : Add First Delete Last, 64

AMRIS : Ad hoc Multicast Routing protocol utilizing Increasing id-numbers, 89

AMRoute : Ad hoc Multicast Routing Protocol, 89

ANEP : Active Network Encapsulation Protocol, 62

AODV : Ad hoc On Demand Distance Vector routing protocol, 85

API : Application programming Interface, 60

AS : Autonomous System, 44

ATM : Asynchronous Transfer Mode, 47

B

BFS : Breath First Search, 57

BRP : Bordercast Resolution Protocol, 85

BS : Base Station, 79

BSR : Backup Source Routing protocol, 85

C

CAMP : Core-Assisted Mesh Protocol, 91

CBF : Core Based Forwarding, 33

CBRP : Cluster Based Routing Protocol, 85

CBS : Core-Binding Server, 47

CBT : Core-Based Tree, 40

CEDAR : Core Extraction Distributed Ad hoc Routing, 85

CGSR : Clusterhead Gateway Switch Routing protocol, 83

D

DAG : Directed Acyclic Graph, 85

DBF : Distributed Bellman-Ford routing protocol, 83

DDM : Differential Destination Multicast, 88

DFAL : Delete First Add Last, 64

DIFFSERV : Differentiated Services, 59

DIS : Distributed Interactive Simulation, 35

DR : Designated Routeur, 48

DREAM : Distance Routing Effect Algorithm for Mobility, 86

DSDV : Distance Source Distance Vector routing protocol, 83

DSR : Dynamic Source Routing protocol, 85

DSR-MB : Simple Protocol for Multicast and Broadcast using DSR, 87

DSRFLOW : Flow State in the Dynamic Source Routing protocol, 85

DTDV : Higly Dynamic Destination-Sequenced Distance Vector routing protocol, 83

DVMRP : Distance Vector Multicast Routing Protocol, 38, 87

F

FLIT : flow control digit, 10

FORP : Flow Oriented Routing Protocol, 85

FSR : Fisheye State Routing protocol, 85

FTP : File Transfer Protocol, 36

G

GLS : Geographic Location Service, 86

GSMP : General Switch Management Protocol, 59
GSR : Global State Routing protocol, 85

H
HARP : Hybrid Ad Hoc Routing Protocol, 85
HLR : Home Location Register, 79
HSLS : Hazy Sighted Link State routing protocol, 83
HSR : Host Specific Routing protocol, 85
HTTP : Hyper-Text Transfer Protocol, 36

I
IAD : Interleaved Add Delete, 64
IARP : Intrazone Routing Protocol, 85
IBA : InfiniBand TM Architecture., 15
IDA : Interleaved Delete Add, 64
IERP : Interzone Routing Protocol, 85
IETF : Internet Engineering Task Force, 38
ISDN : Integrated Services Digital Network, 59
ISUP : ISDN User Part, 59

L
LAM : Lightweight Adaptive Multicast protocol, 87
LAN : Local Area Network, 13, 35
LANMAR : Landmark Routing Protocol for Large Scale Networks, 85
LAR : Location-Aided Routing protocol, 86
LCA : Linked Cluster Architecture, 83
LCM : LSR-based Core Managment, 34, 47
LMR : Lightweight Mobile Routing protcol, 85
LSA : Link State Advertisement, 101
Link State Advertisment, 42
LSR : Link State Routing, 34
LUNAR : Lightweight Underlay Network Ad hoc Routing, 85

M
MANet : Mobile Ad Hoc Network, 83
MAODV : Multicast Ad hoc On-Demand Distance Vector routing, 89
MCEDAR : Multicasting Core-Extraction Distributed Ad Hoc Routing, 91
MOLSR : Multicast Optimized Link State Routing, 88
MOSPF : Multicast Extensions to OSPF, 42
MPLS : Multi Protocol Label Switching, 59
MPR : MultiPoints Relais, 83
MZR : Multicast Zone Routing protocol, 88

N
NACK : Negative ACK, 60
NOW : Network of Workstations, 13
NSMP : Neighbor Supporting Ad hoc Multicast Routing Protocol, 87

O
ODMRP : On-Demand Multicast Routing Protocol, 91
OLSR : Optimized Link State Routing, 83
OSPF : Open Shortest Path First, 47

P
PIM-DM : Protocol Independent Multicast-Dense Mode, 44
PIM-SM : Protocol Independent Multicast-Sparse Mode, 44
PNNI : Private Network-to-Network Interface, 47

Q
QoS : Quality of Service, 61

R
RDMAR : Relative-Distance Micro-discovery Ad hoc Routing protocol, 85
ROST : Receiver-Only Shared Tree, 37
RP : Rendez-vous Point, 44
RSVP : Resource ReSerVation Protocol, 59

S
SAN : System Area Network, 15, 23
SDS : Service Discovery Service, 110
SIMD : Simple Instruction Multiple Data, 12
SLA : Site Level Agregator, 101
SLP : Service Location Protocol, 110

SMP : shared memory multiprocessor, 12
SoC : System Area Network, 24
SRMP : Source Routing-based Multicast Protocol, 87
SRT : Source Rooted Tree, 37
SSR : Signal Stability Routing protocol, 85
SST : Symmetric Shared Tree, 37
STAR : Source Tree Adaptive routing protocol, 83

T
TBC : Total Body Conditionning, 83
TBRPF : Topology Broadcast based on Reverse-Path Forwarding, 84
TC : Topology Control, 84
TLA : Top Level Aggregator, 101
TORA : Temporally-Ordered Routing Algorithm routing protocol, 85

V
VLR : Visitor Location Register, 79

W
WAN : Wide Area Network, 35
WRP : Wireless Routing Protocol, 83
WWW : World Wide Web, 36

X
XMMAN : Extensions for Multicast in Mobile Ad-hoc Networks, 87

Y
YAM : Yet Another Multicast, 45

Z
ZHLS : Zone-Based Hierarchical Link State Routing, 86
ZRP : Zone Routing Protocol, 85

Annexe

Les publications incluses en annexe complètent les résultats présentés dans les différents chapitres de ce manuscrit.

[1] V. Bouchitté, J. Cohen, and E. Fleury. Optimal deadlock-free path-based multicast algorithms in meshes. In *5th International Colloquium on Structural Information and Communication Complexity (SIROCCO'98)*, Amalfi, Italy, June 1998.

[2] E. Fleury, Y. Huang, and P. K. McKinley. On the performance and feasibility of multicast core selection heuristics. *Networks*, 35(2) :145–56, March 2000.

[3] E. Fleury and H. Koubaa. A performance study of a service covering protocol in ad hoc networks. In *International Conference on Networks (ICON) 2001*, Bangkok, Thailand, October 2001. IEEE.

[4] G. Chelius, E. Fleury, and S. Ubéda. Merging ad hoc environment with wireless access : a survey. *ACM Wireless Networks (WINET)*, (submitted), 2002. (short version published in Med-hoc-Net 2002).