



Thèse

présentée pour obtenir le grade de docteur

de l'École Nationale Supérieure
des Télécommunications

Spécialité : Informatique et Réseaux

Jean-Louis Rougier

Routage dynamique hiérarchique unicast et multicast.

Soutenue le 21 Octobre 1999 devant le jury composé de

Catherine Rosenberg	Président
François Baccelli	Rapporteur
Annie Gravey	Rapporteur
Daniel Kofman	Examineur
Paul Kuhn	Examineur
Samir Tohme	Examineur

*A mes parents,
A Newton.*

Remerciements

Je tiens à remercier Daniel Kofman pour avoir accepté d'encadrer cette thèse. Outre ses conseils avisés, j'ai apprécié sa sympathie, sa rigueur, son esprit critique et sa grande pédagogie, dont, j'espère, j'ai su m'inspirer.

Ma profonde reconnaissance va également à Annie Gravey pour ses remarques judicieuses et sa gentillesse. J'ai pu apprécier ses qualités techniques et humaines pendant ces trois années (et toujours aujourd'hui). Je tiens à la remercier également pour avoir bien voulu relire le manuscrit et rédiger un rapport de thèse.

François Baccelli a également tenu une place importante dès le début de cette thèse, par les cours qu'il enseigne à Paris 6 et ses travaux de recherche sur la géométrie aléatoire dont nous avons eu le plaisir de trouver un nouveau champs d'application. J'ai eu beaucoup de plaisir de pouvoir travailler avec lui sur le routage multicast — collaboration très enrichissante et déterminante pour cette thèse. Je tiens également à le remercier pour la relecture attentive du présent document et la rédaction d'un rapport de thèse.

Je tiens à remercier très sincèrement Sergie Fdida, Paul Kuhn, Catherine Rosenberg et Samir Tohme pour m'avoir fait l'honneur de participer au jury de thèse et pour les questions très constructives qu'ils m'ont posées.

Je voudrais également remercier Samir Thome puis Ulrich Finger, pour m'avoir accepté dans le département Informatique et Réseaux. J'ai une pensée particulière pour toute l'équipe du département (mes collègues à présent...), les thésards et stagiaires avec qui j'ai partagé ces trois années avec beaucoup de bonheur et parmi lesquels je compte à présent certains de mes meilleurs amis.

La seconde partie de cette thèse doit beaucoup à Arturo Ragozini, qui m'a beaucoup aidé lors de son séjour sabbatique à l'ENST. Je le remercie vivement pour cette collaboration aussi conviviale que fructueuse.

Un grand merci également à mon grand-père qui a eu beaucoup de patience pour relire ce manuscrit et corriger de nombreuses coquilles et fautes d'orthographe.

J'ai une pensée émue pour mes Parents et toute ma famille, sans qui je n'aurais jamais pu arriver jusque là.

Résumé

Le développement de nouveaux applicatifs multimédia multi-parties évolue à une vitesse vertigineuse de sorte que les réseaux traditionnels sont dépassés. De nombreux développements sont actuellement en cours, particulièrement sur la gestion de la qualité de service et sur les communications de groupe. Les aspects portant sur le routage ont été quelque peu négligés par la communauté scientifique malgré leur importance dans une architecture adaptée aux nouveaux services. Le manque de routage performant, extensible à de grands réseaux et supportant la notion de qualité de service et/ou de communication de groupe (multicast) constitue un frein important pour la démocratisation de nouvelles applications (qui sont elles déjà prêtes).

Nos travaux portent sur l'évaluation de performances de protocoles de routage. Plus exactement, le présent rapport est structuré en deux parties, l'une sur le routage multicast et l'autre sur le routage unicast.

Dans la première partie de la thèse, nous nous intéressons au routage multicast (c'est à dire adapté aux communications de groupe). Le but de ces protocoles est de construire des "arbres" de diffusion interconnectant les membres d'un groupe donné. Nous étudions les performances des arbres hiérarchiques centrés, qui apparaissent comme un candidat très intéressant pour le routage multicast dans de très grands réseaux (routage "externe" IP). Nous donnons en particulier des règles de dimensionnement de ce type d'arbres permettant une optimisation des ressources.

Dans la seconde partie, nous nous intéressons au routage sensible à la qualité de service et à ses performances. Ce type de routage prend en compte l'état du réseau (congestion, délais, etc.) et la qualité de service demandée par l'utilisateur afin de choisir des chemins ayant une plus grande chance d'accepter le flux (ou la connexion). Ces protocoles nécessitent la dissémination d'informations de routage dans le réseau, ce qui représente un sur-coût protocolaire important. Il est donc nécessaire de trouver un compromis entre la qualité des décisions de routage et la quantité d'informations à échanger. Le sur-coût est contrôlé en limitant la fréquence des rafraichissements des informations de routage et en limitant la portée géographique de ces diffusions (on parle alors de routage hiérarchique). Nous étudions les performances de

protocoles de routage sensibles à la qualité de service, dans un réseau plat (une seule aire de routage) puis dans un réseau hiérarchique (plusieurs aires de routage).

L'analyse de performances de protocoles de routage est très difficile avec les techniques habituelles : Les résultats de files d'attente se limitent à des topologies rudimentaires (par exemple trois noeuds en triangle) et la complexité des simulations devient ingérable dans le contexte de réseaux hiérarchiques (vu le nombre de configurations à tester). Nous proposons une modélisation du routage basée sur la géométrie stochastique permettant de représenter la topologie d'un réseau avec un nombre extrêmement limité de paramètres. Les résultats sur les processus ponctuels et le calcul de Palm permettent ensuite d'obtenir des expressions explicites des grandeurs étudiées en fonction de ces paramètres.

Abstract

New multimedia multiparty software are being developed and are evolving very quickly, posing novel constraints to the supporting networks. Today, many such applications are stable and ready to use, but their widespread deployment is limited because today's networks are not well suited to support them — mainly for their lack of multiparty communication support and ability to deliver quality of services to specific real time flows. Intensive research and development efforts are being made in order to allow for QoS provisioning and group communications in packet networks (IP, ATM). However, the routing aspects may have been neglected even though it represents a very important building block of future networks architectures. The development of performant and scalable routing protocols, supporting group communications (multicasting) and/or quality of service is essential for the support of new services.

Our works deal with routing protocols and their performance evaluation. More precisely, this thesis is divided into two parts : The first one is dedicated to multicast (i.e. for group communications) routing, and the second one to unicast (i.e. point to point) routing.

In the first part, we focus on multicast routing protocols, i.e. routing protocols adapted for group communications. The goal of this type of protocol is to build a “tree” connecting the members of a particular group. We study the performances of hierarchical center based trees, which represent a very interesting candidate for scalable wide area multicast routing protocols. In particular, we give dimensioning rules for deploying these type of trees in order to minimize network resources.

In the second part, we analyse the performances of QoS (Quality of Service) sensitive unicast routing protocols. This type of routing protocol takes into account network states (such as congestion, delays) and user QoS requirements in order to choose path which have the best chance to accommodate the incoming flows (or connections). These protocols however require the dissemination of network state information which can represent an important overhead. It is very important to be able to find a good trade-off between routing information accuracy (leading to optimum path selection) and network overhead. The overhead can be controlled by limiting the routing information update frequency and by limiting the scope of routing information broadcasts (in a so-called hierarchical routing protocol). In this thesis,

we study the performance of QoS sensitive routing protocols in a flat network (i.e. in a single routing area) and in a hierarchical network (i.e. with several routing areas).

The study of performance of routing protocols is a very difficult task with legacy analysis methods : queuing results are limited to very simple topologies (such as three nodes forming a triangle) and the complexity of simulations explodes, particularly in hierarchical networks where the number of possible topologies to test grows exponentially with the number of hierarchical levels. In this thesis, we propose a stochastic geometric approach which allows us to model an arbitrary network topology with a very limited number of parameters. Well known results on point processes and Palm calculus leads to simple closed expressions for the studied quantities. Performance analysis and optimization then reduce to simple functional analysis.

Table des matières

1	Introduction générale	1
1.1	Contexte de l'étude	1
1.2	Travaux de recherche effectués	4
1.2.1	Thèmes abordés	4
1.2.2	Outils utilisés	5
1.2.3	Organisation du document	5

I Routage Multicast

2	Le Routage Multicast	9
2.1	Introduction	9
2.1.1	Origine du mot.	9
2.1.2	Avantages du multicast.	10
2.1.3	Le point de vue de la signalisation.	11
2.1.4	Contraintes technologiques.	12
2.2	Fonctions à mettre en œuvre	12
2.2.1	Capacités de Duplication	13
2.2.2	Adressage	13
2.2.3	Protocoles et Signalisation	15
2.3	Principaux types d'arbres	16
2.3.1	Arbre partagé/Arbre spécifique	17
2.3.2	Arbre de plus court chemin (SPT)	18
2.3.3	Arbre de Steiner	19
2.3.4	Arbre centré (CBT)	22
2.3.5	Bilan	23
2.4	Protocoles de Routage Multicast existants	24

2.4.1	DVMRP	25
2.4.2	PIM	27
2.4.3	Autres protocoles	29
2.4.4	Le “M-bone”	29
2.5	Travaux en cours	32
2.5.1	“Policy Routing”	32
2.5.2	Arbres Hiérarchiques	33
3	Modélisation des arbres hiérarchiques centrés	39
3.1	Présentation du modèle	39
3.1.1	Nœuds du réseau	39
3.1.2	Points de Rendez-vous	40
3.1.3	Structure hiérarchique	41
3.1.4	Connectivité	43
3.1.5	Calculs Préliminaires	44
3.1.6	Critères de performances	46
3.2	Coût de l’arbre	47
3.2.1	Analyse	47
3.2.2	Mode épars	49
3.2.3	Mode dense	50
3.2.4	Extensions	54
3.3	Délais	57
3.3.1	Diffusion à partir du RP central.	57
3.3.2	Délais à partir d’un point quelconque	59
3.3.3	Remarques	61
3.4	Coût de construction	61
3.4.1	Coût global	62
3.4.2	Ajout d’une feuille	63
3.5	Arbres de concentration de Delaunay	65
3.5.1	Présentation	65
3.5.2	Quelques propriétés	66
3.6	Conclusion	67

4	Routage dynamique	73
4.1	Routage Dynamique	73
4.1.1	Introduction	74
4.1.2	Routage “Distance Vector”	74
4.1.3	Routage “Link-State”	75
4.1.4	Sélection des chemins.	77
4.2	Routage sensible à la qualité de service	77
4.2.1	Motivations	78
4.2.2	Informations de routage	78
4.2.3	Choix de l’algorithmes de routage	80
4.2.4	Routage sur des informations inexactes.	82
4.3	Routage hiérarchique	83
4.3.1	Fonctionnement	84
4.3.2	Agrégation	87
4.3.3	Algorithmes de routage hiérarchiques.	89
4.4	Organisation de cette partie.	90
5	Analyse d’un réseau plat	93
5.1	Présentation du modèle	93
5.1.1	Structure du réseau	93
5.1.2	Informations de routage.	94
5.1.3	Algorithme de Routage.	95
5.1.4	Crankbacks	95
5.2	Analyse	96
5.2.1	Préliminaires	98
5.2.2	Perte d’appel.	99
5.2.3	Rejet de l’appel au nœud source.	99
5.2.4	Crankback.	100
5.2.5	Surcoût	100
5.2.6	Moyennes.	101
5.2.7	Routage à l’aveugle	102
5.3	Résultats	103
5.3.1	Influence de p	103
5.3.2	Influence de l’acuité des informations de routage.	105
5.3.3	Influence de la fonction GCAC	108
5.4	Conclusion et Remarques	110

6	Agrégation	115
6.1	Introduction	115
6.2	Modélisation	115
6.2.1	Structure Hiérarchique	116
6.2.2	Notations	117
6.3	Tables de routage	118
6.3.1	Analyse du problème	119
6.3.2	Résultats	120
6.3.3	Extensions	122
6.3.4	Discussion	124
6.4	Complexité des calculs de Routage	125
6.4.1	Introduction	125
6.4.2	Modélisation	126
6.4.3	Analyse	128
6.4.4	Résultats	131
6.4.5	Discussion	133
7	Analyse d'un réseau hiérarchique	137
7.1	Modèle	137
7.1.1	Structure hiérarchique	137
7.1.2	Algorithme de Routage.	139
7.1.3	Procédure d'agrégation	139
7.1.4	Crankback	141
7.2	Analyse	141
7.2.1	Préliminaires	141
7.2.2	Perte d'appel.	142
7.2.3	Rejet de l'appel au nœud source.	143
7.2.4	Crankback.	143
7.2.5	Surcoût protocolaire	145
7.2.6	Routage aveugle	147
7.2.7	Déconditionnement	149
7.3	Résultats	151
7.3.1	Influence de l'agressivité.	151
7.3.2	Influence de p_0	156
7.3.3	Influence de q_0	159
7.3.4	Structure hiérarchique	160

7.4	Conclusion et Remarques	165
8	Conclusion générale	173
8.1	Routage Multicast	173
8.2	Routage QoS	173
8.3	Géométrie stochastique	175

III Annexes

A	Processus Ponctuels dans \mathbb{R}^2	181
A.1	Notations	181
A.2	Définition des Processus Ponctuels	182
A.2.1	Mesures ponctuelles	182
A.2.2	Processus ponctuels.	183
A.2.3	Processus stationnaires	184
A.3	Processus de Poisson	185
A.3.1	Définition	185
A.3.2	Simulation	187
A.3.3	Transformée de Laplace	188
A.3.4	Perturbation	189
A.4	Calcul de Palm	191
A.4.1	Formule de Palm	191
A.4.2	Formules élémentaires du Calcul de Palm	194
A.5	Processus Conjointement Stationnaires	196
A.5.1	Préliminaires	197
A.5.2	Calcul de Palm	197
A.5.3	Relation entre ces probabilités.	199
A.6	Bréviaire sur le calcul de Palm	201
B	Pavages du plan	203
B.1	Introduction	203
B.2	Résultats généraux	203
B.2.1	Notations	204
B.2.2	Formules	205
B.3	Pavages de Poisson-Voronoi	208
B.3.1	Définition	208

B.3.2	Propriétés génériques	209
B.3.3	Tailles des Cellules.	210
B.4	Pavages de Delaunay	214
B.4.1	Caractéristiques	214
B.4.2	Graphe de Delaunay	214
C	Quelques distributions	219
C.1	Statistique d'Ordre	219
C.1.1	Introduction	219
C.1.2	Propriétés	220
C.2	Quelques Lois élémentaires	221
C.2.1	Expression Multinomiale	221
C.2.2	Loi Hypergéométrique	222
C.3	Blocage dans un domaine de routage	223
C.3.1	Sélection d'un domaine	223
C.3.2	Traversée d'un domaine sélectionné	223

Table des figures

2.1	Intérêt du Multicast	10
2.2	Point de vue signalétique.	11
2.3	Plan d'adressage IPv4.	14
2.4	Plan d'adressage ATM.	15
2.5	Arbres partagés ou spécifiques à une source.	17
2.6	Arbre de type SPT.	19
2.7	Arbre de type Steiner.	20
2.8	Un cas pathologique.	21
2.9	Arbre de type CBT.	22
2.10	Diffusion "Reverse Path Broadcasting".	26
2.11	Diffusion partielle "Reverse Path Multicasting".	26
2.12	Signalisation du protocole PIM-SM.	28
2.13	Exemple d'utilisation de sdr	31
2.14	Couverture géographique d'une session multicast.	32
2.15	Arbres hiérarchiques centrés à deux niveaux hiérarchiques.	34
3.1	Une réalisation du modèle d'arbre hiérarchique utilisé.	42
3.2	Validation de l'approximation proposée.	46
3.3	Influence de l'ajout d'un niveau hiérarchique supplémentaire.	50
3.4	Coût de l'arbre hiérarchique en fonction de H	53
3.5	Exemple de structure hiérarchique minimisant le coût de l'arbre.	54
3.6	Évolution des délais de diffusion en fonction de H	58
3.7	Évolution du délais subi en fonction de H et de la distance ρ	61
3.8	Coût de construction causé par l'ajout d'une nouvelle feuille.	64
3.9	Arbre de concentration de Delaunay.	66
3.10	Évolution de la participation en fonction de p	67
3.11	Evolution du coût en fonction de p ($\lambda = 100$).	68

4.1	Agrégation non hiérarchique	85
4.2	Exemple de division hiérarchique.	85
4.3	Structure logique construite par PNNI.	86
4.4	Vision topologique à l'intérieur des domaines PG A.1 et PG B.	87
4.5	Méthode d'agrégation dans PNNI.	88
5.1	Influence de p sur les performances du réseau	103
5.2	Influence de p sur le nombre moyen de crankbacks.	105
5.3	Influence de p sur le surcoût signalétique.	106
5.4	Influence de q sur les performances du réseau.	106
5.5	Surcoût associé aux crankbacks en fonction de q	107
5.6	Surcoût et ses différentes composantes.	108
5.7	Perte d'appels selon le choix des routes.	109
5.8	Nombre de crankbacks selon le choix des routes.	110
6.1	Réalisation d'un réseau à 3 niveaux hiérarchiques.	117
6.2	Taille de la table de routage	122
6.3	Modèle hiérarchique de connectivité	127
6.4	Structure minimisant la complexité	132
6.5	Complexité en fonction de H	133
7.1	Réseau à deux niveaux hiérarchiques	138
7.2	Influence de β sur les performances du réseau.	151
7.3	Influence de β sur la perte d'appels ($K_c^1 = 1$).	152
7.4	Influence de β sur la perte d'appels ($K_c^1 = 2$).	153
7.5	Influence de β sur le nombre moyen de crankbacks.	154
7.6	Influence de β sur le surcoût signalétique.	155
7.7	Influence de p_0 sur la perte d'appels.	157
7.8	Influence de l'agressivité sur la perte d'appels en fonction de p_0	158
7.9	Influence de l'agressivité sur le surcoût signalétique en fonction de p_0	158
7.10	Influence de l'agressivité sur la perte d'appels en fonction de q_0	159
7.11	Influence de l'agressivité sur le surcoût signalétique en fonction de q_0	160
7.12	Influence de la structure hiérarchique sur les pertes d'appels.	162
7.13	Influence de la structure hiérarchique sur le nombre moyen de Crankbacks.	163
7.14	Influence de la structure hiérarchique sur le surcoût protocolaire.	164
A.1	Interprétation d'une mesure ponctuelle.	182
A.2	Réalisation d'un processus de Poisson homogène	186

A.3	Réalisation d'un processus de Poisson inhomogène	187
B.1	Cas où $0 \in \alpha_1(\Theta) \cap C_x(\Theta)^*$ pour $x \in \alpha_2(\Theta)$	206
B.2	Cas où $0 \in \alpha_0(\Theta) \cap C_x(\Theta)^*$ pour $x \in \alpha_2(\Theta)$	207
B.3	Un résultat élémentaire de géométrie...	208
B.4	Réalisation d'un pavage du plan de Voronoi	209
B.5	Réalisation d'un pavage du plan de Voronoi et du pavage de Delaunay associé .	210
B.6	Exemple de chemin markovien sur le graphe de Delaunay.	216

Chapitre 1

Introduction générale

1.1 Contexte de l'étude

Les travaux rapportés dans le présent manuscrit ont été effectués au sein du département Informatique et Réseaux de l'ENST et ont été financés par le CNET (Consultations Thématiques Informelles). La consultation portait sur l'étude des nouvelles architectures de réseaux permettant de supporter les nouvelles applications multimédia et multi-parties.

Nous assistons en effet depuis quelques années à l'apparition de nouvelles applications posant de nouvelles contraintes sur les réseaux sous-jacents.

- Applications Multimédia. Tout d'abord, de nombreuses applications requérant le transfert de la voix, de sons et même de vidéo sont apparues. La visio-conférence ou la téléphonie sur poste de travail ne sont que les exemples les plus courants. Les flux générés par ce type d'application sont difficiles à transmettre en raison des débits générés (surtout pour la vidéo) et de leurs fortes contraintes temporelles. Il est apparu nécessaire de pouvoir offrir de meilleures qualités de services (bande passante, délais et gigue¹ garantis).
- Applications Multicast. Les applications font de plus en plus intervenir plusieurs personnes à la fois. Qu'il s'agisse de diffusions audio ou vidéo, de visio-conférence, de travail collaboratif (traitement de texte partagé) ou même de jeux distribués liés à des environnements virtuels, le paradigme un émetteur-un récepteur n'est plus valable. Les réseaux actuels sont basés sur des communications point-à-point. Le besoin est donc apparu de développer des infrastructures "multicast", c'est à dire adaptées aux communications de

¹gigue : Variation du délais.

groupe.

Le développement de nouveaux applicatifs évolue à une vitesse vertigineuse de sorte que les réseaux traditionnels se sont vus dépassés. Historiquement, les capacités des réseaux ont souvent précédé le marché des services les utilisant. Nous nous trouvons à présent dans la situation inverse : les applications multimédia-multiparties atteignent leur maturité industrielle mais des réseaux tels que l'Internet ne les supportent pas encore (ou pas encore efficacement). De nombreux développements sont nécessaires, essentiellement sur la gestion de la qualité de service et sur les communications de groupe.

Le réseau Internet.

L'Internet offre une base de développement logiciel unique, grâce à sa grande ouverture. La popularisation de l'Internet a certainement contribué à la formidable accélération dans l'apparition de nouveaux services. Pourtant, le réseau Internet actuel n'est plus adapté aux besoins des applications qu'il a lui-même générées et doit donc évoluer — on parle d'Internet 2.

- Le réseau mondial actuel ne permet pas d'assurer une quelconque qualité de service aux applications : il est basé sur un paradigme “best-effort” (c.à.d. “je fais de mon mieux”). De nombreux travaux sont en cours pour permettre une qualité de service garantie ou tout au moins une différenciation de services — certains services (temps réels par exemple) pouvant être traités différemment des simples services de transfert de données.
- Le paradigme non orienté connexion (ou “datagramme”) offre une grande souplesse pour le transfert de données mais est mal adapté aux nouveaux services multimédia. Les réseaux basés sur le concept de “flux” (et non plus de “paquet”) apparaissent avec des nouvelles architectures comme MPLS² (ou les architectures IP sur ATM même si elles ne font pas l'unanimité).
- Le paradigme “point-à-point” (ou “unicast”) du réseau Internet n'est plus non plus adapté aux applications multi-parties. La communauté Internet avait très bien anticipé ce nouveau besoin puisque un réseau expérimental “multicast” (permettant de gérer plus efficacement les communications de groupe) existe depuis 1992. Néanmoins, ce réseau n'a pas su évoluer assez rapidement : Les applications sont déjà prêtes, mais il reste de très nombreux travaux à réaliser pour pouvoir supporter les communications de groupe à grande échelle.

²MPLS : MultiProtocol Label Switching.

Les réseaux ATM.

Les besoins en terme de qualité de service des nouvelles applications avaient été très bien anticipées par les opérateurs avec le développement des réseaux ATM — peut être trop, certains reprochent à cette technologie son manque de flexibilité pour le support des données (c.à.d les applications traditionnelles). Les réseaux ATM ont été conçus pour supporter l'intégration de services : Gestion de Trafic, Réservation de ressources, Routage sensible à la qualité de services, etc. sont disponibles depuis de nombreuses années. Dans la pratique cependant, il faut tout d'abord remarquer que les industriels ont eu du mal à suivre le rythme effréné de la normalisation. Des commutateurs aux fonctionnalités vraiment satisfaisantes n'existent que depuis deux à trois années.

Le développement des réseaux ATM a été accompagné d'un développement de nouvelles applications pouvant tirer partie de leurs fonctionnalités quasiment nulles. Ceci est du pour une part au fait que les applications existantes sont basées sur un paradigme "datagramme" alors que les réseaux ATM sont orientés-connexions. Notons également que les APIs³ normalisés existent depuis moins d'un an... Si la technologie est à présent opérationnelle, le dimensionnement de ces réseaux, nécessaire pour un déploiement à grande échelle, reste encore mal maîtrisé, notamment sur les aspects liés au routage.

De nombreux travaux sont également nécessaires pour que cette technologie puisse supporter les nouvelles applications :

- Les applications sont basées sur TCP/IP et ses évolutions. De nombreux travaux ont été menés (et sont encore en cours) afin de pouvoir supporter efficacement les réseaux IP.
- Le réseau ATM actuel est mal adapté aux communications de groupe. Des connexions point-à-multipoints sont disponibles mais manquent de flexibilité (gestion basée à la source). De nombreux travaux sont en cours pour faciliter la gestion de ce type de connexions⁴. Par contre, aucune extension permettant le support de véritables capacités multicast n'est pour l'instant prévues, bien que des recherches en amont soient menées.

³API : Application Part Interface. Interface offerte aux applications, permet d'ouvrir une connexion et de demander une qualité de service particulière par exemple.

⁴Les capacités LIJ (Leaf Initiated Join) sont bien disponibles dans l'UNI 4.0 de l'ATM Forum mais ne sont pas encore supportées par le routage.

1.2 Travaux de recherche effectués

1.2.1 Thèmes abordés

Les travaux de recherche présentés dans ce manuscrit portent sur le routage, c'est à dire la fonction permettant de déterminer le chemin à suivre pour acheminer une information vers son destinataire.

Routage Multicast.

Nous avons vu dans la section précédente que de nombreux travaux sont en cours pour le support de communication de groupe, c.à.d. le multicast. Comme nous l'expliquons dans le chapitre 2, la principale difficulté réside dans le développement de protocoles de routage multicast adaptés aux réseaux de grandes dimensions. Les protocoles multicast existants sont en effet mal adaptés à un réseau aux dimensions aussi grandes que le réseau Internet. L'absence de protocole supportant de grands réseaux retarde la popularisation des applications multiparties. Les solutions basées sur des "arbres hiérarchiques" représentent une solution très prometteuse. Les études ont essentiellement portées sur les aspects algorithmiques. Le manque de résultats sur leurs performances rend le dimensionnement de ces objets difficile. Nous nous intéressons à ce problème dans la première partie de ce rapport.

Routage QoS.

De nombreux travaux ont été menés pour le support de la qualité de service dans les réseaux. L'essentiel des travaux a porté sur les fonctionnalités de bas niveaux (ordonnancement, gestion de trafic, etc.). Le développement du routage sensible à la qualité de service a fait l'objet de moins d'attention, bien qu'il s'agisse d'une pièce indispensable d'une architecture supportant la qualité de service. Dans la seconde partie de ce rapport, nous nous intéressons à ce type de protocoles de routage.

Il a été démontré que l'utilisation d'un routage sensible à la qualité de service peut engendrer des gains intéressants en terme d'utilisation de ressources, sans que le surcoût associé ne soit trop important. Néanmoins, le manque de résultats précis sur les performances de tels protocoles rend leur mise en oeuvre difficile. Dans les études présentées, nous essayons de comprendre l'impact des différents paramètres mis en jeu sur les performances globales du réseau. Nous recherchons également des règles de dimensionnement simples permettant d'utiliser à bon escient cette technologie.

1.2.2 Outils utilisés

L'étude de performances de protocoles de routage s'avère extrêmement délicate avec les méthodes d'analyse traditionnelles.

- Les modèles analytiques classiques (basés sur des files d'attente ou des analyses markoviennes etc.) sont mal adaptés vu la complexité des systèmes considérés. Les quelques résultats disponibles sont basés sur des topologies extrêmement rudimentaires (trois noeuds connectés en triangle par exemple...) [102].
- Les études par simulation sont très délicates. Afin de s'assurer que le protocole de routage étudié fonctionne sur tout type de réseau, les simulations font en général appel à des générateurs de graphes aléatoires. Il est nécessaire de réaliser un grand nombre de simulations (sur des topologies générées aléatoirement) pour obtenir des moyennes significatives. Lors de nos études de performances dans un contexte hiérarchique, le nombre de combinaisons différentes à tester (différentes structures hiérarchiques possibles) rend cette technique irréalisable.

Les études menées dans ce document sont basées sur ces modèles analytiques utilisant la géométrie stochastique. Cette approche nouvelle a l'avantage de permettre de travailler sur des modèles macroscopiques, décrivant la topologie des réseaux avec un nombre de paramètres extrêmement limité. Les résultats assez puissants sur les processus ponctuels (Calcul de Palm) permettent d'obtenir des formules explicites (souvent simples) comme une fonction de ces paramètres.

Nous avons pris connaissance des techniques de modélisation de réseaux par la géométrie stochastique grâce à des travaux récents du CNET et de l'INRIA sur le dimensionnement de la boucle locale et de réseaux mobiles. Des travaux préliminaires nous ont ensuite encouragés à appliquer cette technique à l'étude de protocoles de routage. Nous avons conscience que les concepts mathématiques manipulés ne sont pas encore très connus de la communauté Réseaux. C'est pour cette raison qu'il nous est apparu nécessaire d'inclure en troisième partie (annexes) une brève présentation des concepts mathématiques et des théorèmes utilisés.

1.2.3 Organisation du document

Ce document est organisé en trois parties. La première partie traite du routage multicast. La seconde partie est consacrée au routage sensible à la qualité de service. Finalement, la troisième partie regroupe les annexes résumant les résultats et les concepts nécessaires à la compréhension des modèles utilisés.

Première partie



Routage Multicast

Chapitre 2

Le Routage Multicast

Dans cette partie, nous nous concentrons sur le routage multicast et plus particulièrement sur les algorithmes et protocoles de routage multicast hiérarchiques. Nous commençons par une brève introduction sur le routage multicast dans l’Internet — sauf mention explicite, nous ne parlerons que des réseaux IP, les seuls possédant à ce jour de véritables capacités multicast.

2.1 Introduction

Le multicast est une technique particulière permettant de réduire considérablement les coûts de transmission pour des communications de groupes (c.à.d. un émetteur vers plusieurs récepteurs (“one-to-many”), ou plusieurs émetteurs vers plusieurs récepteurs (“many-to-many”)). Nous commençons cette introduction sur cette nouvelle technique en présentant les avantages du multicast sur le point-à-point ou la diffusion (“broadcast”) et en analysant les contraintes technologiques que cette technique implique.

2.1.1 Origine du mot.

L’introduction de nouvelles techniques de diffusion de groupe a engendré la création de nouveaux mots de vocabulaire. Les communications traditionnelles, faisant intervenir une source et un destinataire, sont parfois qualifiées de communication “unicast”. Ce néologisme a été formé à partir du mot “broadcast” en introduisant le préfixe “uni” pour bien spécifier que cette communication ne fait intervenir qu’une source et un destinataire. Le mot “multicast” est également un néologisme formé à partir du mot anglo-saxon “broadcast”. Le préfixe “multi” spécifie dans ce cas que la communication implique plusieurs récepteurs (et peut-être également plusieurs émetteurs). Ce mot n’est pourtant pas à confondre avec son ancêtre “broadcast” qui

désigne une diffusion totale. En effet, une diffusion multicast ne fait pas intervenir toutes les machines connectés à un réseau mais seulement un sous-ensemble bien défini de ces machines.

2.1.2 Avantages du multicast.

Le multicast a été introduit avec l'avènement des applications multi-parties (conférences sur Internet, etc.) et les applications de travail collaboratif (traitement de texte partagé, simulations réparties, etc.). Autrement dit, le mot multicast est indissociablement lié à la notion de communication de groupe, cette technique ayant été introduite afin de réduire les coûts de communication pour ce genre d'applications.

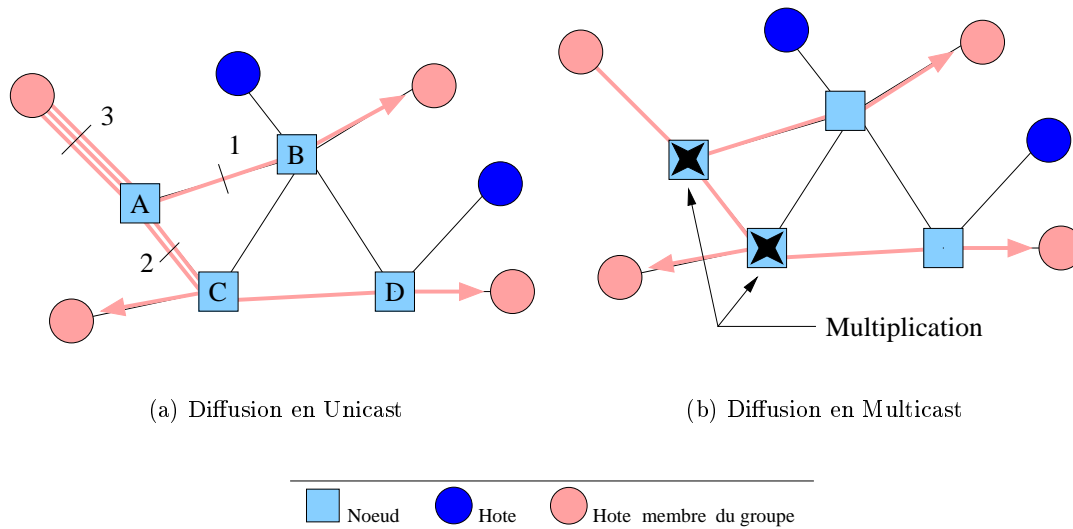


FIG. 2.1 – Intérêt du Multicast

Avec une technique traditionnelle “unicast”, une communication faisant intervenir plusieurs récepteurs nécessite l’émission successive de la même information, et ce autant de fois qu’il y a de destinataires. Il est facile de comprendre le gâchis en terme de bande passante que ces répétitions d’informations strictement identiques représentent. La figure 2.1(a) représente une diffusion partielle vers 3 stations sans capacités multicast, de sorte que la même information est envoyée 3 fois sur le premier lien et 2 fois sur le lien A-C.

Lorsque les nœuds du réseau possèdent des capacités de multiplication, il est possible pour la source d’envoyer un seul message, qui sera copié et envoyé sur les différentes “branches de l’arbre” multicast lorsque ceci est nécessaire. La figure 2.1(b) représente une diffusion partielle lorsque les nœuds du réseau possèdent ces capacités multicast.

2.1.3 Le point de vue de la signalisation.

Du point de vue de la signalisation, le multicast introduit également de nouvelles facilités et de nouveaux concepts. Dans le cas unicast, les différents récepteurs doivent se faire connaître à la source, un à un, afin que celle-ci puisse envoyer l'information avec l'adresse de chaque destinataire. Ceci peut aussi être le cas pour des connexions point-à-multipoints avec Frame-Relay ou ATM, où c'est encore à la source de gérer le groupe : Les nouveaux destinataire se font connaître à la source qui doit alors rajouter une nouvelle branche à son arbre de diffusion. De même à chaque départ d'une feuille, la source doit enlever la branche correspondante. Cette signalisation "basée à la source" est donc adaptée pour des groupes de faible taille, ou lorsque la source désire restreindre l'entrée dans le groupe. Pour des diffusions de très grande envergure¹ cette technique n'est pas réalisable, car la gestion du groupe à proprement parlé pourrait saturer la source.

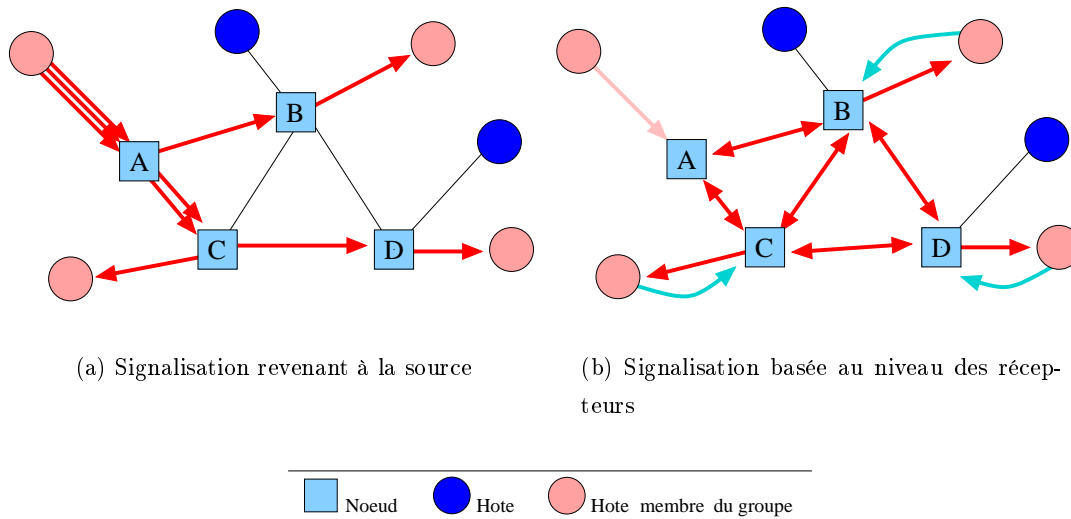


FIG. 2.2 – Point de vue signalétique.

Avec les nouvelles techniques multicast a donc été introduite une signalisation basée au niveau des récepteurs². Dans ce cas, la source n'a absolument pas à se soucier de la taille du

¹Une séquence d'information "CNN Live" par exemple pourrait être suivie par des centaines de milliers de personnes sur l'Internet dans un avenir proche. Pour donner un ordre d'idées, les diffusion vidéo du décollage des fusées de la NASA sont d'ors et déjà suivis par plus d'un millier de personnes en moyenne, même si le réseau expérimental multicast (le "M-bone") est encore très limité — seuls les grandes universités et quelques laboratoires de recherche y sont connectés. Les départs et les arrivées dans ce groupe sont extrêmement fréquents.

²C'est le cas dans les réseaux IP. Bien que l'ATM ne dispose pas de capacités multicast à proprement parlé, une signalisation "basée récepteur" est en cours de développement [64] à l'ATM Forum pour faciliter la

groupe, toute la gestion de l'arbre étant effectuée par le réseau³. Les hôtes désirant participer à la communication utilisent alors des messages spécifiques pour joindre le groupe, le réseau possédant des mécanismes pour permettre de rajouter une nouvelle branche menant vers cette destination. La figure 2.2 illustre la différence entre les approches où l'initiative appartient à la source ou aux récepteurs.

2.1.4 Contraintes technologiques.

Nous avons rapidement présenté les principales motivations qui ont permis de créer le multicast. Cette technique requiert des capacités supplémentaires à mettre en œuvre dans le réseau.

- **Capacités de Multiplication.** Nous avons mis en évidence précédemment que le multicast ne peut fonctionner que si les nœuds du réseau sont pourvus de capacité de duplication des informations reçues.
- **Adressage.** Nous avons jusqu'à présent passé sous silence ce problème. Nous avons vu que si les capacités de duplication sont présentes dans le réseau, il suffit à la source d'envoyer une seule fois cette information pour joindre tous les destinataires. Mais vers quelle adresse est envoyée cette information ? Dans un réseau en mode datagramme, nous pourrions penser à inclure dans les paquets la liste des adresses de tous les destinataires. Ceci ne serait bien sûr pas réalisable pour de grands groupe. Il est donc nécessaire de définir une adresse spécifique identifiant le groupe.
- **Protocoles et Signalisation.** Il est nécessaire de mettre en place des procédures spécifiques afin de construire les arbres de diffusion. On peut distinguer deux types d'interfaces mettant en jeu des procédures différentes :
 - *Hôte-Réseau.* Des messages doivent être définis pour permettre aux hôtes de faire part au réseau de leur désir de joindre ou de quitter un groupe donné.
 - *Réseau-Réseau.* Des procédures doivent ensuite être mises en place pour construire ou modifier l'arbre multicast selon les requêtes des hôtes. On parle de "routage multicast".

2.2 Fonctions à mettre en œuvre

gestion des connexions point-à-multipoints. Il s'agit des capacités LIJ ("Leaf Initiated Join") introduites dans l'UNI 4.0 [65]. Ces capacités ne sont pas encore disponibles car elles ne sont pas encore supportées par PNNI v.1 [63].

³Avec certains protocoles IP, la source ignore même s'il existe des récepteurs dans le groupe multicast qu'il a créé.

Dans cette section, nous revenons plus précisément sur les contraintes technologiques évoquées dans le paragraphe précédent.

2.2.1 Capacités de Duplication

Nous avons vu l'intérêt (en terme de bande passante économisée) d'introduire des capacités de duplication dans le réseau. Cette multiplication peut avoir lieu à deux niveaux :

- **Au niveau des réseaux à accès multiple**. La plupart des segments LAN (ou BMA⁴) ont des capacités diffusantes qui peuvent être utilisées pour éviter de retransmettre plusieurs fois l'information. Le filtrage est alors assuré au niveau des routeurs ou des hôtes. Les segments Ethernet possèdent de véritables capacités multicast (le filtrage⁵ étant alors assuré au niveau 2 au lieu du niveau 3). Certains réseaux WAN (ou NBMA⁶) comme Frame-relay ou ATM possèdent également des liaisons point-à-multipoints⁷ permettant de joindre directement plusieurs destinataires.
- **Au niveau des routeurs**. La plupart des matrices de commutation actuelles permettent de dupliquer un élément d'information vers plusieurs ports de sortie. Cette fonctionnalité est particulièrement facile à mettre en œuvre pour des architectures de commutation basées sur un bus diffusant. La nature diffusante du "fond de panier" se prête en effet très bien à cette tâche — il suffit d'autoriser à plusieurs ports de sortie d'écouter simultanément le médium pendant l'intervalle de temps ("slot") désiré.

En conclusion, les architectures actuelles permettent en général la duplication d'informations. L'introduction de capacités multicast dans le réseau est le plus souvent un problème logiciel, et non matériel.

2.2.2 Adressage

Nous avons vu que l'introduction de capacités multicast nécessite la définition d'un adressage de groupe. A chaque communication multicast est associée une adresse spécifique. Les machines désirant participer à la session multicast doivent joindre l'adresse de cette session. Il faut remarquer que l'introduction de ces adresses impose des comportements qui ne pouvaient exister dans le cas de communications unicast. En particulier,

- Plusieurs stations peuvent avoir la même adresse multicast : en effet, tous les membres du groupe possèdent la même adresse multicast, quelque soit leur localisation sur le

⁴BMA : Broadcast Multiple Access.

⁵Des adresses MAC multicast sont définies.

⁶NBMA : Non Broadcast Multiple Access.

⁷Les liaisons multipoints-à-point ("VC-Merging") sont également en cours de normalisation à l'ATM Forum (PNNI v.2 [64]). Les liaisons multipoints-à-multipoints ne sont par contre pas encore disponibles.

- réseaux (c.à.d. indépendamment de leur adresse unicast).
- Un hôte peut posséder autant d’adresses multicast qu’il désire, suivant le nombre de sessions qu’il veut joindre.

<i>Classe</i>	<i>Représentation binaire</i>	<i>Bornes</i>
A	0 ...	0.0.0.0 à 127.255.255.255
B	1 0 ...	128.0.0.0 à 191.255.255.255
C	1 1 0 ...	192.0.0.0 à 223.255.255.255
D	1 1 1 0 ...	224.0.0.0 à 233.255.255.255
E	1 1 1 1

FIG. 2.3 – Plan d’adressage IPv4.

Adresses IP de groupe.

Pour envoyer des données vers un groupe multicast, il suffit d’envoyer à son routeur par défaut des paquets dont l’adresse destinataire est l’adresse du groupe en question — un émetteur n’a même pas besoin de s’enregistrer comme étant un membre du groupe dans les réseaux multicasts de l’Internet. En IPv4, les adresses de groupe (ou adresses multicast) sont des adresses réservées (ou interdites) du plan d’adressage. Plus précisément, il s’agit d’adresses de classe D (cf. figure 2.3) — classe d’habitude interdite dans l’Internet.

Adresses ATM Multicast.

Cette propriété des réseaux ATM est souvent méconnue, et pourtant les adresses de groupe existent également dans ces réseaux ! L’UNI⁸ version 4.0 de l’ATM-Forum [65] définit en effet ce type d’adresse, même si les réseaux ATM ne supportent pas le multicast pour l’instant. Ces adresses sont utilisées pour le “anycasting” (voir plus loin) présenté dans l’UNI 4.0 et P-NNI version 1 [63, pages 32-34]. Les adresses de groupe ATM sont facilement reconnaissables à leur champ AFI⁹ bien spécifique (un champ AFI de groupe est associé à chaque AFI individuel), comme présenté dans [65, Annexe 5]. La figure 2.4 représente les différents formats d’adresses ATM définies¹⁰.

Remarque 2.1 “Anycast” est encore un nouveau néologisme inspiré de “unicast”, “multicast” et “broadcast”. Comme le préfixe anglo-saxon “any” l’indique, il ne s’agit pas de diffusion mais

⁸UNI : User-Network Interface.

⁹AFI : Authority and Format Identifier.

¹⁰Sur la figure 2.4, DSP signifie “Domain Specific Part”, DCC désigne le “Data Country Code” et ICD le “International Code Designator” — voir [65] pour plus de détails.

d'une transmission vers un membre du groupe (et un seul). Il ne s'agit bien entendu pas de n'importe quel membre du groupe, le routage s'assure en général que la connexion soit ouverte vers le membre du groupe le plus proche de l'appelant. Cette propriété est très utilisée pour contacter des serveurs¹¹.

Type d'adresse	Représentation binaire		
DCC	0x39	DCC	DSP (17 octets)
ICD	0x39	ICD	DSP (17 octets)
E.164	0x39	E.164	DSP (11 octets)
Group (DCC)	0xBD	DCC	DSP (17 octets)
Group (ICD)	0xC5	ICD	DSP (17 octets)
Group (E.164)	0xC7	E.164	DSP (11 octets)

FIG. 2.4 – Plan d'adressage ATM.

2.2.3 Protocoles et Signalisation

Traditionnellement, les hôtes ne sont jamais impliqués dans le routage sauf pour des besoins particuliers¹². Cette séparation des tâches se retrouve également dans le cas du multicast.

Interface Hôte-Réseau.

Dans le réseau Internet, la signalisation multicast entre un hôte et son routeur par défaut est assurée par le protocole IGMP (Internet Group Membership Protocol) [75]. Il s'agit d'un protocole extrêmement simple de "polling". Le routeur en charge d'un sous-réseau IP envoie régulièrement des requêtes (avec un TTL¹³ de 1) pour connaître l'appartenance des hôtes du sous-réseaux aux différents groupes multicast. Les hôtes du sous réseau renvoie (après un temps aléatoire, pour éviter les collisions) au routeur un message dans lequel ils spécifient à quels groupes (c.à.d. à quelles adresses de classe D) ils appartiennent. Des messages supplémentaires viennent se greffer à ces requêtes/réponses dans IGMP version 2 [75] afin de prendre en compte plus rapidement l'arrivée ou le départ d'un membre du groupe.

Interface Réseau-Réseau.

Les routeurs IP disposent grâce au protocole IGMP des informations sur les groupes multicast qu'ils doivent laisser passer sur leurs interfaces — car des hôtes situés derrière ces

¹¹ Serveur LEC ou LECS dans LAN Emulation par exemple.

¹² Redondance en cas de panne d'un routeur, ou routage en mode passif pour des serveurs possédant de nombreuses interfaces.

¹³ TTL : Time To Live.

interfaces désirent participer à ces groupes. Il ne s'agit néanmoins que d'informations locales. Il est nécessaire de mettre en place des procédures entre routeurs afin de construire l'arbre de diffusion multicast. C'est le rôle des protocoles de **routage multicast**. Nous consacrons le reste de ce chapitre à ce sujet.

2.3 Principaux types d'arbres

Avant d'étudier plus en détail les différents protocoles de routage multicast, il nous a semblé nécessaire de présenter les différents types d'arbre de diffusion qu'il est possible de construire. En effet, une des difficultés majeures dans le développement de protocoles de routage multicast est le choix du type d'arbre. Il n'y a en effet pas de choix "canonique" : Certains arbres ont d'excellentes performances du point de vue des délais de bout-en-bout mais utilisent en général beaucoup de ressources. D'autres types d'arbres permettent d'utiliser le minimum de ressources possibles mais peuvent conduire à des délais de bout-en-bout assez médiocres, etc.

Dans cette section, nous présentons les principaux types d'arbres en insistant sur leurs avantages et inconvénients respectifs. Afin de comparer ces arbres, il nous faut d'abord définir les critères de performances les plus significatifs¹⁴ :

- **Complexité.** La construction de l'arbre permettant de joindre les différents membres du groupe multicast nécessite des algorithmes plus au moins complexes selon le type d'arbre sélectionné. La complexité peut être mesurée de deux façon selon que l'on considère un groupe statique ou dynamique.
 - Dans le cas statique, il s'agit de joindre une liste de nœuds donnée.
 - Dans le cas dynamique, nous étudions la capacité de l'arbre à s'adapter à l'arrivée ou au départ d'une feuille avec le moins d'opérations possible.
- **Coût.** Le "coût" de l'arbre est mesurée en fonction des ressources utilisées. En général, il s'agit uniquement du nombre de liens impliqués dans l'arbre construit.
- **Délais.** Les délais de bout-en-bout sont également importants pour les flux à contraintes temporelles ou pour des applications interactives.

Il est difficile de comparer les performances d'un arbre par rapport à un autre, car les résultats obtenus dépendent évidemment de la topologie considérée. Les comparaisons n'ont donc de sens que sur des performances "moyennes" sur des topologies très variées. Les résultats que nous mentionnons dans la suite sont tirés de résultats de simulation [11, 26, 25] obtenus sur un très grand nombre de topologies, tirées grâce à des générateurs de graphes aléatoires [28]. Dans la suite de cette section, nous décrirons donc des comportements globaux (en moyenne sur un grand nombre de topologies possibles).

¹⁴Nous les citons dans un ordre arbitraire.

2.3.1 Arbre partagé/Arbre spécifique

Tout d'abord, il est important de noter que les arbres multicasts peuvent être classés en deux grandes catégories : les arbres partagés ou les arbres spécifiques à une source. Un arbre spécifique est construit à partir d'une source déterminée, de sorte qu'il est nécessaire de construire plusieurs arbres pour un même groupe multicast si ce groupe comprend plusieurs émetteurs. Il s'agit donc d'arbres unidirectionnels (de la source vers les récepteurs). Au contraire, un arbre partagé est établi une fois pour toute pour interconnecter tous les membres du groupe multicast. Il s'agit donc d'arbres bidirectionnels, où il n'y a pas de distinction entre émetteurs et récepteurs.

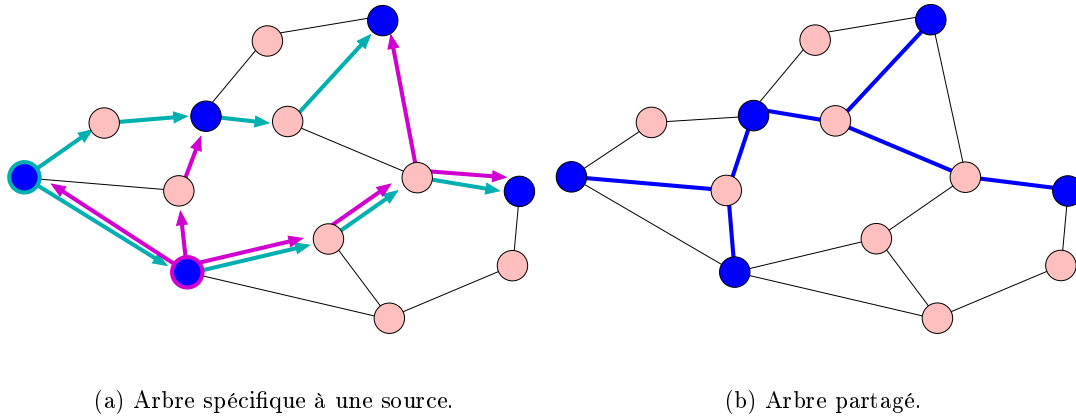


FIG. 2.5 – Arbres partagés ou spécifiques à une source.

Sur la figure 2.5 sont représentés ces deux types d'arbre pour un réseau simple — les membres du groupe multicast étant représentés en bleu. Comme nous pouvons le voir, les arbres spécifiques peuvent utiliser un plus grand nombre de liens. La différence entre ces deux types d'arbre porte sur le nombre d'états à stocker pour maintenir l'arbre en place. Dans le cas d'arbres spécifiques, il y a un arbre à décrire pour chaque source. Autrement dit, les nœuds du réseau doivent stocker un nombre d'états de l'ordre de grandeur $O(GS)$ où G est le nombre de groupes multicast présents dans le réseau et où S désigne le nombre moyen de sources par groupe multicast. Dans le cas des arbres partagés, au contraire, le nombre d'états à stocker ne dépend pas du nombre de sources, de sorte que l'ordre de grandeur est dans ce cas en $O(G)$.

Les arbres partagés sont donc plus adaptés pour des réseaux étendus et des réseaux de transit (devant véhiculer des informations d'un très grand nombre de groupes multicast). Nous reviendrons sur cette remarque lorsque nous aborderons le protocole PIM et les travaux actuels de l'IETF.

Remarque 2.2 Nous n'avons pas traité dans ce paragraphe des phénomènes de concentration

que peuvent engendrer certains types d'arbres. Si les arbres partagés sont plus économiques en terme de ressources (nombre d'états), des phénomènes de congestion peuvent apparaître au niveau de certains liens qui fédèrent le trafic provenant des différentes sources du groupe multicast. A l'opposé, les arbres spécifiques permettent de distribuer le trafic provenant des différentes sources sur des chemins différents, générant ainsi moins de congestion [18, 10]. Tout dépend en fait du choix effectué pour définir un "flux" dans le contexte du multicast : Soit une diffusion d'une source donnée vers des destinataires, ou seulement une communication de groupe, indépendamment de qui émet. Cette différence est bien mise en évidence dans le contexte du protocole RSVP¹⁵ par exemple [68].

2.3.2 Arbre de plus court chemin (SPT)

Les arbres de plus court chemin, nommés SPT (de l'anglais "Shortest Path Tree"), sont les arbres les plus utilisés actuellement.

Construction.

Les arbres SPT sont extrêmement simples à construire : Chaque feuille de l'arbre est connectée à la source en utilisant le plus court chemin défini par le protocole de routage unicast sous-jacent. Néanmoins, les capacités multicast des nœuds du réseau sont utilisées car deux branches qui se réunissent en un point donné sont fusionnées. Par exemple, les branches $D - B$ et $E - B$ se rejoignent au niveau du nœud F sur la figure 2.6.

Remarque 2.3 Afin de pouvoir différencier les différents arbres d'un même groupe, les routeurs doivent consulter l'adresse source des paquets reçus. Avec le routage unicast, les décisions sont basées sur l'adresse du destinataire uniquement. Pour le routage multicast, il est nécessaire de consulter à la fois l'adresse du destinataire (adresse de groupe) et l'adresse source dans le cas des arbres spécifiques.

Performances.

Par construction, les délais de transmission de ce type d'arbre sont minimaux (et ce quelque soit la topologie considérée). Du point de vue de l'utilisation des ressources, les arbres SPT ne sont par contre pas très économiques. Tout d'abord, il s'agit d'arbres spécifiques. D'autre part, chaque branche est construite indépendamment des autres par routage unicast, sans se soucier de l'éventuel proximité de nœuds du même groupe multicast. Il serait possible d'économiser

¹⁵RSVP : Resource reSerVation Protocol. Deux types de réservations sont définies : soit une réservation explicite pour un couple (*Source, Groupe*) ou une réservation partagée pour le groupe, indépendamment de la source, notée (***, *Groupe*).

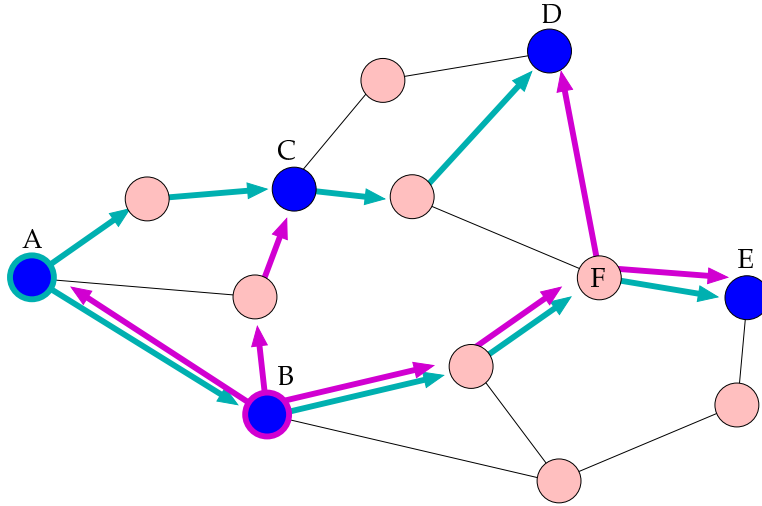


FIG. 2.6 – Arbre de type SPT.

des ressources en rejoignant une branche existante, même si les délais de bout-en-bout en seraient légèrement affectés. Par exemple, la liaison $A - E$ est très coûteuse : En passant par $A - C - F$, une économie d'un lien serait opérée. Par contre les délais pourraient en être affectés puisqu'il faudrait alors 5 "sauts" pour passer de A à E au lieu des 4 sauts préconisés par l'algorithme de plus court chemin¹⁶.

2.3.3 Arbre de Steiner

Les arbres de Steiner ne sont pas encore mis en œuvre¹⁷ mais constitue une approche intéressante du point de vue théorique et extrêmement prometteuse. Un arbre de Steiner est un arbre partagé permettant de connecter les membres du groupe à travers un graphe donné, et ce en minimisant les ressources utilisées. La figure 2.7 montre un exemple d'arbre de Steiner (utilisant un nombre minimal de liens) sur un graphe relativement simple.

Construction.

La construction d'un arbre de Steiner est un problème NP-complet, ce qui le rend extrêmement difficile à mettre en œuvre dans un réseau de grandes dimensions. Il s'agit néanmoins d'un problème bien connu de la théorie des graphes et il existe une littérature abondante sur le sujet (voir [14] par exemple). Des nombreuses heuristiques ont été proposées, l'heuristique KMB [16] (des initiales de ses trois inventeurs) étant l'une des plus connues et des plus uti-

¹⁶Si la métrique choisie est le nombre de sauts.

¹⁷A l'exception de quelques réseaux expérimentaux de faible taille.

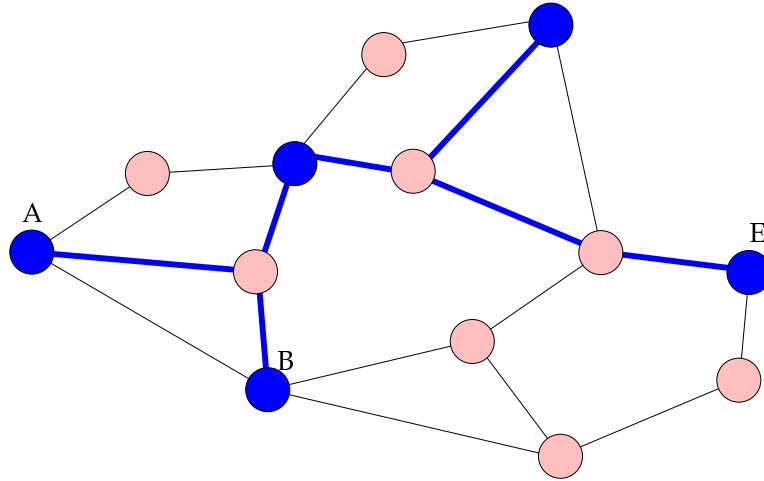


FIG. 2.7 – Arbre de type Steiner.

lisées¹⁸. La plupart des heuristiques se ramènent à un problème de type “arbre couvrant de poids minimum” (ou “minimum spanning tree”¹⁹). Imaginons que les routeurs disposent de la topologie du réseau (un graphe) et de la localisation des membres du groupe multicast. Les heuristiques de Steiner contiennent en général les trois étapes suivantes :

1. Le graphe original est réduit de manière à ne conserver que les nœuds du groupe multicast. Il s’agit d’un graphe maillé dont les arêtes sont marquées par les poids des plus court chemins joignant les deux points considérés.
2. Un algorithme d’arbre couvrant est utilisé sur le graphe réduit : Seule une partie des arêtes est ainsi sélectionnée.
3. Les branches correspondant aux arêtes sélectionnées dans l’arbre couvrant de poids minimum (sur le graphe réduit) sont alors construites.

Il a été prouvé que les arbres construits par ce type d’heuristiques utilisent dans le pire des cas 2 fois de liens qu’un arbre de Steiner — Cette borne est très pessimiste puisque des simulations donnent une inefficacité (ratio coût heuristique / coût Steiner) moyenne de 1,05 pour l’algorithme KMB [25]. Les principales variations parmi les différentes heuristiques portent sur le troisième point, certaines d’entre eux essayant d’inclure dans l’arbre des points du graphe dont la connectivité est importante afin de réduire le coût de l’arbre (les “points de Steiner”).

Remarquons qu’en dehors des considérations de complexité, la construction d’un arbre de Steiner nécessite la connaissance de la localisation de la totalité des membres du groupe : Ces informations ne sont pas toujours présentes car elles requièrent la mise en œuvre d’un

¹⁸Pas dans les réseaux réels, puisque ce type d’arbre n’est quasiment pas utilisé, mais pour les simulations. Le coût d’un arbre est en effet systématiquement comparé à celui d’un arbre de Steiner.

¹⁹Spanning Tree : Il s’agit de rejoindre tous les points d’un graphe en utilisant le moins de ressources

protocole de dissémination d'informations sur les groupes multicast. Nous pouvons donc voir que ce type d'arbre est mal adapté dans un contexte dynamique : Il faut annoncer le départ ou l'arrivée de membres du groupe dans tout le réseau (ou tout au moins dans toute le domaine de routage).

De plus, la littérature porte en général sur le problème “statique” de construction. Des heuristiques permettant de modifier l'arbre afin de rajouter ou de supprimer une feuille ont été peu traitées. Les heuristiques de type Steiner mentionnées précédemment ne sont en général pas itératives : Pour que les performances (en terme de consommation des ressources) soient maintenues malgré les changements qui surviennent dans le groupe, il est nécessaire de recalculer entièrement l'arbre de diffusion à chaque changement. Des heuristiques plus simples à mettre en œuvre et permettant d'ajouter de nouvelles branches sans modifier la totalité de l'arbre ont été étudiées dans [25, 11]. Certaines variantes des heuristiques classiques (type KMB) peuvent également être itératives [29]. Toutes les variantes itératives conduisent à une utilisation des ressources moins intéressante [25, 11].

Performances.

Par construction, les arbres de Steiner sont optimaux vis à vis du coût (c.à.d. de l'utilisation des ressources) — les heuristiques ne permettant bien sûr pas des coûts optimaux. Les délais subis à la traversée ce type d'arbre peuvent par contre être assez médiocres. Sur la figure 2.7 par exemple, il faut 5 sauts pour relier les nœuds B et E alors que 3 sauts suffiraient. Les délais ne sont pas satisfaisants (pour des flux à contraintes temporelles) dans certaines configurations, les arbres de Steiner conduisant parfois à des arbres en “couronne”. Pour illustration, la figure 2.8 montre un cas pathologique ou il est nécessaire de passer par 4 liens pour joindre les deux extrémités, alors que ces deux nœuds sont directement connectés.

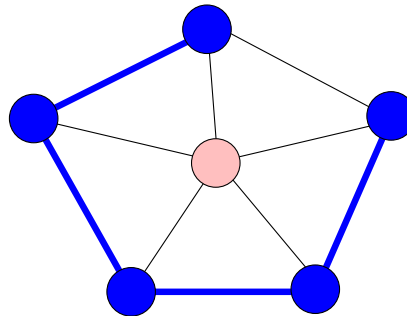


FIG. 2.8 – Un cas pathologique.

Remarquons que de nombreuses études ont été effectuées sur les arbres de Steiner avec contraintes (sur les délais) [20, 29]. Les heuristiques développées permettent d'obtenir de

meilleurs délais de bout-en-bout, au prix d'un coût supérieur.

2.3.4 Arbre centré (CBT)

Nous avons vu l'intérêt des arbres partagés pour réduire le nombre d'états à stocker au niveau des nœuds du réseau. L'arbre de Steiner permet de construire des arbres partagés, mais la construction de ces arbres est très complexe et nécessite la localisation de tous les membres des groupes multicast (information inconnue dans l'Internet actuel²⁰ par exemple). Les arbres centrés permettent de construire des arbres partagés de manière extrêmement simple. De plus, les performances obtenues par ces arbres peuvent être intéressantes.

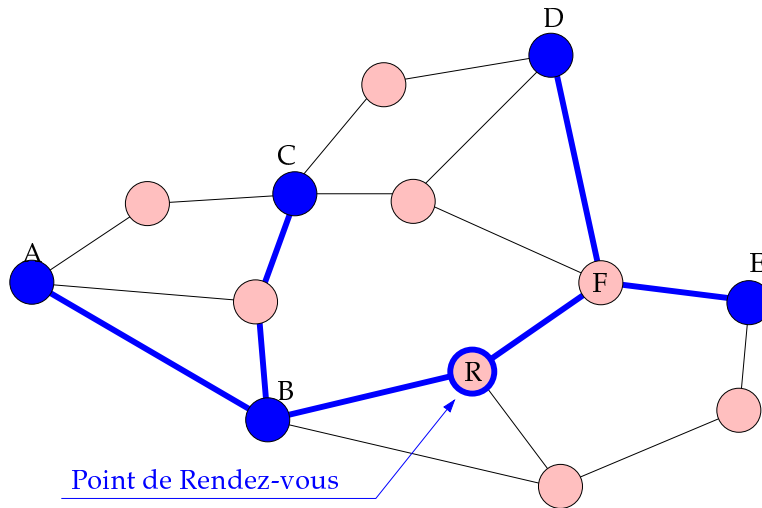


FIG. 2.9 – Arbre de type CBT.

Construction.

Un nœud du réseau est désigné comme “point de rendez-vous” ou un “centre” — d’où la désignation anglosaxonne “Center Based Tree” (CBT). Pour connecter une feuille à l’arbre multicast, il suffit de connecter cette feuille au RP (c.à.d. au point de rendez-vous) par le plus-court chemin (en utilisant le routage unicast sous-jacent). Comme dans le cas d’un arbre SPT, les branches se rejoignant en un même point sont fusionnées. Par exemple, les branches $E - R$ et $D - R$ se rejoignent au niveau du nœud F sur la figure 2.9.

La construction d’un arbre centré est donc extrêmement simple mais elle requiert la connaissance du point de rendez-vous. En général, les points de rendez-vous sont statiques

²⁰Dans l’Internet actuel... Et très certainement dans le futur également, les spécialistes de l’IETF étant tout à fait hostile à la propagation de ce genre d’informations au niveau du routage externe — Ce genre d’information ne pouvant être “agrégé” (voir deuxième partie).

(fixés une fois pour toute par l'administrateur) de sorte que cette information est entrée lors de la configuration des nœuds du réseau. Des mécanismes permettant d'obtenir l'identité d'un point de rendez-vous dynamiquement sont en cours de développement [13]. Afin d'éviter de trop grandes concentrations de trafic au niveau d'un point de rendez-vous, il existe en général plusieurs RPs, chacun étant responsable d'une plage d'adresse multicast. La sélection du RP adéquat est donc extrêmement simple après consultation de l'adresse du groupe que la feuille cherche à joindre (par "hashing").

Performance.

Les performances des arbres centrés dépendent de l'emplacement du point de rendez-vous. Il est facile de voir que le RP a tout intérêt à être situé au "milieu" du réseau (si ceci à un sens...). Il serait tentant de choisir le RP dynamiquement en fonction de la localisation des membres du groupe, afin de minimiser les ressources utilisées (et éventuellement sous contraintes du point de vue des délais du bout en bout). Cependant, cette optimisation nécessite la localisation de tous les membres du groupe²¹. De plus, il s'agit, comme pour le cas de l'arbre de Steiner, d'un problème NP-complet. Des heuristiques très simples, et donc exploitables dans un réseau opérationnel, ont été développées[12]. Néanmoins, ces heuristiques n'ont jamais été véritablement mises en œuvre dans l'Internet. Les RP sont généralement fixés administrativement, indépendamment de la location des membres du groupe. Notons que même lorsque le RP est placé de manière arbitraire, les performances moyennes (délais, coûts) sont raisonnables : Le gain en terme de ressources utilisés peut être très significatif par rapport à des arbres SPT dans le cas d'applications many-to-many, sans que les délais en soient trop affectés [10].

2.3.5 Bilan

Sur la table 2.1 sont résumés schématiquement les caractéristiques moyennes des trois types d'arbres multicast étudiés. Les résultats annoncés sont issus des remarques faites dans les trois paragraphes précédents et d'après des résultats de simulation bien connus dans le monde Internet [26, 11, 10].

Les arbres de Steiner sont donc particulièrement intéressants du point de vue théorique car ils permettent de réduire considérablement les ressources (nombre de liens et états dans les routeurs) utilisés par un groupe multicast. Dans un réseau de grande dimension ou un réseau de transit, cette solution pourrait être très intéressante si sa mise en œuvre n'était pas

²¹Dans la plupart des protocoles construisant ce type d'arbre, les RP ne savent absolument pas combien de machines sont connectés au groupe. En fait, ce nœud ne prend conscience qu'il est un RP que lorsqu'il reçoit une demande de raccordement au groupe.

	SPT	CBT	Steiner
Complexité	Faible	Faible	Grande
Dynamisme	Bon	Bon	Mauvais
Coût	Important	Moyen	Faible
Etats	$O(GS)$	$O(G)$	$O(G)$
Délais	Faible	Moyen	Moyen
Concentration	Faible	Forte	Forte

TAB. 2.1 – Comparaison des principaux types d’arbres.

si complexe. A cause de son manque de dynamisme et de la trop grande complexité associée à sa construction, les arbres de Steiner ne sont pas utilisés à l’heure actuelle.

Les arbres **SPT** sont les plus utilisés pour leur simplicité et leurs excellentes performances pour les flux multimédia ou les flux interactifs (faible délais et faible concentration²²). Néanmoins, ce type d’arbre consomme beaucoup de ressources (nombre d’états, nombre de liens mis en jeu) : Il serait préférable d’utiliser des arbres partagés pour des applications many-to-many sans contraintes temporelles trop exigeantes.

Les arbres **CBT** apparaissent comme une solution intermédiaire par rapport aux deux cas extrêmes précédents : Les délais peuvent être assez bons si le **RP** est bien placé et l’utilisation de ressources est très nettement réduite par rapport à des arbres **SPT** dans le cas de groupes possédant plusieurs sources. Ce type d’arbre est de plus très facile à mettre en œuvre, de sorte que cette solution est préconisée pour des applications “many-to-many” sans contraintes temps réelles trop exigeantes. Étant donné qu’il s’agit d’un arbre partagé (et donc réalisable dans de grand réseaux²³) facile à construire, ce type d’arbre est également envisagé pour des réseaux de transit (c.à.d. pour le routage externe²⁴ IP par exemple).

Remarque 2.4 Nous n’avons présenté dans cette section que les arbres élémentaires. D’autres variantes existent, comme les arbres hiérarchiques dont nous parlons à la fin de ce chapitre et qui seront également l’objet d’étude du chapitre 3.

2.4 Protocoles de Routage Multicast existants

Dans cette section, nous décrivons rapidement l’état de l’art sur les protocoles de routage

²²Concentration : voir remarque 2.2 à la page 17 du trafic.

²³Les anglo-saxons diraient “scalable”.

²⁴Voir partie suivante. En simplifiant, il s’agit du routage permettant de relier des domaines de routage indépendants entre eux.

multicast. Nous nous concentrons sur les protocoles de l'Internet (les autres réseaux étant dépourvus de tels protocoles²⁵).

2.4.1 DVMRP

DVMRP (Distance Vector Multicast Routing Protocol) fût le premier protocole de routage multicast utilisé dans l'Internet. Il s'agit dans sa première version d'un protocole extrêmement basique de diffusion ("broadcast") et non de diffusion sélective : Tout paquet émis est diffusé dans tout le réseau et est donc reçu par tous les routeurs de l'Internet. Le filtrage n'est assuré qu'au niveau local par les routeurs IP en fonction des informations reçues par le protocole IGMP — l'information est transmise sur le réseau local si au moins un hôte participe au groupe multicast.

Pour éviter les problèmes de boucles qui peuvent être créées lors de la diffusion, un protocole spécifique, nommé "Reverse Path Broadcasting" (RPB) est utilisé — on parle aussi de RPF (Reverse Path Forwarding). Lors de la réception d'un paquet multicast, l'adresse source est analysée afin de déterminer si ce paquet doit être relayé vers les autres interfaces ou être détruit. Cette décision est prise en consultant une table de routage unicast : le port de sortie menant (par le plus court chemin) vers la source est déterminé. Si cette interface correspond à celle sur laquelle a été reçue le paquet, l'information est transmise en aval. Dans le cas contraire, le paquet reçu est détruit. Cet algorithme permet donc de construire un arbre couvrant, optimal pour les délais de diffusion lorsque le réseau est symétrique. L'arbre peut être amélioré en analysant les messages échangés par le protocole de routage unicast sous-jacent (il s'agit d'un algorithme "distance vector"²⁶, cf. chapitre 4). Il est alors possible de déterminer s'il est nécessaire ou non de transmettre l'information sur certaines interfaces — selon que le plus court chemin à partir du nœud voisin vers la destination passe par cette interface ou non... La figure 2.10 représente les arbres couvrants obtenus par cette technique.

Cette technique rudimentaire n'est pas très efficace car l'information est reçue par tous les nœuds du réseau, qu'ils désirent participer au groupe ou non. Le protocole DVMRP v.2 permet d'éliminer les diffusions vers des réseaux qui ne font pas partie du groupe par "pruning" : Une diffusion (RPB) est régulièrement effectuée, de sorte que tous les routeurs du réseau reçoivent l'information. Les routeurs ne possédant aucun membre du groupe sur leurs interfaces envoient ensuite un message spécifique en amont afin de ne plus recevoir d'information pour ce groupe. Un routeur renvoie ce message **prune** en amont si toutes ces interfaces désirent

²⁵En ATM par exemple, il n'y a pas à proprement parlé de routage multicast. Les connections point-à-multipoints étant ouvertes branche par branche par le routage unicast.

²⁶D'où le nom choisi de "Distance Vector Multicast Routing Protocol". DVMRP ne fonctionne en fait avec un protocole de routage unicast de type "distance vector".

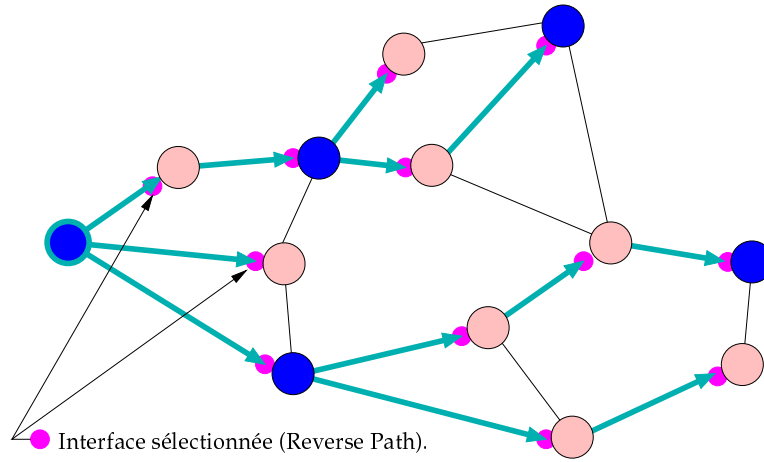


FIG. 2.10 – Diffusion “Reverse Path Broadcasting”.

quitter le groupe. Ce message de “pruning” est donc transféré de proche en proche, éliminant les branches superflues. Après quelques échanges signalétiques, un arbre **SPT** est finalement construit, comme le montre la figure 2.11.

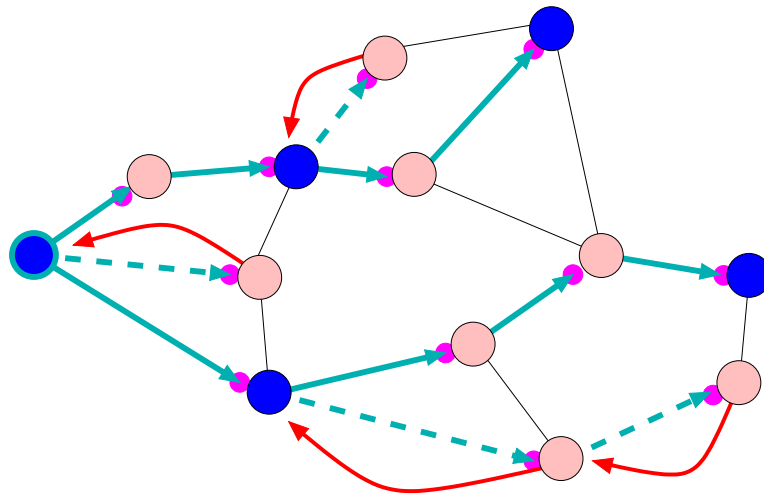


FIG. 2.11 – Diffusion partielle “Reverse Path Multicasting”.

Remarquons que l’arbre **SPT** a été construit de manière extrêmement simple, sans qu’aucune information sur la localisation des membres du groupe multicast n’ait été échangée sur le réseau. En revanche, ce protocole souffre des limitations suivantes :

- **Broadcast périodiques.** Afin de pouvoir s’adapter dynamiquement aux changements dans le groupe multicast (arrivée ou départ de feuilles), il est nécessaire de reconstruire l’arbre périodiquement. Ces reconstructions sont assurées par une diffusion totale par **RPB** suivis de nouveaux messages de “pruning”.

- **Robustesse.** Le protocole de routage unicast intégré (utilisé par l’algorithme **RPF**) est un protocole de type “distance vector” (cf. chapitre 4) assez rudimentaire souffrant de problèmes de convergence (“comptage à l’infini”). Les Métriques utilisées limitent de plus l’étendue du réseau (32 sauts maximum).
- **Surcoût.** Ce protocole fonctionne avec son protocole de routage unicast propre (de type “distance vector”), indépendant du protocole de routage unicast sous-jacent²⁷.
- **Arbres SPT.** Ce protocole ne peut pas construire d’arbres partagés.

Remarque 2.5 Ce protocole [85] est désormais historique, et n’est quasiment plus utilisé — étant remplacé par le protocole **PIM** pour le routage interne. Par contre, le protocole **DVMRP** permet d’utiliser des “tuyaux” (encapsulation IP dans IP) permettant de franchir des routeurs ne supportant pas le multicast. **DVMRP** est encore utilisé dans le M-bone (voir plus loin) pour la définition de ces tunnels — Bien que dans ce cas également, ce protocole soit de plus en plus remplacé par des solutions propriétaires (en attendant l’arrivée de standards) plus souples et moins “bavardes”.

2.4.2 PIM

Le protocole **PIM** [10] doit son nom au fait qu’il peut fonctionner sur n’importe quel protocole unicast sous-jacent (“Protocol Independent Multicast”) — contrairement à son prédécesseur. Il s’agit en fait de deux protocoles distincts qui peuvent être utilisés de paire :

- **Mode dense.** Le protocole **PIM-DM** [72] (PIM-“Dense Mode”) est très proche du protocole **DVMRP**, mais utilisant les tables de routage du protocole de routage unicast sous-jacent (quelqu’il soit). Ce protocole permet donc de construire des arbres **SPT** par diffusion puis élagage des branches inutiles (c.à.d. algorithme **RPF**).
- **Mode Épars.** Le protocole **PIM-SM** [73] (PIM-“Sparse Mode”) permet de construire des arbres **CBT** (par défaut, voir figure 2.12(a)) ou **SPT** (sur demande explicite, voir figure 2.12(b)). Les informations d’un groupe multicast ne sont transmises vers une feuille qu’après l’établissement (une signalisation spécifique étant définie à cette fin) d’une

²⁷Il y a donc deux protocoles de routage unicast tournant en parallèle, ce qui représente un gâchis certain en terme d’utilisation des ressources. Néanmoins ce dédoublement a été voulu afin de contrôler le coût des “tuyaux” (voir remarque 2.5) pour construire une topologie logique (au dessus de la topologie physique et des tunnels). Les métriques peuvent aussi être ajustées pour pouvoir utiliser des chemins différents pour le trafic multicast et le trafic unicast. Si la table de routage unicast existante était utilisée, la totalité des chemins reçus du “tuyau” serait détruite — l’algorithme **RPF** ne sélectionnant pas cette interface comme la plus appropriée pour joindre la source dans ce cas puisque les chemins unicast et multicast sont différents... Des techniques alternatives comme **MGBP** (voir plus loin) sont plus adaptées pour les fournisseurs de service Internet (véritable notion de “police routing”).

branche effectuée — il n’y a aucun phénomène de “diffusion” comme c’est le cas dans PIM-DM.

La figure 2.12 résume brièvement les mécanismes mis en œuvre pour la construction d’un arbre SPT ou CBT pour le protocole PIM-SM — Le fonctionnement de PIM-DM ne sera pas décrit dans ce paragraphe, étant totalement similaire à celui de DVMRP. Avec le protocole PIM-SM, un arbre centré est construit initialement, comme montré sur la figure 2.12(a). Le point de rendez-vous est fixé administrativement lors de la configuration des routeurs. Les récepteurs peuvent ensuite prendre l’initiative de quitter l’arbre centré et de joindre explicitement une source donnée (cf. figure 2.12(b)), si le trafic émis par cette source devient soutenu (un débit seuil est défini par l’administrateur).

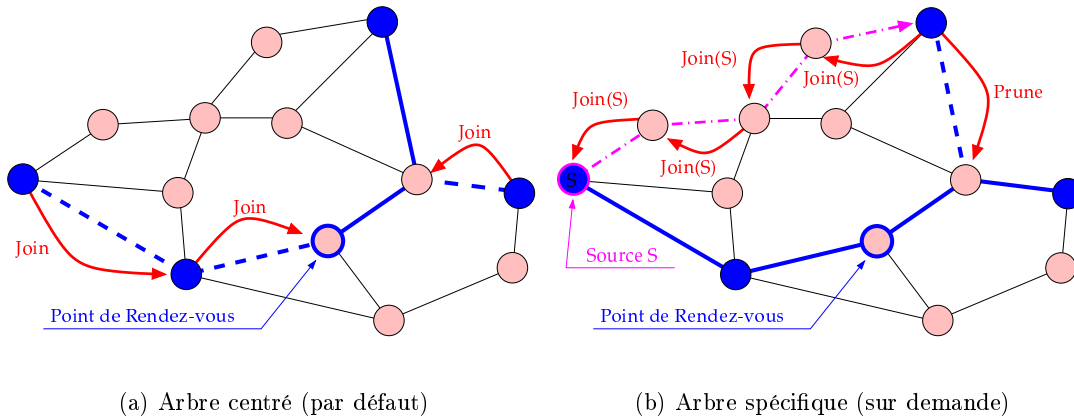


FIG. 2.12 – Signalisation du protocole PIM-SM.

La dénomination “mode épars”/“mode dense” est justifiée par les applications visées par chaque protocole : comme montré dans [27], le mode épars est plus économique en terme d’échanges de signalisation ou d’états à stocker lorsque le groupe est faiblement distribué dans le réseau (c.à.d. lorsque le nombre de membres du groupe est faible par rapport au nombre total de nœud du réseau). Au contraire, le mode dense est plus adapté lorsque les membres du groupe sont fortement concentrés dans le réseau (c.à.d. une large proportion des routeurs du réseau désirent participer au groupe multicast). D’après des résultats de simulation, la frontière entre ces deux modes se situe aux alentours d’une densité de 30 à 40% (c.à.d. 40% des routeurs doivent participer au groupe multicast).

Le protocole PIM est de plus en plus utilisé dans les réseaux locaux, essentiellement sous sa forme dense. Les fournisseurs de service s’intéressent également de plus en plus à ce protocole pour le “sparse mode” qui permet d’éviter les diffusions intempestives dans le réseau et permet également de construire des arbres partagés, plus économiques en termes de ressources. L’avantage majeur de cette solution est d’être indépendante du protocole de routage unicast

sous-jacent et surtout de pouvoir offrir différents types d'arbres (SPT ou CBT) et différentes signalisations (adaptées au mode dense ou épars) dans un même protocole²⁸.

2.4.3 Autres protocoles

Les protocoles les plus utilisés sont indiscutablement les protocoles PIM et DVMRP. D'autres protocoles ont néanmoins été définis et sont parfois disponibles sur le marché. C'est le cas notamment de MOSPF et CBT.

MOSPF.

MOSPF [80] est une extension du fameux protocole de routage unicast OSPF²⁹ — MOSPF signifie "Multicast extension to OSPF". Ce protocole construit des arbres SPT uniquement. Les informations sur les groupes multicast sont échangées dans le réseau (grâce à des messages "Link State Advertisements" spécifiques). Chaque routeur dispose de la vision topologique du réseau et de la localisation des membres du groupe multicast. Il leur est donc possible par un simple calcul (Bellman-ford par exemple) de construire une représentation de l'arbre SPT. Aucune diffusion ni aucun échange signalétique n'est donc nécessaire pour la construction des arbres multicast. Les arbres partagés ne sont pas supportés.

CBT.

Le protocole CBT [9] ("Center Based Tree") construit, comme son nom l'indique, des arbres centrés. Il s'agit d'un protocole peu "bavard" (adapté au mode épars) proche du protocole PIM-SM. La différence majeure avec ce dernier vient de la signalisation. Le protocole CBT permet également de construire des arbres hiérarchiques centrés (voir plus loin) — la création de ces arbres pouvait conduire à des boucles ou des blocages ("dead-locks") dans sa première version [21, 66], de sorte que cette facilité n'est plus présente dans CBT v.3 [66].

2.4.4 Le "M-bone"

Un réseau expérimental, nommé "M-Bone" (pour "Multicast Back-Bone"), a été introduit dans l'Internet depuis 1992. Il relie la plupart des centres de recherches et des campus universitaires mondiaux. Les fournisseurs de service Internet ont été très réticents, au départ, à l'idée d'offrir des services multicast, à l'exception de quelques fournisseurs de services sur la Silicon

²⁸Le mode "sparse/dense" est proposé par certains constructeurs. Dans ce mode de fonctionnement, le mode "sparse" est utilisé si un point de rendez-vous est défini pour l'adresse multicast du groupe. Dans le cas contraire, le mode "dense" est utilisé

²⁹OSPF : Open Shortest Path First. Voir chapitre 4.

Valey. Ce état de l'art est en train de changer puisque les plus grands fournisseurs de services offrent à présent des services multicast ³⁰— sinon, de tels projets sont souvent à l'étude.

Architecture.

Les ressources multicast ne sont donc généralement pas supportées dans l'Internet. La connectivité entre les domaines multicast (appelés les “nuages multicast”) est souvent assurée par des “tuyaux” (encapsulation IP dans IP) permettant de franchir les zones qui ne supportent pas le multicast. Historiquement, les tuyaux étaient construits grâce au protocole *DVMRP* dont nous avons parlé plus haut, mais cette solution est de plus en plus abandonnée au profit de protocoles propriétaires³¹, ce protocole n'étant pas adapté pour des réseaux de trop grande taille³². Le manque de protocole de routage multicast viable à grande échelle, c.à.d adapté au routage inter-domaine (ou “externe” cf. chapitre 4) constitue un frein au développement du multicast. Des solutions sont en cours de développement chez les constructeurs et à l'IETF — nous y consacrons la fin de ce chapitre. A l'intérieur des “nuages multicast”, n'importe quel protocole évoqué plus haut peut être utilisé — Il s'agit le plus généralement de PIM-DM, ou PIM-SM dans une moindre mesure.

Fonctionnement.

Le M-bone est un succès du point de vue du développement de nouvelles applications. Nous décrivons rapidement dans la suite de cette section l'organisation de ce réseau (pour mieux comprendre à quelles fins tous ces protocoles de routage multicast sont utilisés!!!)

Jusqu'à présent, nous n'avons pas parlé d'un problème extrêmement important associé au multicast : la gestion du plan d'adressage et les annonces de sessions multicast. Ces deux fonctions sont assurées par des programmes appelés “Session Directory”³³ utilisant les protocoles *SDP* (Session Directory Protocol [76]) et *SAP* (Session Announcement protocol [78]) . Ces programmes se connectent sur une adresse IP multicast réservée sur laquelle sont annoncées (sous des formats spécifiés par les protocoles précédemment cités) les différentes sessions multicast actuellement en cours sur l'Internet. Au delà de l'attrait pour l'utilisateur (son “pro-

³⁰Voir <http://www.psc.edu/~mathis/contacts.html> pour quelques informations officielles sur les fournisseurs de services multicast.

³¹GRE, MSDP, MBGP pour Cisco par exemple. Voir <http://www.cisco.com>.

³²Le protocole de routage unicast sous-jacent est un algorithme “distance-vector” peu sophistiqué (sujet aux problèmes de comptage à l'infini) et non-hiérarchique. Voir le chapitre 4 pour plus de précisions sur ces notions de routage unicast.

³³Les principaux logiciels sont : le *sd* de Berkeley ou le *sdr* de l'UCL.

Voir http://www.it.kth.se/~e93_mda/mbone/ripegw/resources.html par exemple pour une liste des sites de téléchargement.

gramme télévisé” en quelque sorte), les adresses multicast permettant de joindre ces sessions sont annoncées, avec des informations sur les applications utilisées et les formats des données transmises. Ceci permet alors à **sdr** (ou **sd**) de lancer directement les applications avec les paramètres adéquats pour participer aux sessions sélectionnées. Ces programmes possèdent également des protocoles de création de session et d’invitation (Session Initiation Protocol [77]) afin de pouvoir inviter certains membres à joindre un groupe multicast. Notons qu’il n’y a pas réellement de contrôle de session multicast (pour des groupes privés), la sécurité étant assurée par cryptage des données. La figure 2.13 montre quelques captures d’écrans lors d’une utilisation du programme **sdr**.



FIG. 2.13 – Exemple d’utilisation de **sdr**.

Il est possible de définir des couvertures géographiques pour les sessions créées en jouant sur le champ TTL³⁴ des paquets IP. La figure 2.14 résume les conventions de couverture géogra-

³⁴TTL : Time To Live.

phique du Mbone. Les paquets émis sont filtrés aux passages dans les routeurs en fonction des TTLs afin d’assurer ces couvertures géographiques. Finalement, notons que lors de la création d’un nouveau groupe multicast, l’adresse multicast du groupe est sélectionnée aléatoirement par le “répertoire” (**sdr**) parmi les adresses qui semblent libres (c’est à dire pour laquelle aucune annonce n’a été reçue par le protocole **SAP**). Des architectures d’allocation d’adresses plus élaborées sont en cours d’étude [71, 19, 69]. Le groupe “MALLOC” de l’IETF travaille sur ce problème.

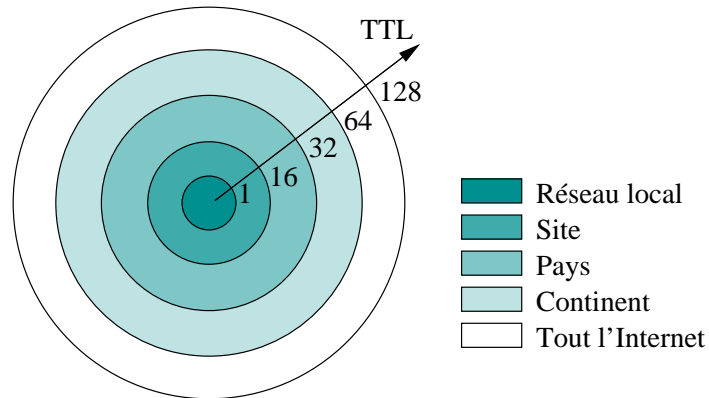


FIG. 2.14 – Couverture géographique d’une session multicast.

2.5 Travaux en cours

Le multicast dans son ensemble est un sujet extrêmement prolifique de développement et de recherche. Nous ne citons dans la suite que quelques problèmes qui nous intéressent directement pour la suite de cette partie, c’est à dire les travaux sur le routage multicast à proprement parler.

Les protocoles de routage multicast intra-domaine sont à présent bien spécifiés et mis en œuvre. Par contre, le manque de protocole adapté aux très grands réseaux est un frein au développement des capacités multicast dans l’Internet. Les travaux sur le routage multicast portent donc essentiellement sur le routage multicast inter-domaine.

2.5.1 “Policy Routing”

Le protocole **DVMRP** est encore utilisé de nos jours pour les “tuyaux” que ce protocole supporte et également grâce aux métriques que ce protocole transporte, indépendamment de la topologie sous-jacente. Grâce à ces métriques propres, il est possible de contrôler le routage sur une topologie logique (au dessus des “tuyaux”). Une application intéressante de

cette capacité est de définir des chemins différents pour les trafics unicast et multicast — ce qui est impossible si le routage unicast standard est utilisé par l'algorithme RPF.

La séparation entre le routage unicast et le routage multicast est une forme particulière d'une notion plus générale de routage que les anglo-saxons appellent "policy-routing"³⁵ : Il ne s'agit plus d'un routage basé sur la topologie uniquement (plus courts chemins), mais qui doit tenir compte d'autres contraintes. Dans l'Internet, les principales contraintes sont causées par les accords commerciaux entre les différents fournisseurs d'accès ou de services Internet : Un fournisseur de service (ISP³⁶ A) accepte, moyennant finance, de propager les informations provenant de l'ISP B. L'ISP B peut être un petit fournisseur d'accès à Internet couvrant une région, et l'ISP A un large fournisseur de service international par exemple. L'ISP B est peut-être également relié à un autre fournisseur national ISP C. Il est probable que dans ces conditions, L'ISP A accepte les paquets provenant de l'ISP B mais refuse les paquets qui auraient pu traverser B en provenance de l'ISP C...

Ces accords commerciaux (entre autres) posent de fortes contraintes sur les politiques de routage. Ces problèmes sont résolus dans le cas du routage unicast par le protocole BGP (Border Gateway Protocol [82]). Il n'existe encore aucun équivalent dans le cadre du multicast — rappelons nous que le M-bone est un réseau expérimental qui n'est donc pas du tout concerné par les problèmes d'accords commerciaux évoqués plus haut.

Les solutions en cours d'étude proposent en général de propager les informations de routage multicast par le protocole BGP de façon à bénéficier de ses capacités à gérer le "policy routing". Ces solutions sont basées sur le protocole BGP 4+ [67], qui permet l'échange d'informations supplémentaires (par rapport aux informations de routage unicast). On peut citer notamment les travaux suivants : [70, 84] et [83].

2.5.2 Arbres Hiérarchiques

Présentation.

De nombreux travaux sont également en cours sur l'utilisation des arbres hiérarchiques. Il est en effet reconnu que l'utilisation d'arbres partagés est inéluctable pour le routage inter-domaine d'un réseau aussi important que l'Internet. Il n'est pourtant pas question d'utiliser des arbres de Steiner, trop complexes et nécessitant des informations sur la localisation des membres des groupes multicasts. Les arbres centrés apparaissent comme un candidat intéres-

³⁵Le terme de "policy routing" utilisé pour l'Internet actuel et BGP 4 pourrait faire sourire certains spécialistes. Il est vrai que les fonctionnalités offertes sont pour l'instant minimalistes. L'architecture MPLS est d'ailleurs très attendu car elle devrait permettre un développement considérable dans ce domaine (plus généralement sur le "traffic engineering" dans l'Internet).

³⁶ISP : Internet Service Provider

sant [10]. Néanmoins, le choix du core et les problèmes associés (voir [15]) rendent difficile leur mise en œuvre à si grande échelle. L'idée d'utiliser un arbre basé sur une hiérarchie de point de rendez-vous est donc apparue.

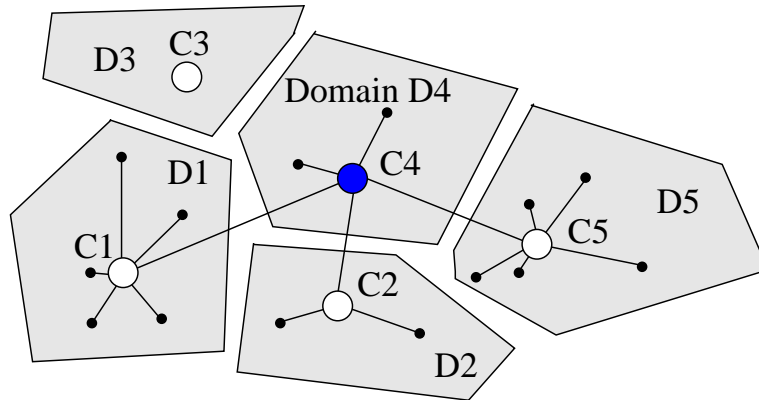


FIG. 2.15 – Arbres hiérarchiques centrés à deux niveaux hiérarchiques.

Sur la figure 2.15 est représenté un arbre centré à deux niveaux hiérarchiques. Le réseau est divisé en domaines de routage multicast dans lesquels sont définis un ou plusieurs points de rendez-vous — chacun d'entre eux étant associé à une plage d'adresse multicast par exemple. Au niveau hiérarchique supérieur, un (ou plusieurs) point(s) de rendez-vous central est (sont) désigné(s) — ce nœud peut très bien être un RP de niveau inférieur également. Les routeurs recevant une demande de participation au groupe multicast (via IGMP) se connectent au RP de leur domaine (correspondant à l'adresse multicast souhaitée). Si ce RP n'appartient pas déjà au groupe, il doit à son tour joindre un RP de niveau hiérarchique supérieur et ainsi de suite jusqu'à atteindre le RP central. Les niveaux hiérarchiques peuvent par exemple représenter les sites d'une entreprise puis l'entreprise, la région, le pays, le continent puis l'Internet global.

L'arbre résultant est un arbre partagé (bidirectionnel) dont la construction n'est pas beaucoup plus complexe que celle d'un arbre CBT — si on suppose la liste des cores et la définition des domaines de routage multicast connues. L'avantage de cette approche est que, malgré sa simplicité, l'arbre hiérarchique s'adapte naturellement à la distribution géographique des membres du groupe multicast. Si les membres du groupe sont tous localisés dans un domaine particulier, l'arbre construit consiste essentiellement en un arbre centré couvrant le domaine et ne fait intervenir aucun autre RP ou nœuds des autres domaines — à l'exception du RP central. Si des nouveaux membres du groupe apparaissent dans d'autres domaines, de nouveaux RP interviennent, etc.

Les performances de ces arbres hiérarchiques ont été néanmoins peu étudiées. La plupart des recherches sur les arbres hiérarchiques ont porté sur les aspects algorithmiques essentiellement. Quelques brèves études par simulation sont disponibles dans [23, 21]. L'impact du

nombre de niveaux hiérarchiques et plus généralement de la structure hiérarchique sur les performances de ces arbres hiérarchiques centrés n'ont pas été étudiés à notre connaissance. Il nous a semblé qu'il s'agit pourtant d'un problème essentiel pour le dimensionnement de réseau — Faut-il, par exemple, utiliser les domaines de routage unicast existants ou utiliser des domaines de routage multicast plus petits (resp. grands), organisés avec plus (resp. moins) de niveaux hiérarchiques ? Dans le chapitre suivant, nous analysons les performances d'arbres hiérarchiques centrés grâce à une étude analytique afin d'apporter quelques réponses à ce problème délicat.

Nous nous sommes également intéressés aux algorithmes de construction des arbres hiérarchiques [7, 3], mais ces travaux sont trop peu avancés pour les inclure dans ce rapport.

Travaux relatifs aux arbres hiérarchiques centrés.

La première architecture basée sur une hiérarchie de points de rendez-vous a été introduite dans le protocole CBT [9]. Plus exactement, ce protocole définit deux niveaux hiérarchiques : Un point de rendez-vous principal est utilisé pour interconnecter les RPs secondaires, où viennent se connecter les routeurs. Néanmoins, les versions récentes de CBT [66] ne possèdent plus qu'un seul niveau hiérarchique à cause de problèmes de boucles et de blocages ("deadlocks") identifiés dans [21]...

La véritable notion d'arbres hiérarchiques apparaît très brièvement dans un rapport de recherche de l'UCL³⁷ [15]. La hiérarchie de points de rendez-vous permet en effet de résoudre de nombreux problèmes liés à la connaissance du point de rendez-vous avec les arbres centrés. Les recherches n'ont hélas pas pu être poursuivies — bien que de nombreuses idées de cette proposition aient été reprises quelques années plus tard pour le développement de l'architecture MASC/BGMP.

Les arbres hiérarchiques centrés ont été étudiés plus en profondeur à l'UCSC³⁸. Dans [21], les problèmes de boucles liées à la construction d'arbres hiérarchiques ont été étudiées. La solution proposée résout les problèmes de boucles et de blocages dans la construction de l'arbre — une preuve formelle du bon fonctionnement de l'algorithme étant donnée. Les performances de ces arbres en terme de délais et de consommation de ressource sont également brièvement étudiées par simulation. Il apparaît que les arbres hiérarchiques sont légèrement plus complexes à construire (en terme de messages échangés) par rapport à un arbre CBT classique. Les délais moyens peuvent par contre être améliorés significativement suivant la structure hiérarchique choisie. Dans [22], une architecture complète permettant de construire des arbres hiérarchiques centrés à l'échelle de l'Internet est proposée par les mêmes auteurs.

³⁷UCL : University College London.

³⁸UCSC : University of California at Santa Cruz.

Vers des Réseaux ATM Multicast ?

Les arbres hiérarchiques centrés ont été également étudiés dans les réseaux ATM. En effet, ces réseaux ne possèdent pas à ce jour de capacité multicast à proprement parler — bien que des connexions point-à-multipoints existent³⁹ Les connexions point-à-multipoints actuelles sont très coûteuses en terme de ressources (arbres de type SPT). Des travaux sont donc en cours pour permettre l'introduction d'arbres partagés dans les réseaux ATM. Les arbres hiérarchiques apparaissent comme un excellent candidat, car la structure hiérarchique du routage (unicast) PNNI, présentée dans le chapitre 4, peut être utilisée aisément pour la création d'arbres hiérarchiques centrés [23]. Nous nous sommes également intéressés à ces problèmes [5, 7], bien qu'il n'ait pas été possible de les inclure dans ce manuscrit, faute de place.

Notons finalement que la structure hiérarchique peut également être utilisée pour la création d'arbres de Steiner hiérarchiques. L'avantage de cette technique est que les calculs liés à la construction des arbres de Steiner sont toujours limités à des domaines de routage de faible taille [24]. Remarquons néanmoins que cette solution n'a pas eu d'échos dans l'Internet, car le nombre de niveaux hiérarchiques du réseau des réseaux est insuffisant pour travailler avec des domaines suffisamment petits⁴⁰.

Développements à l'IETF.

Les développements algorithmiques sur les arbres hiérarchiques ont été suivis avec intérêt par la communauté IETF et ont inspiré très nettement les développements actuels sur l'architecture de routage multicast inter-domaine MASC/BGMP [17].

Dans cette architecture, la notion d'arbre hiérarchique à deux niveaux est utilisée, bien qu'il ne s'agisse pas d'arbres hiérarchiques centrés. Les spécialistes de l'IETF ont en effet reproché à cette solution un manque de robustesse du fait que la connectivité de l'arbre peut être perdue au cas où le RP central deviendrait défaillant. Dans BGMP un arbre centré est construit au niveau des domaines de routage de sorte que le point de rendez-vous est un domaine de routage et non un routeur. La connectivité de l'arbre est donc assurée tant qu'il existe une route (par routage unicast) menant vers ce domaine.

Le point de rendez-vous est fixé comme le domaine créateur du groupe multicast. Un

³⁹Les connexions point-à-multipoints sont construites branche par branche par le routage unicast, sans utiliser les adresses de groupe. Il ne s'agit donc pas de multicast. Les capacités de "Leaf-Initiated-Join" [64] peuvent être considérées comme une solution "multicast" bien que les adresses de groupe ATM ne soient pas utilisées. Le L.I.J. est supporté par l'UNI 4.0 mais n'est pas encore opérationnel : Il faudra attendre la sortie de PNNI version 2...

⁴⁰Ces notions de taille de domaines et de niveaux hiérarchiques sont étudiés en détail dans le chapitre 6

protocole d'allocation d'adresse spécifique (MASC [71]) permet d'associer à chaque domaine des plages d'adresses multicast, de sorte que la détermination du domaine de rendez-vous est aisée. Au niveau de chaque domaine, les informations sont relayées grâce à un protocole de routage multicast traditionnel (DVMRP, PIM-DM, PIM-SM, etc.). Il existe donc une hiérarchie d'arbres multicast : Un arbre centré reliant les domaines entre eux, et des arbres hétérogènes au niveau des domaines.

Les travaux sur MASC/BGMP ont pour ambition de créer une architecture capable d'étendre les capacités multicast au niveau de l'Internet mondial. Ces développements sont effectués à l'IETF, conjointement dans les groupes BGMP et MALLOC et devraient aboutir⁴¹ d'ici la fin de l'an 2000.

⁴¹D'après le cahier des charges de ces groupes. Voir <http://www.ietf.org>.

Chapitre 3

Modélisation des arbres hiérarchiques centrés

Comme nous l'avons montré dans le chapitre précédent, les arbres hiérarchiques offrent des propriétés particulièrement intéressantes pour le support du multicast dans de grands réseaux. Dans ce chapitre, nous nous intéressons spécifiquement aux arbres hiérarchiques centrés, dont nous étudions les performances grâce à une étude analytique basée sur les processus ponctuels et le calcul de Palm — ces concepts étant présentés brièvement en annexe A. Cette étude a été menée en partie en collaboration avec François Baccelli et a fait l'objet d'une présentation à Infocom [3].

Dans la section suivante, le modèle utilisé pour cette étude est présenté. Les différents critères de performances sont ensuite évalués dans les sections suivantes : L'utilisation de ressources de l'arbre hiérarchique est étudié dans la section 3.2, les délais moyens à la section 3.3 et les coûts de construction à la section 3.4. Quelques remarques et conclusions viennent enfin clore ce chapitre.

3.1 Présentation du modèle

3.1.1 Nœuds du réseau

Les nœuds du réseau seront représentés par des points tirés aléatoirement sur le plan \mathbb{R}^2 . La nature aléatoire de cette approche nous permet de faire le minimum d'hypothèses sur la structure du réseau — les grandeurs étudiées dans les sections suivantes étant des moyennes par rapport à toutes les réalisations possibles du réseau.

Plus précisément, les nœuds du réseau (c.à.d. les routeurs par exemple) seront représentés par un processus de poisson π_0 homogène sur \mathbb{R}^2 d'intensité λ_0 (cf. paragraphe A.3 en annexe). Parmi tous ces nœuds, seule une proportion donnée désire participer au groupe multicast étudié. Nous utiliserons donc un processus π_0^* construit par amincissement de π_0 avec la probabilité p_0^* . Autrement dit, un tirage de Bernoulli de paramètre p_0^* est effectué sur les différents points de $\text{supp}(\pi_0)$ ¹ : seuls les points de $\text{supp}(\pi_0^*)$ seront membres du groupe multicast. Bien sûr, le processus obtenu est également homogène et son intensité est $\lambda_0^* = \lambda_0 p_0^*$. Remarquons que des modèles plus complexes, permettant de prendre en compte des phénomènes d'inhomogénéité dans la distribution spatiale seraient également possibles — bien que les calculs ultérieurs seraient bien plus complexes. Nous considérerons par mesure de simplicité des processus homogènes ; le cas de processus plus complexes est discuté à la fin de cette section.

Afin d'observer un réseau de taille finie, nous limiterons notre analyse à un ensemble W de \mathbb{R}^2 de surface p.s. finie. Autrement dit, seuls les points contenus dans cette fenêtre d'observation seront considérés. Nous reviendrons dans la suite sur cette remarque. Notons dès à présent que cette restriction spatiale définit un modèle non homogène — tous les nœuds considérés étant placés dans une fenêtre W . Cependant, nous conservons un modèle “localement homogène” au sein de W , le processus observé étant une restriction d'un processus homogène. Nous négligerons d'ailleurs quelques “effets de bord” à la frontière de cette fenêtre.

3.1.2 Points de Rendez-vous

La construction des arbres multicast centrés nécessite la définition de nœuds spécifiques appelés “points de rendez-vous”, notés RP². Comme précédemment, ces nœuds spécifiques seront représentés par des points de processus de Poisson. Si un arbre à H niveaux hiérarchiques est utilisé, il nous faut définir H processus ponctuels : $\pi_1, \pi_2, \dots, \pi_{H-1}$, d'intensités respectives $\lambda_1, \lambda_2, \dots, \lambda_{H-1}$ vérifiant

$$\lambda_0 > \lambda_1 > \dots > \lambda_{H-1}$$

Les nœuds de π_0 sont connectés à leur RP de niveau 1, c'est à dire à des points de $\text{supp}(\pi_1)$. Certains points de $\text{supp}(\pi_1)$ ont donc en charge la distribution d'informations vers d'autres routeurs, d'autres restant au contraire inactifs si ils n'ont reçu aucune demande de participation au groupe. Ceci nous permet de définir un second processus π_1^* défini par amincissement de π_1 en ne gardant que les RPs “actifs”. A leur tour, les points de $\text{supp}(\pi_1^*)$ doivent se connecter à un RP de niveau 2 pour rejoindre le groupe de sorte qu'il est possible de définir un processus

¹ $\text{supp}(\pi)$ désigne le support du processus π , c'est à dire l'ensemble des points de \mathbb{R}^2 qui sont associés à ce processus. Voir le chapitre A pour plus de précisions sur ces notations.

²RP : Rendez-vous Point.

π_2^* . Ce procédé se répète à chaque niveau hiérarchique, définissant les processus π_k^* pour tout $k = 1, 2, \dots, H - 1$.

Nous supposons dans la suite que les processus π_1, \dots, π_{H-1} sont des processus de Poisson homogènes, indépendants entre eux et indépendants de π_0 . Ce choix est justifié par le fait qu'en général, les RPs sont choisis administrativement, c.à.d. indépendamment de la distribution spatiale des membres du groupe multicast³.

Nous introduirons les probabilités de Palm associés aux processus π_0, \dots, π_{H-1} que nous noterons $\mathbb{P}_0^{\pi_0}, \dots, \mathbb{P}_0^{\pi_{H-1}}$. Ces probabilités sont définies en annexe A. Les grandeurs moyennes calculées sous ces probabilités seront notées $\mathbb{E}_0^{\pi_0}[\cdot], \dots, \mathbb{E}_0^{\pi_{H-1}}[\cdot]$.

3.1.3 Structure hiérarchique

Afin de construire l'arbre hiérarchique centré, des informations sur l'identité des différents RPs doivent être dissimulées dans le réseau. Pour que les membres du groupe ou les RPs puissent être connectés à un RP de niveau supérieur. Il s'agit en général du RP de niveau supérieur qui est le plus proche (par rapport aux distances définies par le routage unicast) ou d'un RP prédéfini administrativement pour un domaine de routage particulier.

Dans notre modèle, nous utiliserons les pavages de Poisson-Voronoi (voir chapitre B) pour définir les domaines de routage multicast. Un RP x de $\text{supp}(\pi_k)$ (pour $k = 1, \dots, H - 1$) sera en charge d'une zone délimitée par la cellule de Voronoi, centrée en x et générée par π_k (et notée $C_x(\pi_k)$ comme dans l'annexe B). Par construction,

- Un nœud de $\text{supp}(\pi_k^*)$ est connecté au nœud de $\text{supp}(\pi_{k+1})$ le plus proche (par rapport à la distance euclidienne).
- Un nœud $x \in \text{supp}(\pi_k^*)$ est connecté au RP $y \in \text{supp}(\pi_{k+1})$ si et seulement si $x \in C_y(\pi_{k+1})$.

Nous utilisons également les cellules de Voronoi pour définir notre fenêtre W et le comportement de l'arbre hiérarchique au niveau le plus haut. Nous définissons en effet le processus de Poisson homogène π_H , indépendants de π_0, \dots, π_{H-1} , et d'intensité $\lambda_H = 1$. Les RPs actifs de $\text{supp}(\pi_{H-1}^*)$ sont connectés au point de $\text{supp}(\pi_H)$ le plus proche, comme montré précédemment. Ceci définit donc un modèle comportant une infinité d'arbres hiérarchiques à H niveaux. Nous nous concentrerons sur les caractéristiques moyennes (par rapport à tous ces

³Des recherches sur des algorithmes adaptatifs permettant une bonne adaptation de l'arbre à la distribution spatiale du groupe multicast sont également en cours, en collaboration avec François Baccelli. Quelques algorithmes ont notamment été proposés dans [3].

arbres), c.à.d. à des grandeurs du type :

$$\frac{1}{\lambda_H |B|} \mathbb{E} \left[\sum_{x \in \text{supp}(\pi_H) \cap B} f(x) \right]$$

où $f(x)$ est une grandeur caractéristique de l'arbre centré en x au niveau hiérarchique le plus haut et où B est une fenêtre arbitraire (par stationnarité des processus). Comme montré au paragraphe A.4, ceci revient à considérer la cellule de Voronoi centrée à l'origine sous la probabilité de Palm $\mathbb{P}_0^{\pi_H}$ — cette cellule “typique” (sous Palm) représente donc la “fenêtre d'observation” W dont nous avons parlé précédemment.

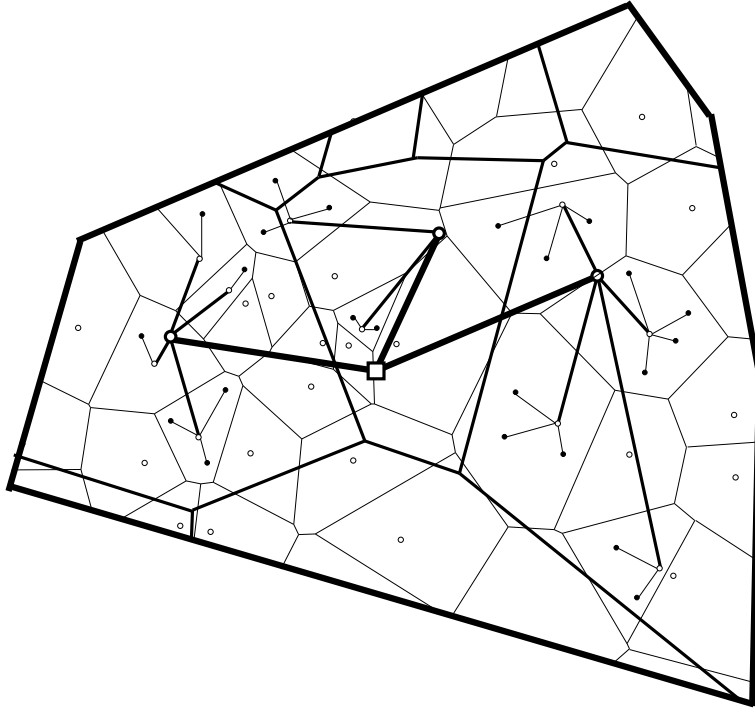


FIG. 3.1 – Une réalisation du modèle d'arbre hiérarchique utilisé.

Une représentation d'un tel modèle est donnée sur la figure 3.1 pour $H = 3$ niveaux hiérarchiques. Afin de mieux décrire la structure hiérarchique obtenue, nous introduirons les paramètres

$$m_k = \mathbb{E}_0^{\pi_{k+1}} [\pi_k(C_0(\pi_{k+1}))],$$

pour $k = 0, 1, \dots, H - 1$. Ces grandeurs représentent le nombre moyen de routeurs désirant participer au groupe multicast⁴ (pour $k = 0$) ou de RPs actifs (pour $k > 0$) dont un point de rendez-vous est responsable. Autrement dit, ce paramètre permet de mesurer la “taille”

⁴Ces routeurs ont reçus des demandes de participation au groupe via IGMP.

moyenne des domaines multicast d'un niveau hiérarchique donné. Remarquons que ces quantités peuvent être exprimées aisément en fonction des intensités des processus π_1, \dots, π_{H-1} :

Proposition 3.1

$$m_k = \frac{\lambda_k}{\lambda_{k+1}}$$

pour $k = 0, 1, \dots, H - 1$.

Preuve. Il s'agit d'une application directe de la formule d'échange de Neveu (cf. paragraphe A.6 en annexe). \square

Remarque 3.1 Remarquons également que ces grandeurs sont reliées par la relation suivante :

$$\prod_{i=0}^{H-1} m_i = \frac{\lambda_0}{\lambda_1} \frac{\lambda_1}{\lambda_2} \dots \frac{\lambda_{H-1}}{\lambda_H} = \lambda_0$$

avec la convention $\lambda_H = 1$.

Ce modèle étant défini, il est possible d'étudier quelques aspects intéressants sur les performances de ces arbres hiérarchiques. Comme nous l'avons évoqué dans le chapitre précédent, un des principaux critères d'évaluation de performance est le "coût" de l'arbre permettant d'évaluer la quantité de ressources utilisées par l'arbre de diffusion multicast. Ce critère de performance nous amène à définir la connectivité des nœuds du réseau et une fonction de coût associée à ces connections.

3.1.4 Connectivité

Nous avons jusqu'à présent parlé de connectivité entre un routeur ou un RP actif et son RP de niveau supérieur, sans faire aucune hypothèse sur la nature de cette connectivité.

Si le domaine considéré possède des capacités multicast, tous les nœuds ou RP du domaine sont pourvus de capacités de duplication. Dans ce cas, un arbre centré (ou arbre de concentration de Delaunay dans ce modèle particulier, cf. section 3.5) peut être établi pour relier les routeurs ou RPs actifs à leur point de rendez-vous. Il suffit à ces nœuds d'envoyer un message de signalisation⁵ vers le RP en suivant le routage unicast. Si deux branches se réunissent en un point donné, les capacités multicast de ce point seront mises en œuvre.

Il n'est pas toujours possible de construire des arbres de concentration à tous les niveaux hiérarchiques. En effet, la totalité des nœuds ne dispose pas toujours de capacités multicast. Dans ce cas, une technique dite de "tunnelling" est utilisée pour relier les nœuds multicast

⁵Comme c'est le cas dans PIM ou CBT — cf. chapitre précédent.

à travers un domaine unicast. Cette technique consiste à ouvrir des “tuyaux” (c.à.d. une encapsulation IP dans IP) entre un membre et son RP. L’inconvénient de cette technique est que, si deux branches se rejoignent en un point donné, l’information doit être transmise deux fois à partir du RP (vers l’une et l’autre des branches).

Dans la suite, nous considérerons le cas où les ressources multicast ne sont pas disponibles dans tout le réseau — les routeurs et les RPs actifs étant donc connectés par des “tuyaux” . La fonction de coût que nous utiliserons correspondra à la distance séparant les deux extrémités de ce tuyau. Ceci ne veut absolument pas dire que nous considérons que ces nœuds sont directement connectés : le “tuyau” passe vraisemblablement à travers plusieurs nœuds intermédiaires (suivant le routage unicast sous-jacent), mais nous mesurons le coût de ce tuyau comme étant sa longueur. Dans le cas de graphes de Delaunay et du routage markovien (cf. paragraphe B.4.2 en annexe), cette hypothèse est justifiée puisque le nombre de sauts nécessaire à la connection de deux points donnés est proportionnel à la distance les séparant (théorème de Møller).

Remarquons que le réseau multicast expérimental actuel de l’Internet, le “M-bone”, est constitué de domaines multicast interconnectés par ces fameux “tuyaux” IP-dans-IP. L’analyse que nous menons dans la section suivante peut donc être appliquée au réseau internet actuel, les nœuds (au niveau physique) et les RPs actifs (au niveaux hiérarchiques supérieurs) pouvant être vus comme les “nuages” multicast.

3.1.5 Calculs Préliminaires

Dans la suite de ce chapitre, nous aurons besoin de l’expression des intensités des processus $\pi_1^*, \dots, \pi_{H-1}^*$. Rappelons que l’intensité de π_0^* , notée $\lambda_0^* = \lambda_0 p_0^*$, est connue puisqu’il s’agit d’un paramètre de notre modèle. Similairement, nous noterons par $\lambda_1^*, \dots, \lambda_{H-1}^*$ les intensités des processus $\pi_1^*, \dots, \pi_{H-1}^*$, avec $\lambda_1^* = \lambda_1 p_1^*, \dots, \lambda_{H-1}^* = \lambda_{H-1} p_{H-1}^*$.

Notons qu’un point de rendez-vous de niveau k devient actif s’il existe au moins un membre du groupe (pour $k = 1$) ou un RP actif de niveau hiérarchique $k - 1$ (pour $k > 1$) dans la cellule de Voronoi dont il a la charge.

Proposition 3.2 *La probabilité p_k^* est donnée par l’expression suivante :*

$$p_k^* = \mathbb{P}_0^{\pi_k} \{ \pi_{k-1}^* (C_0(l\pi_k)) > 0 \}$$

Preuve. Il s’agit d’une simple application de la formule de Campbell (cf. théorème A.5 en annexe). Il est possible en effet d’écrire, pour tout $B \subset \mathbb{R}^2$,

$$\lambda_k^* |B| \stackrel{\circ}{=} \mathbb{E} \left[\int_B \mathbf{1}_{\{\pi_{k-1}^*(C_x(\pi_k)) > 0\}} d\pi_k \right] = \lambda_1 \mathbb{P}_0^{\pi_k} \{ \pi_{k-1}^* (C_0(\pi_k)) > 0 \} |B|$$

□

Il est impossible d'exprimer cette probabilité sous une forme explicite car la distribution des processus $\{\pi_k^*\}_{k=1,\dots,H-1}$ nous est inconnue. Il ne s'agit pas de processus de Poisson (contrairement aux processus $\{\pi_k\}_{k=1,\dots,H-1}$ dont ils sont issus) — les tailles de deux cellules de Voronoi voisines étant corrélées.

Dans le cas de processus de Poisson, il est possible d'obtenir une formule approchée, comme nous le montrons dans la proposition suivante.

Proposition 3.3 *Si les processus $\{\pi_k^*\}_{k=1,\dots,N-1}$ étaient des processus de Poisson indépendants de π_k , nous pourrions écrire :*

$$p_k^* \approx 1 - \left(\frac{\alpha \lambda_k}{\lambda_{k-1}^* + \alpha \lambda_k} \right)^\alpha$$

où $\alpha \approx 3.57$.

Preuve. Dans le cas de processus de Poisson,

$$\begin{aligned} p_k^* &= 1 - \mathbb{P}_0^{\pi_k} \{ \pi_{k-1}^*(C_0(\pi_k)) = 0 \} \\ &= 1 - \mathbb{E}_0^{\pi_k} [\exp(-\lambda_{k-1}^* |C_0(\pi_k)|)]. \end{aligned}$$

La distribution de la taille d'une cellule typique de Voronoi n'est pas connue explicitement. Néanmoins, nous avons à notre disposition d'excellentes approximations qui ont été obtenues par "fitting" à partir de simulations. Comme présenté dans annexe B à la page 212, $|C_0(\pi_k)|$ suit, sous $\mathbb{P}_0^{\pi_k}$, une distribution proche de

$$f(x) = \frac{b^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-bx)$$

où $b \approx \alpha \lambda_k$ et $\alpha \approx 3.57$. Après intégration, nous arrivons donc au résultat annoncé. □

Ce résultat n'est donc pas exact dans la mesure où les processus considérés ne sont pas des processus de Poisson. Néanmoins, des résultats de simulations montrent que la formule précédente donne d'excellents résultats (cf. figure 3.2) et constitue donc un très bonne approximation. Nous utiliserons cette identité dans la suite de ce chapitre.

Nous achèverons ces remarques préliminaires avec la proposition suivante sur les dérivées très particulières des intensités des RPs actifs par rapport à l'intensité des feuilles de l'arbre multicast.

Proposition 3.4

$$\frac{\partial \lambda_k^*}{\partial \lambda_0^*} = \prod_{i=1}^k (1 - p_i^*)$$

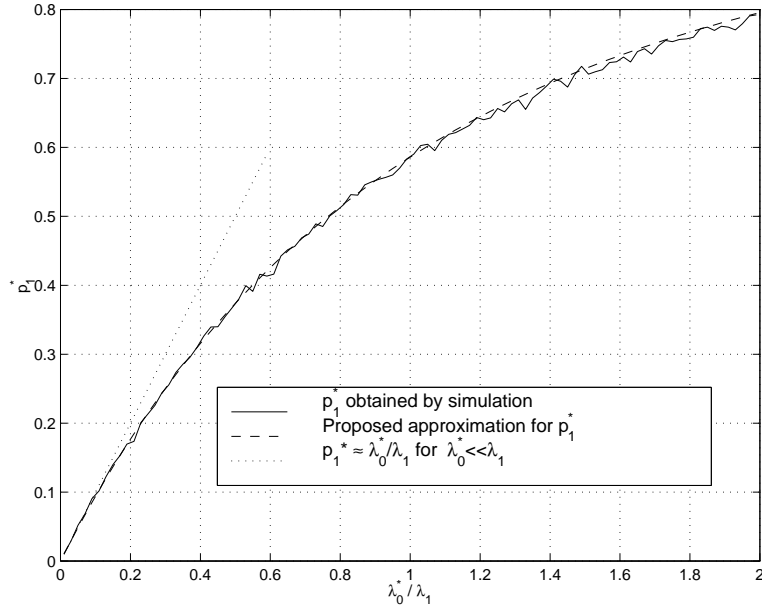


FIG. 3.2 – Validation de l'approximation proposée.

Preuve. Remarquons tout d'abord que $p^* = 1 - \left(1 + \frac{\lambda_{k-1}^*}{\alpha \lambda_k}\right)^{-\alpha}$, de sorte qu'une simple dérivation conduit à la relation de récurrence :

$$\frac{\partial \lambda_k^*}{\partial \lambda_0^*} = \frac{\partial \lambda_{k-1}^*}{\partial \lambda_0^*} (1 - p_k^*)$$

□

3.1.6 Critères de performances

Le modèle présenté nous permet d'étudier les performances des arbres hiérarchiques centrés, sous différents aspects :

- **Coût de l'arbre.** Il s'agit de la quantité de ressources utilisées pour la distribution des informations vers les différents membres du groupe multicast. Comme c'est en général le cas dans les simulations, nous utiliserons comme coût la somme des longueurs des différents liens (c.à.d des "tuyaux" dans ce cas) mis en jeu.
- **Délais.** Les délais moyens sont également des critères importants de performance pour des flux multimédia ou interactifs. Nous évaluerons ces délais comme le nombre de tuyaux traversés pour aller d'un point à un autre (ce qui revient à considérer que les différents liens ont des capacités et des états de congestion indépendants et identiquement distribués).

- **Coût de construction.** Le surcoût protocolaire et signalétique associé au maintient et à la construction de l'arbre est également un critère important de performance.

3.2 Coût de l'arbre

3.2.1 Analyse

Soit \mathcal{I} le coût moyen⁶ de l'arbre hiérarchique construit au sein de la fenêtre d'observation W et soit \mathcal{I}_k le coût moyen engendré par les tuyaux connectant les routeur RPs actifs de $\text{supp}(\pi_k^*)$ à leur RP de niveau $k + 1$ — pour $k = 0, \dots, H - 1$. Par construction, nous avons bien sûr :

$$\mathcal{I} = \sum_{k=0}^{H-1} \mathcal{I}_k$$

Avec les hypothèses effectuées (notamment sur W et l'utilisation de “tuyaux”), nous pouvons écrire :

$$\mathcal{I}_k = \mathbb{E}_0^{\pi_H} \left[\int_{C_0(\pi_H)} \int_{C_y(\pi_{k+1})} \|z - y\| \pi_k^*(dz) \pi_{k+1}(dy) \right] \quad (3.1)$$

où les notations propres aux processus ponctuels et présentées dans l'annexe A sont utilisées. Ces expressions prennent une expression fort simple en utilisant le calcul de Palm.

Proposition 3.5

$$\mathcal{I}_k = \frac{\lambda_k^*}{2\sqrt{\lambda_{k+1}}}$$

pour tout $k = 0, \dots, H - 1$.

Preuve. En appliquant la formule d'échange de Neveu (corollaire A.3 en annexe), l'équation (3.1) peut être simplifiée,

$$\mathcal{I}_k = \lambda_{k+1} \mathbb{E}_0^{\pi_{k+1}} \left[\int_{C_0(\pi_{k+1})} \|z\| \pi_k^*(z) \right]$$

Introduisons alors z_0^k le point de $\text{supp}(\pi_k)$ le plus proche de l'origine. Une application de la formule de Neveu (corollaire A.3 en annexe) permet alors d'écrire, pour tout $k = 0, \dots, H - 1$:

$$\begin{aligned} \mathcal{I}_k &= \lambda_k^* \mathbb{E}_0^{\pi_k^*} \left[\|z_0^{k+1}\| \right] \\ &= \lambda_k^* \mathbb{E}_0^{\pi_k} \left[\|z_0^{k+1}\| \mid \pi_{k-1}^*(C_0(\pi_k)) > 0 \right] \end{aligned}$$

⁶Nous utilisons un 'I', comme “Infrastructure”, pour désigner ce coût.

où la dernière relation découle de l'interprétation de la probabilité de Palm comme une probabilité conditionnelle (cf. paragraphe A.5 en annexe). Remarquons que $\|z_0^{k+1}\|$ est une fonctionnelle des processus π_{k+1} , alors que l'événement $\{\pi_{k-1}^*(C_0(\pi_k)) > 0\}$ ne dépend que des processus $\pi_k, \pi_{k-1}, \dots, \pi_1$ et π_0^* , qui sont tous indépendants de π_{k+1} par construction. Nous arrivons donc à l'identité suivante :

$$\begin{aligned} \mathcal{I}_k &= \lambda_k^* \mathbb{E}_0^{\pi_k} \left[\|z_0^{k+1}\| \right] \\ &= \lambda_k^* \mathbb{E} \left[\|z_0^{k+1}\| \right] \end{aligned}$$

où la dernière relation découle de l'indépendance de π_k et π_{k+1} . En appliquant le lemme suivant, nous arrivons finalement au résultat annoncé. \square

Lemme 3.1 *Soit z_0^k le point de $\text{supp}(\pi_k)$ le plus proche de l'origine (pour tout $k = 0, \dots, H-1$). Ce point obéit à l'identité suivante :*

$$\mathbb{E} \left[\|z_0^k\| \right] = \frac{1}{2\sqrt{\lambda_{k+1}}}$$

Preuve. D'après [90]. Notant par $\mathcal{C}(x, r)$ le disque ouvert de centre $x \in \mathbb{R}^2$ et de rayon r , il est possible d'écrire :

$$\begin{aligned} \mathbb{E} \left[\|z_0^k\| \right] &= \mathbb{E} \left[\int_E \|x\| \mathbf{1}_{\{\pi_k(\mathcal{C}(0, \|x\|)) = 0\}} \pi_k(dx) \right] \\ &= \lambda_k \int_E \|x\| \mathbb{P}_0^{\pi_k} \{ \pi_k(\mathcal{C}(-x, \|x\|)) = 0 \} dx \end{aligned}$$

où nous avons utilisé la formule de Campbell (théorème A.5 en annexe). Finalement, d'après le théorème de Slivnyak (théorème A.4 de l'annexe A),

$$\mathbb{P}_0^{\pi_k} \{ \pi_k(\mathcal{C}(-x, \|x\|)) = 0 \} = \mathbb{P} \{ \pi_k(\mathcal{C}(-x, \|x\|)) = 0 \}$$

cette dernière expression pouvant s'écrire $\mathbb{E} \left[\prod_{x \in \text{supp}(\pi_k)} \mathbf{1}_{\{x \notin \mathcal{C}(-x, \|x\|)\}} \right]$. Il suffit alors d'appliquer le théorème A.2 (page 188) pour achever ce calcul :

$$\begin{aligned} \mathbb{E} \left[\|z_0^k\| \right] &= \lambda_k \int_E \|x\| \exp(-\lambda_k \pi \|x\|^2) dx \\ &= 2\pi \lambda_k \int_0^\infty r^2 \exp(-\lambda_k \pi r^2) dr \\ &= \int_0^\infty \exp(-\lambda_k \pi r^2) dr \end{aligned}$$

\square

Corollaire 3.1 *Le coût de l'arbre hiérarchique s'exprime simplement sous la forme :*

$$\mathcal{I} = \sum_{k=0}^{H-1} \frac{\lambda_k p_k^*}{2\sqrt{\lambda_{k+1}}}$$

où les $\{p_k^*\}_{k=1, \dots, H-1}$ peuvent être calculés récursivement comme montré dans la proposition 3.3.

Le coût de l'arbre hiérarchique prend donc une expression extrêmement simple, en fonction des intensités des différents processus et de la proportion de nœuds du réseau désirant participer au groupe multicast. Il est donc possible de déterminer l'impact de la structure hiérarchique sur le coût de l'arbre. Néanmoins, les relations de récursivité entre les p_k^* ($k = 1, \dots, H-1$) ne sont guère faciles à manipuler de sorte que des évaluations numériques sont nécessaires. Dans deux cas particuliers cependant, l'expression du coût de l'arbre hiérarchique prend une forme simplifiée : il nous est alors possible d'étudier le comportement de cet arbre analytiquement. Ces deux cas particuliers seront dénommés "mode dense" et "mode épars" par référence au vocabulaire introduit dans le protocole de routage multicast PIM :

- **Mode épars.** Ce cas correspond à une distribution clairesemée des membres du groupe dans le réseau, que nous modéliserons par la relation $p_0^* \approx 0$.
- **Mode dense.** Ce cas correspond à une forte participation des nœuds du réseau au groupe multicast, que nous modéliserons par la relation $p_0^* \approx 1$.

3.2.2 Mode épars

Proposition 3.6 *En mode épars, c.à.d. sous l'hypothèse $p_0^* \rightarrow 0$, la fonction de coût de l'arbre hiérarchique prend la forme dégénérée suivante :*

$$\mathcal{I} = \sum_{k=0}^{H-1} \frac{\lambda_0^*}{2\sqrt{\lambda_{k+1}}}$$

Preuve. L'hypothèse $p_0^* \ll 1$ conduit également aux relations $p_k^* \ll 1 \forall k = 1, \dots, H-1$ (cf. proposition 3.3). De plus, nous pouvons alors écrire : $p_1^* \approx \frac{\lambda_0}{\lambda_1} p_0^*$ et par itération

$$p_k^* \approx \frac{\lambda_0^*}{\lambda_k}$$

pour tout $k = 1, \dots, H-1$. Le coût exprimé précédemment prend alors l'expression annoncée. □

Corollaire 3.2 *Dans ces conditions, il n'est pas dur de voir qu'en mode épars, le coût est minimum lorsque l'arbre ne comporte qu'un seul niveau hiérarchique ($H = 1$), c.à.d. un seul point de rendez-vous.*

La structure optimale à un seul niveau hiérarchique s'explique aisément par les arguments géométriques suivants.

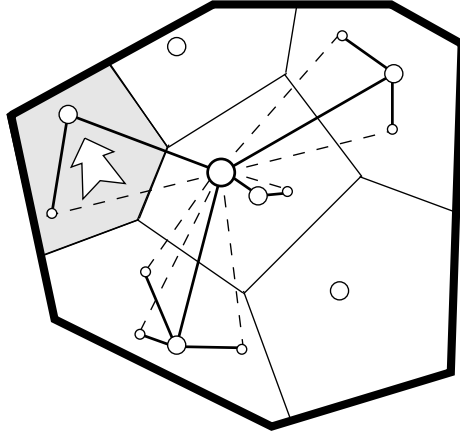


FIG. 3.3 – Influence de l'ajout d'un niveau hiérarchique supplémentaire.

Remarque 3.2 Considérons l'arbre multicast représenté sur la figure 3.3. Les lignes pointillées représentent les tuyaux IP-dans-IP reliant les membres au point de rendez-vous central lorsqu'un seul niveau hiérarchique est utilisé. Lorsqu'un niveau hiérarchique supplémentaire est rajouté (lignes épaisses), deux phénomènes distincts peuvent être observés :

- Le chemin reliant un membre au core central devient plus long puisqu'il est nécessaire de passer par des points de rendez-vous intermédiaires. C'est le cas notamment dans la cellule grisée de la figure 3.3.
- Un tuyau joignant un point de rendez-vous au RP central peut être partagé entre plusieurs membres du même domaine multicast. Ce phénomène de "concentration" permet de réduire considérablement le coût de l'arbre.

Ces deux phénomènes sont donc antagonistes du point de vue de leur influence sur la fonction de coût. Dans le cas général, il existe un compromis entre ces deux phénomènes. Dans le cas du mode épars, le premier effet est prédominant, car la probabilité que deux membres appartiennent à la même cellule est négligeable. Il est donc préférable de joindre directement le point de rendez-vous central.

3.2.3 Mode dense

Proposition 3.7 *Sous l'hypothèse $p_0^* \approx 1$, c.à.d. en "mode dense", le coût de l'arbre hiérarchique prend la forme suivante :*

$$\mathcal{I} = \sum_{k=0}^{H-1} \frac{\lambda_k}{2\sqrt{\lambda_{k+1}}}$$

Preuve. $p_0^* \approx 1$ entraîne par itération $p_k^* \approx 1$ pour $k = 1, \dots, H - 1$ et l'expression du coût (cf. corollaire 3.1) prend la forme simplifiée annoncée. \square

Cette fonction de coût n'est pas monotone par rapport aux intensités $\lambda_1, \dots, \lambda_{H-1}$ et admet un minimum que nous pouvons déterminer aisément par dérivation.

Structure Optimale.

Proposition 3.8 *A l'optimum, la structure hiérarchique obéit aux relations suivantes, pour tout $k < H - 1$,*

$$\begin{aligned} m_k &= 2\sqrt{m_{k+1}} \\ \mathcal{I}_k &= 2\mathcal{I}_{k+1} \end{aligned}$$

Preuve. Après dérivation de \mathcal{I} par rapport à λ_k , on arrive à :

$$\sqrt{\frac{\lambda_k}{\lambda_{k+1}}} = \frac{1}{2} \frac{\lambda_{k-1}}{\lambda_k}$$

qui peut être reformulée sous une forme plus parlante en utilisant les m_k exprimés dans la proposition 3.1, ou les \mathcal{I}_k exprimés au début de cette analyse. \square

Corollaire 3.3 *Pour un nombre H de niveaux hiérarchiques donnés, la structure optimale minimisant le coût de l'arbre est décrite par les identités suivantes :*

$$\begin{aligned} m_0 &= 4(4^{-H} \lambda_0)^{\frac{1}{2^H-1}} \\ m_k &= 4(4^{-H} \lambda_0)^{\frac{2^k}{2^H-1}} \quad \text{pour } k = 1, \dots, H - 1. \end{aligned}$$

Preuve. En utilisant les relations de la proposition 3.8, il est facile d'arriver à l'expression ;

$$m_k = 4 \left(\frac{m_0}{4} \right)^{2^k}$$

La remarque 3.1 de la page 43 permet alors d'achever ce calcul. \square

Corollaire 3.4 *A l'optimum, les intensités des processus représentant les points de rendez-vous sont donnés par la relation suivante :*

$$\lambda_k = 4^{-k} (4^H)^{\frac{2^k - 1}{2^H - 1}} (\lambda_0)^{\frac{2^H - 2^k}{2^H - 1}}$$

Preuve. Il suffit de remarquer que $\prod_{i=0}^{k-1} m_i = \frac{\lambda_0}{\lambda_k}$ pour $k = 0, \dots, H - 1$. \square

Les relations précédentes donnent la structure hiérarchique permettant de minimiser le coût de l'arbre hiérarchique pour un nombre de niveaux hiérarchiques donné. Le paragraphe suivant étudie spécifiquement l'influence du nombre de niveaux hiérarchiques H pour une structure optimale.

Nombre de niveaux hiérarchiques optimal.

Corollaire 3.5 *Pour une structure hiérarchique optimale, le coût de l'arbre s'écrit :*

$$\mathcal{I} = 2 (1 - 2^{-H}) (2^{-H})^{\frac{1}{2^H - 1}} \left(\sqrt{\lambda_0} \right)^{1 + \frac{1}{2^H - 1}}$$

Preuve. D'après la proposition 3.8, $\mathcal{I} = 2\mathcal{I}_0 (1 - 2^{-H})$. L'expression de \mathcal{I}_0 se déduit alors aisément de celle de m_0 donnée dans le corollaire 3.3. \square

Cette évolution est représentée sur la figure 3.5 pour différentes valeurs de λ_0 (c.à.d le nombre total de membres du groupe présents dans W). Nous pouvons voir que ces courbes admettent un minimum pour un nombre de niveaux hiérarchique bien choisi. Dans la suite, nous ferons comme ci H était un nombre réel. Ceci nous permettra de déterminer approximativement le nombre optimal de niveaux hiérarchiques à utiliser.

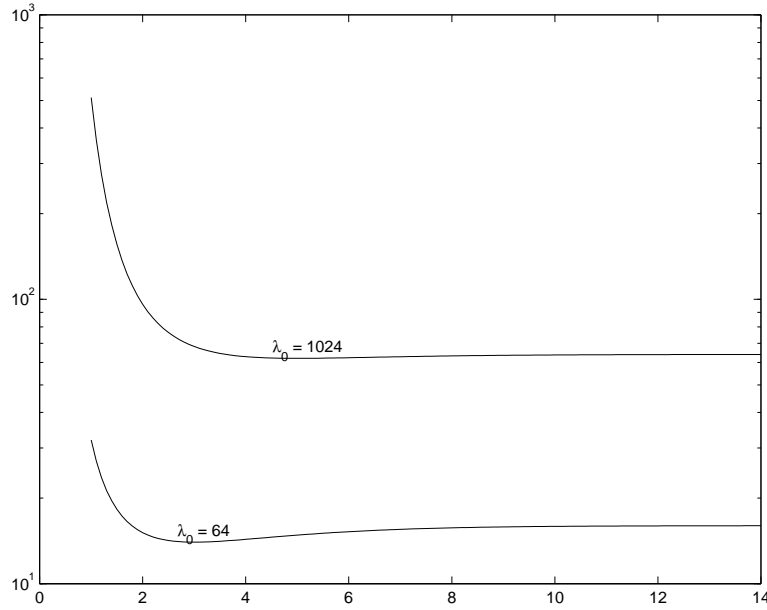
Proposition 3.9 *Le coût de l'arbre est minimisé pour la structure hiérarchique décrite plus haut et pour un nombre de niveaux hiérarchiques proche de :*

$$H^* = \frac{1}{2} \log_2 (\lambda_0)$$

Preuve. Posant $K = 2^H - 1$, on trouve

$$\mathcal{I} = 2 \frac{K}{K + 1} (K + 1)^{-\frac{1}{K}} \left(\sqrt{\lambda_0} \right)^{1 + \frac{1}{K}}$$

Par une simple dérivation de \mathcal{I} , on arrive alors à la relation $K + 1 = \sqrt{\lambda_0}$. \square

FIG. 3.4 – Coût de l'arbre hiérarchique en fonction de H .

Corollaire 3.6 *La structure optimale est approximativement décrite par les égalités suivantes, pour tout $k = 0, \dots, H - 1$:*

$$\begin{aligned} m_k &= 4 \\ \lambda_k &= \lambda_0 4^{-k} \end{aligned}$$

Preuve. Il suffit de remarquer que

$$m_k = \left[2 \left(\frac{\sqrt{\lambda_0}}{K+1} \right)^{\frac{2^k}{K}} \right]^2$$

et $K + 1 = \sqrt{\lambda_0}$ à l'optimum... □

Remarques sur les résultats obtenus.

Remarque 3.3 L'optimum est donc atteint pour un grand nombre de niveaux hiérarchiques, avec des domaines de routage multicast de très faibles tailles. Une telle structure hiérarchique est difficilement réalisable dans un réseau réel. Néanmoins, les courbes de la figure 3.4 semblent indiquer qu'il suffit d'un nombre extrêmement limité de niveaux hiérarchiques pour limiter de manière significative le coût de l'arbre. On voit même que la courbe $H \mapsto \mathcal{I}(H)$ est extrêmement "plate" autour de l'optimum. Même pour des groupes multicast de très grande

taille, le coût n'évolue plus significativement pour $H > 3$ — ce qui est tout à fait compatible avec le nombre de niveaux hiérarchiques de l'Internet actuel.

Remarque 3.4 Lorsqu'un faible nombre de niveaux hiérarchiques, nous avons pu déterminer que la structure hiérarchique optimale requiert de grands domaines multicast aux niveaux hiérarchiques supérieurs, et de petits domaines aux niveaux inférieurs (cf. proposition 3.8). La figure 3.5 représente un arbre hiérarchique possédant cette structure optimale pour trois niveaux hiérarchiques et un nombre total de nœuds présents dans W égal à $\lambda_0 = 10^3$.

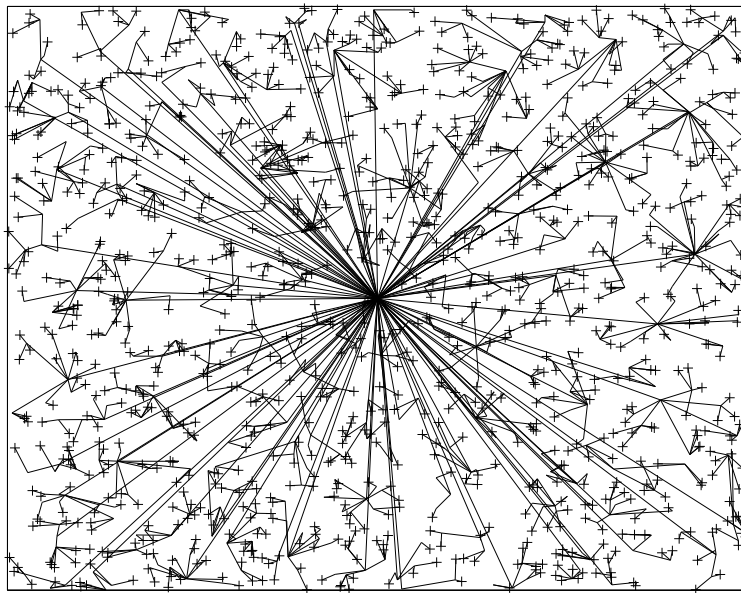


FIG. 3.5 – Exemple de structure hiérarchique minimisant le coût de l'arbre.

3.2.4 Extensions

Cas général

Dans les paragraphes précédents, nous avons étudié les structures hiérarchiques optimales permettant de minimiser le coût de l'arbre de diffusion multicast dans des cas particuliers (mode “dense” et mode “épars”). Nous avons notamment pu mettre en évidence qu'un nombre extrêmement limité de niveaux hiérarchiques est suffisant pour obtenir une bonne utilisation du réseau dans le cas “dense”. Or il faut bien comprendre que le “mode dense” constitue un pire cas. C'est en effet lorsque tous les nœuds du réseau désirent participer au groupe multicast que les phénomènes de concentration sont les plus importants. C'est donc bien dans ce cas que le nombre de niveaux hiérarchiques à utiliser doit être maximum, pour encourager ces concentrations au niveau des RPs intermédiaires. En résumé, dans le cas général (c.à.d

pour une proportion p_0^* de membres du groupe multicast quelconque), le nombre de niveaux hiérarchiques à utiliser reste toujours très limité.

Fonction de Coût.

Dans les calculs précédents, nous avons considéré une fonction de coût proportionnelle à la longueur des tuyaux. Il s'agit en effet d'un choix désormais classique pour l'analyse de performances des protocoles de routage multicast, dans les outils de simulation. Dans le cas de graphes de Delaunay et de chemins markoviens (voir paragraphe B.4), cette hypothèse est également justifiée par le théorème de Møller : Le nombre de liens physiques traversés étant, en moyenne, proportionnel à la distance euclidienne séparant les deux extrémités. Le théorème B.2 de l'annexe B permet également de relier la somme des longueurs des liens physiques utilisés avec la longueur euclidienne du "tuyau", ces deux quantités étant asymptotiquement proportionnelles d'un facteur $\frac{4}{\pi}$ lorsque les deux extrémités sont suffisamment éloignées.

Nous aurions pu néanmoins choisir une fonction de coût plus générique. En fait, notre modèle permet d'obtenir des expressions analytiques pour toute fonction de coût du type

$$c(l) = a.l^b$$

où l est la longueur euclidienne du "tuyau" considéré. La seule difficulté résiderait dans la généralisation du lemme 3.1 pour le calcul de $\mathbb{E}[a||z_0^k||^b]$ — or ce calcul a déjà été effectué lors de précédents travaux du CNET et de l'INRIA sur la géométrie aléatoire (voir [87]).

Les structures optimales mises en évidence dans l'étude précédente engendrent de fortes concentrations autour des points de rendez-vous des niveaux hiérarchiques supérieurs. Ces structures ne peuvent donc pas toujours être réalisées si des contraintes existent sur le nombre maximal de "tunnels IP" qu'un RP peut supporter. Il est néanmoins possible de reprendre l'analyse précédente en effectuant une minimisation de la fonction de coût avec des contraintes sur les grandeurs :

$$c_k = \mathbb{E}_0^{\pi_k} [\pi_{k-1}^*(C_0(\pi_k))] < C_k$$

par exemple, avec $k = 1, \dots, H$. Notons que d'après la formule d'échange de Neveu, $c_k = \frac{\lambda_{k-1}^*}{\lambda_k}$ pour tout $k = 1, \dots, H$.

Homogénéité.

Nous avons dû, dans les calculs précédents, considérer des processus homogènes afin de simplifier notre analyse. Les membres du groupe sont alors distribués de manière homogène dans le réseau. Notre modèle ne permet donc pas d'étudier les cas où les membres d'un groupe seraient cloisonnés dans certaines parties du réseau uniquement. L'utilisation de modèle

inhomogènes restent néanmoins difficile, car tous les résultats sur le calcul de Palm, énumérés en annexe A, ne pourraient être utilisés tels quels⁷. Des extensions utilisant des processus de Poisson doublement stochastiques (ou encore appelés processus de Cox) paraissent par contre prometteuses. Comme présenté dans [94], ces processus sont stationnaires (et permettent donc l'utilisation du calcul de Palm) tout en permettant de prendre en compte des phénomènes d'inhomogénéité. Des études sont en cours pour l'utilisation de ce type de processus.

Hypothèse “tunnelling”.

L'étude précédente a été effectuée sous l'hypothèse “tunnelling”, c'est à dire pour modéliser un réseau où les capacités multicast sont rares. Les membres du groupe et les points de rendez-vous sont donc connectés par des “tuyaux” (encapsulation IP dans IP ou GRE⁸).

Les capacités multicast sont de plus en plus déployées dans le réseau Internet et les “tuyaux” sont voués à disparaître à long terme. Si les capacités multicast sont disponibles dans tout le réseau (et non pas seulement dans certains “nuages” multicast), l'analyse menée ci-avant n'est plus valable. En effet, les phénomènes de concentration (partage d'une partie d'une branche entre plusieurs feuilles) ne pouvaient avoir lieu qu'au niveau des points de rendez-vous, alors que ces phénomènes peuvent prendre place au niveau de n'importe quel nœud dans un réseau multicast.

A la fin de ce chapitre, un nouvel objet, nommé “arbre de concentration de Delaunay”, est défini afin d'étudier les phénomènes de concentration pouvant intervenir au niveau des nœuds intermédiaires. Cet objet s'est hélas révélé fort complexe et nous n'avons donc pas pu achever l'étude sur les performances des arbres hiérarchiques utilisant des arbres de concentration.

Des résultats préliminaires de simulations [5, 7] permettent néanmoins de donner quelques pistes. Les résultats obtenus par simulation sur des graphes aléatoires sont très différents suivant le générateur de graphe utilisé. Pour des graphes purement aléatoires ou aucune notion de localité n'existe — deux points quelconques ont une probabilité p d'être connectés, quelque soit la distance les séparant — il s'est avéré que l'optimum est atteint pour un seul niveau hiérarchique seulement, et ce quelque soit la densité du groupe multicast. En revanche, lorsque les graphes générés présentent une certaine localité — la probabilité que deux points soient reliés est une fonction décroissante de la distance qui les séparent — l'optimum est atteint pour un grand nombre de niveaux hiérarchiques, afin de faciliter l'effet de concentration. Nous avons d'ailleurs trouver une structure optimale très proche de celle mise en évidence ci-avant

⁷Le projet RNRT GEORGES (CNET, INRIA, ENST) vient d'être monté pour travailler sur le développement et l'utilisation d'outils basés sur les processus ponctuels inhomogènes.

⁸GRE : Generic Routing Encapsulation.

avec :

$$H^* = \log_2(N)$$

$$m_0 = \dots = m_{H-1} = 2$$

où N est le nombre de nœuds du réseau considéré. Nous renvoyons le lecteur au rapport rédigé à la suite de ces travaux [7] pour plus de détails.

3.3 Délais

Le délai est également un critère extrêmement important de performance, d'autant plus qu'il existe souvent un compromis à trouver entre coût de l'arbre et délai — comme nous l'avons vu dans le chapitre précédent. Néanmoins, la notion de délai est assez difficile à définir dans un contexte “many-to-many”, c'est à dire lorsque chaque membre du groupe multicast est à la fois un récepteur et un émetteur potentiel.

3.3.1 Diffusion à partir du RP central.

Remarquons qu'une diffusion basée à partir du point de rendez-vous central est assez facile à étudier : H tuyaux IP-dans-IP sont utilisés pour joindre un membre quelconque du groupe. Si nous considérons les délais rencontrés à la traversée d'un “tuyau” proportionnels à sa longueur euclidienne, nous obtenons la variable suivante :

$$\mathcal{D}_c = \sum_{k=1}^H \mathbb{E} \left[\|z_0^k\| \right]$$

où $\mathbb{E}[\|z_0^k\|]$ représente la longueur moyenne d'un tuyau reliant un point de $\text{supp}(\pi_{k-1}^*)$ à son RP le plus proche. Comme montré dans le lemme 3.1, ce délai peut être exprimé simplement en fonction des intensités des différents processus.

Proposition 3.10 *Le délai subi du RP vers un membre quelconque du groupe est, en moyenne :*

$$\mathcal{D}_c = \sum_{k=1}^H \frac{1}{2\sqrt{\lambda_k}}$$

Remarque 3.5 Les délais de diffusion sont donc bien sûr minimisés pour $H = 1$, c.à.d. un seul niveau hiérarchique — le délais étant alors de $\frac{1}{2}$. Comme nous l'avons déjà remarqué (cf. remarque 3.2 à la page 50), l'ajout de niveaux hiérarchiques supplémentaires rallonge le chemin entre une feuille et la racine de l'arbre, car ce chemin doit passer par des RPs intermédiaires.

Dans la suite de cette brève analyse des délais, nous étudions l'évolution du délais de transmission en fonction du nombre de niveaux hiérarchiques suivant la structure utilisée. En particulier, il est très intéressant de pouvoir observer l'évolution de ce délais pour une structure minimisant le coût de l'arbre (étudié dans la section précédente) dans le cas "dense".

Délais observés pour une structure minimisant le coût de l'arbre.

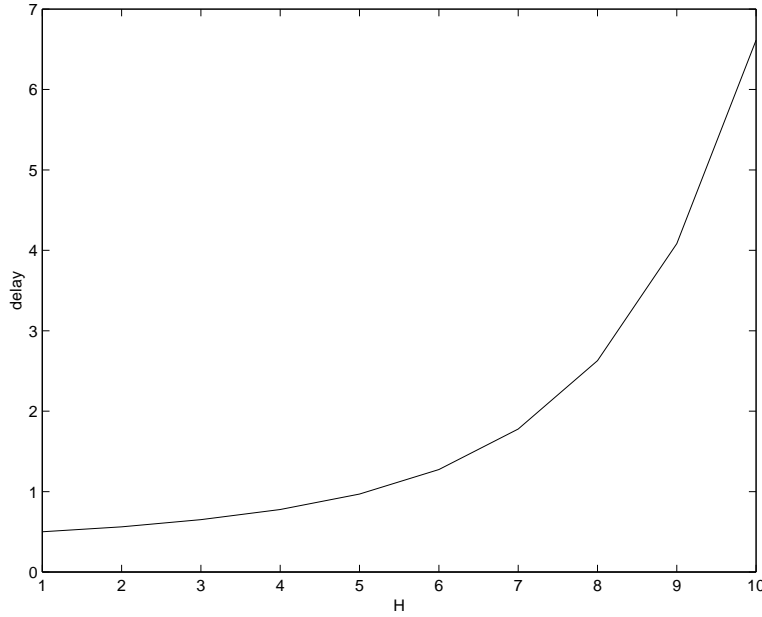


FIG. 3.6 – Évolution des délais de diffusion en fonction de H .

Il est facile de tracer l'évolution de \mathcal{D}_c en fonction du nombre de niveaux hiérarchiques H en utilisant les expressions des intensités $\{\lambda_k\}_{k=1,\dots,H-1}$ à l'optimum (cf. corollaire 3.4 à la page 52). La figure 3.6 représente cette évolution pour $\lambda_0 = 1024$. Nous pouvons observer une évolution de type géométrique : Les délais évoluent très peu lorsque H est assez faible (en fait pour $H < H^*$). Les délais deviennent par contre très importants pour de grandes valeurs de H . Les propositions suivantes permettent de mieux comprendre cette évolution.

Proposition 3.11 *Pour $H \approx H^*$ et $\lambda_0 \gg 1$, le coût de diffusion est donné par $\mathcal{D}_c \approx 1$, c.à.d. le double du coût de diffusion rencontré pour un seul niveau hiérarchique.*

Preuve. Si à l'optimum (par rapport à la fonction de coût), nous savons que les intensités obéissent aux égalités suivantes : $\lambda_k \approx \lambda_0 4^{-k}$ de sorte que la fonction de délai s'écrit :

$$\mathcal{D}_c \approx \frac{2^H - 1}{\sqrt{\lambda_0}}$$

Il suffit alors de remarquer que l'optimum est atteint pour $2^H = \sqrt{\lambda_0}$ d'où

$$\mathcal{D}_c \approx 1 - \frac{1}{\sqrt{\lambda_0}}$$

□

Proposition 3.12 *Pour un très grand nombre de niveaux hiérarchiques ($H \gg 1$), la fonction de délai évolue géométriquement avec H puisque*

$$\mathcal{D}_c \approx \frac{2^H - 1}{\sqrt{\lambda_0}}$$

Preuve. Pour $H \gg 1$, les expressions des λ_k minimisant le coût (cf. corollaire 3.4) peuvent être approchées au premier ordre par $\lambda_k \approx \lambda_0 4^{-k}$ de sorte que la fonction de délai prend la forme approchée annoncée. □

Remarque 3.6 Le compromis entre coût et délai est donc très bien mis en valeur dans ce cas puisque les fonctions de coût (cf. figure 3.5) et de délai (cf. figure 3.6) sont respectivement décroissantes et croissantes pour $H < H^*$. Par contre, les résultats précédents tendent à montrer qu'il est préférable d'utiliser un faible nombre de niveaux hiérarchiques : Le coût est réduit très significativement avec un faible niveau hiérarchique et n'évolue que très peu ensuite. De plus, les délais ne sont pas trop affectés lorsque le nombre de niveaux hiérarchiques reste faible (c.à.d. $H < H^*$). Enfin, l'utilisation d'un trop grand nombre de niveaux hiérarchiques peut être catastrophique, car un tel choix pourrait à la fois conduire à un coût supérieur à l'optimum et surtout à des délais de diffusion inacceptables (croissance géométrique des délais avec H pour $H \gg 1$).

3.3.2 Délais à partir d'un point quelconque

Lorsque l'arbre hiérarchique est utilisé par des applications "many-to-many", c'est à dire pour lesquelles tout membre du groupe est à la fois un récepteur et une source potentielle. Les délais sont alors plus difficiles à appréhender. Nous pouvons néanmoins utiliser la fonction Q (cf. proposition B.10 en annexe B) pour mesurer la distance à parcourir entre deux points quelconques du réseau.

Proposition 3.13 *Soit $N(\rho)$ le niveau hiérarchique le plus haut qu'il est nécessaire d'emprunter pour joindre deux points quelconques distants de ρ . $N(\rho)$ suit alors la distribution suivante :*

$$\mathbb{P}\{N(\rho) = k\} = Q\left(\rho\sqrt{\lambda_k}\right) \prod_{i=1}^{k-1} \left(1 - Q\left(\rho\sqrt{\lambda_i}\right)\right) \quad \text{pour } k = 1, \dots, H - 1,$$

$$\mathbb{P}\{N(\rho) = H\} = \prod_{i=1}^{H-1} \left(1 - Q(\rho\sqrt{\lambda_i})\right)$$

Preuve. Par définition, deux points distants de ρ appartiennent à la même cellule de Voronoi engendré par π_k avec la probabilité $Q(\rho\sqrt{\lambda_k})$ (voir proposition B.10 en annexe). Le résultat en découle directement. \square

Corollaire 3.7 *Il est nécessaire d'emprunter $2\mathbb{E}[N(\rho)]$ tuyaux en moyenne pour rejoindre deux points distants de ρ . Si le délais de traversée d'un tuyau est supposé proportionnel à sa longueur euclidienne, le délais subis pour rejoindre ces deux points est donné par l'identité suivante :*

$$\mathcal{D}(\rho) = \sum_{k=1}^H \frac{1}{\sqrt{\lambda_k}} \prod_{i=1}^{k-1} \left[1 - Q(\rho\sqrt{\lambda_i})\right]$$

Preuve. Il suffit de remarquer que la longueur moyenne d'un tuyau de niveau k est donné par $\mathbb{E}[|z_0^k|] = \frac{1}{2\sqrt{\lambda_k}}$. On arrive alors à

$$\mathcal{D}(\rho) = \sum_{k=1}^H \mathbb{P}\{N(\rho) = k\} \sum_{i=1}^k \frac{1}{\sqrt{\lambda_i}}$$

qui peut être simplifié en utilisant les expressions de $\mathbb{P}\{N(\rho) = k\}$. \square

Délais observés pour une structure optimale.

Sur la figure 3.7 est représenté l'évolution du délais subis en fonction de la distance ρ séparant deux points donnés et en fonction du nombre de niveaux hiérarchiques. Les intensités des processus ont été calculées en utilisant les expressions de la section 3.2 pour une structure minimisant le coût de l'arbre hiérarchique dans le cas "dense" et pour $\lambda_0 = 1024$.

Le délai subi pour joindre un membre du groupe situé à proximité (c.à.d. pour $\rho \ll 1$) diminue avec le nombre de niveaux hiérarchiques H . En effet, lorsque H augmente, la taille d'un domaine multicast de niveau physique diminue, de sorte que la distance à parcourir pour joindre le point de rendez-vous est plus faible. Il est en effet facile de vérifier que pour $\rho \ll 1$, $\mathcal{D} \approx \frac{1}{\sqrt{\lambda_1}}$.

Lorsqu'au contraire la destination observée est très éloignée de l'émetteur (c.à.d. $\rho \gg 1$), le délais est de plus en plus mauvais lorsque le nombre de niveaux hiérarchiques augmente, car le chemin menant à ce point doit faire de nombreux détours par des RPs intermédiaires. Lorsque $\rho \gg 1$, on observe en effet $\mathcal{D} \approx \sum_{k=1}^H \frac{1}{\sqrt{\lambda_k}}$. Cette valeur vaut d'ailleurs $\mathcal{D} \approx \frac{2}{\sqrt{\lambda_0}} (2^H - 1)$ pour $H \gg H^*$ comme nous l'avons montré dans la section précédente (cf. délais de diffusion à partir du point de rendez-vous central).

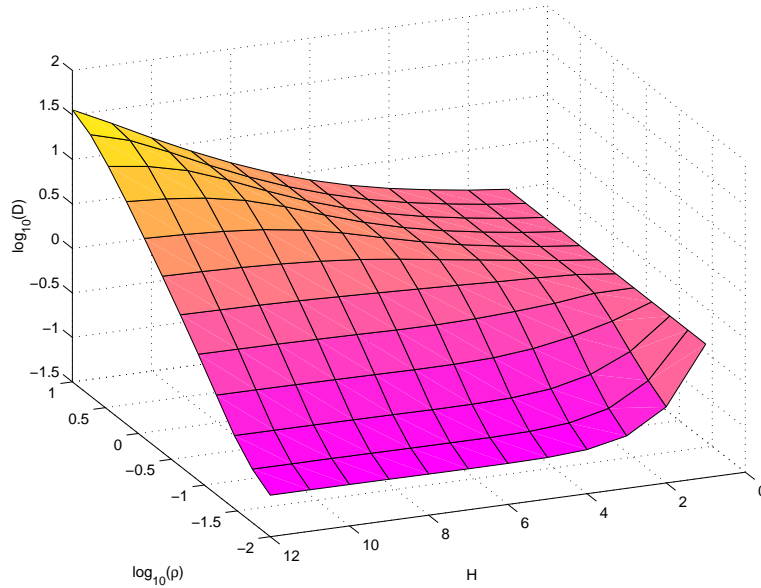


FIG. 3.7 – Évolution du délais subi en fonction de H et de la distance ρ .

Remarque 3.7 Les résultats précédents montrent qu’il est préférable de ne pas choisir un nombre trop important de niveaux hiérarchiques — comme nous l’avons déjà mis en évidence dans le cas d’une diffusion à partir du RP central. Lorsque $\rho \ll 1$, les délais ne diminuent plus significativement dès $H = 3$, et ce même pour de grand réseaux. En revanche, les délais augmentent géométriquement (pour $H \gg H^*$) avec le nombre de niveaux hiérarchiques utilisés, lorsque $\rho \gg 1$.

3.3.3 Remarques

L’étude précédente a été effectuée sous l’hypothèse d’un réseau aux ressources multicast rares, utilisant des “tuyaux”. Les résultats présentés dans cette section ne sont donc valables que dans ce cas précis. Lorsque les capacités multicast sont disponibles dans la totalité du réseau, les phénomènes de concentration peuvent avoir lieu au niveau de tous les nœuds du réseau — au lieu des seuls RPs. Il est alors possible qu’un message envoyé sur l’arbre ne passe par aucun RP pour atteindre certaines destinations — un nœud intermédiaire effectuant une diffusion locale. Les délais peuvent donc être significativement plus court lorsque $\rho \ll 1$. Dans le cas $\rho \gg 1$ au contraire, l’intuition nous pousse à penser que les phénomènes de concentration jouent un rôle négligeable, le chemin devant de toute façon passer par les RPs à chaque niveau hiérarchique.

3.4 Coût de construction

Dans les deux sections précédentes, nous nous sommes intéressés aux performances (ressources utilisées et délais) des arbres hiérarchiques une fois ceux-ci construits. Nous n'avons par contre pas encore considéré la construction de ces arbres. Les aspects algorithmiques liés à la construction de ces arbres ne seront pas traités dans ce chapitre (nous en avons brièvement parlé dans le chapitre précédent). On peut néanmoins remarquer que la construction et le maintien de ces arbres nécessitent des échanges de messages de signalisation — quelque soit la procédure utilisée pour définir les domaines multicast et faire connaître la localisation des différents points de rendez-vous. En particulier, un nœud du réseau désirent joindre le groupe multicast doit le faire savoir à son point de rendez-vous local. De plus, si ce RP n'appartient pas déjà à l'arbre de diffusion (c.à.d s'il n'est pas encore "actif"), ce dernier doit alors joindre à son tour le RP de niveau supérieur et ainsi de suite. Notre modèle permet de calculer explicitement le nombre moyen de messages échangés et d'autres grandeurs associées à la construction de ces arbres.

3.4.1 Coût global

Nous considérons dans cette section le coût global engendré par la construction d'un arbre hiérarchique joignant un nombre donné de feuilles. Plus exactement, étant donné λ_0^* , le nombre moyen de nœuds du réseau désirent participer au groupe multicast, nous nous intéressons au coût de construction de l'arbre joignant ces différentes feuilles.

Soit N_G^k le nombre moyen de messages échangés entre les feuilles (pour $k = 0$) ou les RPs de niveau k et leur point de rendez-vous le plus proche — où l'on suppose qu'un message suffit pour qu'un membre ou un RP joigne son RP de niveau supérieur. Il est également possible de prendre en compte la distance parcourue par chacun de ces messages⁹, en définissant un coût global \mathcal{C}_G comme la somme des distances parcourues par tous les messages de signalisation échangés. Ces grandeurs sont faciles à déterminer en utilisant les résultats du paragraphe 3.1.5 et de la section 3.2.

Proposition 3.14

$$\begin{aligned} N_G^k &= \lambda_k^* \quad \text{pour } k = 0, \dots, H-1 \\ \mathcal{C}_G &= \sum_{k=0}^{H-1} \frac{\lambda_k^*}{2\sqrt{\lambda_{k+1}}} \end{aligned}$$

avec $\lambda_k^* = \lambda_k p_k^*$, les $\{p_k^*\}_{k=1, \dots, H}$ étant exprimés à la proposition 3.3.

⁹Rappelons nous que ce message passe à travers des liens intermédiaires, dont le nombre est proportionnel à la distance dans le cas de chemins markoviens sur le graphe de Delaunay.

Remarque 3.8 Nous pouvons donc observer qu’avec la définition choisie, le coût de construction est identique au coût \mathcal{I} étudié dans la section 3.2. Il est donc possible de minimiser le coût de construction avec une structure hiérarchique spécifique.

La plupart des groupes multicast ne sont pas figés — de nouveaux membres pouvant rejoindre ou quitter la session à tout moment. Il est donc important de s’assurer que l’arbre diffusant peut s’adapter efficacement à l’arrivée ou au départ de membres du groupe multicast. Dans la suite de ce chapitre, nous nous intéressons au coût de maintenance de l’arbre hiérarchique lorsqu’un nouveau rejoint le groupe.

3.4.2 Ajout d’une feuille

Définissons \mathcal{C}_Δ , le coût causé par l’ajout d’une nouvelle feuille, comme la somme des longueurs des nouveaux “tuyaux” créés pour interconnecter la nouvelle feuille à l’arbre. Dans le cas de chemins markoviens sur le graphe de Delaunay (cf. paragraphe B.4.2 en annexe B), ce coût correspond donc, à un facteur multiplicatif $\frac{4}{\pi}\sqrt{\lambda_0}$ près, au nombre de liens traversés par des messages de signalisation — des informations devant être stockées sur ces liens afin de construire l’arbre.

Proposition 3.15 *La charge signalétique causé par l’ajout d’une feuille est donné par l’expression suivante :*

$$\mathcal{C}_\Delta = \sum_{k=1}^H \frac{1}{2\sqrt{\lambda_k}} \prod_{i=1}^k (1 - p_i^*)$$

où les $\{p_k^*\}_{k=1, \dots, H-1}$ peuvent être calculés récursivement comme montré dans la proposition 3.3.

Preuve. Il faut tout d’abord joindre son RP local situé à une distance moyenne de $\mathbb{E}[||z_0^1||] = \frac{1}{2\sqrt{\lambda_1}}$ (voir section 3.2). Si ce point de rendez-vous n’appartient pas déjà à l’arbre, avec la probabilité $(1 - p_1^*)$, il est nécessaire que ce RP joigne à son tour le RP de niveau supérieur le plus proche — à distance moyenne de $\mathbb{E}[||z_0^2||] = \frac{1}{2\sqrt{\lambda_2}}$. Et ainsi de suite jusqu’au niveau hiérarchique le plus haut. \square

Remarque 3.9 Il est possible de retrouver ce résultat de manière plus formelle. Soit U un point aléatoire tiré uniformément sur notre fenêtre d’observation W . Si $\mathcal{C}_G(\pi_0^*)$ est le coût de construction de l’arbre hiérarchique reliant les points de $\text{supp}(\pi_0^*)$, nous nous intéressons donc à la grandeur :

$$\mathcal{C}_\Delta = \mathcal{C}_G(\pi_0^* + \delta_U) - \mathcal{C}_G(\pi_0^*)$$

avec la convention¹⁰ $\text{supp}(\pi_0^* + \delta_U) = \text{supp}(\pi_0^*) \cup \{U\}$. Il est alors possible d'utiliser les résultats sur la perturbation des processus de Poisson (cf. paragraphe A.3.4 en annexe) : Nous reconnaissons en effet :

$$c_\Delta = \frac{\partial \mathcal{C}_G}{\partial \lambda_0^*}$$

En utilisant l'expression de \mathcal{C}_G (cf. proposition 3.14), nous arrivons à l'identité

$$c_\Delta = \sum_{k=1}^H \frac{1}{2\sqrt{\lambda_k}} \frac{\partial \lambda_{k-1}^*}{\partial \lambda_0^*}$$

La proposition 3.4 permet alors de retrouver l'expression annoncée précédemment.

Résultats pour une structure spécifique.

Le coût engendré par l'ajout d'une nouvelle feuille dépend donc de la densité p_0^* du groupe multicast (on s'attend bien sûr à voir ce coût diminuer lorsque la densité augmente) et de la structure hiérarchique utilisée. Dans la suite, nous considérons le cas d'une structure optimale mise en évidence dans la section 3.2.3, c'est à dire minimisant les ressources utilisées par l'arbre hiérarchique dans le cas "dense". Sur la figure 3.8 est représenté le coût signalétique causé par l'ajout d'une nouvelle feuille en fonction de la densité p_0^* et du nombre de niveaux hiérarchiques utilisés.

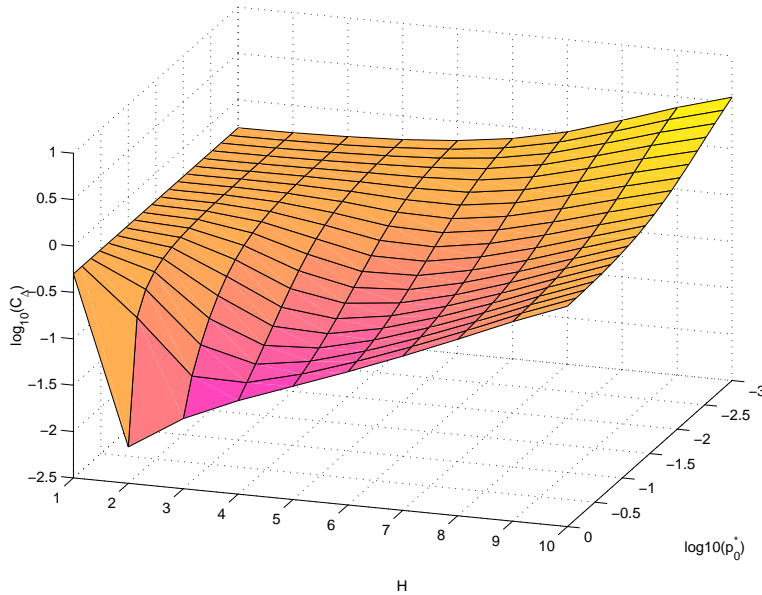


FIG. 3.8 – Coût de construction causé par l'ajout d'une nouvelle feuille.

¹⁰Nous utilisons une notation liée aux processus ponctuels, π_0^* étant une mesure sur \mathbb{R}^2 — voir annexe A.

3.5 Arbres de concentration de Delaunay

Nous finissons cette étude consacrée aux arbres hiérarchiques par une brève présentation d'un nouveau modèle mathématique : l'arbre de concentration de Delaunay. Cet objet permet en effet de modéliser un réseau centré (c.à.d. de type CBT) dans le cas où tous les nœuds sont pourvus de capacités multicast. Cet objet pourrait donc permettre d'étudier les arbres centrés tout d'abord, et d'étudier les arbres hiérarchiques centrés ensuite, lorsque l'hypothèse de "tunnelling" est relâchée (cf. paragraphe 3.1.4 à la page 43).

3.5.1 Présentation

Soit π un processus ponctuel de Poisson homogène (voir annexe A) sur \mathbb{R}^2 et soit λ son intensité (supposée finie). On s'intéresse dans toute la suite à la restriction de ce processus sur une fenêtre d'observation finie W_r . Par mesure de simplicité, nous supposons parfois que W_r est un disque de surface unitaire, c'est à dire de rayon $r = \frac{1}{\sqrt{\pi}}$. Le choix d'un disque est justifié par l'homogénéité du processus.

Les points de $\text{supp}(\pi)$ représentent les nœuds du réseau (c.a.d. les routeurs or commutateurs). Nous supposons que ces nœuds sont interconnectés par le graphe de Delaunay engendré par π (défini dans le chapitre B).

Un second processus ponctuel, π^* , peut être construit à partir de π par amincissement, avec une probabilité de Bernouilli p . Il est clair, qu'alors, π^* est également un processus ponctuel de Poisson d'intensité $\lambda^* = p\lambda$. Les points $x \in \pi^*$, représentent les nœuds du réseau qui souhaitent participer au groupe multicast. On suppose qu'un arbre de type CBT est utilisé, de sorte que ces nœuds vont être connectés à l'origine, qui représente le "point de rendez-vous".

Chaque "branche" connectant au core un nœud de π^* passe par des nœuds intermédiaires, selon le chemin markovien décrit dans [89] (et brièvement décrit en annexe dans le paragraphe B.4.2). Certains nœuds sont donc impliqués malgré eux dans ce groupe multicast, car ils devront garder des états sur ce groupe afin d'assurer l'acheminement des paquets de ce groupe multicast.

Nous supposons enfin que tous les nœuds du réseau considérés possèdent des capacités multicast : Lorsqu'une nouvelle feuille cherche à joindre le RP, il est possible que la nouvelle branche passe par un nœud qui appartient déjà à l'arbre (une ou plusieurs branches passent déjà par ce point). Dans ce cas, la nouvelle branche se termine au niveau de ce point de croisement, qui utilisera ses capacités de duplication pour faire parvenir les informations d'une branche vers les autres.

Sur la figure 3.9 est représenté une réalisation d'un arbre de concentration sur le graphe de Delaunay pour une fenêtre rectangulaire, avec $\lambda = 100$ et $p = 0.2$.



FIG. 3.9 – Arbre de concentration de Delaunay.

L'étude analytique de cet objet s'est avérée fort complexe et nous n'avons pu obtenir de résultats analytiques probants dans l'immédiat. Nous présentons néanmoins dans la suite quelques résultats obtenus par simulation qui permettent de mieux appréhender cet objet.

3.5.2 Quelques propriétés

Participation à l'arbre

Sur la figure 3.10 est représenté la proportion de nœud du réseau participant (volontairement ou involontairement) au groupe multicast en fonction de p pour $\lambda = 100$ et $|W| = 1$.

Le résultat important est bien sûr l'évolution non linéaire de cette participation, et ce grâce aux phénomènes de concentration (c.à.d. deux branches se réunissant en un point donné). L'évolution de cette courbe est quasi-linéaire pour $p \ll 1$, car la probabilité que des branches se croisent est très faible (il y a trop peu de membres du groupe multicast dans le réseau). En suite, l'augmentation de la participation diminue très fortement grâce aux phénomènes de concentration.

Coût de l'arbre

L'évolution du coût de l'arbre en fonction de p est représenté sur la figure 3.11 pour $\lambda = 100$ et $|W| = 1$. Dans ce cas encore, ce coût n'évolue pas linéairement — contrairement à un arbre utilisant des “tunnels” où le coût serait proportionnel à λp , comme montré dans la section 3.2.

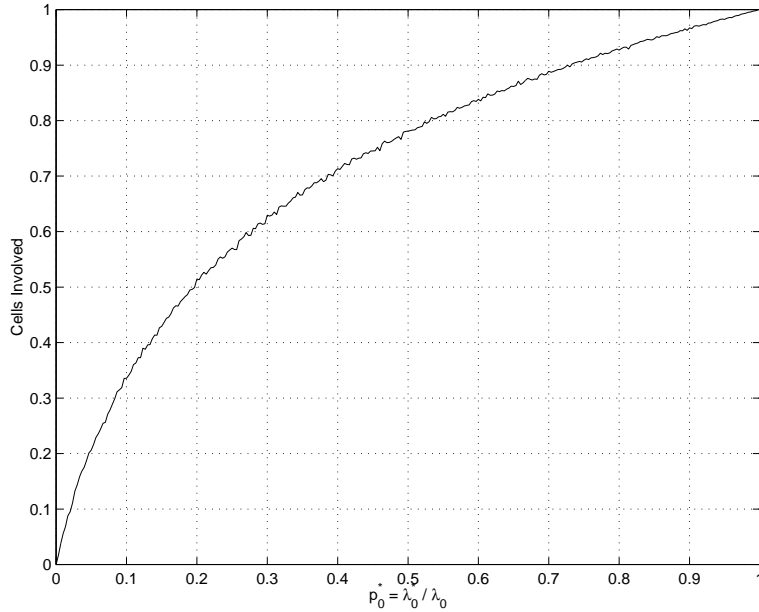


FIG. 3.10 – Évolution de la participation en fonction de p .

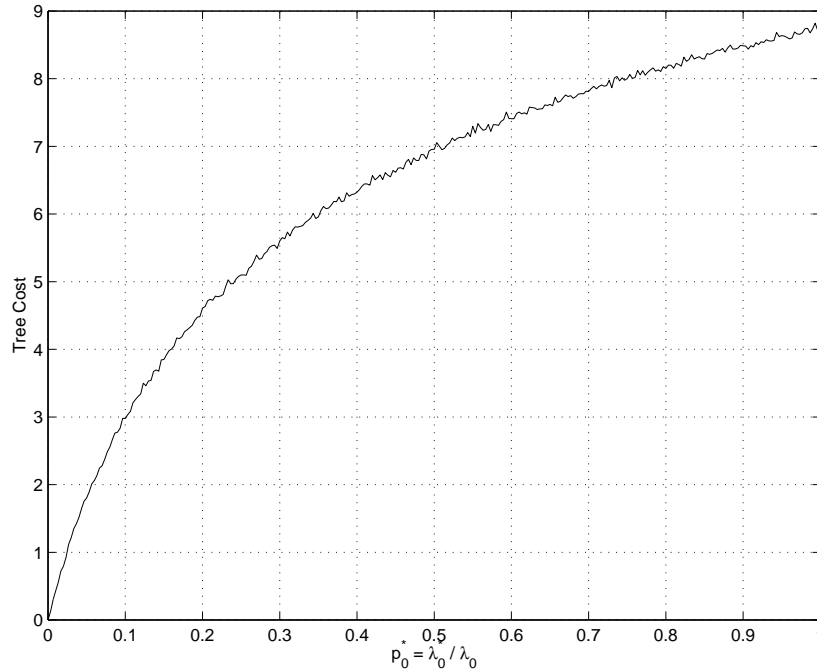
Cette évolution concave est due aux phénomènes de concentration qui peuvent avoir lieu au niveau de tous les nœuds du réseau.

Remarque 3.10 Il est également intéressant de noter que le coût de l'arbre prend une forme en $O(\sqrt{\lambda})$ pour $p = 1$ et $|W| = 1$.

Preuve. Lorsque $p \approx 1$, quasiment tous les nœuds du réseaux font partie du groupe multicast. Lorsqu'une nouvelle feuille arrive, la nouvelle branche a toutes les chances de s'arrêter au premier nœud rencontré (en direction du point de rendez-vous) — celui-ci faisant très certainement parti du groupe multicast. L'arbre est donc constitué de λ arêtes. Or des considérations géométriques permettent de montrer que la longueur moyenne d'une arête est proportionnelle à $\frac{1}{\sqrt{\lambda}}$. Le coefficient de proportionnalité nous est hélas inconnu car il dépend du routage un peu particulier qui permet à chaque nœud de rejoindre le RP¹¹. \square

3.6 Conclusion

¹¹Ce coefficient vaut 1 dans le cas du routage markovien présenté en annexe. Les phénomènes de concentration sont tels que les chemins joignant un point au RP ne sont plus markoviens. Nous observons que la concentration conduit à des longueurs d'arrêtes plus courtes (coefficient ≈ 0.85).

FIG. 3.11 – Evolution du coût en fonction de p ($\lambda = 100$).

Arbres Hiérarchiques.

Dans ce chapitre, nous nous sommes intéressés aux performances des arbres hiérarchiques centrés. Nous avons pu étudier ces objets analytiquement lorsque des “tunnels” (encapsulation IP dans IP) sont utilisés — c’est à dire lorsque le réseau dispose de peu de ressources multicast. Nous avons pu mettre en évidence une structure hiérarchique permettant de minimiser la quantité de ressources utilisées par l’arbre multicast. Nous avons montré qu’un nombre très limité de niveaux hiérarchiques suffit à réduire significativement le coût de l’arbre : même dans le pire cas (le mode “dense”), il est en effet inutile de dimensionner les domaines multicast de manière optimale, c.à.d. avec un nombre de niveaux hiérarchiques égal à

$$H^* \approx \frac{1}{2} \log_2(M)$$

où M désigne le nombre total de nœuds du réseau, le coût de l’arbre n’évoluant quasiment plus après $H > 2$, même pour de grands réseaux.

Nous avons également pu mettre en évidence le compromis qui existe entre l’utilisation de ressources et les délais de bout-en-bout. En effet, ces deux quantités sont respectivement décroissantes et croissantes avec le nombre de niveaux hiérarchiques (lorsque $H < H^*$). Nous avons montré que les délais restent néanmoins raisonnables lorsque le nombre de niveaux hiérarchiques est limité : En particulier, les délais de diffusion à partir de la racine de l’arbre

sont bornés par

$$\mathcal{D}_H < 2\mathcal{D}_1$$

lorsque $H < H^*$ pour une structure minimisant les ressources de l'arbre (dans le mode "dense"). Nous avons pu également constater que les délais suivent une évolution géométrique avec le nombre de niveaux hiérarchiques (pour cette même structure optimale) pour $H > H^*$ de sorte qu'il est essentiel de ne pas utiliser un trop grand nombre de niveaux hiérarchiques.

L'outil : Géométrie aléatoire.

Au delà des résultats sur les arbres hiérarchiques évoqués ci-dessus, cette étude nous semble positive du point de vue du modèle à proprement parlé. L'utilisation des processus ponctuels et de résultats de géométrie aléatoire s'est avérée très adaptée à l'étude de protocoles de routage multicast. Nous avons pu obtenir des formules explicites pour la plupart des grandeurs caractérisant les performances de l'arbre hiérarchique : le coût (consommation en terme de ressources), les délais (à partir du RP central ou d'une feuille quelconque) et les coûts de construction. Dans certains cas, ces résultats ont permis de déterminer des structures optimales permettant de donner des indications pour le dimensionnement de réseaux.

Par contre, nous avons du faire quelques hypothèses afin de pouvoir obtenir ces résultats analytiques. En particulier, nous n'avons pu appréhender cette objet que sous l'hypothèse "tunnelling". Des travaux sont en cours avec François Baccelli sur l'étude des arbres de concentration de Delaunay (cf. section 3.5), afin de d'analyser le cas où les ressources multicast sont présentes dans tous les nœuds du réseau. Nous avons également considéré des graphes fortement connectés¹². Il a été prouvé [10, 28] que la connectivité des graphes influe de manière significative sur les performances des protocoles de routage multicast. Remarquons que les réseaux de Delaunay sont également planaires, ce qui n'est pas toujours le cas dans les réseaux réels.

L'étude analytique présentée ne peut donc pas se substituer aux outils de simulations utilisés habituellement pour l'évaluation de protocoles de routage multicast [41, 5], si ces outils sont utilisables — malgré la complexité du problème et le nombre extrêmement important de possibilités différentes à tester dans le cadre d'un réseau hiérarchique... Quelques résultats préliminaires de simulation ont permis de mettre en évidence que les structures hiérarchiques optimales (c.à.d. minimisant la quantité de ressources utilisées) mises en évidence dans ce chapitre semblent assez proches de la réalité pour une large catégorie de graphes [5, 7].

Le modèle présenté offre surtout un outil très puissant permettant d'appréhender quels sont les phénomènes en jeu et l'influence des différents paramètres. Nous avons notamment

¹²Les graphes de Delaunay utilisés ont une connectivité moyenne de 6. La connectivité des réseaux fédérateurs de grands fournisseurs de service se situe plutôt aux alentours de 3.

pu appréhender l'influence de la structure hiérarchique et du nombre de niveaux hiérarchiques sur les différents critères de performance de assez simple (par dérivation!), pour des réseaux de dimension arbitraire. Cette étude est extrêmement complexe à mener par simulation étant donné le nombre de structures hiérarchiques possibles¹³...

¹³Notre propre expérience de simulation [5, 7] nous a montré qu'il est impossible d'aller au delà d'un réseau de 100 noeuds pour mener à bien cette étude dans des délais raisonnables, à cause du nombre de structure hiérarchiques possibles d'une part, et à cause du générateur de graphe aléatoire d'autre part.

Deuxième partie



Routage QoS

Chapitre 4

Routage dynamique

Dans ce chapitre, nous introduisons les concepts généraux de routage utilisés dans les réseaux privés (Réseaux IP et réseaux ATM normalisés par l'ATM-Forum) — nous ne traiterons pas des techniques de routage spécifiques utilisées par les opérateurs dans les réseaux publics. Dans la section suivante, la notion de routage dynamique est introduite. La section 4.2 traite du routage dynamique sensible à la qualité de service. Finalement, les notions de hiérarchie et d'agrégation sont introduites dans la section 4.3.

4.1 Routage Dynamique

La fonction de routage est utilisée pour calculer les tables de routage (ou table de commutation) qui seront utilisées par la matrice de commutation pour acheminer les informations reçues vers leur destinataire. Il existe deux formes de routage :

- **Routage statique.** Les routes sont précalculées par l'opérateur du réseau et sont ensuite rentrées dans les commutateurs ou routeurs. Ces machines n'ont pas la possibilité de modifier les tables de routage dont elles disposent. Ce type de routage est peu répandu dans les réseaux privés mais reste par contre très utilisé par les opérateurs téléphoniques européens par exemple. Pour ces derniers, les calculs de routage sont basés sur des mesures de trafic et permettent d'assurer une qualité de service donnée aux utilisateurs tout en maximisant les revenus estimés de l'opérateur — le dimensionnement des liens et le calcul des routes étant optimisés conjointement en général — voir [36].
- **Routage dynamique.** Les tables de routage sont calculées dynamiquement par les routeurs ou commutateurs. Cette technique nécessite l'échange d'informations de routage. Comme nous l'expliquons plus loin, cette approche est très utilisée dans les réseaux privés.

4.1.1 Introduction

Le but du routage dynamique est de pouvoir s'adapter automatiquement à un environnement changeant rapidement. Si un nouveau sous-réseau est ajouté, l'existence de ce sous-réseau est propagée automatiquement par le protocole de routage dynamique. Des calculs de routage permettent alors aux routeurs (resp. commutateurs) de déterminer le meilleur chemin permettant de joindre la nouvelle destination. Par opposition, un tel événement aurait nécessiter la configuration de la totalité des routeurs (resp. commutateurs) du réseau dans le cas d'un routage statique.

L'introduction d'un protocole de routage dynamique est également motivée par des soucis de robustesse. Si une liaison tombe en panne, la rupture de communication sera détectée et les tables de routage seront automatiquement re-calculées afin d'éviter le lien défectueux¹.

En un mot, le routage dynamique permet de s'adapter automatiquement aux changements qui peuvent intervenir dans le réseau. Les protocoles de routage dynamique ont connu un très fort développement dans les réseaux privés (réseaux IP², Novell, etc.) et dans tout particulièrement dans l'Internet — Il est facile de comprendre l'intérêt que représente cette solution dans un tel réseau où les modifications dans la topologie sont si fréquentes³.

4.1.2 Routage “Distance Vector”

La technique dite “distance-vector” que nous décrivons brièvement dans ce paragraphe correspond à un algorithme de Bellman-Ford distribué. Les tables de routage sont construites grâce à des échanges périodiques (ou exceptionnels) de vecteurs de distance, d'où cette nomination.

Fonctionnement.

A l'allumage, les routeurs n'ont pas connaissance des destinations lointaines car aucune information n'a encore été échangée. Par contre, l'administrateur réseau a préalablement configuré chaque routeur en précisant quels sont les réseaux (préfixes+masques dans les réseaux IP) qui lui sont directement raccordés. Ces informations sont transmises telles quelles par chaque routeur à tous ses voisins. Les routeurs adjacents prennent alors connaissance de l'existence

¹Dans le cas du routage statique, les routes alternatives permettant de faire face à des pannes doivent être rentrées dans les commutateurs également.

²IP : Internet Protocol.

³On estimait déjà dans les années 80 qu'il y avait près d'une dizaine de nouveaux préfixes d'adresses annoncés par jour dans l'Internet. Et ne parlons pas des pannes qui peuvent également générer des événements de routage...

des réseaux connectés à leurs voisins directs, et ceux-ci vont rajouter une entrée dans leur table de routage (avec une métrique de un saut).

Les tables de routage sont transmises périodiquement. Les informations de ces tables étant une liste de réseaux accessibles et les distances correspondantes, les informations échangées sont des vecteurs de distances, d'où la nomination de cet algorithme. Grâce à ces échanges, les routeurs apprennent l'existence de réseaux distants. De plus, par comparaison des vecteurs de distance reçus, un routeur est aussi capable de détecter s'il existe un chemin plus court (nombre de sauts inférieur) menant vers une destination. Il est facile de voir que la convergence est assurée après un nombre de périodes borné par le diamètre du réseau (en nombre de sauts).

Points forts et limitations.

L'avantage principal de cet algorithme est qu'il est distribué et extrêmement simple. Il souffre en revanche de nombreuses limitations :

- **Métriques limitées.** Cet algorithme ne peut pas, par nature, considérer plus d'un critère de sélection. Il s'agit de poids administratifs ou de nombre de sauts. Le choix de ces paramètres est souvent limité à cause du problème de "comptage à l'infini" (cf. [42]).
- **Convergence.** Cette technique souffre de problèmes de convergence. Dans certaines situations pathologiques (voir [42]), la convergence de l'algorithme peut prendre jusqu'à 2 ou 3 périodes de rafraîchissement avec les variantes⁴ les plus efficaces — ce qui représente 1 minute 30 secondes avec les paramètres traditionnellement recommandés.
- **Échanges d'informations.** Les échanges périodiques de la totalité des tables de routage pose un gros problème de faisabilité pour de grands réseaux.

Cette technique reste encore très utilisée dans l'Internet, grâce au succès du protocole RIP (Routing Information Protocol [79]). La tendance actuelle est tout de même plutôt aux protocoles "link-state" (cf. paragraphe suivant) pour leur meilleure convergence.

4.1.3 Routage "Link-State"

Fonctionnement.

Les problèmes des algorithmes du type "distance-vector" sont dus au fait que les routeurs ne disposent d'aucune vision d'ensemble du réseau et doivent donc se baser sur les informations transmises par leurs voisins — informations éventuellement obsolètes. Dans la technologie dite "Link-State", les informations de routage sont diffusées⁵ dans tout le réseau. Grâce à

⁴"poisonous reverse", "path holddown", "route poisoning", voir [42].

⁵Plus exactement, il s'agit d'échanges multicast sur une adresse de groupe spécifique (`All_OSPF_Routers` par exemple pour OSPF) afin de ne pas déranger les hôtes qui ne désirent pas participer au routage.

ces échanges, les routeurs peuvent construire une base de données topologique contenant la localisation des routeurs et l'état des liens qui les relie entre eux. Il suffit alors à chaque routeur d'utiliser un algorithme de recherche de plus court chemin (Dijkstra⁶) sur cette base de données pour créer sa table de routage.

Points forts et limitations.

Remarquons que cette technique est exempte des problèmes de "comptage à l'infini" : Lorsqu'un lien tombe, cette information est diffusée sur tout le réseau. Tous les routeurs peuvent alors mettre à jour leur base de données et recalculer leur table de routage en conséquence (évitant ainsi la liaison défectueuse). L'avantage de cette technique réside donc essentiellement dans sa vitesse de convergence (l'ordre de grandeur donné par les constructeurs est d'une seconde environ).

En revanche, les calculs de routage sont complexes et nécessitent le stockage en mémoire d'une base de données topologique. Les *broadcasts* nécessaires aux échanges d'informations de routage représentent également une charge importante pour le réseau⁷. Des problèmes de faisabilité à grande échelle⁸ sont donc également posés par cette technique de routage.

OSPF (Open Shortest Path First [81]) est le protocole de type "Link-State" le plus utilisé dans l'Internet actuellement. Ce protocole a également inspiré l'ATM Forum pour le développement du protocole de routage de PNNI (Private Network Network Interface [63]).

Remarque 4.1 Nous avons présenté ici les deux grandes "familles" de protocoles de routage. Il existe de nombreuses variantes des techniques présentées, et d'autres techniques qui ne peuvent rentrer dans aucune des deux catégories "link-state" ou "distance-vector" : C'est le cas de l'algorithme DUAL⁹ de J.J.Garcia-Luna-Aceves [35] par exemple. Voir [42, chapitre 6] pour plus de détails.

⁶L'algorithme de Dijkstra est également dénommé "Shortest Path First" par les anglo-saxons, ce qui explique la dénomination du principal protocole "link-state" de l'Internet : OSPF (Open Shortest Path First).

⁷Remarquons cependant que les informations de routage peuvent n'être transmises qu'exceptionnellement, lorsqu'un changement est détecté dans le réseau (défaillance d'un routeur ou d'un lien etc.). La plupart des protocoles font tout de même appel à des échanges d'information de routage périodiques par mesure de sécurité — mais avec des fréquences faibles par rapport au routage "distance vector", toutes les 30 minutes au lieu de toutes les 30 secondes par exemple.

⁸Les anglo-saxons parlent de "scalability".

⁹L'algorithme DUAL est utilisé par le protocole propriétaire Cisco E-IGRP (Enhanced Interior Gateway Routing Protocol). Voir <http://www.cisco.com>.

4.1.4 Sélection des chemins.

Les tables de routage sont construites dans le but de minimiser les ressources utilisées par chaque flux afin d'utiliser au mieux les ressources du réseau. Des algorithmes de plus courts-chemins (Bellman-Ford, Dijkstra) sont donc utilisés. Dans RIP, les calculs de plus court chemin sont effectués sur le nombre de sauts (nombre de liens utilisés). Il n'est pas alors possible de distinguer une liaison par modem à 56.6Kb/s d'un réseau local à haut débit... Les protocoles plus récents comme OSPF sont basés sur des "poids administratifs" permettant de donner une idée de la bande passante disponible sur les différents éléments du réseaux.

Remarquons qu'avec les protocoles de routage dynamique actuels de l'Internet (RIP, OSPF), les décisions de routage que nous avons décrites précédemment sont basées sur des données topologiques uniquement¹⁰. Les informations sont donc acheminées sur les chemins décrits dans les tables de routage sans aucune considération sur l'état de congestion des liens traversés... Le routage sensible à la qualité de service que nous présentons dans la section suivante permet de prendre en compte d'autres informations comme l'état de congestion ou les délais de traversée des liens.

Notons également que par défaut, une seule route est utilisée pour joindre une destination donnée dans les réseaux datagrammes. Il serait préférable, du point de vue des performances globales, de partager la charge sur plusieurs chemins distincts [34]. Cependant ce partage pourrait conduire à des problèmes de boucles de routage dans les réseaux en mode non connecté. Le partage de charge n'est effectué que sur des routes ayant exactement les mêmes coûts (somme des poids administratifs des liens utilisés) — il est possible de prouver que les risques de bouclage sont alors évités. On parle de partage de charge de type ECMP (Equal Cost MultiPath).

4.2 Routage sensible à la qualité de service

Les protocoles de routage dynamique présentés dans la section précédente ne sont pas adaptés aux nouvelles architectures de réseau à intégration de service : Les décisions de routage ne sont pas basées sur l'état de congestion des liens traversés. Le routage sensible à la qualité de service (que nous appellerons "routage QoS¹¹" tout simplement) permet de tenir compte des demandes spécifiques de l'utilisateur (en terme de QoS) et de l'état du réseau (congestion,

¹⁰Le protocole IGRP définit une métrique composite qui peut prendre en compte la charge des liens. Néanmoins, cette facilité n'est jamais utilisée dans la pratique — A titre d'exemple, le constructeur *cisco* déconseille tout simplement son utilisation dans ses manuels de configuration. Cette option peut en effet conduire à des phénomènes oscillatoires dégradants les performances globales.

¹¹QoS : Quality of Service.

délais etc.).

4.2.1 Motivations

Le routage sensible à la qualité de service, ou “routage QoS”, permet d’accroître significativement la probabilité que le réseau puisse satisfaire les besoins en terme de qualité de service des utilisateurs. Les chemins sont en effet sélectionnés en fonction de la demande des utilisateurs et de l’état de congestion des différents liens du réseau. Ce type de routage requière par contre la dissémination d’informations sur les liens du réseau, comme leur bande passante résiduelle, leur délais de traversée ou leur taux de perte. Remarquons que ces échanges représentent un surcoût beaucoup plus important par rapport à la seule propagation des poids administratifs — les métriques nécessaires changeant constamment à chaque arrivée ou départ d’un flux (resp. d’une connexion).

4.2.2 Informations de routage

Choix des paramètres échangés.

Les protocoles de routage QoS sont basés sur la technique “distance-vector” présentée plus haut. Un nombre arbitraire de paramètres peuvent ainsi être échangés et mis à la disposition de l’algorithme de routage pour prendre les décisions les plus fines possibles. Ces paramètres peuvent être classés en deux sous-ensemble, en suivant la dénomination choisie par l’ATM Forum pour PNNI :

- **Attributs.** Un attribut est un paramètre décrivant un lien que l’algorithme de routage peut considérer individuellement pour décider si un lien est approprié pour accepter la connexion (ou le flux). Il s’agit par exemple de la bande passante résiduelle.
- **Métriques.** Une métrique est un paramètre décrivant un lien. L’algorithme de routage doit combiner les métriques des différents liens d’un chemin pour décider si ce chemin est approprié ou non pour accepter la connexion (ou le flux). Les métriques sont en général additives (poids administratifs, délais, gigue¹²) ou multiplicatives (taux de perte...).

Le choix des paramètres échangés est difficile, car leur propagation sur le réseau représente un surcoût protocolaire important. Le nombre de paramètres échangés par le protocole PNNI¹³ [63] est assez important : poids administratif, bande passante maximale, bande passante résiduelle¹⁴, délais de traversée, variation du délais de traversée (ou “gigue”), taux de perte.

¹²Les figures des différents liens ne devraient pas être additionnées car le résultat final est très pessimiste.

Des solutions plus réalistes sont considérées à l’ATM Forum.

¹³PNNI : Private Network Network Interface.

¹⁴La bande passante maximale correspond au débit du lien (en cellules par secondes). La bande passante résiduelle correspond à la quantité de bande passante non réservée qui peut être offerte aux nouvelles connexions.

Dans QOSPF¹⁵ [62], seuls la bande passante résiduelle et le délais de traversée sont échangés. Cette approche pragmatique est motivée par la grande complexité des algorithmes de routage prenant en compte plusieurs métriques (cf. plus loin).

Rafraîchissement des informations de routage.

Les attributs et métriques permettant un routage QoS (bande passante résiduelle, délais, ...) varient en fonction du temps. Il serait souhaitable que les algorithmes de routage soient mis au courant de l'évolution exacte de ces paramètres afin de prendre les décisions de routage les plus adaptées. Une telle approche pourrait néanmoins engendrer des échanges de routage quasi-incessants. Il est donc nécessaire de mettre en place des politiques de rafraîchissement des informations de routage permettant de limiter la quantité d'information échangée sans trop nuire à la qualité des décisions de routage. Ces politiques sont très délicates à définir car il existe un compromis à trouver entre surcoût protocolaire (taux de messages échangés etc.) et l'acuité des informations de routage sur lesquelles sont basées les décisions de routage.

La solution la plus simple est certainement basée sur des échanges périodiques des informations de routage, comme c'est le cas pour l'instant dans QOSPF. Le surcoût protocolaire est alors déterministe et dépend directement de la fréquence des échanges choisie. Cette solution est par contre assez mal adaptée pour répondre à des arrivées en chaîne ("bursts" à l'échelle de temps d'une connexion ou d'un flux). De plus, les changements en terme de bande passante sont plus ou moins significatifs selon leurs valeurs : Des informations précises sont nécessaires lorsque le lien est congestionné alors que des informations très grossières suffisent lorsque le lien est très peu chargé.

Des techniques de rafraîchissement "différentielles" peuvent être utilisées : L'idée est de ne transmettre les informations de routage que lorsqu'un "changement significatif" a été observé. Dans PNNI par exemple, un changement significatif est défini comme une variation relative dépassant un certain seuil (par exemple +/- 30 %). Des extensions sont également à l'étude pour supporter ce type de politique de rafraîchissement dans QOSPF (voir [62]). Notons qu'en réalité, le protocole PNNI est une combinaison des deux approches (périodique et différentielle).

Recherches actuelles.

Les politiques de rafraîchissement définies des protocoles de routage actuels [63] et [62] offrent à l'administrateur une grande latitude dans le choix des paramètres, ce qui permet un dimensionnement précis en fonction du surcoût protocolaire toléré et des performances attendues. Néanmoins, l'impact des mécanismes de rafraîchissement sur les performances globales

¹⁵QOSPF : QoS Extensions to OSPF.

(utilisation des ressources, etc.) restent encore mal connues. De nombreuses études ont été menées ces dernières années afin de mieux comprendre ce que l’acuité des informations de routage signifie [37] et quel en est son impact sur les performances globales dans un contexte IP (QOSPF) [38] ou ATM¹⁶(PNNI) [40, 57, 58].

Les premiers résultats montrent que les rafraîchissements périodiques peuvent causer des phénomènes oscillatoires si les fréquences de rafraîchissement sont trop faibles. Le taux de rejet des appels (ou flux) peuvent alors être catastrophiques, de sorte qu’une technique différentielle semble plus adaptée. Des techniques spécifiques pour les approches différentielles sont proposées dans [31]. Les études citées sont basées sur des simulations sur des topologies fixes et sur des graphes aléatoires [61]. Dans le chapitre suivant, nous proposons un modèle analytique simple permettant d’étudier comment l’acuité des informations de routage influe sur les performances globales d’un réseau.

Remarquons finalement que des recherches sont en cours afin de limiter la quantité d’informations de routage à échanger sur le réseau. Ces recherches partent du principe que seul une fraction des flux nécessitent un routage QoS (requérant la connaissance de métriques supplémentaires), les flux “best-effort” pouvant être routés par les protocoles usuels. Il a été également constaté qu’une grande partie du trafic Internet est créée par une faible proportion de flux de longue durée, de sorte qu’il suffirait d’utiliser un routage utilisant les informations de congestion pour router ces flux pour améliorer significativement les performances globales de l’Internet. Des techniques sont donc en cours d’étude afin de transmettre le minimum d’informations de routage (uniquement les informations susceptibles d’être utilisées pour le routage de certains flux¹⁷).

4.2.3 Choix de l’algorithmes de routage

Le choix d’une politique de rafraîchissement des informations de routage n’est pas le seul problème rencontré lors du développement de protocoles de routage QoS. Utiliser ces informations à bon escient est également délicat.

¹⁶Les recherches menées sur un protocole sont généralement applicables à un autre. La seule différence vient du fait que le routage IP est effectué “par sauts” (hop-by-hop) alors que PNNI définit un routage basé à la source (“source-routing”). Il ne semble pas non plus exister d’équivalent au “crankbacks” PNNI dans le monde IP.

¹⁷Pour reprendre un vocable introduit dans la liste de discussion IETF QoR (QoS Routing), il s’agit d’une approche “pull” (recherche d’informations de QoS à la demande) alors que les approches “link-state” sont qualifiées de “push” (dissémination d’informations de QoS par défaut).

Complexité.

Il a été prouvé que la détermination d'un chemin prenant en compte deux paramètres de routage (ou plus !) est un problème NP-complet [60, 44, 50]. Les algorithmes de routage mis en oeuvre dans les réseaux actuels considèrent rarement tous les paramètres mis à leur disposition. Dans QOSPF, les délais ne sont pas considérés par exemple — seule la bande passante résiduelle et le nombre de sauts sont pris en compte. La norme PNMI ne spécifie pas d'algorithme de routage particulier, chaque constructeur étant libre de ses choix. Les algorithmes proposés dans la littérature [44, 52, 50, 51] n'utilisent en général que la bande passante — voir uniquement les poids administratifs¹⁸. Notons que le choix de la seule bande passante se justifie dans des contextes spécifiques où des ordonnancements basés sur les débits¹⁹ sont utilisés [38] — les délais et gigues étant alors inversement proportionnels à la bande passante.

Remarquons de plus que les algorithmes de routage QoS devraient être utilisés à chaque arrivée de connexion (resp. flux), pour prendre en compte au mieux leurs demandes de qualité de service. Une telle approche ne semble pas réalisable lorsque le nombre de connexions ou flux à manipuler est grand. L'utilisation de **tables de routage précalculées** est donc préconisée [62, 63]. Avec cette méthode, des chemins vers chaque destination possible et pour une catégorie de service spécifiée (en terme de bande passante en général) sont calculées périodiquement, ou exceptionnellement (après la réception de nouvelles informations de routage par exemple). Ces routes sont ensuite utilisées par les connexions ou flux entrants. Il a été prouvé que le pré-calcul des tables de routage permet de réduire considérablement les ressources (CPU) nécessaires sans que les performances globales n'en soient trop affectées [57, 30].

Impact sur les performances globales du réseau.

Nous avons vu que la plupart des algorithmes actuels utilisent uniquement la bande passante résiduelle pour sélectionner les chemins. Les algorithmes utilisant ce paramètre doivent faire face à un compromis délicat :

- Soit la priorité est donnée à la bande passante (aux dépens du nombre de liens utilisés). Cette approche permet un excellent partage de la charge sur les liens les moins congestionnés et conduit à une meilleure utilisation des ressources pour des charges faibles à modérées [50]. Par contre, les chemins sélectionnés peuvent passer par un plus grand nombre de liens de sorte que l'utilisation du réseau devient très mauvaise à forte charge.
- Soit la priorité est donnée au nombre de liens utilisés. Dans ce cas, le partage de charge peut être moins bien assuré mais l'utilisation des ressources est préservée, même à forte

¹⁸La fonction GCAC (Generic Connexion Admission Control) étant ensuite utilisée pour sélectionner les routes disposant des ressources suffisantes pour accepter la connexion.

¹⁹Rate Based Schedulers.

charge.

Les algorithmes privilégiant le nombre de sauts sont en général préférés afin de préserver les ressources. Il a été montré également que ce type d'algorithme est moins affecté par l'inacuité des informations de routage (dû aux mécanismes de rafraîchissements) [50]. L'algorithme le plus connu, dénommé "shortest-widest" fonctionne de la manière suivante : Un algorithme de plus court chemin est tout d'abord utilisé. Une liste de plus-courts chemins menant à la destination est alors retournée, parmi lesquelles le chemin possédant le plus de bande passante est finalement choisi [50].

Des algorithmes plus complexes, souvent nommés algorithmes de **routage "compétitifs"** sont également très intéressants car ils peuvent permettre d'obtenir un compromis intéressant entre partage de charge et utilisation du réseau. Les calculs sont basés sur une fonction de coût prenant à la fois compte du nombre de lien et de la bande passante résiduelle : Les liens sont marqués par une quantité $f(w)$ où w est la bande passante résiduelle où f est une fonction décroissante de w . On trouve par exemple : $f(w) = \frac{1}{w}$. Le but est alors de trouver un chemin P minimisant la somme $\sum_{i \in P} f(w_i)$. Les performances sont variables selon la bande passante requise [49].

4.2.4 Routage sur des informations inexactes.

De nombreuses recherches sont en cours sur la création d'algorithme de routage robustes permettant de minimiser l'impact des informations de routage inexactes. Notons que les algorithmes donnant la priorité au nombre de sauts sont moins affectés par ces problèmes (utilisant dans une moindre mesure les informations de QoS). Des algorithmes spécifiques sont également développés pour prendre en compte les imprécisions dues au mécanismes de rafraîchissement [37, 32].

Fonction GCAC

Une connexion ou un flux peut être refusé sur le chemin sélectionné si les ressources restantes au niveau d'un lien sont insuffisantes²⁰. Le routage QoS permet en théorie de sélectionner des chemins possédant des ressources suffisantes afin de diminuer le taux de blocage. Dans la pratique, nous avons vu qu'il faut souvent faire appel à des tables de routage précalculées. Nous avons également vu que les algorithmes de routage QoS ne tiennent généralement pas compte de la totalité des informations de routage, de sorte qu'il peut être nécessaire de vérifier que les chemins sélectionnés répondent bien aux critères de qualité de service désirés.

Pour les réseaux ATM, le protocole PNNI définit une fonction générique, appelée GCAC²¹,

²⁰Par le contrôle d'admission, appelé CAC (Connexion Admission Control) dans les réseaux ATM.

²¹GCAC : Generic Connexion Admission Control.

permettant d'anticiper les décisions des algorithmes d'admission d'appels sur les liens choisis²². Notons qu'il ne s'agit pas d'une garantie absolue de conformité, puisque la fonction **GCAC** utilise des informations de QoS plus ou moins exactes — selon la politique de rafraîchissement choisie. Nous montrons dans le chapitre suivant que les chemins sélectionnés ont tout de même une plus forte probabilité d'accepter la connexion de sorte que cette fonction permet de réduire les taux de blocage²³ tout en évitant d'échanger trop de messages de signalisation inutiles.

Crankback.

Malgré les précautions prises lors du routage, il est possible que le flux ou la connexion soit refusé sur le chemin sélectionné — Un lien ne dispose pas de ressources suffisantes. Ce phénomène est d'autant plus probable avec des politiques de rafraîchissement des informations de routage très conservatrices. Il peut être souhaitable d'autoriser un re-routage du flux ou de la connexion afin de diminuer leur taux de perte.

Le protocole PNNI définit une procédure signalétique, appelée "crankback", permettant de libérer les ressources réservées sur le chemin défectueux jusqu'au premier commutateur. Ce dernier peut alors décider de re-router la connexion sur un chemin alternatif.

4.3 Routage hiérarchique

Nous avons mis en évidence que le routage dynamique engendre un coût supplémentaire sur le réseau, en terme de bande passante (nécessité d'échanger des informations de routage) et des ressources matérielles (calculs de routage et/ou stockage des tables précalculées). Quelque soit la technologie ("Link State" ou "Distance Vector") utilisée, les sur-coûts mentionnés deviennent rapidement excessifs lorsque la taille du réseau augmente. Le routage hiérarchie est un élément clé permettant de réduire la quantité d'information de routage à manipuler tout en assurant un routage cohérent de bout-en-bout.

L'idée sous-jacente est fort simple : Il s'agit de diviser récursivement (à chaque niveau hiérarchique) le réseau en aires de routage indépendantes. Le routage est assuré par un protocole de routage dynamique classique à l'intérieur des domaines — ceci étant de taille raisonnable après partition. Les informations de routage d'un domaine donné sont **agrégées**, c'est à dire résumées, afin d'être passées au niveau hiérarchique supérieur. Ce mécanisme d'agrégation a pour but de limiter la quantité d'information échangée tout en assurant un routage de bout-en-bout cohérent.

²²Remarquons qu'en ATM, les algorithmes de **CAC** ne sont pas normalisés. La définition d'une fonction **GCAC** standard était donc nécessaire, pour travailler dans un environnement multi-constructeur.

²³Lorsque le re-routage n'est pas autorisé ou tout au moins contrôlé.

4.3.1 Fonctionnement

Dans l'Internet.

Le réseau Internet est muni de deux niveaux hiérarchiques. Le réseau mondial est divisé en aires de routage indépendantes, appelées ASs²⁴. Le routage intra-domaine est assuré par les protocoles de routage dynamiques évoqués au début de ce chapitre (RIP, OSPF, ...). Ces protocoles permettent de calculer les plus courts chemins entre différents sous-réseaux. Les informations de routage internes ne sont pas transmises en dehors des frontières d'un AS.

Au niveau hiérarchique supérieur, la connectivité entre domaines est assurée par un protocole de routage externe (BGP²⁵ en général). Ce protocole permet de déterminer les chemins entre domaines. Ce protocole utilise des informations agrégées puisqu'aucune information sur la topologie interne des ASs ne lui est connue²⁶. Le routage externe sélectionne les domaines (c.à.d AS) à traverser, les paquets étant acheminés par les protocoles de routage interne afin de traverser les domaines sélectionnés.

Cette solution n'est plus suffisante depuis quelques années vues les dimensions prises par le réseau Internet. Le nombre de réseaux existants dans l'Internet devient trop important à gérer, de sorte qu'il faut avoir recours à une agrégation manuelle aux niveaux des routeurs externes. Les informations de routage échangées sont agrégées dès que ceci est possible (sans considération de niveaux hiérarchiques), en jouant sur les préfixes et les masques des adresses IP²⁷. Sur la figure 4.1 est représentée une application de cette technique d'agrégation. Les informations de routage sont échangées de proche en proche (technique "distance-vector") : Le routeur central R2 reçoit ainsi des informations sur les réseaux A.0 et A.1. Les routeurs situés à gauche de R2 n'ont pas besoin de connaître l'existence de deux réseaux distincts A.0 et A.1, de sorte que le routeur R2 n'annonce à R1 que le préfixe A (regroupant les réseaux A.0 et A.1). Cette agrégation doit être administrée avec précaution pour éviter d'éventuels problèmes de routage (deux routeurs peuvent par exemple annoncer une accessibilité vers les mêmes préfixes d'adresses suite à des agrégations inadéquates).

Dans l'ATM avec le protocole PNNI.

Le protocole PNNI [63] est certainement le protocole de routage dans lequel la notion de hiérarchie a été le plus adroitement mise en application. Ce protocole permet par exemple

²⁴AS : Autonomous System.

²⁵BGP : Border Gateway Protocol [82].

²⁶L'élément d'information manipulé par le protocole de routage externe est le numéro d'AS : Un AS contient éventuellement plusieurs réseaux mais aucune information sur les sous-réseaux n'est connue.

²⁷Cette technique est connue sous le nom de CIDR (Classless Inter-Domain Routing) [42, chapitre 9].

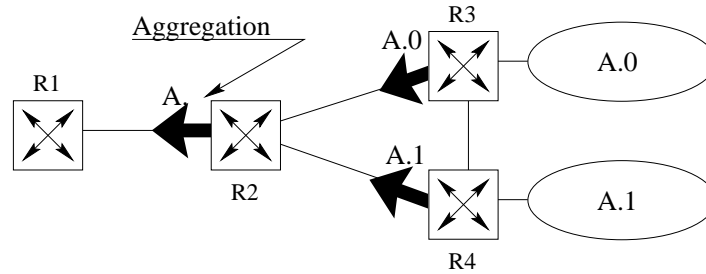


FIG. 4.1 – Agrégation par "sauts" dans un réseau non hiérarchique.

de définir un nombre de niveaux hiérarchiques arbitraire²⁸. Dans PNNI, le réseau est divisé en domaines de routage appelés "peer-groups"²⁹ (notés PGs par commodité). Les domaines sont récursivement sous-divisés en sous-domaines selon le nombre de niveaux hiérarchiques souhaité. Par exemple, sur la figure 4.2, le réseau est divisé en deux domaines (notés PG A et PG B) ; le peer-group PG A étant lui même sous-divisé en deux peer-groups (notés PG A.1 et PG A.2).

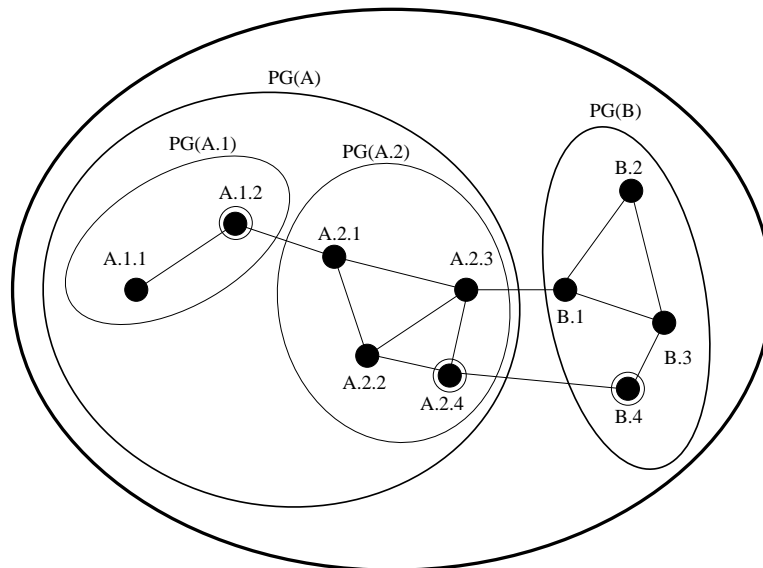


FIG. 4.2 – Exemple de division hiérarchique.

A l'intérieur d'un domaine, les informations de routage sont échangées selon une technique de type "link-state" mais ne dépassent pas les frontières du Peer-Group. Ces échanges de

²⁸En réalité, le format des adresses ATM ne permet pas la définition de plus de 104 niveaux hiérarchiques... Mais nous verrons dans le chapitre 6 qu'il ne s'agit absolument pas d'une restriction, un nombre de niveaux hiérarchiques très limité suffisant à réduire considérablement les tailles des tables de routage.

²⁹Peer-Group : Les commutateurs d'un même PG doivent posséder un préfixe d'adresse commun (pour que l'agrégation soit efficace) d'où la notion de "pairs" (peers).

messages sont également utilisés par un mécanisme assez classique d'élection afin qu'à tout moment un noeud de chaque PG soit désigné pour assurer les fonctions de "peer-group leader" (PGL). Par exemple, le noeud A.1.2 est le PGL du peer-group PG A.1 sur la figure 4.2. Le PGL est chargé d'agrèger les informations de routage du domaine dont il a la charge et de disséminer les informations agrégées au niveau hiérarchique supérieur. Le peer-group agrégé est représenté par un noeud logique au niveau hiérarchique supérieur. Les noeuds logiques échangent des informations de routage agrégées avec leurs pairs et font descendre les informations reçues à l'intérieur du peer-group dont ils sont issus.

La structure hiérarchique résultante est représentée sur la figure 4.3. Le domaine PG A.1 est représenté par le noeud logique A.1, qui informe ses pairs (c.à.d les noeuds logiques A.2) des informations agrégées de son domaine. Les échanges d'informations entre ces noeuds sont également utilisés pour élire un PGL (le noeud logique A.2 en l'occurrence). Ce noeud agrège à son tour les informations³⁰ de routage du peer-group qu'il représente (sous la forme du noeud logique A) au niveau supérieur.

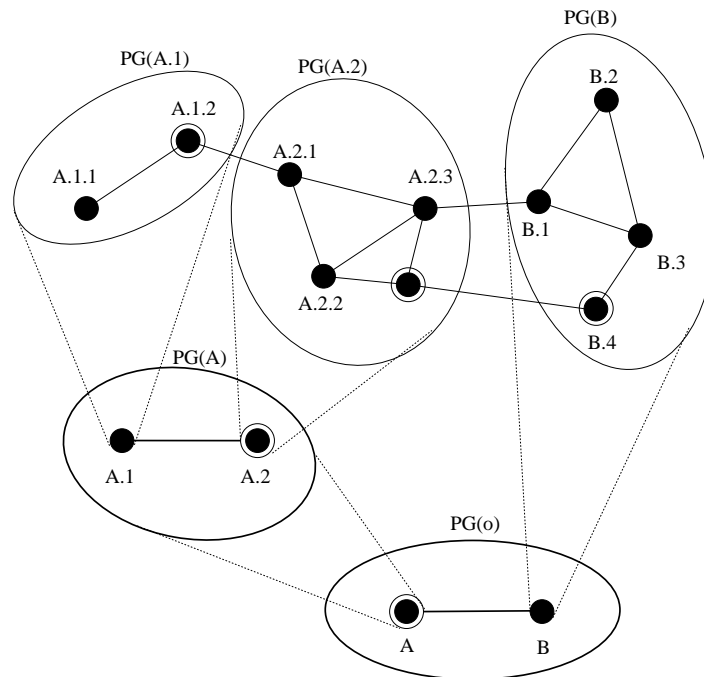


FIG. 4.3 – Structure logique construite par PNNT.

Avec une analogie manageriale, il est possible de comparer le PGL à un chef d'une équipe (à un niveau hiérarchique donné) : sa fonction le rend en particulier responsable de la montée et de la descente d'informations. Après une réunion avec ses subordonnés, il résume les informations

³⁰Informations déjà agrégées...

importantes et en fait part à ses collègues lors d’une réunion au niveau hiérarchique supérieur. Il informe ensuite ses subordonnés des informations échangées pendant cette réunion³¹.

Les mécanismes de routage sont donc totalement récursifs et permettent une réduction significative des informations de routage échangées globalement. La figure 4.4 représente les bases de données obtenues grâce aux échanges d’information dans la structure hiérarchique de la figure 4.3. Les noeuds du PG A.1 connaissent ainsi la topologie de leur domaine dans les détails; Il savent également qu’il est possible de joindre le peer-group PG A.2 (via le noeud A.1.2). Le PG B est également accessible via le peer-group PG A.2. Avec la structure hiérarchique choisie, les noeuds du PG B ne connaissent que l’existence du peer-group A et ignorent tout des sous-divisions (PG A.1 et PG A.2) internes de ce domaine.

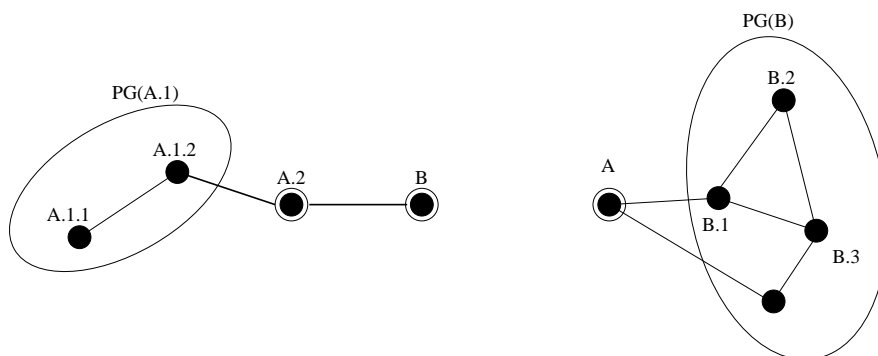


FIG. 4.4 – Vision topologique à l’intérieur des domaines PG A.1 et PG B.

4.3.2 Agrégation

Comme nous l’avons vu, un des principaux rôles de la fonction d’agrégation est de résumer les informations d’accessibilité (c’est à dire les différentes adresses accessibles dans le domaine de routage) afin de limiter la taille des tables de routage. Ceci est assez facile à effectuer si le plan d’adressage s’y prête en utilisant les préfixes d’adresses associés à des masques³². La structure des adresses IP et ATM le permet par exemple. L’impact de la structure hiérarchique sur la taille de la table de routage a été étudié par L.Kleinrock dans [46]. Nous reprenons également cette étude avec un modèle différent dans le chapitre 6.

Il est par contre beaucoup plus difficile d’agréger les informations topologiques comme le nombre de sauts ou le poids administratif afin de donner un ordre de grandeur du “coût” associé à la traversée d’un domaine.

³¹L’analogie s’arrêtant là, car il n’y a pas de “filtrage” lors de la descente d’information dans PNNI — ce qui est loin d’être le cas dans un contexte managérial!

³²Nous considérons superflu de définir ses notions bien connues.

Agrégation Topologique.

Dans les réseaux IP, les routages internes et externes sont totalement séparés et les coûts associés aux différents domaines sont fixés administrativement. Les métriques disparates (et incompatibles) utilisées par les différents protocoles de routage interne rend de toute façon cette l'idée d'une agrégation topologique impossible à réaliser. Dans le contexte du routage PNMI, le problème est par contre posé. Ce protocole propose deux méthodes de représentation (cf. figure 4.5) des peer-groups [63] basées sur les travaux de W.C.Lee [47] :

- **Simple Node Representation.** Cette approche, utilisée par défaut, permet de représenter un domaine entier avec un seul jeu de paramètres (un poids administratif, une bande passante, etc.) Il s'agit en fait d'une représentation en étoile symétrique — les extrémité des branches de l'étoile représentant les points de sortie du peer-group. Les branches sont toutes marquées par le même jeu de paramètre.
- **Complex Node Representation.** Il s'agit d'une représentation plus fine en étoile asymétrique — chaque branche de l'étoile possédant son propre jeu de paramètres. Il est également possible de définir exceptionnellement des “raccourcis” entre deux ports (sans passer par le coeur de l'étoile) afin de mieux représenter les réseaux fortement asymétriques.

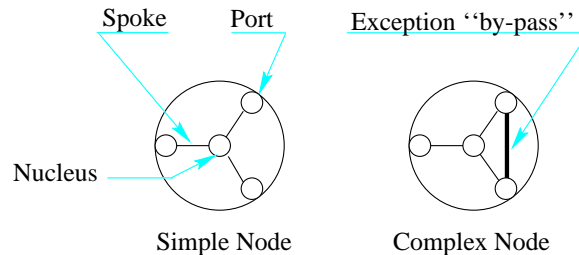


FIG. 4.5 – Méthode d'agrégation dans PNMI.

La définition d'algorithme d'agrégation topologique est difficile car il existe un compromis à trouver entre la quantité d'informations utilisées et l'acuité de la représentation finale. De nombreuses recherches ont été effectuées sur le sujet [47, 48, 43, 55]. La plupart des algorithmes construisent tout d'abord un maillage sur un graphe réduit ou seuls les noeuds périphériques sont conservés (les “ports” dans le vocabulaire PNMI). Les arrêtes sont marquées par les poids des plus court-chemins (selon l'algorithme de routage utilisé) reliant les ports considérés. Ce graphe maillé est ensuite réduit en une étoile symétrique, asymétrique, avec ou sans exception selon les algorithmes. Le document [48] offre un état de l'art assez complet sur le sujet.

Agrégation et QoS.

La recherche sur les algorithmes d'agrégation (évoquée dans le paragraphe précédent) porte essentiellement sur l'agrégation topologique — portant donc sur un nombre de sauts ou sur des poids administratifs. Dans le cadre d'un routage QoS, il est également nécessaire d'agrégier d'autres attributs et métriques. Le problème d'agrégation multi-critères a été très peu étudié [48, 43]. Les métriques et attributs sont donc souvent résumées en suivant les étapes d'un processus d'agrégation basé sur les poids administratifs... Cette technique n'est bien sûr pas cohérente lorsque le routage sous-jacent privilégie le partage de charge (routage basé prioritairement sur la bande passante résiduelle en dépit du nombre de sauts, voir section précédente).

L'agrégation de paramètres de QoS pose également la question du choix d'un niveau d'agressivité. Lorsque plusieurs chemins possibles existent entre deux "ports", il est par exemple possible de choisir la bande passante disponible sur chemin le plus congestionné, ce qui définit une stratégie "conservatrice". Il est possible au contraire de choisir la bande-passante disponible sur le chemin le moins congestionné, où l'on obéit alors à une politique "agressive". Des résultats de simulations [43] semblent montrer qu'une politique agressive est préférable afin de réduire les taux de pertes d'appels entrants. Dans le chapitre 7, nous revenons sur la notion d'agressivité du processus d'agrégation. Nous proposons un modèle permettant de quantifier cette notion pour en étudier l'impact³³ sur les performances globales du réseau. Nous verrons que les résultats trouvés montrent également que le choix d'une stratégie agressive est préférable du point de vue de l'utilisation des ressources.

4.3.3 Algorithmes de routage hiérarchiques.

Fonctionnement.

N'oublions pas que le routage doit être assuré malgré la structure hiérarchique et les pertes d'informations topologiques dues à l'agrégation. Le protocole PNNI définit une procédure générique permettant d'assurer un routage cohérent (même s'il est éventuellement sous-optimal) dans une hiérarchie complexe. La procédure d'ouverture de connexion est en fait distribuée. Notons tout d'abord que le protocole de routage de PNNI est de type "source-routing" c'est à dire que la décision de routage est effectuée par le premier commutateur. La liste des noeuds à traverser (DTL³⁴) est incluse dans le message d'ouverture de connexion. Les noeuds intermédiaires suivent ensuite le chemin décrit dans ce message. Dans une structure hiérarchique, le noeud source détermine les domaines à traverser. Cette information de routage est ensuite

³³L'impact de la fonction GCAC et du reroutage sont également traités.

³⁴DTL : Designated Transit List

raffinée à chaque entrée dans un nouveau domaine, ou le chemin permettant de traverser le nouveau peer-group (en direction du prochain domaine, selon les indications du noeud source) est sélectionné.

Imaginons par exemple le cas de la figure 4.3 où le noeud A.1.1 désire communiquer avec le noeud B.3. Le DTL généré par le premier commutateur sera le suivant :

$$[[A.1.1 A.1.2] [A.1 A.2] [A B]]$$

A l'entrée dans le peer-group PG A.2, le DTL est analysé et un chemin permettant d'atteindre le PG B est sélectionné : [A.2.1 A.2.3]. Une fois arrivé au noeud B.1, le chemin [B.1 B.3] est finalement choisi.

Dans [54, 43], différents algorithmes de routage adaptés au protocole PNNI sont discutés. Une approche particulièrement élégante (décrite en détail dans [54]) consiste tout d'abord à déterminer le peer-group de niveau le plus haut que la source et la destination ont en commun (on parle "d'ancêtre" commun par analogie avec un arbre généalogique). A l'intérieur du peer-group ainsi sélectionné, un premier calcul de routage permet de sélectionner les domaines du niveau hiérarchique inférieur $n - 1$ qu'il faudra traverser — [A B] dans l'exemple précédent. A l'intérieur du premier domaine de niveau $n - 1$, les domaines de routage de niveau $n - 2$ permettant d'atteindre le second domaine de niveau $n - 1$ peuvent alors être déterminés — [[A.1 A.2] [A B]]. La construction du DTL se poursuit récursivement en descendant les niveaux hiérarchiques jusqu'à atteindre le niveau physique — nous arrivons à [[A.1.1 A.1.2] [A.1 A.2] [A B]] dans l'exemple précédent. Il est possible de trouver une description d'un approche non récursive dans [63, Annexe H].

Il a été montré dans [53] que la complexité globale associée au routage d'une connexion est directement affectée par la structure hiérarchique choisie. Des structures spécifiques permettant de minimiser la complexité de ces calculs ont été découvertes par le même auteur. Nous reviendrons sur ces notions dans le chapitre 6.

Crankback.

Remarquons finalement qu'il est possible qu'une connexion soit refusée sur le chemin sélectionné par l'algorithme de routage, à cause des inexactitudes des informations de routage (dues aux processus de rafraîchissement et d'agrégation). Dans le protocole PNNI, une procédure de **crankback** permet de libérer les ressources réservées sur le chemin défectueux (jusqu'au dernier noeud ayant pris une décision de routage). Il est alors possible de choisir une route alternative afin d'éviter le lien bloquant³⁵.

³⁵Des algorithmes de routage permettant de choisir des routes disjointes ou possédant le moins de lien en commun est étudié dans [45].

4.4 Organisation de cette partie.

Motivations.

Les trois chapitres suivants traitent de problèmes différents liés au routage dynamique) que nous présentons rapidement dans cette section. Tous les aspects sont bien sûr entremêlés même si nous avons dû les traiter séparément par souci de clarté — et pour obéir au second principe cartésien. Même si ces études diffèrent par l'analyse choisie ou les problèmes traités, remarquons qu'ils répondent tous à une même problématique globale évoquée en introduction : celle du dimensionnement de réseau. Nous cherchons à comprendre l'impact des paramètres des protocoles de routage sur les performances globales du réseau. Nous cherchons des règles simples et génériques permettant de trouver une paramétrisation optimale conduisant au compromis performance/surcoût désiré.

Analyses menées.

Comme nous l'avons vu tout au long de ce chapitre, de nombreuses questions se posent sur les performances de réseau utilisant des protocoles de routage dynamique. Nous avons mis en évidence l'impact de la politique de rafraîchissement des informations de routage sur la qualité des décisions de routage. Nous avons vu également que les résultats de simulations disponibles aujourd'hui donnent de nombreuses indications sur les politiques à adopter. Pourtant, les études actuelles ne permettent pas encore de choisir précisément les paramètres mis en jeu (période des échanges ou taux différentiel etc.) selon les performances et le surcoût protocolaire attendu.

Dans le chapitre suivant, nous proposons un modèle simple dans lequel nous étudions l'impact de l'acuité des informations de routage sur les performances globales du réseau. Cette étude analytique permet également d'étudier l'impact du reroutage ou de la fonction GCAC dans le contexte du routage PNNI. Le modèle sous sa forme actuelle n'est pourtant pas totalement achevé. Il s'agit d'une première étape dans la réalisation de méthodes analytiques permettant une paramétrisation adéquate de protocoles de routage QoS en fonction du mode de fonctionnement désiré.

Nous avons également mis en évidence l'importance jouée par la hiérarchie et l'agrégation pour réduire le sur-coût associé aux protocoles de routage dynamique. Dans le chapitre 6, nous étudions comment la structure hiérarchique choisie agit sur la quantité d'information à manipuler ou sur la complexité des calculs de routage. Ces résultats quantitatifs permettent de choisir une structure hiérarchique adaptée aux contraintes technologiques (capacités de

traitement, quantité de mémoire disponible) du matériel utilisé.

Finalement, nous avons vu la difficulté que représente le développement d’algorithmes d’agrégation dans un réseau offrant des garanties de QoS. Nous avons pu notamment souligner l’impact important joué par l’agressivité de la politique d’agrégation sur les performances du réseau. Dans le chapitre 7, nous proposons un modèle simple et générique permettant de modéliser un processus d’agrégation par son agressivité. Il nous est alors possible d’observer les performances globales du réseau en fonction de ce paramètre — conjointement à d’autres paramètres contrôlant le re-routage ou la structure hiérarchique par exemple. Les résultats obtenus semblent aller en faveur de politiques agressives (liées à des contrôles sévères sur le re-routage) et mettent en cause le bon fonctionnement de la fonction **GCAC**³⁶ dans un réseau hiérarchique.

Contexte de l’étude.

Il est important de noter que les chapitres suivants se placent dans le cadre du protocole PNMI. Les résultats obtenus sont pourtant génériques, les différences qui existaient entre le monde ATM et le monde Internet diminuant considérablement avec le temps.

La seule véritable différence repose en fait sur le fait que nous supposons un protocole de routage basé à la source (et non pas par sauts comme c’est actuellement le cas dans l’Internet). Remarquons qu’il est essentiel d’utiliser un routage basé à la source dans un réseau supportant la QoS afin de préserver la réservation de ressources effectuée sur le chemin initial, malgré d’éventuels changements des tables de routage — on parle de “pinned route” à l’IETF. De nombreux travaux sont donc en cours pour permettre de réaliser un routage à la source dans l’Internet.

Le protocole **SDR**³⁷ permet à un flux d’être acheminé suivant un chemin déterminé — le chemin est placé dans l’en-tête du paquet, comme c’était le cas dans les réseaux token-ring. Ce protocole est encore peu développé dans le réseau IPv4. Il fait par contre partie intégrante des systèmes basés sur IPv6. Les nouvelles architectures IP en cours de définition, comme **MPLS**³⁸ vont permettre également de faciliter la gestion et l’acheminement de “flux”. Le routage QoS basé à la source (à l’entrée dans le réseau **MPLS**) devrait connaître un essor très important avec ce type d’architecture.

³⁶ou du “pruning” dans des solutions proposées pour IP.

³⁷SDR : Source Demand Routing [74].

³⁸MPLS : MultiProtocol Label Swapping, voir <http://www.ietf.org/html.charters/mpls-charter.html>.

Chapitre 5

Analyse d'un réseau plat

Dans ce chapitre, nous nous concentrons sur un modèle simple de réseau plat, c.à.d. non hiérarchique. Nous nous plaçons dans le cadre du routage PNNI [63] évoqué dans le chapitre précédent, bien que certains aspects de cette étude restent valable pour le monde IP (cf. chapitre précédent). Une étude analytique simple est donnée, afin d'étudier spécifiquement l'impact du mécanisme de rafraîchissement des informations de routage sur les performances du réseau considéré. Ce travail a été réalisé en collaboration avec Arturo Ragozini, pendant ses deux séjours à Paris.

Le chapitre présent est organisé de la manière suivante : Le modèle utilisé est tout d'abord présenté dans la section 5.1. Les différentes variables décrivant les performances du système sont ensuite déterminées analytiquement dans la section 5.2. Enfin, les résultats obtenus sont présentés et analysés dans la section 5.3. Quelques commentaires et une conclusion viennent alors clore ce chapitre.

5.1 Présentation du modèle

Le modèle présenté ci-dessous s'appuie souvent sur des résultats de géométrie aléatoire. Nous ferons appel à de nombreuses références à l'annexe B qui présente de manière détaillée les concepts sur les pavages du plan utilisés dans cette étude.

5.1.1 Structure du réseau

Les nœuds des réseau — c.à.d. des commutateurs ATM dans le contexte de PNNI — seront représentés par un processus ponctuel homogène de Poisson (cf. section A.3 en annexe), noté π , d'intensité supposée finie λ . Sans perte de généralité, nous considérerons uniquement les

points de ce processus qui appartiennent à disque de rayon $\frac{1}{\sqrt{\pi}}$ centré à l'origine. Le choix de ce rayon permet de disposer d'une surface d'aire unitaire — il y a donc λ nœuds dans le réseau en moyenne.

La connectivité entre ces nœuds est modélisée par le graphe de Delaunay (décrit en annexe à la section B.4). Une représentation graphique d'un tel graphe est donnée par la figure B.5 à la page 210.

5.1.2 Informations de routage.

Les liens du graphe de Delaunay sont marqués par la bande passante résiduelle qu'ils peuvent offrir. Les autres métriques usuelles, comme le délai, la jige ou le taux de perte, ne seront pas considérés. Un lien arbitraire du graphe de Delaunay indexé par i est donc marqué par une variable aléatoire X_i à valeur dans $[0, C]$, dans chaque direction — Notons que ceci sous-entend que nous considérons des liens de même capacité.

Comme expliqué dans le chapitre précédent, il n'est pas possible de transmettre continuellement les valeurs des bandes passantes des différents liens aux autres nœuds du réseau. Ces informations sont donc transmises périodiquement (avec une fréquence pas trop élevée) ou uniquement lors d'un “événement majeur”. Les nœuds du réseau n'ont donc à leur disposition que des estimations, notées \tilde{X}_i , pour prendre leurs décisions de routage.

Nous supposons les variables $\{\tilde{X}_i\}$ i.i.d. Il apparaît clairement que la distribution conjointe de X_i et \tilde{X}_i est une caractéristique du mécanisme de rafraîchissement des informations de routage. Dans toute la suite, aucune hypothèse sur cette distribution n'est nécessaire¹. Nous verrons en fait que les résultats ne dépendent que des deux paramètres suivants (dépendants de cette distribution) : $p = \mathbb{P}\{\tilde{X} > b\}$ et $q = \mathbb{P}\{X > b | \tilde{X} > b\}$ si b désigne la bande passante demandée par l'appelant. Cette bande passante b représente aussi bien un débit crête, un débit moyen ou une bande passante effective (de l'anglais “effective bandwidth” [105]), en fonction de la politique d'admission d'appels choisie.

Afin d'utiliser le modèle présenté pour le dimensionnement de réseau, il faut pouvoir estimer ces deux paramètres en fonction des caractéristiques du mécanisme de rafraîchissement. Cet aspect est discuté à la fin de ce chapitre. On peut aussi tout simplement mesurer ces paramètres, pour un b donné, par des simulations ou sur des réseaux réels.

¹Nous supposons par contre qu'il existe un état stationnaire pour qu'une telle distribution existe. des extensions pour des régimes “cyclo-stationnaires” sont envisagées.

5.1.3 Algorithme de Routage.

Comme expliqué dans le chapitre précédent, il n'est pas toujours possible de lancer un algorithme de routage QoS "à la demande" à chaque arrivée d'appel (ou de flux dans le cas de réseaux IP). Des tables de routage précalculées, correspondant à différentes plages de bande passante, et pour chaque destination possible, sont donc stockées en mémoire. Nous considérerons un mécanisme de routage fonctionnant à base de routes précalculées uniquement. Autrement dit, si aucune route ne correspond, l'appel est directement rejeté, sans faire appel à un éventuel calcul de routage propre pour cet appel.

Nous avons aussi montré dans le chapitre 4 la difficulté du choix d'un algorithme de routage QoS (et nous savons que dans PNNI, ils ne sont pas spécifiés). Les algorithmes donnant le plus d'importance au nombre de sauts (par rapport à la bande passante disponible) sont préférables pour une bonne utilisation des ressources. Nous considérerons en fait l'approche extrême² où les chemins sont précalculés en fonction des poids administratifs uniquement. Autrement dit, il s'agit d'un algorithme classique de plus court chemin, qui ne tient pas compte des paramètres de QoS.

Les paramètres de QoS sont ensuite utilisés par le premier nœud afin de déterminer l'ensemble des chemins "admissibles" a priori : Grâce à la fonction de GCAC, une route r est considérée comme "admissible" si elle dispose, a priori, de suffisamment de bande passante — c.à.d. si $\tilde{X}_i > b \forall i \in r$, où b désigne, cette fois encore, la quantité de bande passante désirée par l'utilisateur.

Notons finalement que les graphes de Delaunay utilisés ont un fort degré de connectivité³, de sorte qu'il existe un grand nombre de chemins possibles entre chaque nœud source et destination. Nous ferons l'hypothèse qu'il est toujours possible de trouver K chemins parfaitement disjoints et de même longueur (longueur minimale en terme de nombre de sauts) pour traverser le réseau. On peut voir sur quelques exemples que cette hypothèse est vraie lorsque $K = 2$ ou 3 et constitue une approximation pour de plus grandes valeurs de K — approximation d'autant meilleure que K est faible, bien sûr.

5.1.4 Crankbacks

Comme nous l'avons vu, les routes sont choisies par le premier nœud d'après les estimations sur la congestion des liens (forcément inexactes) dont ils disposent, c.à.d. d'après les variables $\{\tilde{X}_i\}$. Il est donc possible que le chemin sélectionné ne dispose finalement pas des ressources suffisantes, auquel cas une procédure de "crankback" est mise en œuvre. Le nœud d'entrée

²Approche extrême, certes, mais bien réelle. Il semble, par exemple, que cette solution ait la préférence des développeurs d'un grand constructeur suédois par exemple.

³Par connectivité, il faut entendre le nombre moyen de voisins directement connectés.

dispose alors de la possibilité de sélectionner une route alternative admissible, si une telle route existe. Afin de limiter le surcoût protocolaire lié à la signalisation, le nombre de crankbacks doit être contrôlé. Nous introduisons donc le paramètre K_c , représentant le nombre maximal de crankbacks autorisé pour un appel donné.

Le procédé régissant le routage et le re-routage d'un appel est résumé sous la forme d'un pseudo-code dans les tables 5.1 et 5.2. Dans cette description, \mathbf{R} (retourné par la fonction `get_precomputed_path()`) correspond à une liste de chemins, c'est à dire une liste de liste. Si p est un index, alors $r = R[p]$ correspond donc au $p^{\text{ième}}$ chemin. Il s'agit d'une liste des nœuds à traverser. Enfin, `call` est une structure complexe que nous ne spécifierons pas et qui décrit les différents paramètres nécessaires à l'ouverture de la connexion. En particulier, `call.dest` identifie le destinataire souhaité et `call.QoS` la qualité de service désirée.

A la réception d'une demande d'ouverture de connexion (message `set_up`) par une station, le nœud en question récupère tout d'abord la liste des chemins précalculés en direction de la destination souhaitée (grâce à la pseudo-commande `get_precomputed_path()`). De ces chemins ne sont conservés que les chemins "admissibles", la sélection étant effectuée par la fonction `GCAC`. Ensuite, le nœud s'assure qu'il existe un chemin dont le premier lien a assez de ressources pour accepter l'appel (fonction CAC^4) grâce à la fonction `find_suitable_path()`. Si un tel chemin existe, les ressources sont réservées et la demande de connexion transmise au prochain nœud. Le "context" (routes admissibles et quelques variables utiles au routage de l'appel) est alors stocké afin de pouvoir traiter rapidement un éventuel message de crankback pour cet appel⁵. Si aucun chemin approprié n'a pu être déterminé, l'appel est rejeté.

A la réception d'un message `set_up` par un nœud sélectionné, la fonction `CAC` est utilisée afin de s'assurer que les ressources disponibles sont effectivement suffisantes pour accepter l'appel. Dans le cas contraire, un message de `crankback` est généré et envoyé en direction du nœud source, libérant les ressources réservées sur son passage.

Enfin, si le nœud source reçoit un message de `crankback`, le contexte de l'appel est récupéré afin de déterminer si une route alternative existe. Si ce n'est pas le cas, l'appel est rejeté.

5.2 Analyse

Dans cette section, les principaux critères de performance sont exprimés en fonction des paramètres du modèle — en particulier, en fonction de $q = \mathbb{P}\{X > b \mid \tilde{X} > b\}$ qui représente l'acuité des informations de routage.

⁴CAC : Call Admission Control.

⁵Ces variables sont libérées à la réception du message `connect` (spécifiant le succès de l'ouverture de la connexion) ou si l'appel est rejeté.

```

switch(received_mesg)
case uni.set_up(call)
  p=1; i=1; R=GCAC(call.QoS,get_precomputed_path(call.dest));
  found=find_suitable_path(call.QoS,R,p);
  if found==FAIL then %%% reject Call %%%
  else
    %%% Reserve ressources %%%
    r=R[p];
    send mni.set_up(call,r,i+1) to r[i+1]; store_context(call)(R,path,0);
  endif
case mni.crankback(call,r,1)
  %%% Release Ressources %%%
  (R,p,c)=get_context(call); p=p+1; c=c+1; found=find_suitable_path(call.QoS,R,p);
  if c<=Kc and found==OK then
    %%% Reserve ressources %%%
    send mni.set_up(call,r,i+1) to r[i+1]; store_context(call)(R,p,c);
  else %%% reject Call %%% release_contest(call);
  endif
case ...

function boolean find_suitable_path(QoS,R,p)
  r=R(p);
  while p <= length(R) and not CAC(QoS,r[2]) do
    p=p+1; r=R[p]; end
  if p>length(R) return(FAIL) else return(OK) endif
endfunction

```

TAB. 5.1 – Pseudo-Code : Comportement du nœud source.

```

switch(received_mesg)
case nni.set_up(call,r,index)
  if CAC(call.QoS,r[index+1])) then
    %% Reserve ressources %%
    send nni.set_up(call,r,index+1) to r[index+1];
  else
    send nni.crankback(call,r,index-1) to r[index-1];
  endif
case nni.cranckback(call,r,index)
  %% Release ressources %%
  send nni.crankback(call,r,index-1) to r[index-1];
case ...

```

TAB. 5.2 – Pseudo-Code : Comportement d'un nœud sélectionné.

5.2.1 Préliminaires

Soit M le nombre minimum de sauts nécessaire à la traversée du réseau considéré (calculé par un algorithme de plus court chemin). Les K chemins précalculés pour la destination sont supposés être disjoints et posséder le même nombre de sauts, à savoir M . Il faut noter que M est une variable aléatoire (étant donné que nous considérons des graphes aléatoires) dont la distribution est analysée en annexe à la section B.4.2. Dans la suite des calculs, nous considérerons d'abord M donné et calculerons donc des moyennes et des probabilités conditionnelles. Ensuite, ces différentes quantités seront intégrées par rapport à la distribution de M afin d'obtenir les résultats recherchés.

Les différents chemins étant considérés disjoints, posons $P = p^M$ et $Q = q^M$. Ces variables représentent, respectivement, la probabilité qu'une route soit sélectionnée par la fonction **GCAC** et la probabilité que l'appel soit accepté sur le chemin choisi.

Proposition 5.1 *Soit Υ_r l'événement "GCAC trouve r routes admissibles (sur les K routes disponibles)". Alors,*

$$\mathbb{P}(\Upsilon_r) = \binom{K}{r} P^r (1 - P)^{K-r}$$

Preuve. Comme montré précédemment, la fonction **GCAC** sélectionne une route avec une probabilité $P = p^M$. De plus, les K routes sont supposées disjointes, c.à.d. indépendantes, d'où la distribution binomiale présentée. \square

Corollaire 5.1 *Soit R le nombre de routes admissibles sélectionnées par la fonction GCAC et \mathcal{R} sa moyenne. On a donc :*

$$\mathcal{R} = KP$$

5.2.2 Perte d'appel.

Soit \mathcal{P}_b la probabilité qu'un appel soit bloqué, quelqu'en soit la raison. Cet événement peut se produire parce que le nœud source ne peut trouver aucun chemin admissible ou parce que l'appel a subis un (ou plusieurs) crankback(s) et qu'aucune route alternative n'a pu être trouvée. Cette probabilité correspond à l'action `reject Call` dans la table 5.1.

Proposition 5.2 *La probabilité de blocage est donnée par l'expression suivante :*

$$\mathcal{P}_b = \sum_{r=0}^K \binom{K}{r} P^r (1-P)^{K-r} (1-Q)^{r \wedge K_c} \quad (5.1)$$

où $a \wedge b = \min\{a, b\}$.

Preuve. Il suffit de remarquer que, sachant Υ_r ,

$$\mathcal{P}_b | \Upsilon_r = \mathbb{P}\{N_c = r \wedge K_c | \Upsilon_r\} = (1-Q)^{r \wedge K_c}$$

□

Remarque 5.1 Remarquons que pour $K_c = K$, l'expression précédente prend la forme simple suivante : $\mathcal{P}_b = (1-PQ)^K$.

5.2.3 Rejet de l'appel au nœud source.

Comme nous l'avons vu, un appel peut être directement rejeté par le premier nœud si jamais la totalité des chemins précalculés sont éliminés par la fonction GCAC.

Proposition 5.3 *La probabilité de rejet est donnée par l'expression suivante :*

$$\mathcal{P}_r = (1-Pq)^K \quad (5.2)$$

Preuve. \mathcal{P}_r correspond à l'événement Υ_0 , c.à.d. "aucune route n'est admissible", d'où la forme $(1-P)^K$. Le terme q doit ensuite être rajouté dans la mesure où un chemin n'est véritablement sélectionné que si son premier lien dispose effectivement d'assez de bande passante — comme le montre la table 5.1 où fonction CAC est appelée par la fonction `find_suitable_path()`. □

5.2.4 Crankback.

Un appel peut être bloqué sur un chemin sélectionné (c.à.d. sur un chemin qui disposait, à priori, des ressources nécessaires). Dans ce cas, une procédure de crankback est initialisée. Soit \mathcal{P}_c la probabilité qu'un appel subisse au moins un crankback (qu'il soit ensuite rejeté ou non). Introduisons également \mathcal{N}_c , le nombre moyen de crankbacks subi par un appel.

Proposition 5.4 *Le nombre moyen de crankbacks subi par un appel est donné par l'équation suivante :*

$$\mathcal{N}_c = \frac{1-Q}{Q} (1 - \mathcal{P}_b) \quad (5.3)$$

Preuve. Il est facile de voir que

$$\mathbb{P}\{N_c = n | \Upsilon_r\} = (1-Q)^n Q \mathbf{1}_{\{n < r \wedge K_c\}} \mathbf{1}_{\{n \leq r \wedge K_c\}}$$

ce qui conduit à la moyenne conditionnelle suivante :

$$\mathbb{E}[N_c | \Upsilon_r] = \frac{1-Q}{Q} \left[1 - (1-Q)^{r \wedge K_c} \right] = \frac{1-Q}{Q} [1 - \mathcal{P}_{b|\Upsilon_r}]$$

où nous avons identifié l'expression de la probabilité de blocage, conditionnellement à Υ_r (exprimée dans la preuve de la proposition 5.2). Le résultat annoncé est alors obtenu après intégration par rapport à Υ_r . \square

Finalement, un calcul trivial conduit à l'expression de la probabilité de crankback.

$$\mathcal{P}_c = \mathbb{P}\{N_c > 0\} = 1 - Q \quad (5.4)$$

5.2.5 Surcoût

Afin de mieux évaluer le surcoût causé par les crankbacks, nous introduirons une nouvelle quantité correspondant aux nombres de liens visités inutilement pendant la phase d'ouverture de connexion. Autrement dit, cette quantité représente le nombre de liens sur lesquels une réservation de ressource a été effectuée, avant d'être ensuite libérée par une procédure de Crankback. Pour mieux appréhender le surcoût que ces réservations inutile représente par rapport au nombre de liens qu'il est nécessaire de traverser pour atteindre la destination, nous nous intéresserons au ratio η , normalisé par le paramètre M .

Lemme 5.1 *Le nombre moyen de liens visités sur une route donnée est :*

- M , si l'appel est accepté sur cette route.
- $\mathcal{N}_l < M$, si l'appel est refusé et subit donc un crankback, avec

$$\mathcal{N}_l = \frac{q}{1-q} - M \frac{q^M}{1-q^M}$$

Preuve. Le premier point est évident. Pour le second, il faut tout d'abord remarquer que \mathcal{N}_l correspond à une moyenne conditionnelle, par rapport au fait que l'appel n'est pas accepté. Il faut donc normaliser par le terme $1 - q^M$. On voit alors que N_l suit une loi géométrique tronquée :

$$\mathbb{P}\{N_l = i | \text{Crankback}\} = \frac{q^i (1-q)}{1-q^M} \quad \forall i = 0, \dots, M-1$$

Il suffit alors de calculer la moyenne associée à une telle distribution. □

Proposition 5.5 *Le ratio de sur-coût η obéit à la relation suivante :*

$$\eta = \frac{\mathcal{N}_c \mathcal{N}_l}{M} \tag{5.5}$$

où $\mathcal{N}_c, \mathcal{N}_l$ sont exprimés à la proposition 5.4, au lemme 5.1 respectivement.

Preuve. A chaque fois que l'appel est bloqué sur un chemin, la procédure de crankback va être utilisée pour libérer les ressources réservées — sur \mathcal{N}_l liens en moyenne. D'où le résultat annoncé. □

5.2.6 Moyennes.

Comme mentionné dans l'annexe B.4.2, la distribution de M n'est pas connue. Par contre, le théorème limite centrale peut s'appliquer à ces chemins, de sorte que la variance de M décroît avec la taille du réseau. Nous considérerons que le réseau est suffisamment "large"⁶ pour considérer M constant. Sous cette approximation, la moyenne des expressions précédentes est obtenue en les évaluant pour $M \approx \sqrt{\lambda} \frac{2}{\sqrt{\pi}}$, où rappelons le, $\frac{2}{\sqrt{\pi}}$ est le diamètre du réseau considéré. Le facteur $\frac{4}{\pi} \sqrt{\lambda}$ se justifie lorsqu'un algorithme de routage markovien est utilisé — cf. paragraphe B.4.2 en annexe.

Les résultats obtenus par ces calculs sont présentés dans la section suivante.

⁶Pour une intensité du processus π donnée.

5.2.7 Routage à l'aveugle

Afin de mettre en évidence le rôle de la fonction **GCAC**, nous comparerons les résultats obtenus précédemment avec ceux d'un algorithme de routage "aveugle", c'est à dire n'utilisant aucunement les informations de congestion émises par le protocole de routage. Autrement dit, un appel entrant est envoyé aléatoirement sur un des chemins précalculés disponibles, sans considérer l'état de congestion de ce chemin. Au cas où la connexion est refusée, une route alternative peut être tirée aléatoirement parmi les $K - 1$ routes restantes, et ainsi de suite.

Remarque 5.2 La difficulté de l'étude du routage "aveugle" consiste dans le fait que la probabilité $\mathbb{P}\{X > b\}$ nous est inconnue — par construction, seule les probabilités $p = \mathbb{P}\{\tilde{X} > b\}$ et $q = \mathbb{P}\{X > b | \tilde{X} > b\}$ sont à notre disposition. Dans la suite, nous considérerons un régime stationnaire⁷ pour lequel $p_0 = \mathbb{P}\{\tilde{X} > b\} = \mathbb{P}\{X > b\}$. Autrement dit, nous considérerons que la bande passante réelle X et son estimée \tilde{X} correspondent à des observations d'un même processus, à des instants différents (plus ou moins proches selon la politique de rafraîchissement). En régime stationnaire, les distributions de X et de \tilde{X} sont donc identiques. Par contre ces deux variables sont bien dépendantes l'une de l'autre de sorte et le paramètre q est en général non nul.

Nous utiliserons les mêmes notations que précédemment à l'exception d'un "chapeau" qui signifie que nous avons à faire au routage "aveugle". Les expressions des différents critères de performance seront donnés sans aucune démonstration — leur expression étant aisée à déterminer en utilisant les approximations et les calculs présentés ci-avant pour le cas où la fonction **GCAC** est utilisée.

Proposition 5.6 *Lorsque la fonction **GCAC** n'est pas utilisée, les performances du réseau sont décrites par les identités suivantes :*

$$\begin{aligned}\hat{\mathcal{P}}_b &= (1 - P)^{K_c} \\ \hat{\mathcal{P}}_r &= (1 - p)^K \\ \hat{\mathcal{P}}_c &= 1 - P \\ \hat{\mathcal{N}}_c &= \frac{1 - P}{P} (1 - \hat{\mathcal{P}}_b) \\ \hat{\eta} &= \frac{\hat{\mathcal{N}}_c \hat{\mathcal{N}}_l}{M} \\ \hat{\mathcal{N}}_l &= \frac{p}{1 - p} - M \frac{P}{1 - P}\end{aligned}$$

où $P = p^M$.

⁷L'inconvénient de cette hypothèse est que nous ne pouvons appréhender le cas où des phénomènes oscillatoires [57] apparaissent. Des extensions considérant des processus cyclo-stationnaires sont envisagées.

5.3 Résultats

5.3.1 Influence de p

Nous considérons dans cette section que l'acuité des informations de routage n'est pas affectée par p . Autrement dit, nous étudions l'impact de p sur les différents critères de performance pour q fixé.

Performances du réseau.

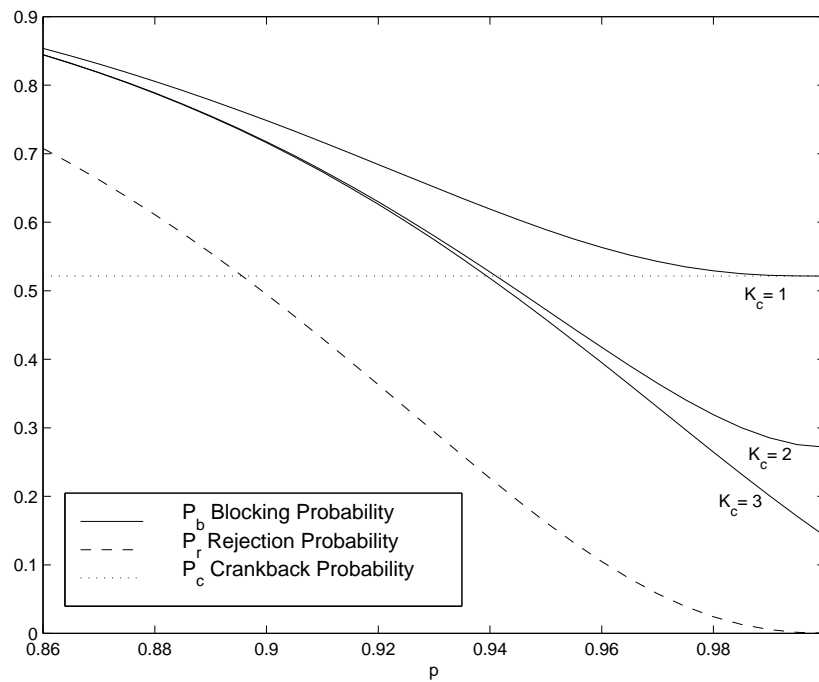


FIG. 5.1 – Influence de p sur les performances du réseau .

La figure 5.1 représente l'évolution de \mathcal{P}_b en fonction de p pour $\lambda = 100$, $K = 3$ et $q = .95$. Ces courbes permettent de mettre en évidence quelques résultats assez intuitifs :

- Lorsque $p \rightarrow 1$, la probabilité de rejet à la source devient nul. En effet, les bandes passantes annoncées sont très importantes de sorte que la fonction GCAC ne refuse jamais aucun appel entrant (on le voit également sur la figure 5.2 où le nombre de routes admissibles devient égal au nombre de routes disponibles, c.à.d $K = 3$). Les appels ne peuvent être refusé qu'après crankback, si le nombre maximal de crankback autorisé (noté K_c) est atteint. Il est donc logique de constater que la probabilité de blocage est très nettement affectée par le paramètre K_c . Pour $K_c = 1$ et $p = 1$, on retrouve

bien $\mathcal{P}_b = \mathcal{P}_c$. Cette quantité décroît significativement si on autorise un crankback supplémentaire. Par contre, l'utilité d'un troisième crankback est plus discutable, sauf pour des valeurs extrêmes (c'est à dire $p \approx 1$).

- Lors p décroît, \mathcal{P}_r (la probabilité de rejet par la fonction **GCAC**) atteint rapidement un niveau inacceptable : $\mathcal{P}_r \approx 0.7$ pour $p \approx .85$ par exemple. Le rejet à la source devient alors la cause majeure de pertes des appels. L'influence des crankbacks n'est plus très significative. On voit d'ailleurs que le paramètre K_c n'influe quasiment plus sur les valeurs prises par \mathcal{P}_b .

Il faut noter que le paramètre p dépend de deux facteurs : La charge du réseau et la quantité de bande passante b demandée par les appels entrants. Nous pouvons donc remarquer que pour une charge donnée, autoriser plusieurs crankbacks peut permettre d'accepter un plus grand nombre d'appels — dans la limite du possible, selon le surcoût protocolaire que l'on désire. Néanmoins, pour des appels exigeants en bande passante, il n'est peut-être pas nécessaire d'autoriser un grand nombre de crankbacks. En effet, la probabilité de blocage de ces appels n'en serait pas améliorer significativement et le surcoût dû aux crankbacks ne se justifierait donc plus.

Surcoût protocolaire.

Nous avons pu souligner l'impact bénéfique du crankback sur les probabilités de blocage des appels dans le paragraphe précédent. Les crankbacks engendrent cependant un surcoût protocolaire, en terme de messages de signalisation à échanger sur le réseau et à traiter par les commutateurs. De plus, la bande passante bloquée inutilement jusqu'à l'arrivée du **crankback** aurait pu être utilisée par d'autre appels. Le nombre moyen de crankbacks (\mathcal{N}_c) et la proportion de liens traversés inutilement (η) permettent d'évaluer le surcoût générés par les procédures de crankback.

La figure 5.2 représentée l'évolution de \mathcal{N}_c en fonction de p pour plusieurs valeurs de K_c . Il peut paraître surprenant au premier abord de constater que \mathcal{N}_c diminue lorsque p diminue. En effet, la probabilité de crankback ne dépend pas de p . Cette évolution s'explique par les rejets d'appels à la source : rappelons nous qu'il ne peut y avoir plusieurs crankback que s'il existe des routes alternatives (admissibles). Le nombre de crankback est nul lorsque l'appel est directement rejeté par la source, ce qui explique les valeurs très faibles constatés lorsque $p \ll 1$.

La remarque précédente est également mise en lumière par la similarité des évolutions de \mathcal{N}_c et de \mathcal{R} (nombre moyen de routes admissibles vues par la fonction **GCAC**). L'influence de K_c est donc logiquement peu significative pour $p \ll 1$. En revanche, l'influence de K_c est très importante pour $p \approx 1$. Pour $K_c \approx 3$ et $p \approx 1$, un appel subit en moyenne presque 2

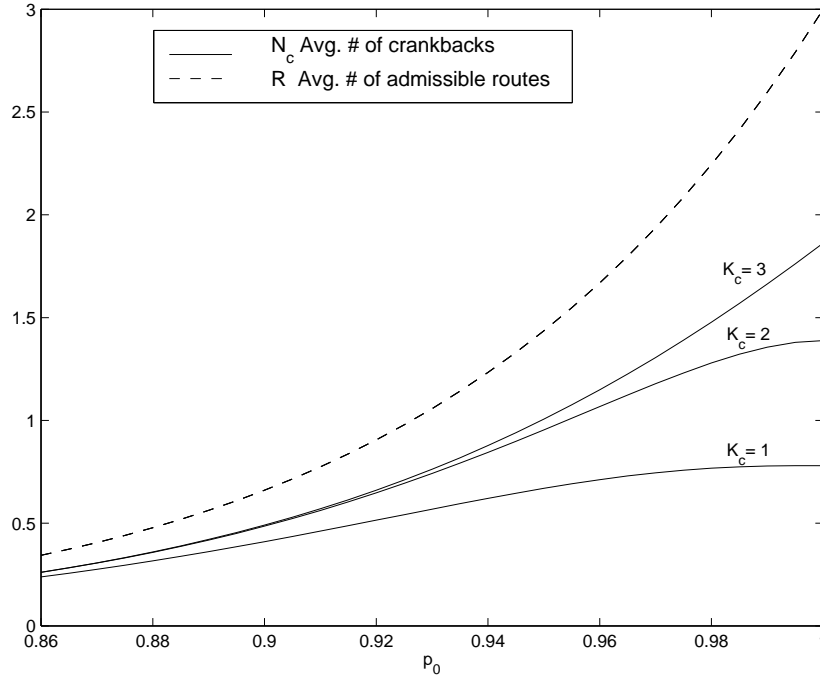


FIG. 5.2 – Influence de p sur le nombre moyen de crankbacks.

crankbacks, qui n'est certainement pas très raisonnable.

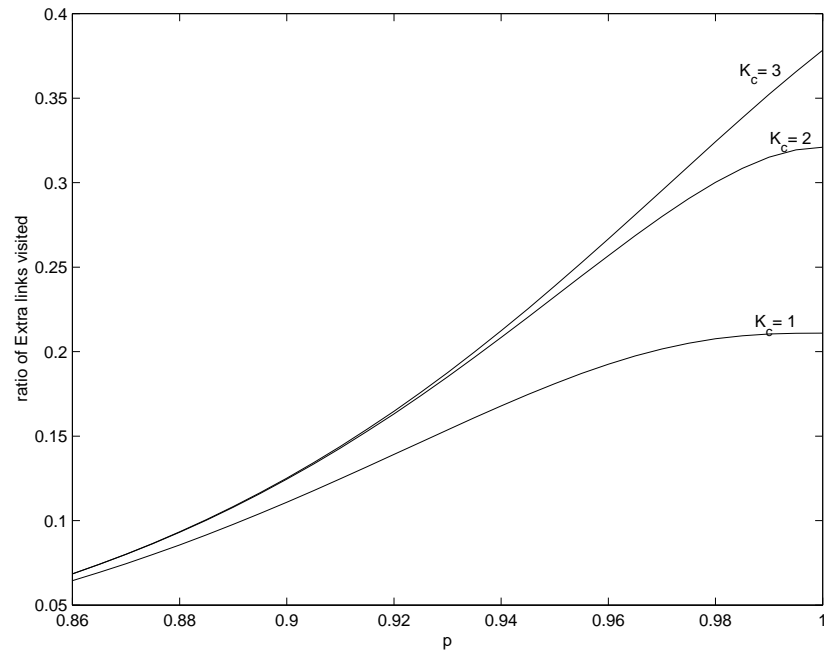
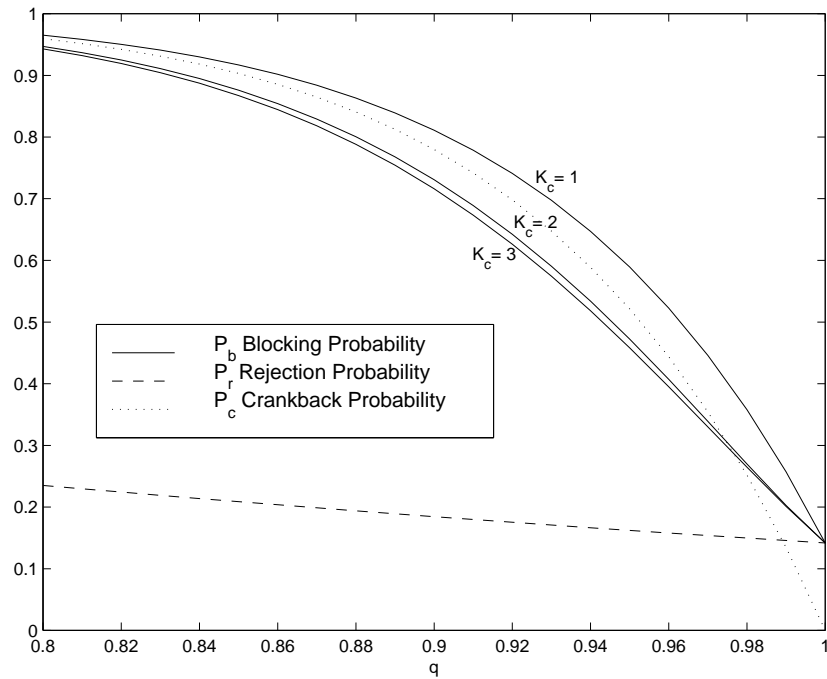
La figure 5.3 montre comment p influe sur le surcoût signalétique η pour plusieurs valeurs de K_c . Là encore, on constate un surcoût faible pour $p \ll 1$ puisque les appels sont majoritairement rejetés à la source par la fonction GCAC. L'influence du paramètre K_c ne devient significative que lorsque l'appel dispose de plusieurs routes admissibles à sa disposition, c'est à dire lorsque p est proche de 1. Cette courbe peut permettre de choisir K_c en fonction de la proportion de liens traversés inutilement que l'architecte du réseau peut tolérer. La probabilité de blocage associée (cf. figure 5.1) peut être alors déterminée.

5.3.2 Influence de l'acuité des informations de routage.

Dans cette section, nous considérons à présent p constant et faisons varier le paramètre q . Autrement dit, nous considérons une charge du réseau donnée et un type d'appel déterminé (à la bande passante fixée) et étudions l'impact des incertitudes engendrées par la politique de rafraîchissement des informations de routage sur le réseau.

Performances.

Sur la figure 5.4 sont représenté les évolutions de \mathcal{P}_b , \mathcal{P}_r et \mathcal{P}_c en fonction de q , pour $\lambda = 100$, $p = 0.95$ et $K = 3$ pour différentes valeurs de K_c . L'influence de K_c n'est pas très

FIG. 5.3 – Influence de p sur le surcoût signalétique.FIG. 5.4 – Influence de q sur les performances du réseau.

significative dans ce cas. Nous avons vu dans la section précédente que l'influence de K_c est grandissante lorsque p tend vers 1. Pour une valeur plus importante de p , nous avons pu mettre en évidence une influence nettement plus significative de K_c .

Pour $q \rightarrow 1$, la probabilité de blocage devient égal à la probabilité de rejet à la source — la probabilité de crankback devenant nulle. Pour $q \ll 1$, la probabilité de crankback atteint des valeurs catastrophiques, causant la perte des appels après avoir testé tous les chemins admissibles (bornés par K_c). Le crankback dans une situation où les informations de routage sont extrêmement grossières ne se justifie plus : Cette procédure ne permet pas d'obtenir de meilleurs taux de blocage, en introduisant néanmoins un surcoût en terme de signalisation (paragraphe suivant).

Surcoût protocolaire.

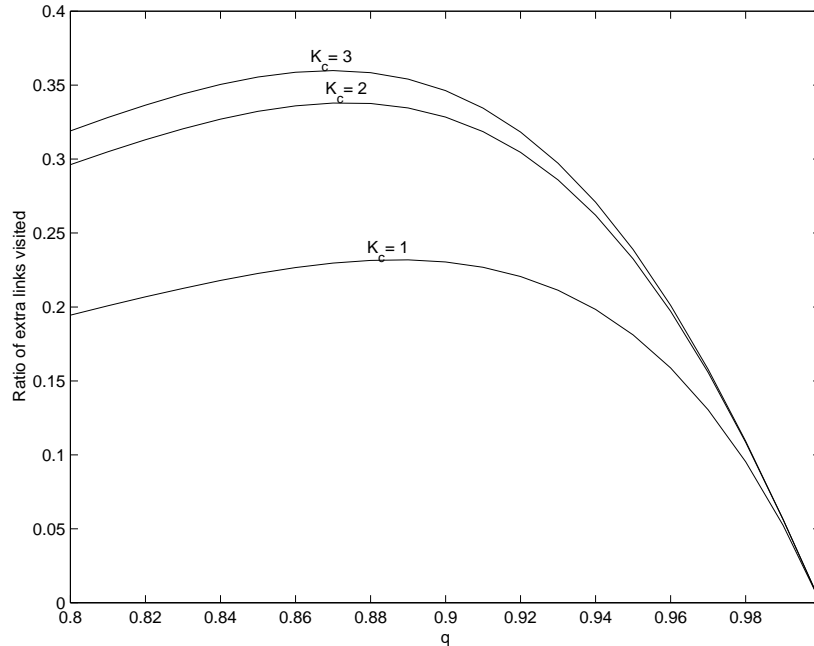


FIG. 5.5 – Surcoût associé aux crankbacks en fonction de q .

Sur la figure 5.5 est représenté l'évolution du surcoût η en fonction de q pour $\lambda = 100$, $p = 0.95$ et $K = 3$ pour diverses valeurs de K_c . Comme nous pouvons le constater, ces courbes ne sont pas monotones et possèdent un maximum avec les valeurs choisies. Ceci s'explique par le fait que η est le produit des deux quantités \mathcal{N}_l et \mathcal{N}_c dont l'une est croissante (et quasi linéaire) et l'autre décroissante. Comme le montre la figure 5.6, \mathcal{N}_c est quasiment constant pour $q \ll 1$ de sorte que η épouse la croissance de \mathcal{N}_l . Lorsque q tend vers 1, la courbe d'évolution de \mathcal{N}_c est de plus en plus incurvée de sorte que $M.\eta$ devient décroissant — \mathcal{N}_l

restant lui quasiment linéaire⁸. L'extremum de η peut être déterminé numériquement. En revanche, il ne peut être obtenu par une équation simple, à cause des expressions analytiques relativement complexes de \mathcal{N}_c et \mathcal{N}_l .

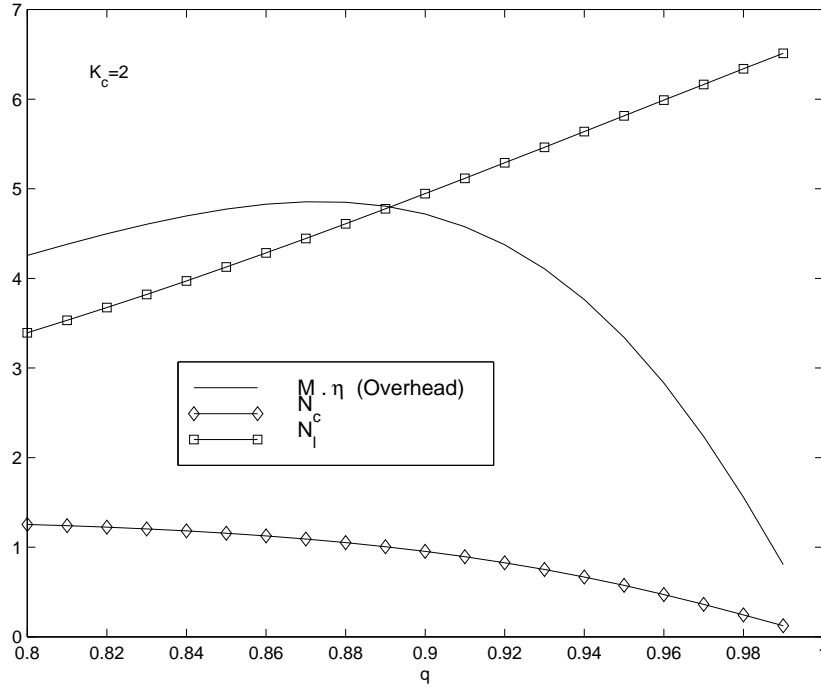


FIG. 5.6 – Surcoût et ses différentes composantes.

5.3.3 Influence de la fonction GCAC

Pour clore cette analyse, nous comparons à présent les performances d'un réseau selon que la fonction GCAC est utilisée ou non. Rappelons que les performances du réseau lorsque cette fonction n'est pas utilisée ont été traitées au paragraphe 5.2.7.

Perte d'appels selon l'acuité q .

Sur la figure 5.7 est représentée l'évolution des probabilités de perte d'appels, selon que la fonction GCAC est utilisée ou non. Les paramètres sont fixés comme suit : $\lambda_0 = 100$, $p = 0.9$, $K = 3$.

Nous pouvons voir que les performances obtenues selon le choix de l'algorithme de routage varie énormément selon l'acuité des informations de routage : Lorsque l'acuité est très grossière, les taux de perte de connexions entrantes sont bien supérieures lorsque la fonction GCAC

⁸Avec les valeurs choisies

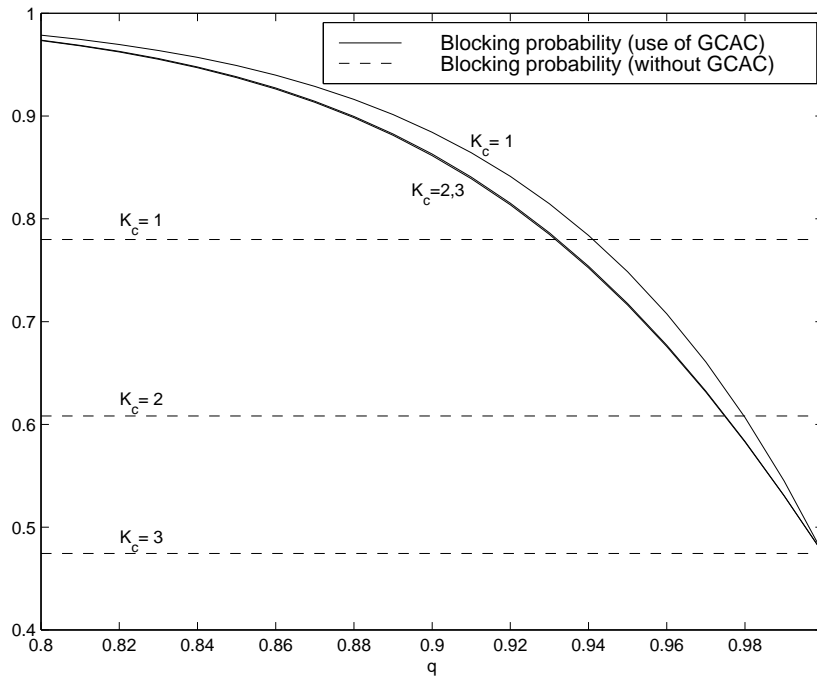


FIG. 5.7 – Perte d’appels selon le choix des routes.

est utilisée. En effet, la préselection de chemins “admissibles” n’est pas judicieuse car les informations sur lequel la sélection est effectuée sont bien trop grossières. Remarquons en revanche que les performances obtenues avec la fonction GCAC peuvent être très intéressantes (inférieures à celle obtenues par le routage aveugle) lorsque l’acuité est bonne. C’est particulièrement le cas lorsque le nombre de crankback est limité ($K_c = 1$ par exemple) : Les connexions sont routées sur les chemins “admissibles” uniquement, c’est à dire sur les chemins ayant une plus grande probabilité d’accepter la connexion. Lorsque $K_c = K$ en revanche — c’est à dire lorsque le nombre de re-routage n’est pas contrôlé, le routage aveugle se comporte toujours mieux. En effet, cette solution a l’avantage de tester la totalité des chemins pré-calculés, contrairement au cas où la fonction GCAC est utilisée (dans ce cas, seules les routes “admissibles” sont sélectionnées). Le routage “aveugle” conduit néanmoins à des probabilités de crankbacks beaucoup plus importantes, comme nous le montrons dans le paragraphe suivant.

Crankbacks et surcoûts associés, en fonction de q .

Sur la figure 5.8 est représenté le nombre moyen de crankbacks subis par les connexions entrantes en fonction du paramètre q . Les traits pointillés correspondent au cas où la fonction GCAC n’est pas utilisée. Les paramètres sont identiques à ceux de la figure précédente.

Comme attendu, le nombre moyen de crankbacks représenté sur la figure 5.8 est très

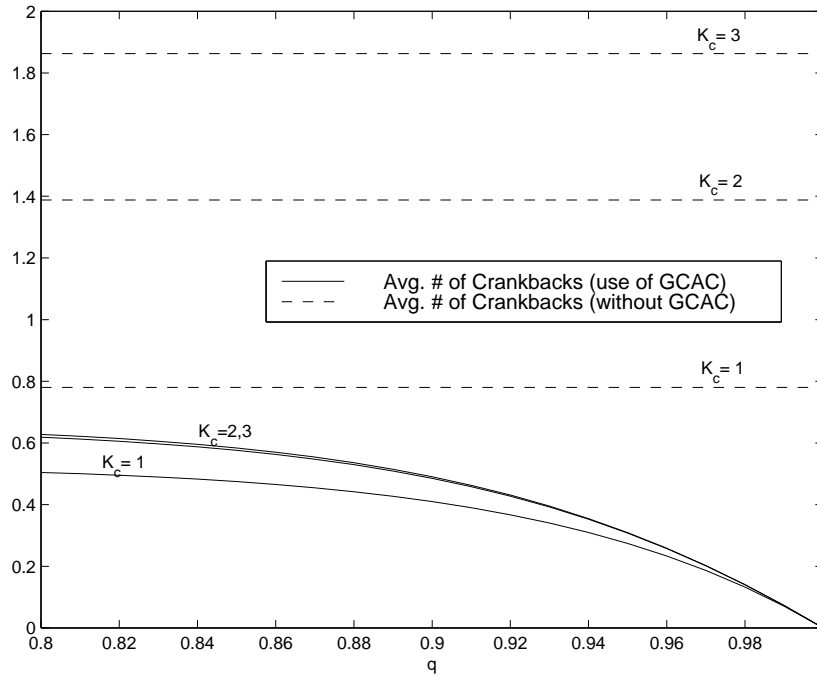


FIG. 5.8 – Nombre de crankbacks selon le choix des routes.

nettement limité grâce à la fonction **GCAC**. Dans cet exemple, nous obtenons un nombre de crankback très limité même pour des informations de routage assez grossières. Remarquons que pour certaines configurations ($p \approx 1$), nous avons pu observer que le nombre moyen de crankbacks subis par les connexions entrantes peut être supérieur lorsque la fonction **GCAC** est utilisée, pour une acuité des informations de routage est très mauvaise ($q < 0.9$). Il faut cependant noter qu'il s'agit d'un cas extrême où q est très faible malgré une probabilité p très élevée ($p = 0.95$). Il nous semble peu réaliste de choisir des valeurs trop grossières d'acuité pour de telles valeurs de p .

5.4 Conclusion et Remarques

Dans ce chapitre, nous avons déterminé l'influence du procédé de rafraîchissement des informations de routage sur les performances globales du réseau. Nous avons en effet identifié en introduction qu'il existe un compromis à trouver entre la qualité du routage et le surcoût associé au protocole de routage dynamique. L'utilisation d'un protocole de routage QoS peut permettre une utilisation optimale des ressources du réseau. En revanche, la qualité des décisions de routage dépend directement de l'acuité des informations échangées — en particulier de la fréquence de ces échanges. Or, plus les échanges sont fréquents et plus la proportion de trafic de contrôle devient importante.

Approche choisie.

Tout d'abord il faut remarquer que les résultats ont été obtenus sous des hypothèses bien spécifiques : Une réseau très fortement connecté (graphe de Delaunay dont la connectivité moyenne est de 6), des chemins disjoints (ce qui n'est pas toujours possible) et une approximation pour permettre de moyenniser les résultats par rapport au graphe aléatoire. Les résultats obtenus ne peuvent donc pas, dans l'état actuel des travaux, être utilisés comme seul outil de dimensionnement de réseau. Des simulations sur des graphes génériques sont nécessaires pour vérifier dans quelles mesures nos hypothèses sont contraignantes. L'approche présentée a l'avantage de reposer sur un modèle extrêmement simple, avec un faible nombre de paramètres, de manière à pouvoir appréhender de manière globale un système si complexe. Les expressions obtenues sont peu complexes et permettent néanmoins de mettre en évidence les rôles joués par l'acuité des informations de routage, par le reroutage ou la fonction GCAC sur les performances du réseau.

Acuité des informations de routage.

Dans ce chapitre, nous avons donné une définition précise et quantifiable à la notion, quelque peu abstraite, "d'acuité" des informations de routage, afin de pouvoir en étudier l'influence globale sur les performances d'un réseau. Cette notion a été modélisée par la probabilité conditionnelle : $q = \mathbb{P}\{X > b \mid \tilde{X} > b\}$.

Les courbes 5.4 et 5.5 permettent de déterminer l'influence de ce paramètre sur le taux de pertes d'appels d'une part et sur le surcoût signalétique d'autre part. Grâce aux équations données précédemment, il est donc possible d'avoir un ordre de grandeur des performances attendues pour une acuité donnée.

Par contre, le modèle, sous sa forme actuelle, est assez difficile à utiliser. En effet, l'acuité q n'est pas un paramètre présent dans les protocoles de routage existants, mais une conséquence de phénomènes complexes. Les paramètres configurables sont plutôt la fréquence d'échange des informations de routage par exemple. Il est donc impossible de prévoir directement les performances d'un réseau en fonction d'une paramétrisation donnée du protocole de routage.

Dans l'attente de résultats sur cet aspect de notre modèle, il est donc nécessaire d'utiliser une approche expérimentale, en évaluant q soit par simulation, soit par des mesures sur des réseaux réels (en comparant la bande passante annoncée avec la bande passante réelle, si ces données sont disponibles).

Des études permettant d'évaluer les paramètres p et q en fonction de la politique de rafraîchissement des informations de routage sont en cours d'étude. Quelques résultats préliminaires [8] ont été obtenus lors d'un stage de fin d'étude qui s'est déroulé à l'ENST. Ces travaux, basés

sur une étude markovienne [103, 104] sont consacrés à l'étude du compromis entre utilisation du réseau et la fréquence de rafraîchissement des informations de routage, dans le cas d'une politique de rafraîchissement proportionnelle (c.à.d apériodique). Nous renvoyons le lecteur intéressé au document cité pour plus de précisions.

Crankback et reroutage.

Au delà de l'acuité des informations de routage, nous avons également pu prendre en compte d'autres facteurs critiques pour les performances du réseau. En particulier, l'influence des crankbacks et du reroutage a pu être étudiée grâce aux paramètres K (nombre de routes précalculées) et K_c (nombre maximal de crankbacks autorisés).

Il apparaît qu'il est possible de réduire très significativement le taux de perte en autorisant qu'un appel soit re-routé au moins un fois (c.à.d pour $K_c > 1$). En revanche, l'apport engendré par les reroutages ultérieurs ($K_c = 3$) est beaucoup moins significatif. Autoriser le reroutage n'est pas sans conséquence sur le surcoût protocolaire puisqu'il est alors possible qu'un appel subisse deux crankbacks successifs. Le choix de ce paramètre doit donc être contraint par le surcoût signalétique qu'il est possible de tolérer.

A la lumière de cette étude, il apparaît également que le re-routage n'est vraiment utile que dans des conditions bien particulières, à savoir $p \approx 1$ et $q \approx 1$. Lorsque la probabilité p est trop faible, le nombre de routes admissibles est trop limité et le reroutage n'est souvent pas possible. Ceci veut dire, en particulier, que le reroutage de connexions friandes en bande passante n'est pas forcément nécessaire. L'acuité des informations de routage joue également un rôle important : Lorsque que ces données deviennent trop grossières, l'apport du reroutage diminue considérablement.

Fonction GCAC.

La comparaison des performances du réseau obtenues avec et sans la fonction GCAC a également pu mettre en évidence le bon fonctionnement de cette technique. La fonction GCAC permet de réduire le surcoût protocolaire — en réduisant la probabilité de crankback. Ceci est dû à la sélection des routes effectuée par cette fonction : Seules les routes ayant une forte chance d'accepter la connexion sont admissibles pour supporter la connexion entrante. Lorsque le nombre de crankbacks est contrôlé (c.à.d. pour $K_c < K$), cette technique peut permettre de réduire le taux de perte des appels lorsque l'acuité des informations de routage le permet — la sélection des chemins permettant de diminuer la probabilité qu'une connexion subisse plusieurs crankbacks (au quel cas, elle serait rejetée).

Le bon comportement de la fonction GCAC contraste énormément avec les pauvres perfor-

mances obtenues dans le cas d'un réseau hiérarchique (cf. chapitre 7). Nous verrons donc que cette fonction n'est pas adaptée à toutes les circonstances. Dans le cas du réseau plat, il faut admettre que cette solution est extrêmement simple et semble donner des résultats satisfaisants lorsque les informations de routage sont suffisamment précises. La réduction du surcoût protocolaire que du point de vue de l'utilisation des ressources.

Notons également que notre étude souffre très certainement d'une lacune. En effet, une fois les routes admissibles sélectionnées, la connexion est routée sur une de ces routes de manière aléatoire. Un algorithme de routage du type "shortest-widest" serait certainement plus judicieux : La connexion serait alors prioritairement routée sur la route annonçant la plus grande bande passante. On peut anticiper que les probabilités de crankbacks seraient alors encore plus faibles. Des extensions de cette étude permettant de modéliser cette politique de routage reste pourtant beaucoup plus difficile. Nous avons pu nous contenter jusqu'à présent d'un unique paramètre $q = \mathbb{P}\{X > b \mid \tilde{X} > b\}$, facile à estimer, alors que l'algorithme "shortest-widest" nécessite la connaissance de la distribution conditionnelle $\mathbb{P}\{X > b \mid \tilde{X} > w\}$ où w est la bande passante disponible sur le chemin choisi ($\forall w > b$).

Chapitre 6

Routage Hiérarchique et Agrégation

Dans ce chapitre, nous nous intéresserons spécifiquement à l'agrégation dans le routage hiérarchique et à l'impact de ces techniques sur le surcoût protocolaire associé au routage dynamique (complexité des calculs, taille des tables de routage à échanger etc.). L'influence de la hiérarchie sur l'utilisation du réseau est analysée dans le chapitre suivant. Le modèle générique de réseau hiérarchisé est présenté dans la section 6.2. L'impact de l'agrégation sur la taille des tables de routage est étudié dans la section 6.3. L'impact de l'agrégation sur la complexité des calculs de routage est enfin étudié dans la section 6.4.

6.1 Introduction

Comme présenté dans le chapitre 4, l'agrégation permet de limiter la quantité d'informations à échanger et à traiter par les nœuds du réseau, tout en assurant un routage de bout-en-bout cohérent. Nous supposons que l'agrégation des informations de routage est purement hiérarchique, c'est à dire que les procédures d'agrégation ne sont utilisées qu'à l'interface entre deux niveaux hiérarchiques (au niveau du *Peer-Group Leader* dans le cas de PNNI par exemple). Nous avons vu en effet dans la section précédente¹ que certains protocoles de routage proposent en effet d'agréger les informations au niveau de chaque nœud dès que ceci est possible (sans avoir recours à la définition d'une quelconque structure hiérarchique).

Il est important de bien comprendre l'impact de l'agrégation hiérarchique sur la quantité d'information échangée, sur la taille des tables de routage et la complexité des calculs de routage, afin de pouvoir dimensionner les aires de routage de manière adéquate.

¹Voir page 85.

6.2 Modélisation

6.2.1 Structure Hiérarchique

Dans ce chapitre, les nœuds du réseau seront représentés par des points distribués aléatoirement sur le plan — bien que seuls les points gisant dans une fenêtre d’observation finie seront observés afin d’obtenir un réseau de taille finie. Les nœuds du réseau seront groupés en aires de routage en utilisant des pavages de Poisson-Voronoi (définis à la page 208).

Plus précisément, soit π_0 un processus de Poisson homogène sur \mathbb{R}^2 d’intensité λ_0 (c.f. section A.3 de l’annexe A). Le support² de ce processus représente les nœuds du réseau considéré. La fenêtre d’observation sera considérée d’aire unitaire, de sorte que le réseau comportera λ_0 nœuds en moyenne.

Définissons alors $\pi_1, \pi_2 \dots \pi_H$ une suite de H processus de Poisson homogènes sur \mathbb{R}^2 , d’intensité respective $\lambda_1, \dots \lambda_H$. Ces processus correspondent aux nœuds logiques de niveau supérieur dans un réseau hiérarchique, à l’exception du dernier processus π_H , si on fait l’hypothèse que ce réseau comporte H niveaux hiérarchiques. Le dernier processus π_H joue en effet un rôle assez spécifique, évoqué plus loin.

Avec ce modèle, un point $x \in \text{supp}(\pi_1)$ correspond à un nœud logique de niveau 1. Ce nœud sera donc utilisé par le protocole de routage pour représenter une aire de routage entière. Dans ce modèle, l’aire de routage dont x est responsable correspond à l’ensemble des nœuds physiques (c.à.d. les points du support de π_0) qui appartiennent à la cellule de Voronoi générée par π_1 et centrée en x et notée $C_x(\pi_1)$ (voir annexe B.3).

Les nœuds du niveau 1 sont à leur tour regroupés dans des aires de routage distinctes, que seront également représentées par les cellules de Voronoi générées par π_2 . Ce procédé est répété itérativement jusqu’au niveau hiérarchique le plus haut, qui est défini les cellules de Voronoi générées par π_H . Nous obtenons donc une infinité de réseaux hiérarchiques (un réseau par cellule de Voronoi généré par π_H). Les grandeurs que nous observerons correspondront à des moyennes calculées sur toutes ces réalisations. Comme montré en annexe A.4, ceci revient à considérer la cellule générée par π_H centrée à l’origine, sous la probabilité de Palm associés à π_H . Cette cellule “typique” représente la fenêtre d’observation W .

Nous poserons $\lambda_H = 1$ pour obtenir une fenêtre d’observation dont l’aire moyenne est unitaire (sans perte de généralité, car il suffit de jouer sur un facteur d’échelle). En effet, l’aire moyenne d’une cellule “typique” de Voronoi est égale à $\frac{1}{\lambda_H}$, comme montré dans l’annexe B.3.3. Un exemple de réalisation de ce modèle est donné sur la figure 6.1. Sur ce graphique, les

²C’est à dire les points de \mathbb{R}^2 associés à ce processus, voir chapitre A.

nœuds physiques sont représentés par de petites étoiles, les nœuds de $\text{supp}(\pi_1)$ par de petits cercles et les points de $\text{supp}(\pi_2)$ par des cercles plus épais.

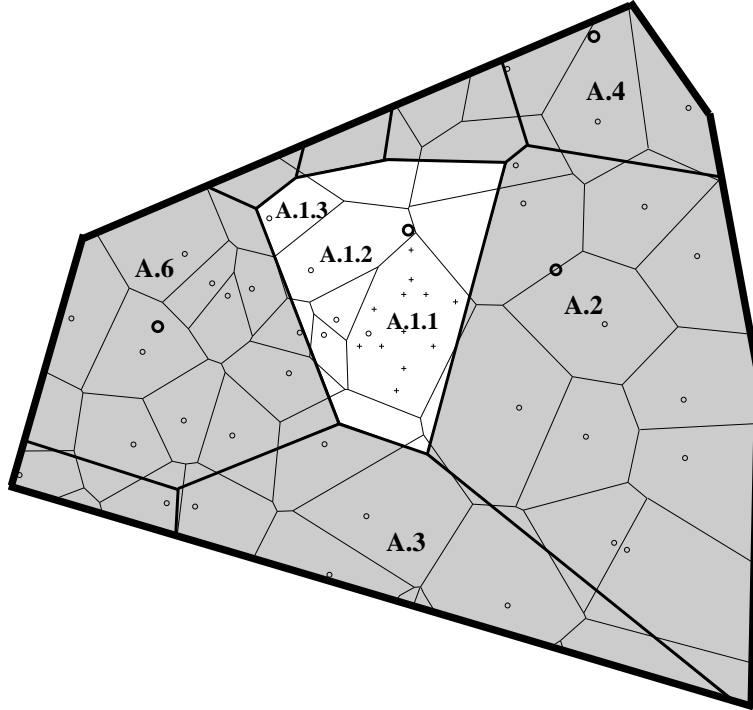


FIG. 6.1 – Réalisation d'un réseau à 3 niveaux hiérarchiques.

Remarque 6.1 Dans la suite de ce chapitre, nous ne ferons aucune hypothèse (sauf mention explicite) sur la connectivité des nœuds de ce réseau. Il n'est en effet pas nécessaire de considérer un graphe de Delaunay, comme nous l'avons fait dans la section précédente. Pour le moment, la seule donnée des nœuds du réseau est suffisante.

6.2.2 Notations

Comme nous l'avons mentionné plus haut, le réseau étudié comporte $N = \lambda_0$ nœuds physiques en moyenne. Dans la suite de cette section, nous introduisons d'autres notations utiles à cette étude.

Soit $m_k(x)$ le nombre de nœuds de l'aire de routage de niveau k contenant le point x (pour tout $x \in \text{supp}(\pi_k)$ et $k = 0, \dots, H$). D'après le modèle utilisé, $m_k(x)$ correspond au nombre de points de $\text{supp}(\pi_k)$ qui appartiennent à la cellule de Voronoi $C_x(\pi_{k+1})$, pour $k = 0, \dots, H - 1$. D'où la relation

$$m_k(x) \doteq \pi_k(C_x(\pi_{k+1}))$$

La moyenne de $m_k(x)$ calculée sur tous les points de $\text{supp}(\pi_k)$ sera notée m_k (c.à.d. sans mention aucune d'un centre d'une cellule de Voronoi particulière). Cette moyenne empirique correspond, d'après la définition A.7 en annexe, à une moyenne sous Palm pour la cellule "typique", c'est à dire :

$$m_k = \mathbb{E}_0^{\pi_{k+1}}[\pi_k(C_0(\pi_{k+1}))] \quad \text{pour } k = 0, \dots, H-1$$

Proposition 6.1 *Pour tout $i = 0, \dots, H-1$,*

$$m_i = \frac{\lambda_i}{\lambda_{i+1}}$$

avec, rappelons le $\lambda_H = 1$.

Preuve. Par une application directe de la formule d'échange de Neveu (voir A.6 à la page 201). □

Remarque 6.2 Les variables m_k sont reliées par la relation suivante : relation :

$$\prod_{i=0}^{H-1} m_i = \lambda_0$$

Preuve. Il suffit d'identifier

$$\begin{aligned} \prod_{i=0}^{H-1} m_i &= \frac{\lambda_0 \lambda_1}{\lambda_1 \lambda_2} \dots \frac{\lambda_{H-1}}{\lambda_H} \\ &= \frac{\lambda_0}{\lambda_H} \end{aligned}$$

et par convention, $\lambda_H = 1$. □

6.3 Tables de routage

Dans cette section, nous reprenons l'étude de Kleinrock [46] sur l'impact de la hiérarchie sur la taille des tables de routage avec ce modèle de géométrie aléatoire. La taille de la table de routage est évaluée comme le nombre de nœuds vus par le routage, c'est à dire le nombre d'entrées dans la table de routage. D'autres possibilités d'évaluation sont discutées à la fin de la section.

6.3.1 Analyse du problème

Notons T_x la taille de la table de routage d'un nœud x du réseau (c.à.d. $x \in \pi_0$).

Proposition 6.2 *Avec les notations précédentes, la taille de la table de routage peut s'écrire :*

$$T_x \doteq \sum_{i=1}^H \pi_{i-1}(C_x(\pi_i))$$

Preuve. Remarquons que la vision d'un nœud x de π_0 est limitée aux nœuds de son domaine, c'est à dire aux points de π_0 appartenant à la cellule $C_x(\pi_1)$. Le nœud x a donc connaissance de $\pi_0(C_x(\pi_1))$ nœuds au niveau physique. Des informations agrégées parviennent également à ce nœud. En particulier, les $\pi_1(C_x(\pi_2))$ nœuds logiques de niveau 1, puis les $\pi_2(C_x(\pi_3))$ nœuds de niveau 3 etc. Finalement, au niveau le plus haut de la hiérarchie, il y a $\pi_{H-1}(C_0(\pi_H))$ partitions. D'où l'expression annoncée. \square

Le problème qui se pose est donc de minimiser la taille moyenne des tables de routage des différents nœuds du réseau — représentés par les points de $\text{supp}(\pi_0)$ appartenant à la cellule $C_0(\pi_H)$, sous Palm. On est donc amené à considérer la grandeur

$$\mathcal{T} \doteq \frac{1}{\lambda_0} \mathbb{E}_0^{\pi_H} \left[\int_{C_0(\pi_H)} T_x d\pi_0(x) \right] \quad (6.1)$$

Cette expression peut être simplifiée en utilisant le calcul de Palm (voir annexe A.6). On arrive finalement à la proposition suivante :

Proposition 6.3

$$\mathcal{T} = 1 + \gamma \sum_{i=1}^H \frac{\lambda_{i-1}}{\lambda_i} \quad (6.2)$$

où $\gamma \approx 1.280$.

Preuve. L'identité (6.1) peut s'écrire

$$\mathcal{T} = \mathbb{E}_0^{\pi_0}[T_0] = \mathbb{E}_0^{\pi_0} \left[\sum_{i=1}^H \pi_{i-1}(C_0(\pi_i)) \right]$$

, d'après la formule d'échange de Neveu (corollaire A.3). En appliquant le théorème de Slivnyak (A.4 en annexe A), cette équation devient :

$$\begin{aligned} \mathbb{E}_0^{\pi_0}[\pi_0(C_0(\pi_1))] &= 1 + \mathbb{E}[\pi_0(C_0(\pi_1))] \\ \mathbb{E}_0^{\pi_0}[\pi_{i-1}(C_0(\pi_i))] &= \mathbb{E}[\pi_{i-1}(C_0(\pi_i))] \quad \text{pour } i > 1 \end{aligned}$$

Notant ensuite par $\sigma(C_0(\pi_i))$ une σ -algèbre générée par $C_0(\pi_i)$, il est possible d'écrire :

$$\begin{aligned} \mathcal{T} &= 1 + \sum_{i=1}^H \mathbb{E}[\pi_{i-1}(C_0(\pi_i)) | \sigma(C_0(\pi_i))] \\ &= 1 + \sum_{i=1}^H \lambda_{i-1} \mathbb{E}[|C_0(\pi_i)|]. \end{aligned}$$

Finalement, la proposition B.6 de l'annexe B permet de conclure. \square

6.3.2 Résultats

Problème d'optimisation.

L'équation (6.2) nous permet de déterminer simplement la structure optimale du réseau hiérarchique pour minimiser la taille de la table de routage. Le problème revient en effet à :

Minimiser :

$$\mathcal{T} = 1 + \gamma \sum_{i=0}^{H-1} m_i$$

Sous contraintes :

$$\prod_{i=0}^{H-1} m_i = \lambda_0$$

où les $\{m_i\}_{i=0, \dots, H-1}$ ont été définis dans la section précédente.

Le problème d'optimisation obtenu est donc très proche de celui de L.Kleinrock dans [46]. La seule différence réside en fait dans le facteur multiplicatif γ , qui est dû à la nature aléatoire du modèle choisi — face au modèle purement déterministe de Kleinrock. En effet, comme montré dans la proposition B.5 à la page 210, γ est relié au coefficient de variation de la taille d'une cellule "typique" de Voronoi. Néanmoins, ce facteur n'a aucune influence sur la structure optimale, qui est identique à celle déterminée par L.Kleinrock. Ceci est tout à fait cohérent puisque nous ne manipulons que des grandeurs moyennes (i.e. déterministes).

Structure optimale.

Proposition 6.4 *Pour un nombre de niveaux hiérarchiques H donné, l'optimum est atteint pour :*

$$m_0 = \dots = m_{H-1} = (N)^{\frac{1}{H}} \quad (6.3)$$

$$\lambda_k = (\lambda_0)^{\frac{H-k}{H}} \quad k = 0, \dots, H-1 \quad (6.4)$$

avec $N = \lambda_0$.

Preuve. L'expression $\mathcal{T} = \sum_{i=0}^{H-1} m_i$ montre bien les rôle complètement symétriques joués par les $\{m_i\}$. Il est alors clair que l'optimum est atteint pour $m_0 = m_1 = \dots = m_{H-1}$ — si ce n'est pas clair, une bonne dérivation vous en convaincra. La contrainte $\prod_{i=0}^{H-1} m_i = \lambda_0$ permet alors d'exprimer ces variables. \square

Corollaire 6.1 *A l'optimum, la taille de la table de routage s'écrit :*

$$\mathcal{T} = 1 + H\gamma(N)^{\frac{1}{H}}$$

Notons que cette fonction n'est pas monotone et admet un minimum. Afin d'identifier approximativement pour quelle valeur de H l'optimum est atteint, nous supposerons que $H \in \mathbb{R}^+$. Une simple dérivation de cette expression permet alors de trouver le nombre optimal de niveaux hiérarchiques.

Proposition 6.5 *Le nombre optimal de niveaux hiérarchiques, H^* , est donnée par l'expression suivante :*

$$H^* = \log(N) \tag{6.5}$$

Corollaire 6.2 *La structure optimale à l'optimum est atteinte pour :*

$$m_k = e \quad \text{for } k = 0, \dots, H-1 \tag{6.6}$$

$$\lambda_k = \lambda_0 e^{-k} \quad \text{for } k = 1, \dots, H-1. \tag{6.7}$$

Remarque 6.3 Comme on peut le voir dans la proposition 6.5 et le corollaire 6.2, l'optimum est atteint pour un très grand nombre de niveaux hiérarchiques et pour des domaines de dimension extrêmement réduite (deux à trois nœuds). Ces résultats ne sont donc pas applicables pour le dimensionnement de réseaux. Par contre, la taille de la table de routage diminue très rapidement avec le nombre de niveaux hiérarchiques, comme le montre la figure 6.2. Il est donc possible de choisir un nombre de niveaux hiérarchiques beaucoup plus faible, sans que les performances du routage n'en soient trop affectées. On voit même que la courbe est extrêmement plate autour de l'optimum : L'ajout de niveaux hiérarchiques supplémentaires ne se justifie alors plus.

Remarque 6.4 Il est intéressant aussi de constater que pour cet optimum, la taille de la table de routage prend la forme :

$$\mathcal{T} = O(\ln(N)) \tag{6.8}$$

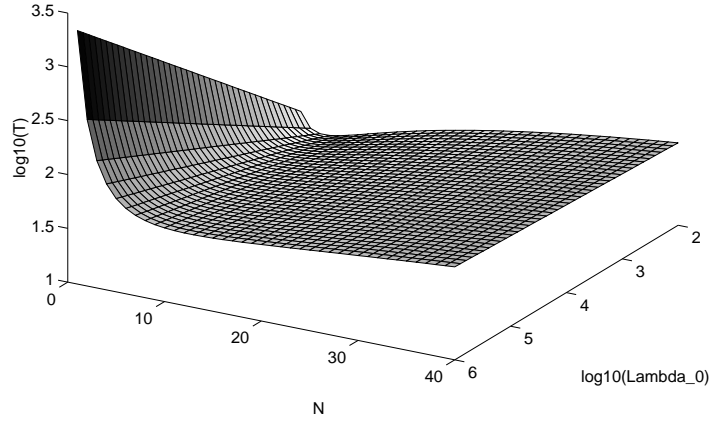


FIG. 6.2 – Taille moyenne de la table de routage en fonction du nombre de niveaux hiérarchiques.

6.3.3 Extensions

Nous n’avons fait aucune hypothèse sur la structure interne du réseau et notamment sur la connectivité entre les différents nœuds. Ceci devient nécessaire si l’on veut évaluer la taille de la base de données topologique, pour un algorithme de type “link-state” par exemple, où le nombre de liens est certainement le facteur critique.

Graphe de Delaunay.

Si l’on considère un graphe de Delaunay (voir section B.4 en annexe), comme dans le modèle du chapitre précédent, le nombre de liens est proportionnel au nombre de nœuds (équation B.11 à la page 209). La structure optimale est donc identique.

Une hypothèse plus pessimiste pourrait consister à considérer que le nombre de liens est proportionnel au carré du nombre de nœuds (réseau maillé ou partiellement maillé). Ce cas est brièvement traité dans la suite.

Graphe Maillé.

Dans le cas d’un réseau maillé, on est amené à s’intéresser à l’expression :

$$\mathcal{L} \doteq \mathbb{E}_0^{\pi_0} \left[\sum_{i=1}^H (\pi_{i-1}(C_0(\pi_i)))^2 \right] \quad (6.9)$$

Proposition 6.6 *Sous l'hypothèse d'un grand réseau (c.à.d. $\lambda_0 \rightarrow \infty$), le nombre de liens considérés à l'optimum prend la forme approchée suivante :*

$$\mathcal{L} \approx 1 + \delta \sum_{i=0}^{H-1} \left(\frac{\lambda_i}{\lambda_{i+1}} \right)^2$$

avec $\delta \approx 2.0$.

Preuve. L'hypothèse d'un grand réseau permet de considérer que

$$\lambda_0 \gg \lambda_1 \gg \dots \gg \lambda_H$$

Ceci est du aux relations géométriques qui relient ces grandeurs à l'optimum — similaire aux relations (6.4) et (6.7) observées précédemment dans le cas du nombre de nœuds. Ceci implique que $\frac{\lambda_i}{\lambda_{i+1}} \gg 1$ pour $i = 0, \dots, H-1$.

D'après Slivnyak (théorème A.4 en annexe), on peut écrire :

$$\begin{aligned} \mathbb{E}_0^{\pi_0} \left[\sum_{i=2}^H (\pi_{i-1}(C_0(\pi_i)))^2 \right] &= \mathbb{E} \left[(\pi_{i-1}(C_0(\pi_i)))^2 \right] \\ \mathbb{E}_0^{\pi_0} \left[(\pi_0(C_0(\pi_1)))^2 \right] &= \mathbb{E} \left[(1 + \pi_0(C_0(\pi_1)))^2 \right] \end{aligned}$$

En utilisant les propriétés bien connues des lois de Poisson, on identifie aisément

$$\mathbb{E} \left[(\pi_{i-1}(C_0(\pi_i)))^2 \right] = \lambda_{i-1} \mathbb{E}[|C_0(\pi_i)|] + \lambda_{i-1}^2 \mathbb{E}[|C_0(\pi_i)|^2]$$

Le dernier terme de cette expression est proportionnel à $\left(\frac{\lambda_{i-1}}{\lambda_i} \right)^2$ alors que le premier terme est proportionnel à $\frac{\lambda_{i-1}}{\lambda_i}$. Sous l'hypothèse d'un grand réseau, nous ne garderons donc que le second terme (c.à.d. le terme dominant). En utilisant la proposition B.7 (à la page 211), on arrive finalement à l'expression annoncée. \square

La structure optimale est donc très proche de celle de Kleinrock observée pour le nombre de nœuds.

Proposition 6.7 *Pour un niveau hiérarchique H donné, la structure optimale est identique à celle de la proposition 6.4.*

Preuve. Remarquons que \mathcal{L} s'écrit $\mathcal{L} = 1 + \delta \sum_{i=0}^{H-1} m_i^2$. La symétrie entre les m_i est donc conservée et l'on conserve $m_0 = m_1 = \dots = m_{H-1}$ à l'optimum. Remarquons que les relations entre les intensités, explicitées par l'équation (6.4), justifient bien a posteriori l'approximation effectuée pour $\lambda_0 \gg 1$. \square

Prenant comme dans le cas précédent $H \in \mathbb{R}^+$, on peut alors identifier dans quels ordres de grandeur se situe l'optimum :

Proposition 6.8 *Le nombre de liens est minimum, pour un réseau maillé de très grande dimension ($\lambda_0 \rightarrow \infty$), pour la structure optimale suivante :*

$$\begin{aligned} H^* &= 2 \ln N \\ m_0 &= m_1 = \dots = m_{H-1} = \sqrt{e} \\ \lambda_i &= \lambda_0 e^{-i/2} \text{ pour } i = 0, \dots, H. \end{aligned}$$

Remarque 6.5 Dans ce cas également, le nombre de niveaux hiérarchiques est extrêmement grand et les domaines de routage sont de taille très réduites. Néanmoins, la taille de la table de routage garde une forme similaire à celle de la figure 6.2 (fonction du type $H \mapsto HK^{1/H}$ pour un K donné). Un nombre de niveaux hiérarchiques plus raisonnable suffit donc à limiter considérablement le nombre de liens à manipuler.

6.3.4 Discussion

Remarque 6.6 La taille des tables de routage a été étudiée dans cette section. Nous avons pu mettre en évidence la structure optimale permettant de minimiser la taille de ces tables de routage (que l'on considère cette taille par rapport au nombre de nœuds ou de liens qui la composent). La structure identifiée possède un nombre important de niveaux hiérarchiques avec des domaines de routage de faibles tailles.

Il est important néanmoins de nuancer notre propos. Les mémoires des systèmes informatiques actuels sont telles que la plupart des routeurs ou commutateurs sont à même de stocker de grandes quantités d'informations³. Pour fixer les idées, les routeurs de fournisseurs de service Internet (routeurs BGP) manipulent des tables de routages d'environ 80,000 entrées⁴ (avec un grand nombre d'informations sur chaque entrée, ce qui requiert plusieurs dizaines de Megaoctets de mémoire). Dans le domaine du routage interne, citons également que le leader des constructeurs de routeurs IP a coutume de recommander qu'il est inutile de faire appel à la hiérarchie⁵ (avec OSPF) pour des réseaux contenant moins de 400 sous-réseaux environ... On est donc loin des deux ou trois sous-réseaux de l'optimum théorique !

³Les travaux de L.Kleinrock précédemment cités se déroulant eux dans un tout autre contexte : La naissance de l'Internet.

⁴Les routeurs de troisième génération sont censés pouvoir supporter des tables de routage d'environ 250,000 entrées...

⁵Il faut cependant souligner que les experts techniques semblent assez réticents à utiliser la hiérarchie définie dans le protocole OSPF pour ses nombreuses limitations et son manque de flexibilité. En effet, la hiérarchie n'est vraiment intéressante que pour des topologies très spécifiques, rarement rencontrées dans la pratique.

Le point le plus important de cette étude est, du point de vue pratique, la décroissance très rapide de la taille de la table de routage en fonction du nombre de niveaux hiérarchiques. Les fonctions sont en effet du type $H \mapsto H(N^K)^{1/H}$ avec $K = 1, 2$ selon le critère retenu ; ce qui prouve bien qu'un nombre très limité de niveaux hiérarchiques suffit à réduire considérablement la quantité d'information à manipuler.

Remarque 6.7 Il est également frappant de constater que dans l'Internet actuel, le nombre de réseaux connectés évolue (ou évoluait) exponentiellement alors que la taille des tables (BGP 4) agrégées évolue à peu près linéairement. Cette évolution peut être mise en relation avec l'équation (6.8) identifiée pour une structure optimale.

Les techniques d'agrégation de BGP 4 sont certes très différentes de celles utilisées dans notre modèle. L'Internet ne dispose en fait que deux niveaux hiérarchiques (routage interne / routage externe). Au niveau du routage externe (BGP 4 en général), l'agrégation des adresses est assurée par configuration, au niveau de chaque nœud dès que celle-ci est possible (voir figure 4.1 au chapitre précédent). En ce sens, on peut considérer que l'on a bien à faire à une agrégation "optimale" (on agrège dès que l'on peut) et il n'est donc pas surprenant de retrouver un comportement proche de celui des structures hiérarchiques optimales décrites ci-avant.

6.4 Complexité des calculs de Routage

6.4.1 Introduction

Dans cette section, nous nous intéressons à la complexité des calculs de routage et à son évolution par rapport à la structure hiérarchique choisie. Nous reprenons l'étude de Piet Van Mieghem [53] avec un modèle stochastique — ce qui nous permet d'affiner les résultats. En effet, cet auteur a pu mettre en évidence que dans le cas de protocoles de routage hiérarchiques et basés à la source, on peut identifier une structure optimum minimisant la complexité des calculs de routage. Cette structure correspond à un compromis approprié entre les deux critères suivants :

- **Complexité.** A l'entrée d'un domaine de routage, la complexité du calcul dépend de la taille du domaine à traverser.
- **Nombre de calculs.** A chaque entrée dans un nouveau domaine, un nouveau calcul de routage doit être effectué.

Plus les domaines sont petits et moins les calculs de routage sont complexes. En revanche, le nombre de domaines traversés pour aller d'un point à un autre augmente — le nombre de calculs augmentant alors également.

Nous nous intéressons spécifiquement au cas d'un algorithme de routage de type "source-routing" (comme par exemple PNNI). Autrement dit, les décisions de routage sont prises au niveau du nœud d'entrée dans le réseau (ou dans le domaine de routage) — décisions qui sont ensuite suivies par les nœuds intermédiaires. Le type d'algorithme considéré correspond à un "calcul à la demande" pour un flux ou une connexion donnée. Le cas des tables de routage précalculées est traité en remarque à la fin de cette section.

Nous nous intéresserons à un algorithme de routage purement hiérarchique, comme présenté dans [54] par exemple. Autrement dit, le nœud d'entrée commence par déterminer une route dans le réseau logique du niveau hiérarchique le plus haut (suivant l'emplacement de la destination). Ensuite, un chemin dans le niveau inférieur est déterminé afin de joindre le prochain domaine sélectionné, et ainsi de suite. Nous considérerons le cas d'un réseau de transit, c'est à dire où la source et la destination sont situés à la périphérie du réseau — de sorte que tout le réseau doit être traversé.

6.4.2 Modélisation

Nous utiliserons le modèle générique présenté dans la section 6.2. Des hypothèses supplémentaires sur la structure du réseau sont néanmoins nécessaires. En effet, aucune hypothèse sur la connectivité des nœuds n'était nécessaire dans la section précédente. Dans le cas présent, il est indispensable de connaître le nombre de domaines de routage traversés (dénommés *peer-groups* dans le cas de PNNI) afin de pouvoir déterminer l'optimum.

Dans [53], un modèle déterministe est utilisé par P. Van Mieghem, et aucune structure n'est sous-entendue. Une hypothèse pire cas est donc utilisée afin de déterminer la structure optimale : La totalité des domaines de routage doivent être traversés pour atteindre la destination. Cette hypothèse, peu réaliste, conduit donc forcément à des résultats extrêmes.

Dans la suite, nous considérerons des graphes de Delaunay. Les nœuds du niveau physique, représentés par les points du processus π_0 , sont donc interconnectés par le graphe de Delaunay généré par π_0 . Le réseau est segmenté récursivement en aires de routage par des pavage de Possoin-Voronoi générés par les processus π_1, \dots, π_{H-1} . Les points de π_k , appelés nœuds logiques dans PNNI, sont également interconnectés par un graphe de Delaunay ($k = 1, \dots, H-1$). Ces connections représentent la connectivité entre domaines voisins. Une représentation d'un tel réseau est illustré sur la figure 6.3 (à l'intérieur d'une aire de routage donné).

Dans cette partie, nous nous intéressons à évaluer la complexité globale des calculs de routage nécessaires à l'acheminement d'un flux ou d'une connexion donnée à travers le réseau. Nous supposons, comme dans [53], que la complexité d'un calcul peut s'exprimer sous la forme KN^2 si N représente le nombre de nœuds du domaine considéré et où K est une constante donnée. Cette valeur représente en effet le terme le plus significatif de la complexité

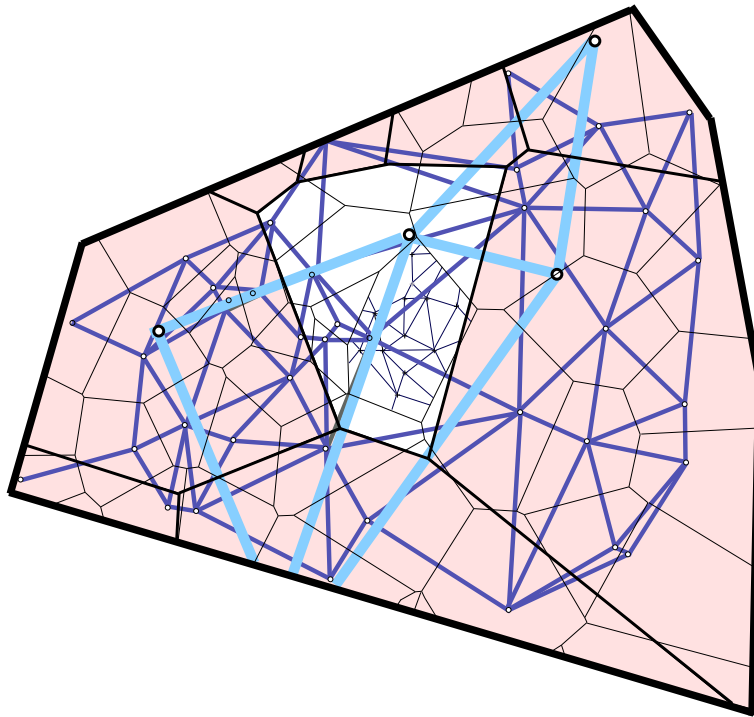


FIG. 6.3 – Connectivité dans le réseau hiérarchique étudié.

du fameux algorithme de Dijkstra⁶ (qui s'exprime en $O(N^2)$). Remarquons que dans un réseau à intégration de service, un calcul de routage tenant en compte la QoS pourrait être souhaitable. Or la prise en compte de plusieurs métriques ou attributs de routage conduisent à des problèmes *NP*-difficiles. Cependant, la plupart des heuristiques se ramènent à des complexités comparables à celle de Dijkstra.

Finalement, nous considérerons des chemins markoviens (voir paragraphe B.4.2 en annexe) : Étant donnés les deux points d'entrée et de sortie du réseau, une droite Δ reliant ces deux points peut être tracée. Les cellules qui sont coupées par cette droite seront sélectionnés pour acheminer le flux ou la connexion.

⁶La complexité d'un tel algorithme peut en fait être réduite à $O(M + N \log N)$ si M correspond au nombre de liens. Nous préférons néanmoins la première expression pour sa simplicité analytique et parcequ'elle correspond à un "pire cas".

6.4.3 Analyse

Processus Intersection.

Soit π_k^Δ le processus ponctuel⁷ sur la droite Δ dont le support (cf. chapitre A) est défini par les points d'intersection de Δ avec le pavage de Voronoi engendré par π_k .

Proposition 6.9 *L'intensité du processus π_k^Δ , notée λ_k^Δ , vérifie l'égalité :*

$$\lambda_k^\Delta = \frac{4}{\pi} \sqrt{\lambda_k} \quad (6.10)$$

Preuve. Il s'agit d'une application directe du théorème de Møller (théorème B.1 à la page 215). \square

Remarque 6.8 A chaque point de $\text{supp}(\pi_k^\Delta)$ correspond donc une nouvelle intersection entre le pavage engendré par π_k et la droite considérée Δ . Autrement dit, chaque point de $\text{supp}(\pi_k^\Delta)$ correspond à une entrée dans un nouveau domaine de niveau k . Sur une distance Δ , le nombre de domaines de niveau k traversés est, en moyenne, $1 + \lambda_k^\Delta \delta$ (en prenant en compte le domaine initial).

Niveau de Parenté.

A l'arrivée d'un flux ou d'une connexion, le nœud source doit tout d'abord déterminer à quel niveau hiérarchique le calcul de routage doit commencer. Par exemple, la destination appartient au même domaine physique, un calcul de routage sur les noeuds et les liens physiques ce domaine suffit. Plus généralement, si la destination appartient au même domaine de niveau k , le routage mettra en jeu les niveaux hiérarchiques $0, \dots, k$, pour $k = 0, \dots, H - 1$.

Proposition 6.10 *Le calcul de routage met en jeu les niveaux hiérarchiques $0, \dots, k$ (avec $k \in \{0, \dots, H - 1\}$) avec la probabilité $p_k(\rho)$ obéissant aux relations :*

$$\begin{aligned} p_0(\rho) &= Q(\rho\sqrt{\lambda_1}) \\ p_k(\rho) &= Q(\rho\sqrt{\lambda_{k+1}}) \prod_{l=1}^k (1 - Q(\rho\sqrt{\lambda_l})) \quad (0 < k < H - 1) \\ p_{H-1}(\rho) &= \prod_{l=1}^{H-1} (1 - Q(\rho\sqrt{\lambda_l})) \end{aligned}$$

où ρ est la distance entre les points sources et destinataires et où nous avons utilisé la fonction Q introduite dans la proposition B.10 à la page 203.

⁷Notons bien qu'il s'agit d'un processus sur la droite réelle et non pas dans le plan.

Preuve. Deux nœuds appartiennent au même domaine de niveau k si, dans notre représentation, les deux points de \mathbb{R}^2 les représentant appartiennent à la même cellule de Voronoi engendrée par π_k . Par définition de la fonction Q (cf. proposition B.10), nous arrivons alors aux expressions annoncées. \square

Proposition 6.11 *Soit $N_k^\Delta(\rho)$ le nombre moyen de calculs de routage de niveaux k mis en jeu pour relier deux points distants de ρ . Cette grandeur vérifie l'identité :*

$$N_k^\Delta(\rho) = p_k(\rho) + (1 + \lambda_k^\Delta \rho) \left(1 - \sum_{l=0}^k p_l(\rho) \right)$$

Preuve. Si les deux points appartiennent au même domaine de niveau k , avec la probabilité $p_k(\rho)$, il est clair que $N_k^\Delta(\rho) = 1$. Si le niveau de Parenté est strictement supérieur à k , avec la probabilité $1 - \sum_{l=0}^k p_l$, alors plusieurs calculs de niveaux k sont nécessaires : La remarque 6.8, permet alors de conclure. \square

Corollaire 6.3 *Lorsque $\rho \rightarrow \infty$, le calcul de routage fait presque sûrement intervenir les H niveaux hiérarchiques. De plus,*

$$\begin{aligned} N_k^\Delta &\approx (1 + \lambda_k^\Delta \rho) \quad \text{pour } k = 1, \dots, H-1 \\ N_H^\Delta &\approx 1 \end{aligned}$$

Une hypothèse simplificatrice : Dans la suite, nous considérons le cas d'un réseau de transit : Les points sources et destinations sont donc situés à la périphérie du réseau. Plus précisément, nous considérons une droite arbitraire Δ infinie de \mathbb{R}^2 . Il existe donc une infinité de structures hiérarchiques coupées par la droite Δ . Les grandeurs considérées correspondront, comme précédemment, à des grandeurs moyennes calculées sur toutes ces réalisations. Ceci revient à considérer une cellule "typique" sous la probabilité de Palm $\mathbb{P}_0^{\pi_H^\Delta}$ présenté en annexe à la page 216.

Nous supposons de plus que le réseau est suffisamment étendu pour pouvoir admettre que la longueur ρ de l'intersection de Δ avec la cellule considérée vérifie $\rho \gg 1$. Nous utiliserons donc les expressions simplifiées du corollaire 6.3.

Complexité globale

Dans la suite, on s'intéresse à l'étude de la complexité des calculs de routage nécessaire à la traversée de notre réseau. Nous nous intéressons donc à une complexité moyenne donnée

par la formule suivante :

$$\mathcal{C} \doteq \mathbb{E}_0^{\pi_H^\Delta} [f(\pi_{H-1}(C_0(\pi_H)))] + \sum_{k=1}^{H-1} N_k^\Delta(\rho) \mathbb{E}_0^{\pi_k^\Delta} [f(\pi_{k-1}(C_0(\pi_k)))]$$

où $f(x) = K.x^2$ caractérise la complexité des calculs de routage et où les moyennes $\mathbb{E}_0^{\pi_H^\Delta}[\cdot]$ obéissent à la définition B.5 de l'annexe B.

Remarque 6.9 Le premier terme de cette expression correspond au calcul de routage dans le niveau hiérarchique le plus haut. Nous utilisons en effet les résultats du corollaire 6.3, sous l'hypothèse $\rho \gg 1$, de sorte qu'un tel calcul est nécessaire presque sûrement. Sous cette même hypothèse, la traversée du réseau considérée met également en jeu $N_k^\Delta(\rho)$ (dont l'expression est donnée dans le corollaire 6.3) domaines de niveau k . Nous considérons une grandeur moyenne par rapport à tous les points de π_H coupés par Δ , c'est à dire sous $\mathbb{P}_0^{\pi_H^\Delta}$ par définition (cf. page 216).

Proposition 6.12 *L'expression de la complexité globale d'un calcul de routage peut être simplifiée de la sorte :*

$$\mathcal{C} = \delta_1 \lambda_{H-1} + \delta_2 (\lambda_{H-1})^2 + \sum_{k=1}^{H-1} (1 + \lambda_k^\Delta \rho) \left[\delta_1 \frac{\lambda_{k-1}}{\lambda_k} + \delta_2 \left(\frac{\lambda_{k-1}}{\lambda_k} \right)^2 \right] \quad (6.11)$$

où $\delta_1 \approx 1.12$ et $\delta_2 \approx 1.62$.

Preuve. Par une simple application de la proposition B.12. □

Corollaire 6.4 *Sous l'hypothèse d'un grand réseau ($\lambda_0 \rightarrow \infty$), la complexité globale à l'optimum prend une expression simplifiée :*

$$\mathcal{C} \approx \delta_2 (\lambda_{H-1})^2 + \sum_{k=1}^{H-1} \frac{4\rho\delta_2}{\pi} \frac{(\lambda_{k-1})^2}{(\lambda_k)^{3/2}}$$

Preuve. L'hypothèse d'un grand réseau se traduit par la relation

$$\lambda_0 \gg \lambda_1 \gg \dots \lambda_{H-1} \gg \lambda_H = 1$$

Cette relation est vérifiée à posteriori par les relations géométriques reliant ces grandeurs à l'optimum (voir plus loin). Ne gardant alors que les termes significatifs, nous arrivons bien à l'expression annoncée. □

6.4.4 Résultats

Structure optimale.

Proposition 6.13 *Pour un nombre de niveaux hiérarchiques H donné, la structure optimale permettant de minimiser la complexité globale d'un flux est caractérisée par les relations géométriques suivantes :*

$$m_{H-1} = \left(\frac{3\rho}{\pi}\right)^{2/3} (m_{H-2})^{4/3} \quad (6.12)$$

$$m_k = \left(\frac{3}{4}\right)^{2/3} (m_{k-1})^{4/3} \text{ pour } k = 1, \dots, H-2. \quad (6.13)$$

Preuve. On vérifie que les fonctions $\lambda_k \mapsto \mathcal{C}(\lambda_k)$ ne sont pas monotones et admettent un minimum. Par dérivation sur les $\{\lambda_k\}_{k=1, \dots, H-1}$, on obtient alors les relations :

$$\lambda_{H-1} = \frac{3\rho \lambda_{H-2}^2}{\pi \lambda_{H-1}^{5/2}} \quad (6.14)$$

$$\frac{\lambda_k}{\lambda_{k+1}^{3/2}} = \frac{3}{4} \frac{\lambda_{k-1}^2}{\lambda_k^{5/2}} \quad (6.15)$$

□

Corollaire 6.5 *A l'optimum,*

$$\begin{aligned} m_0 &= \alpha^2 (\kappa \alpha^{-2H} \lambda_0)^{\frac{\alpha-1}{\alpha^{H-1}}} \\ m_k &= \alpha^2 (\kappa \alpha^{-2H} \lambda_0)^{\frac{\alpha^{k+1} - \alpha^k}{\alpha^{H-1}}} \text{ pour } k = 1, \dots, H-2. \\ m_{H-1} &= \frac{\alpha^2}{\kappa} (\kappa \alpha^{-2H} \lambda_0)^{\frac{\alpha^H - \alpha^{H-1}}{\alpha^{H-1}}} \end{aligned}$$

avec $\kappa = \left(\frac{\pi}{4\rho}\right)^{2/3}$, $\alpha = \frac{4}{3}$ et $\beta = \left(\frac{3}{4}\right)^{2/3}$.

Preuve. Les relations géométriques de la proposition précédente permettent d'écrire

$$m_k = \beta^{\frac{\alpha^k - 1}{\alpha - 1}} m_0^{\alpha^k}$$

où $\alpha = 4/3$ et $\beta = (3/4)^{2/3}$ et pour $k = 1, \dots, H-2$. D'après la remarque 6.2, nous arrivons finalement à

$$\lambda_0 = \left(\frac{4\rho}{\pi}\right)^{2/3} \beta^{\frac{-H}{\alpha-1}} \beta^{\frac{\alpha^H - 1}{(\alpha-1)^2}} m_0^{\frac{\alpha^H - 1}{\alpha - 1}}$$

D'où les relations annoncées. □

Remarque 6.10 L'expression des intensités peut également être déduit des expressions précédentes — en se rappelant que $m_k = \frac{\lambda_k}{\lambda_{k+1}}$ pour tout $k = 0, \dots, H - 1$. Sur la figure 6.4 est représenté l'évolution de λ_1 et λ_2 en fonction de λ_0 (pour $H = 3$). Nous remarquerons que la structure optimale approchée (sous l'hypothèse de grand réseau) est très proche de la structure optimale réelle, obtenue par des méthodes numériques. Ceci justifie bien, à posteriori, l'hypothèse simplificatrice qui a été faite.

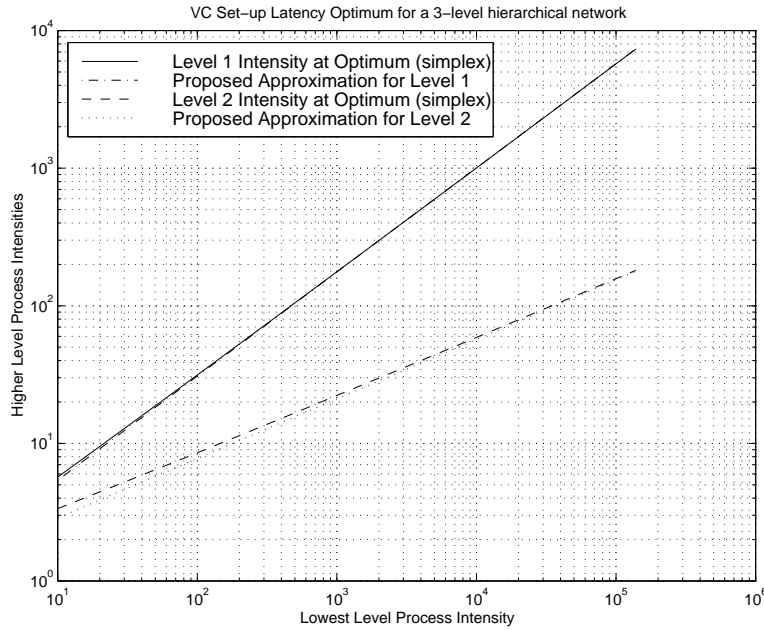


FIG. 6.4 – Évolution des intensités à l'optimum pour $H = 3$

Nombre de niveaux hiérarchiques.

Notons \mathcal{C}_k la complexité des calculs de niveau k pour tout $k = 0, \dots, H - 1$. D'après le corollaire 6.4, leurs expressions simplifiées sont données par :

$$\begin{aligned} \mathcal{C}_k &\approx \frac{4\rho\delta_2}{\pi} \frac{(\lambda_k)^2}{(\lambda_{k+1})^{3/2}} \text{ pour } k < H - 1. \\ \mathcal{C}_{H-1} &\approx \delta_2 (\lambda_{H-1})^2 \end{aligned}$$

Proposition 6.14 *A l'optimum, la relation suivante est vérifiée pour tout $k = 1, \dots, H - 1$:*

$$\mathcal{C}_k = \frac{3}{4} \mathcal{C}_{k-1}$$

De plus, à l'optimum, la complexité s'écrit :

$$\mathcal{C} = \frac{1 - \alpha^{-H}}{1 - \alpha^{-1}} \alpha^3 \alpha^{\frac{-H}{\alpha^{H-1}}} \delta_2 \left(\frac{4\rho}{\pi} \right)^{\frac{\alpha^H - \alpha}{\alpha^{H-1}}} \left(\sqrt{\lambda_0} \right)^{\frac{\alpha^H}{\alpha^{H-1}}}$$

Preuve. Les dérivées partielles de \mathcal{C} par rapport aux intensités (équations (6.14) et (6.15)) peuvent être aisément reformulées pour obtenir l'identité $\mathcal{C}_k = \frac{3}{4}\mathcal{C}_{k-1}$. La complexité globale s'écrit donc

$$\mathcal{C} = \sum_{k=0}^{H-1} \mathcal{C}_k = \frac{1 - \alpha^{-H}}{1 - \alpha^{-1}} \mathcal{C}_0$$

L'expression de \mathcal{C}_0 peut être aisément déterminée en fonction de celle de m_0 (corollaire 6.5).

□

Remarque 6.11 Lorsque $H \rightarrow \infty$, la complexité prend un ordre de grandeur du type $O(\rho\sqrt{\lambda_0})$. La complexité de calculs de routage, pour une structure optimale, est tracée sur la figure 6.5 en fonction du nombre de niveaux hiérarchiques. Il est important de noter que la complexité n'admet pas toujours un minimum et peut donc être minimale pour $H \rightarrow \infty$. en revanche, la courbe devient très plate assez rapidement, de sorte qu'un nombre très limité de niveau hiérarchique suffit à diminuer significativement la complexité des calculs de routage.

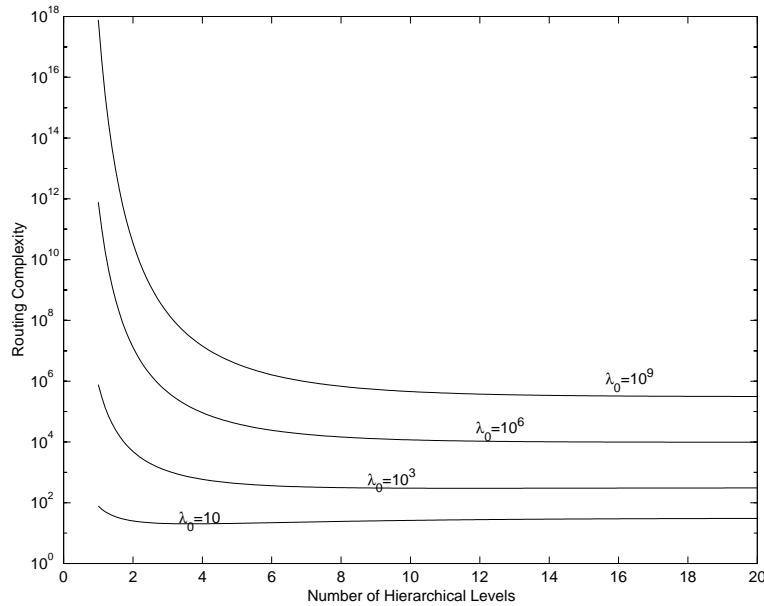


FIG. 6.5 – Evolution de la complexité en fonction du nombre de niveaux hiérarchiques.

6.4.5 Discussion

Structure optimale.

Dans les paragraphes précédents, nous avons étudié la structure optimale permettant de minimiser la complexité des calculs lors du routage d'un flux ou d'un appel. La structure

observée est très particulière, avec des domaines de routage de taille limitée au niveau physique, et des domaines de grande dimension pour les niveaux hiérarchiques les plus haut — comme le montre la proposition 6.13. Ce résultat est somme toute logique : Les calculs de routage aux niveaux hiérarchiques bas interviennent plus fréquemment que ceux des niveaux hauts. Il est donc préférable de dimensionner le réseau de manière à obtenir de petits domaines au niveaux bas, limitant ainsi la complexité globale.

Comme le montre la figure 6.5, cette optimisation conduit également à un nombre de niveaux hiérarchiques assez important. Par contre, il faut remarquer que la complexité est réduite très significativement, même pour un nombre très limité de niveaux (par exemple $H = 2$ pour $\lambda_0 = 1000$). Ce résultat met encore une fois en évidence la puissance de la hiérarchie pour la réduction de la complexité associée au routage dynamique.

Dans [53], une approche pire cas sur des réseaux déterministes est utilisée. Cette approche conduisait à un résultat extrême, avec

$$m_k = \frac{1}{2} m_{k-1}^2 \quad (6.16)$$

au lieu de $m_k = \beta m_{k-1}^{4/3}$ trouvé précédemment.

Les domaines de niveau le plus bas sont donc de taille extrêmement réduite (leur taille tend rapidement vers 2 nœuds) et des domaines de niveau élevé très importants (prenant des valeurs vite déraisonnables). Ces résultats extrêmes étaient dû à l'hypothèse extrêmement grossière considérée : Il faut passer par tous les domaines de tous les niveaux pour passer d'un point à un autre. La validité des résultats obtenus était donc critiquable.

Notre étude permet de reprendre ces calculs de complexité avec des hypothèses beaucoup plus réalistes sur le nombre de nœuds à traverser. Ceci à été possible grâce aux nombreux résultats existants sur la géométrie aléatoire. Cette étude confirme bien qu'il faut de petits domaines aux niveaux bas et de grands domaines aux niveaux hauts pour minimiser la complexité des calculs de routage (résultat opposé aux règles d'ingénierie actuelles).

Validité des résultats

Les résultats de cette section ne peuvent être appliqués à n'importe quel type de réseau. En particulier, nous avons utilisé des graphes de Delaunay. Ces réseaux possèdent une forte connectivité. Il est fort possible qu'il faille passer par un plus grand nombre de nœuds intermédiaires dans un réseau faiblement connecté. La taille des domaines doit alors être encore plus petite au niveau physique et encore plus grande aux niveaux hauts — l'extremum correspondant à la structure de Piet Van Mieghem (équation 6.16).

A l'opposé, on peut imaginer un réseau très fortement connecté. Imaginons le cas où il est possible de traverser n'importe quel domaine en un saut, c'est à dire le cas où tous les

domaines sont totalement maillés (à tous les niveaux hiérarchiques). Il est facile de voir que dans ces conditions, la complexité est de l'ordre de

$$\mathcal{C} \sim 2^{H-1} m_0^2 + \dots + 2^{H-1-k} m_k^2 + \dots + m_{H-1}^2$$

On arrive alors à une structure du type :

$$m_k = \sqrt{2} m_{k-1}$$

pour tout $k = 1, \dots, H - 1$.

En conclusion, il semble que pour minimiser les calculs de routage, il soit préférable de choisir une structure hiérarchique du type :

$$m_0 < m_1 < \dots < m_{H-1}$$

avec

$$m_k = a (m_{k-1})^b$$

où $a > 0$ et b varie entre 1 et 2 selon la connectivité du réseau considéré — plus la connectivité est grande et plus b est proche de 1.

Autres techniques de routage.

Les calculs précédents ont été menés sous l'hypothèse de calculs "à la demande". Autrement dit, nous nous sommes intéressés au cas où, à l'arrivée d'un flux (resp. connexion), un calcul de routage spécifique est utilisé afin d'acheminer ce flux (resp. connexion). On conçoit bien qu'il ne serait pas techniquement réalisable de lancer un calcul spécifique à l'arrivée de chaque connexion (resp. flux, même s'il s'agit de flux agrégés). Dans le cas de PNNI par exemple, il semble que de nombreux constructeurs fassent appels à des tables de routage précalculées. Ainsi, seule une fraction très faible des flux bénéficient de calculs de routage spécifiques.

La construction des tables de routage précalculées consiste, pour une classe de QoS donnée, à déterminer le meilleur chemin vers toutes les destinations possibles du réseau. Autrement dit, la complexité s'exprime de cette manière, en utilisant notre modèle géométrique (présenté au paragraphe 6.2) :

$$\mathcal{C}' = \sum_{k=0}^{H-1} \mathbb{E}_0^{\pi_0} [f(\pi_k(C_0 \pi_{k+1}))]$$

où f caractérise la complexité du calcul de routage. En posant, comme précédemment, $f(x) = x^2$, nous reconnaissons l'équation (6.9) traitant de la taille des tables de routage (en considérant le nombre de liens) sur un graphe maillé. Les structures optimales de ces deux problèmes sont donc identiques (cf. proposition 6.8). Remarquons que cette structure est extrêmement symétrique ($m_k = m_{k-1}$) contrairement à la précédente. Il est donc essentiel de bien connaître la procédure de routage choisie afin de choisir une structure appropriée.

Chapitre 7

Analyse d'un réseau hiérarchique.

Dans ce chapitre, les résultats obtenus lors de l'analyse des performances d'un réseau plat sont généralisés à un réseau à deux niveaux hiérarchiques. Nous travaillons également dans le contexte du protocole PNNI, bien que cette analyse soit valable pour tout protocole de routage hiérarchique basé à la source. Le processus d'agrégation est modélisé de manière abstraite et générique afin d'en étudier l'impact sur les performances globales du réseau — c.à.d en terme de probabilité de blocage d'appel et de sur-coût protocolaire. Cette étude a été menée avec Arturo Ragozini lors de ses deux séjours à l'ENST.

Le modèle utilisé est présenté dans section suivante. Les différents critères de performances sont exprimés dans la section 7.2. Les résultats sont en suite analysés dans la section 7.3. Enfin, des conclusions sont tirées et des possibilités d'extension de ce modèle sont discutées dans la section 7.4.

7.1 Modèle

Dans ce chapitre, nous limiterons notre analyse au cas d'un réseau possédant deux niveaux hiérarchiques uniquement. Les possibles extensions permettant de prendre un compte un nombre arbitraire de niveaux hiérarchique sont discutées à la fin de ce chapitre (section 7.4).

7.1.1 Structure hiérarchique

Comme dans les modèles précédents, les nœuds du réseaux seront représentés par un processus de Poisson π_0 d'intensité λ_0 (cf. section A.3). De manière à conserver un réseau de dimension finie, nous nous considérerons uniquement les points de π_0 qui appartiennent à une

surface W de dimension finie. Sans perte de généralité, nous supposons que W est un disque d'aire unitaire (c.à.d. de rayon $\frac{1}{\sqrt{\pi}}$). Par construction, le réseau comportera donc λ_0 nœuds en moyenne.

Les nœuds de ce réseau sont interconnectés selon un graphe de Delaunay (cf. section B.4). Les domaines de routage sont représentés par des cellules de Voronoi (cf. section B.3) générées par un second processus de Poisson π_1 , d'intensité λ_1 . Au niveau hiérarchique supérieur, les domaines sont représentés par des nœuds logiques — correspondant aux centres des cellules de Voronoi dans notre modèle. La connectivité entre ces domaines est également modélisée par un graphe de Delaunay : Deux domaines sont directement reliés si des deux cellules de Voronoi qui les représentent possèdent une arête commune. La figure 7.1 représente une réalisation de ce modèle.

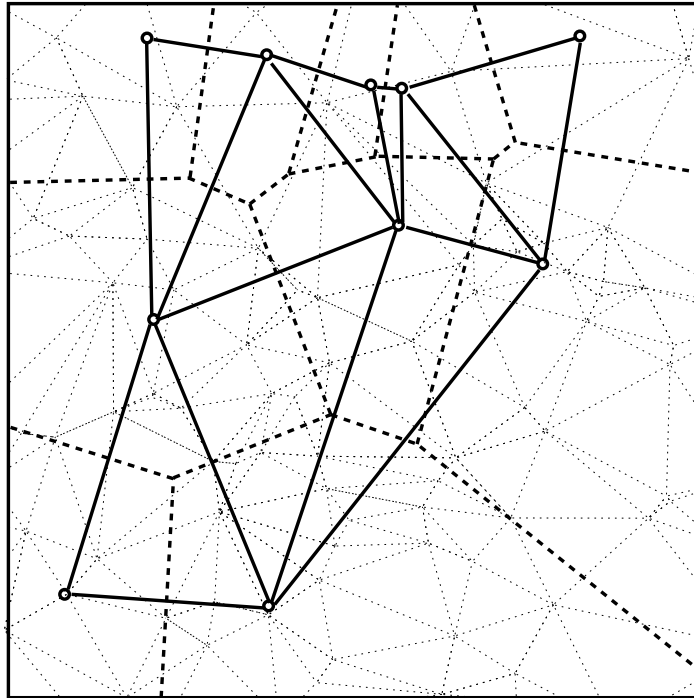


FIG. 7.1 – Représentation d'un réseau à deux niveaux hiérarchiques.

Comme dans le cas du réseau plat (chapitre 5), les liens physiques seront marqués par leur bande passante disponible (les autres attributs et métriques n'étant pas considérés), noté X_i^0 (pour tout lien i). Le processus de rafraîchissement des informations de routage peut introduire des imprécisions entre les valeurs disponibles dans les domaines de routage, notées $\{\tilde{X}^0\}$, et les valeurs réelles. Nous supposons les bandes passantes des liens indépendantes et identiquement distribuées. Notons $p_0 = \mathbb{P}\{\tilde{X}^0 > b\}$ et $q_0 = \mathbb{P}\{X^0 > b \mid \tilde{X}^0 > b\}$ pour

une bande passante b donnée¹ Au niveau hiérarchique supérieur, les nœuds logiques, représentant les domaines de routage, disposent également de marques qui représentent des bandes passantes agrégées — comme expliqué un peu plus loin.

7.1.2 Algorithme de Routage.

L'algorithme de routage sera considéré récursif ("top-down"), comme celui présenté dans [54] par exemple — comme nous l'avons également supposé dans les chapitres précédents. Autrement dit, le premier nœud rencontré débute par déterminer les domaines du plus haut niveau hiérarchique qu'il est nécessaire de traverser pour atteindre la destination. Ensuite, un autre calcul est lancé au niveau hiérarchique inférieur (sur les nœuds de $\text{supp}(\pi_0)$) pour déterminer un chemin permettant de joindre le prochain domaine sélectionné.

Comme dans la chapitre 5, nous considérerons que les nœuds du réseau font appel à des tables de routage précalculées, et ce à tous les niveaux hiérarchiques. Parmi toutes ces routes, la fonction GCAC² sélectionne les chemins admissibles, c'est à dire les chemins qui pourront, à priori, recevoir l'appel considéré. De plus, nous considérerons que tous les chemins précalculés sont disjoints et de même longueur (cf. chapitre 5). Au niveau physique, nous noterons par K_0 le nombre de routes disponibles pour traverser un domaine de routage donné en direction du prochain domaine. Au niveau hiérarchique supérieur, nous noterons par K_1 le nombre de routes disponibles pour atteindre une destination donnée. Comme dans le cas du réseau plat, nous considérons des routes minimisant le nombre de liens utilisés et ne faisons pas appel aux calculs de routage compétitifs ou autres algorithmes plus évolués.

7.1.3 Procédure d'agrégation

Notion d'agressivité.

Il existe de nombreux algorithmes permettant d'agréger les informations de routage. Dans PNNI, par exemple, les procédures d'agrégation n'ont pas été standardisées, de sorte que chaque constructeur possède sa propre manière de procéder. Il est donc indispensable de pouvoir modéliser l'agrégation de manière générique pour que les résultats soient valables pour une large gamme d'algorithmes.

Dans la suite, nous allons définir une procédure d'agrégation assez générique, caractérisant les différents algorithmes par un paramètre uniquement : leur agressivité. Ce concept d'agressivité se révèle en effet essentiel dans l'analyse des performances d'un réseau hiérarchique. Au

¹Il pourrait s'agir d'un débit crête, débit moyen ou d'une bande passante effective ("effective bandwidth") suivant la stratégie d'admission des appels.

²Generic Call Admission Control

delà de la modélisation, le concepteur d'algorithmes d'agrégation se voit confronté à des choix où l'agrégation apparaît très nettement. Par exemple, la plupart des algorithmes d'agrégation construisent un graphe maillé (un jeu de paramètres entre chaque point d'entrée et de sortie du domaine), qu'il est ensuite nécessaire de condenser. Si un algorithme du type "simple node representation" (dans PNNI) est utilisé, un seul paramètre est annoncé pour représenter toutes ces données. Faut-il mieux être "agressif" en utilisant la moyenne de toutes les bandes passantes du graphe maillé ou même le maximum ? Ou faut-il plutôt être "conservateur" (afin d'éviter les crankbacks) et annoncer la bande passante de la branche la plus congestionnée ?

Modélisation.

Dans notre modèle, un domaine entier est condensé en un nœud simple, c'est à dire en seul jeu de paramètres : Un nœud logique, disons $u \in \pi_1$ (représenté par le centre d'une cellule de Voronoi $C_u(\pi_1)$) est ainsi marqué par un unique paramètre, noté X_u^1 . Cette grandeur représente la bande passante dont une connexion pourrait espérer bénéficier à la traversée du domaine considéré. Des techniques d'agrégation plus complexes, comme le "complex node representation" de PNNI, sont traitées brièvement à la fin de ce chapitre.

Une hypothèse conservatrice consisterait à choisir X_u^1 comme la bande passante minimale disponible dans le réseau, c.à.d. $X_u^1 = \min\{X_i^0 : i \in C_u(\pi_1)\}$ où i indexe les liens du niveau physiques. Un approche plus flexible et générique consiste à utiliser les statistiques d'ordre (voir annexe C.1). Étant donné le paramètre $\beta \in [0, 1[$, la bande passante agrégée X_u^1 sera choisie comme le β -quantile (cf. définition C.3 à la page 220) de l'échantillon $\{X_i^0 : i \in C_u(\pi_1)\}$ (avec les mêmes conventions sur l'index i). Ce choix implique qu'une proportion β des liens du domaine ont une bande passante inférieure à X_u^1 . Pour $\beta = 0$, la valeur agrégée choisie correspond au minimum des bandes passantes disponibles dans le domaine — représentant donc une politique d'agrégation extrêmement conservatrice. pour $\beta \rightarrow 1$ au contraire, la valeur annoncée correspond à la bande passante du lien le moins congestionné — représentant une politique d'agrégation très agressive. Le cas $\beta = \frac{1}{2}$ correspond à la bande passante "médiane".

Notons que la procédure de rafraîchissement des informations de routage induit forcément des erreurs sur les bandes passantes agrégées dont disposent les autres domaines. Les nœuds ne peuvent avoir connaissance de la valeur exacte de X_u^1 mais ne disposent que d'une estimation, notée \tilde{X}_u^1 . Dans la suite, nous concentrerons nos efforts sur les procédures d'agrégation spécifiquement, en négligeant ce phénomène. Nous poserons donc $\tilde{X}_u^1 = X_u^1 \forall u \in \pi_k$. Cette approximation peut être justifiée par les faibles variances observées pour X^1 par rapport à celles X^0 : Un théorème central limite s'applique d'ailleurs à valeur agrégée X^1 (cf. proposition C.2 en annexe C) dont la variance évolue donc asymptotiquement en $\frac{1}{\sqrt{L_0}}$ (où $L_0 \rightarrow \infty$ désigne le nombre de liens du domaine considéré). De plus, l'agrégation est certainement le

facteur le plus critique affectant la qualité des informations de routage (surtout dans notre modèle ou un algorithme “simple node” est utilisé).

7.1.4 Crankback

Il est possible qu’une connexion soit refusée sur le chemin choisi. En effet les calculs de routage sont basés sur des informations parfois agrégées (au niveau hiérarchique supérieur) ou obsolètes (à cause du mécanisme de rafraîchissement) et peuvent donc conduire à la sélection de chemins peu appropriés. Une procédure de crankback est alors utilisée pour libérer les ressources sur le chemin défectueux et choisir éventuellement une route alternative.

La procédure de routage et de crankback est donc similaire à celle présentée dans les tables 5.1 et 5.2 dans l’analyse d’un réseau plat. Cependant, dans ce chapitre, on peut discerner deux types de crankbacks suivant le niveau hiérarchique mis en jeu. Dans notre modèle à deux niveaux hiérarchiques, un appel subissant un crankback revient forcément au nœud d’entrée dans le domaine. Si aucune autre alternative n’est possible, alors le crankback revient jusqu’au nœud source et une route alternative (au niveau logique) peut être déterminée.

Afin de limiter les échanges de message de signalisation et plus généralement le surcoût associé au routage, nous supposons que le nombre de crankbacks subis par un appel est contrôlé. Plus exactement, nous imposerons une limite K_c^1 sur le nombre de crankbacks revenant au nœud source. De plus, un appel ne pourra subir plus de K_c^0 crankbacks internes lors de la traversée d’un peer-group.

7.2 Analyse

7.2.1 Préliminaires

Hypothèses simplificatrices.

Dans tout la suite, les différentes grandeurs permettant d’évaluer les performances du réseau seront dans un premier temps calculées pour une réalisation du réseau hiérarchique donnée. Autrement dit, ces grandeurs correspondront à des moyennes ou des probabilités conditionnelles. Nous proposerons ensuite des approximations permettant d’intégrer ces variables afin d’obtenir les moyennes et les probabilités recherchées.

Nous supposons que les cellules de Voronoi sont indépendantes les unes des autres³. Nous

³Ce qui est certainement faux car les tailles des cellules sont corrélées (d’après la manière dont elles sont construites). Néanmoins, le facteur de corrélation reste modéré (≈ 1.10) et des simulations nous ont montré que cette hypothèse conduit à des résultats significatifs.

supposerons donc que $N_0(i)$, le nombre de nœuds de la cellule i et que $L_0(i)$, le nombre de liens de la cellule i , sont des variables aléatoires indépendantes et identiquement distribuées — nous omettrons d'ailleurs l'indice i dans la suite de cette étude pour cette raison. De plus, nous supposerons que le nombre de cellules rencontrées pour traverser le réseau et leurs tailles respectives sont indépendants.

Notations.

Soit $p_{0,1}$ la probabilité de traverser avec succès un domaine arbitraire (représenté par une cellule). Cette probabilité met en jeu les deux événements suivants :

- La bande passante agrégée est suffisante pour recevoir la connexion, c.à.d. $X^1 > b$ si b est la bande passante demandée par l'utilisateur.
- Un chemin comportant suffisamment de bande passante a pu être trouvé au niveau physique afin de traverser le domaine.

Pour une bande passante donnée b , posons $p_1 = \mathbb{P}\{X^1 > b\}$. Cette grandeur correspond à la probabilité que la bande passante agrégée corresponde aux besoins de QoS de l'utilisateur. Nous noterons également $p_{0|1} = \frac{p_{0,1}}{p_1}$ la probabilité conditionnelle de passer avec succès à travers une cellule (c.à.d. une aire de routage) sachant que cette cellule dispose a priori d'assez de bande passante, c.à.d. sachant $X^1 > b$.

Notons M_0 le nombre de liens nécessaires à la traversée d'un domaine et M_1 le nombre de domaines qu'il est nécessaire d'emprunter pour traverser le réseau.

La détermination de p_1 , $p_{0,1}$ et $p_{0|1}$ est quelque peu technique et a donc été placée en annexe C.3.

7.2.2 Perte d'appel.

Soit \mathcal{P}_b la probabilité qu'un appel soit bloqué, quelqu'en soit la raison. Cet événement peut se produire parce que le nœud source ne peut trouver aucun chemin admissible ou parce que l'appel a subi un (ou plusieurs) crankback(s) et qu'aucune route alternative n'a pu être trouvée.

Proposition 7.1 *La probabilité de perte d'un appel est donnée par l'expression :*

$$\mathcal{P}_b^1 = \sum_{r=0}^{K_1} \binom{K_1}{r} P_1^r (1 - Q_1)^{r \wedge K_1^c} (1 - P_1)^{K_1 - r} \quad (7.1)$$

avec $P_1 = p_1^{M_1}$ et $Q_1 = p_{0|1}^{M_1}$ et où p_1 et $p_{0|1}$ sont exprimés aux théorèmes C.1 et C.3 en annexe C.3.

Preuve. Comme nous supposons que les cellules sont indépendantes et identiquement distribuées, le problème est en fait exactement similaire à celui d'un réseau plat (cf. proposition 5.2 à la page 99), en identifiant $p_0 \Leftrightarrow p_1$ (probabilité de sélectionner un domaine) et $q_0 \Leftrightarrow p_{0|1}$ (probabilité de traverser un domaine sachant que ce domaine a été sélectionné). \square

7.2.3 Rejet de l'appel au nœud source.

Proposition 7.2 *La probabilité qu'un appel soit directement rejeté au nœud d'entrée dans le domaine est donnée par :*

$$\mathcal{P}_r^1 = \left(1 - P p_{0|1}^*\right)^{K_1} \quad (7.2)$$

où $P = p_1^{M_1}$, $p_{0|1}^*$ est déterminé dans le corollaire C.2 de l'annexe C.3.

Preuve. Cette fois encore, ce problème est très similaire à celui posé dans un réseau plat (proposition 5.3 à la page 99), en identifiant $p_0 \Leftrightarrow p_1$ (probabilité de sélectionner un peer-group) et $q_0 \Leftrightarrow p_{0|1}^*$ dans la proposition 5.3. \square

7.2.4 Crankback.

Dans ce modèle hiérarchique, nous pouvons introduire plusieurs variables afin de mieux appréhender le surcoût causé par les crankbacks. Soit \mathcal{N}_c^0 le nombre moyen de crankbacks internes à un peer-group (que l'appel traverse finalement la cellule ou non) et soit \mathcal{N}_c^1 le nombre de crankbacks mettant en jeu le niveau hiérarchique supérieur (c.à.d pour lequel le crankback retourne jusqu'au nœud source). Ces second crankbacks peuvent être bien plus coûteux d'un point de vue signalétique. Finalement, \mathcal{N}_c représentera le nombre total de crankbacks subis par un appel.

Résultats partiels.

Proposition 7.3

$$\mathcal{N}_c^0 = \frac{1 - Q}{Q} p_{0|1} \quad (7.3)$$

$$\mathcal{N}_c^1 = \frac{1 - Q_1}{Q_1} (1 - \mathcal{P}_b^1) \quad (7.4)$$

avec $Q = q_0^{M_0}$ et $P = p_{0|1}^{M_1}$. $p_{0|1}$ est exprimé dans le théorème C.3 à la page 226 et \mathcal{P}_b^1 dans la proposition 7.1 ci-dessus.

Preuve. La démonstration est similaire à celle de la proposition 5.4 dans le cas d'un réseau plat, en identifiant $1 - \mathcal{P}_b \Leftrightarrow p_{0|1}$ pour \mathcal{N}_c^0 et $q_0 \Leftrightarrow p_{0|1}$ pour \mathcal{N}_c^1 . \square

Résultats globaux.

Lemme 7.1 Soient \mathcal{N}_{cp}^0 et \mathcal{N}_{cb}^0 le nombre moyen de crankbacks subis par une connexion avant que celle-ci passe finalement à travers le peer-group ou soit finalement bloquée (c.à.d renvoyée au nœud source), respectivement. Ces variables s'expriment de cette manière :

$$\mathcal{N}_{cb}^0 = \frac{1}{1 - p_{0|1}} \mathbb{E} \left[R_0 \wedge K_0^c (1 - Q)^{R_0 \wedge K_0^c} \mid X^1 > b \right] \quad (7.5)$$

$$\mathcal{N}_{cp}^0 = \frac{1 - Q}{Q} - \mathcal{N}_{cb}^0 \frac{1 - p_{0|1}}{p_{0|1}} \quad (7.6)$$

où R_0 représente le nombre de routes admissibles pour traverser un domaine donné et où $Q = q_0^{M_0}$. \mathcal{N}_{cb}^0 peut être aisément évalué comme montré dans le corollaire C.1 en annexe.

Preuve. Bien évidemment, ces grandeurs sont reliées par la relation :

$$\mathcal{N}_{cp}^0 p_{0|1} + \mathcal{N}_{cb}^0 (1 - p_{0|1}) = \mathcal{N}_c^0$$

qui conduit à l'expression de \mathcal{N}_{cp}^0 . Ensuite, remarquons que, puisque l'appel est refusé, le nombre de crankbacks est égal au nombre total de routes admissibles (limité à K_0^c) — d'où l'expression de \mathcal{N}_{cb}^0 . \square

Lemme 7.2 Le nombre de peer-groups traversés sur un chemin donné (de niveau 1) est :

- M_1 , si l'appel est accepté sur cette route.
- $\mathcal{N}_{PG} < M_1$, si l'appel est refusé et subit un crankback, avec

$$\mathcal{N}_{PG} = \frac{p_{0|1}}{1 - p_{0|1}} - M_1 \frac{p_{0|1}^{M_1}}{1 - p_{0|1}^{M_1}}$$

Preuve. Il suffit de suivre les indications de la démonstration du lemme 5.1 sur l'analyse d'un réseau plat en identifiant $q \Leftrightarrow p_{0|1}$ et $M_0 \Leftrightarrow M_1$. \square

Proposition 7.4 Utilisant les lemmes 7.1 et 7.2, le nombre moyen de crankbacks qu'un appel subit est donné par l'expression :

$$\mathcal{N}_c = \mathcal{N}_c^1 [\mathcal{N}_{cb}^0 + \mathcal{N}_{PG} \mathcal{N}_{cp}^0] + M_1 \mathcal{N}_{cp}^0 (1 - \mathcal{P}_b^1) \quad (7.7)$$

Preuve. Soit Υ_r^1 l'événement "GCAC sélectionne r routes admissibles (à partir des K_1 routes précalculées)". Alors, il est facile de voir que :

$$\mathbb{E}[N_c | N_c^1 = n, \Upsilon_r^1] = n (\mathcal{N}_{PG} \mathcal{N}_{cp}^0 + \mathcal{N}_{cb}^0) + M_1 \mathcal{N}_{cp}^0 \mathbf{1}_{\{n < r \wedge K_1^c\}}$$

Il suffit alors de déconditionner pour conclure. \square

Soit \mathcal{P}_c la probabilité qu'une demande d'ouverture de connexion subisse un crankback au moins une fois. De même, soit \mathcal{P}_c^1 la probabilité qu'un appel subisse un crankback qui le ramène jusqu'au nœud source au moins une fois (sans considérer les crankbacks internes aux peer-groups). Enfin, soit \mathcal{P}_c^0 la probabilité qu'un appel subisse au moins un crankback interne lors de la traversée d'un domaine de routage.

Proposition 7.5

$$\begin{aligned} \mathcal{P}_c^0 &= 1 - q_0^{M_0} \\ \mathcal{P}_c^1 &= 1 - p_{0|1}^{M_1} \\ \mathcal{P}_c &= 1 - \left[\left(1 - p_{0|1}^* \right) q_0^{M_0-1} \right]^{M_1} \end{aligned}$$

où $p_{0|1}^*$ est exprimé en annexe dans le corollaire C.2.

Preuve. Les expressions de \mathcal{P}_c^0 et de \mathcal{P}_c^1 sont triviales — elles correspondent à $1 - \mathbb{P}\{N_c^0 = 0\}$ et $1 - \mathbb{P}\{N_c^1 = 0\}$ respectivement. Pour \mathcal{P}_c , remarquons que la connexion est acceptée sur le chemin sélectionné si, au niveau des M_1 peer-groups sélectionnés,

- L'appel n'est pas rejeté à l'entrée dans le domaine, c.à.d. s'il existe au moins une route admissible et si le premier lien de cette route accepte l'appel (fonction CAC), avec la probabilité $1 - p_{0|1}^*$ (voir corollaire C.2 en annexe).
- Tous les M_0 liens du chemin (interne) sélectionné acceptent l'appel, avec la probabilité q_0 . Notons cependant que le premier lien a déjà été pris en compte (fonction CAC) dans le terme $p_{0|1}^*$.

\square

7.2.5 Surcoût protocolaire

Comme nous l'avons fait dans l'analyse d'un réseau plat, introduisons η , le nombre moyen de liens traversés inutilement pendant la phase d'établissement de la connexion, divisés par la "longueur" de cette connexion, c.à.d. $M_1 M_0$. Introduisons également η^1 , le nombre moyen de peer-groups visités inutilement, divisé par M_1 , et η^0 , le nombre moyen de liens visités inutilement lors d'une tentative de traversée d'un peer-group, divisé par M_0 .

Résultats partiels.**Proposition 7.6**

$$\eta^0 = \frac{\mathcal{N}_c^0 \mathcal{N}_l}{M_0} \quad (7.8)$$

$$\eta^1 = \frac{\mathcal{N}_c^1 \mathcal{N}_{PG}}{M_1} \quad (7.9)$$

Dans l'expression de η^0 , \mathcal{N}_l obéit à la même expression que celle du lemme 5.1 à la page 101 et \mathcal{N}_c^0 est explicité dans la proposition 7.3. Dans l'expression de η^1 , \mathcal{N}_{PG} obéit au lemme 7.2 et \mathcal{N}_c^1 à la proposition 7.3.

Preuve. Nous retombons dans un cas similaire à celui de la proposition 5.5 de la page 101 pour un modèle plat. \square

Résultats globaux.**Proposition 7.7**

$$\eta = \mathcal{N}_c^1 \frac{\mathcal{N}_l \mathcal{N}_{cb}^0 + \mathcal{N}_{PG} (\mathcal{N}_l \mathcal{N}_{cp}^0 + M_0)}{M_1 M_0} + (1 - \mathcal{P}_b^1) \frac{\mathcal{N}_l \mathcal{N}_{cp}^0}{M_0} \quad (7.10)$$

Preuve. Rappelons-nous que le nombre moyen de liens visités sur une route admissible est :

- M_0 , si la connexion est acceptée sur cette route,
- $\mathcal{N}_l < M_0$, si la connexion est refusé par un nœud de cette route (par la fonction **CAC**), où \mathcal{N}_l possède la même expression que celle donnée dans l'étude d'un réseau plat.

Soit alors \mathcal{N}_l^p le nombre moyen de liens visités (lors de l'établissement de la connexion) à l'intérieur d'un peer-group que l'appel réussie finalement à traverser et soit \mathcal{N}_l^b le nombre moyen de liens visités dans un peer-group bloquant la connexion. Il est alors aisé de voir que :

$$\mathcal{N}_l^p = \mathcal{N}_l \mathcal{N}_{cp}^0 + M_0$$

$$\mathcal{N}_l^b = \mathcal{N}_l \mathcal{N}_{cb}^0$$

où l'expression de \mathcal{N}_l est donnée dans le lemme 5.1 de la page 101, et où \mathcal{N}_{cp}^0 et \mathcal{N}_{cb}^0 sont déterminés dans le lemme 7.1. Finalement, ce calcul se termine de manière similaire à celui de la proposition 5.5 à la page 101. Les mêmes arguments conduisent à l'expression :

$$\eta = \mathcal{N}_c^1 \frac{\mathcal{N}_l^b + \mathcal{N}_{PG} \mathcal{N}_l^p}{M_0 M_1} + (1 - \mathcal{P}_b^1) \frac{(\mathcal{N}_l^p - M_0) M_1}{M_1 M_0}$$

\square

7.2.6 Routage aveugle

Motivations.

Pour comparaison, nous introduisons une autre procédure de routage, dite “aveugle” car la fonction **GCAC** n’est pas utilisée au niveau logique. En d’autres termes, l’établissement de la connexion est envoyé sur la première route logique (d’après les tables de plus court chemin précalculées) en faisant abstraction de toute information agrégée. La fonction de **GCAC** n’est utilisée qu’au niveau physique lors de la traversée des peer-groups sélectionnés⁴. Si la connexion est rejetée par un peer group de ce chemin, une procédure de crankbacks est mise en œuvre pour libérer les ressources. L’appel revient alors jusqu’à la source et une autre route est choisie — encore une fois, sans faire appel à l’état de congestion agrégé des peer-groups de ce chemin. Cet algorithme s’apparente donc avec les algorithmes de routage existants dans le réseau internet par exemple, où aucune information sur la congestion n’est prise en compte pour le calcul des chemins.

Bien entendu, une telle solution aura pour conséquence fâcheuse d’augmenter très significativement le nombre moyen de crankback subis par les demandes d’établissement de connexions. Cependant, cette approche est justifiée à posteriori par les très fortes probabilités de rejet à la source observées lorsque la fonction de **GCAC** est utilisée sur des informations agrégées de manière conservatrice.

Préliminaires.

La probabilité de traverser avec succès un peer-group sera notée $\hat{p}_{0,1}$ lorsque l’algorithme de routage “aveugle” est utilisé. Cette probabilité prend une expression beaucoup plus simple, comme le montre la proposition suivante :

Proposition 7.8

$$\hat{p}_{0,1} = 1 - \sum_{r=0}^{K_0} \binom{K_0}{r} P^r (1-P)^{K_0-r} (1-Q)^{r \wedge K_0^c}$$

où $P = p_0^{M_0}$ et $Q = q_0^{M_0}$.

Preuve. Puisque aucune information sur les bandes passantes agrégées n’est prise en compte par cet algorithme de routage pour sélectionner les chemins, $1 - \hat{p}_{0,1}$ obéit à la même expression que la probabilité de perte dans un réseau plat (cf. proposition 5.2 à la page 99). \square

Dans toute la suite, nous donnons toutes les expressions des mesures de performances sans démonstration. Ces expressions sont déterminées simplement à partir des résultats précédents.

⁴Nous avons vu dans le chapitre précédent que la fonction **GCAC** permet d’obtenir une très bonne utilisation du réseau tout en limitant le surcoût signalétique à l’intérieur d’un domaine de routage.

Perte d'appel.

Soit $\widehat{\mathcal{P}}_b^1$ la probabilité de perte d'une connexion lorsque la fonction GCAC n'est pas utilisée au niveau hiérarchique supérieur. Cette grandeur obéit à la relation :

$$\widehat{\mathcal{P}}_b^1 = \left(1 - \widehat{p}_{0,1}^{M_1}\right)^{K_c^1}$$

Rejet à la source.

Bien sûr, avec cette procédure de routage "aveugle", il n'y a pas de rejet de connexion au niveau hiérarchique supérieure. Par contre, la fonction GCAC est toujours utilisée au niveau physique, et nous avons,

$$\begin{aligned}\widehat{\mathcal{P}}_r^0 &= (1 - Pq_0)^{K_0} \\ \widehat{\mathcal{P}}_r^1 &= \left[\widehat{\mathcal{P}}_r^0\right]^{K_1}\end{aligned}$$

avec $P = p_0^{M_0}$.

Crankback.

Comme précédemment, nous introduisons plusieurs variables, à savoir $\widehat{\mathcal{N}}_c^0$, $\widehat{\mathcal{N}}_c^1$, $\widehat{\mathcal{N}}_c$ et $\widehat{\mathcal{P}}_c^0$, $\widehat{\mathcal{P}}_c^1$, $\widehat{\mathcal{P}}_c$.

$$\begin{aligned}\widehat{\mathcal{N}}_c^0 &= \frac{1 - q_0}{q_0} \widehat{p}_{0,1} \\ \widehat{\mathcal{N}}_c^1 &= \frac{1 - \widehat{p}_{0,1}}{\widehat{p}_{0,1}} \left(1 - \widehat{\mathcal{P}}_b^1\right) \\ \widehat{\mathcal{P}}_c^0 &= 1 - q_0^{M_0} \\ \widehat{\mathcal{P}}_c^1 &= 1 - \widehat{p}_{0,1}^{M_1}\end{aligned}$$

En utilisant les mêmes conventions que le lemmes 7.1 et 7.2, les identités suivantes sont vérifiées :

$$\begin{aligned}\widehat{\mathcal{N}}_{cb}^0 &= \frac{1}{1 - \widehat{p}_{0,1}} \sum_{r=0}^{K_0} \binom{r}{K_0} (r \wedge K_0^c) Q^r (1 - Q)^{K_0 - r} \\ \widehat{\mathcal{N}}_{cp}^0 &= \frac{1 - q_0}{q_0} - \frac{1 - \widehat{p}_{0,1}}{\widehat{p}_{0,1}} \widehat{\mathcal{N}}_{cb}^0 \\ \widehat{\mathcal{N}}_{PG} &= \frac{\widehat{p}_{0,1}}{1 - \widehat{p}_{0,1}} - M_1 \frac{\widehat{p}_{0,1}^{M_1}}{1 - \widehat{p}_{0,1}^{M_1}}\end{aligned}$$

où $P = p_0^{M_0}$ et $Q = q_0^{M_0}$.

$$\begin{aligned}
\widehat{\mathcal{N}}_c &= \widehat{\mathcal{N}}_c^1 \left(\widehat{\mathcal{N}}_{cb}^0 + \widehat{\mathcal{N}}_{PG} \widehat{\mathcal{N}}_{cp}^0 \right) + M_1 \widehat{\mathcal{N}}_{cp}^0 \left(1 - \widehat{\mathcal{P}}_b^1 \right) \\
\widehat{\mathcal{P}}_c &= 1 - \left[(1 - \widehat{p}_{0,1}^*) q_0^{M_0-1} \right]^{M_1} \\
\widehat{p}_{0,1}^* &= \left(1 - p_0^{M_0} q_0 \right)^{K_0}
\end{aligned}$$

Surcoût.

En utilisant les notations introduites au paragraphe 7.2.5, les identités suivantes sont vérifiées :

$$\begin{aligned}
\widehat{\eta}^0 &= \frac{\widehat{\mathcal{N}}_c^0 \mathcal{N}_l}{M_0} \\
\widehat{\eta}^1 &= \frac{\mathcal{N}_c^1 \widehat{\mathcal{N}}_{PG}}{M_1} \\
\widehat{\eta} &= \widehat{\mathcal{N}}_c^1 \frac{\mathcal{N}_l \widehat{\mathcal{N}}_{cb}^0 + \widehat{\mathcal{N}}_{PG} \left(\mathcal{N}_l \widehat{\mathcal{N}}_{cp}^0 + M_0 \right)}{M_1 M_0} + \left(1 - \widehat{\mathcal{P}}_b^1 \right) \frac{\mathcal{N}_l \widehat{\mathcal{N}}_{cp}^0}{M_0}
\end{aligned}$$

7.2.7 Déconditionnement

Les expressions précédentes ont été calculées sous des hypothèses d'indépendance (bande passante des liens i.i.d. et cellules indépendantes) et pour une réalisation des graphes aléatoires donnée. En effet, nous avons écrit toutes les expressions pour des quantités L_0 , M_0 et M_1 connues. Ces quantités sont en fait des variables aléatoires dépendant des paramètres λ_0 et λ_1 .

Hélas, peu de résultats sont disponibles sur ces variables aléatoires. Si un algorithme de routage "markovien" est utilisé (cf. paragraphe B.4.2 en annexe), nous savons d'après le théorème de Møller, que $\mathbb{E}[M_1] = \frac{4}{\pi} \sqrt{\lambda_1} d$ où d est le diamètre du réseau à traverser ($d = \frac{2}{\sqrt{\pi}}$ dans notre cas). En revanche, la distribution exacte de M_1 ne nous est pas connue. Il est par contre clair que les variations de M_1 sont de plus en plus faibles (par rapport à sa moyenne) lorsque le diamètre d augmente. En l'absence de résultats plus précis, nous considérerons que le réseau étudié est suffisamment étendu pour poser

$$M_1 \approx \frac{4}{\pi} \sqrt{\lambda_1} \frac{2}{\sqrt{\pi}}$$

Le cas de M_0 est bien plus complexe. Si le théorème de Møller peut également être appliqué, donnant $\mathbb{E}[M_0] = \frac{4}{\pi} \sqrt{\lambda_0} d_0$, le diamètre de traversée d'une cellule est également une variable aléatoire. Des arguments géométriques montrent cependant que d_0 est proportionnel à $\frac{1}{\sqrt{\lambda_1}}$.

En l'absence de résultats plus précis, nous poserons, de manière assez grossière,

$$M_0 \approx \frac{4}{\pi} \sqrt{\frac{\lambda_0}{\lambda_1}}$$

Finalement, L_0 , le nombre de liens présents dans un domaine de routage doit être déterminé. Nous savons que le nombre moyen de nœuds d'un domaine est donné par l'expression

$$\mathbb{E}[N_0] = \frac{\lambda_0}{\lambda_1}$$

comme montré à la page 118 dans la proposition 6.1 par exemple. En identifiant chaque lien par son milieu, il est possible de construire un processus ponctuel. Comme montré au paragraphe B.3.2 en annexe, l'intensité du processus ainsi construit est égale à $\frac{3}{2}\lambda_0$. Nous poserons donc

$$L_0 = \left\langle \frac{3}{2} \frac{\lambda_0}{\lambda_1} \right\rangle$$

où $\langle x \rangle$ désigne le plus grand entier naturel $n \in \mathbb{N}$ vérifiant $n \leq x$.

Il ne s'agit donc pas vraiment d'un déconditionnement puisque, par manque de résultats plus précis, il faut se contenter de remplacer les variables aléatoires par leurs moyennes respectives. Cette approche permet néanmoins de raisonner avec des grandeurs à peu près réalistes. Le paragraphe 7.3.4 présente une technique de déconditionnement plus complexe en vue d'étudier spécifiquement l'impact de la structure hiérarchique.

7.3 Résultats

7.3.1 Influence de l'agressivité.

Comportement global.

Sur la figure 7.2, sont représentées les probabilités de perte d'un appel, de crankback et de rejet à la source en fonction de l'agressivité de l'algorithme d'agrégation. Les traits pleins représentent ces grandeurs lorsque la fonction **GCAC** est utilisée et les pointillés pour le routage à l'aveugle. Les paramètres ont été fixés comme suit : $L_0 = 75$, $M_0 = 7$, $M_1 = 3$, $K_0 = 3$, $K_c^0 = 2$, $K_1 = 2$, $K_c^1 = 1$, $p = 0.9$ et $q = 0.9$.

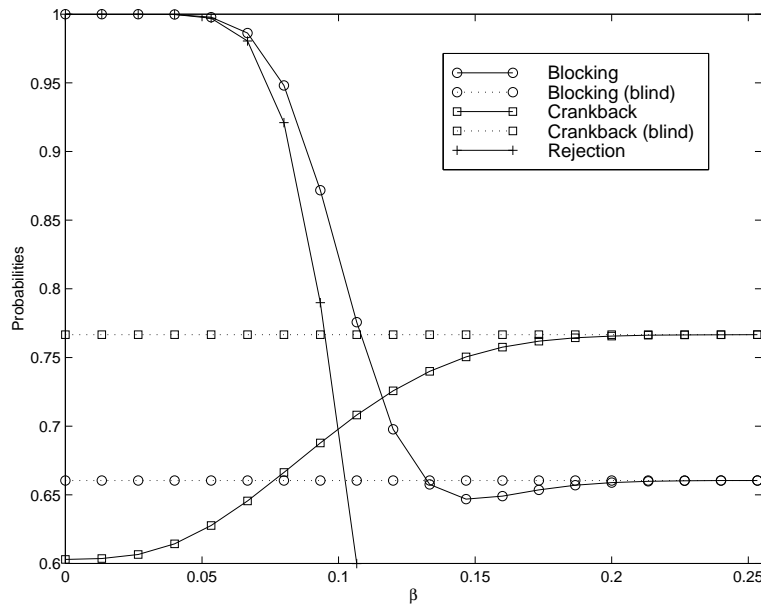


FIG. 7.2 – Influence de β sur les performances du réseau.

Lorsque $\beta \rightarrow 0$, la probabilité de rejet à la source est très proche de 1. Ceci est dû à la politique d'agrégation fort conservatrice sur un domaine de routage relativement important (75 liens en moyenne). La probabilité de perte des appels entrants est donc proche de 1 car ces demandes d'établissement de connexion seront quasi-systématiquement rejetées directement par le noeud source. Bien sûr, la probabilité de crankback est faible. Les crankbacks sont dus, dans ce cas précis, aux erreurs induites par le processus de rafraîchissement des informations de routage — Rappelons que $q = 0.9$.

Lorsque $\beta \rightarrow 1$ au contraire, la probabilité de rejet à la source devient faible. Les appels sont rejetés au niveau physique principalement (la probabilité de crankback est maximale) :

Les valeurs agrégées étant très optimistes, il est pratiquement toujours possible de trouver un chemin au niveau hiérarchique le plus haut. En revanche, la probabilité de crankback devient très importante car les valeurs agrégées sont trop optimistes...

Pertes d'appels.

La figure 7.3 donne la probabilité de perte d'appel selon l'agressivité et selon le nombre de crankbacks K_c^0 autorisés au niveau physique — les autres paramètres étant identiques à ceux de la figure 7.2. Le rôle du reroutage est particulièrement explicite sur ce graphe. Par contre, comme pour le cas d'un réseau plat, il apparaît que la possibilité d'un second re-routage n'est pas essentielle car les performances n'en sont pas améliorées significativement.

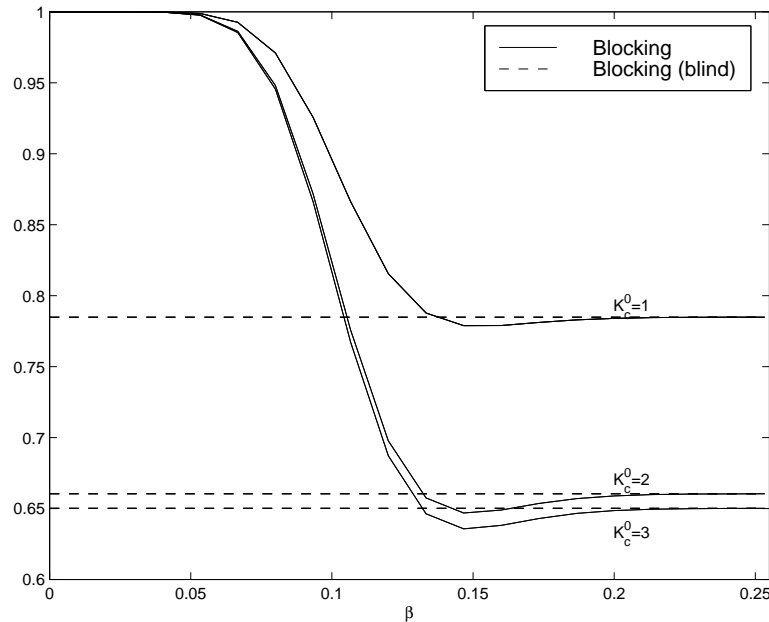


FIG. 7.3 – Influence de β sur la perte d'appels ($K_c^1 = 1$).

L'influence de K_c^1 sur les performances est moins importante, comme il est possible de constater en comparant les figures 7.3 et 7.4.

Les courbes précédentes permettent de mettre en évidence le compromis qui existe entre le rejet d'appels à la source, lorsque la politique d'agrégation est trop conservatrice, et les pertes d'appels après Crankbacks, lorsqu'au contraire la politique d'agrégation est trop agressive. Lorsque le reroutage n'est pas autorisé au niveau physique, l'impact du crankback influe beaucoup plus sur les pertes d'appels (cf. figures 7.4 et 7.3) de sorte que la probabilité de perte n'est pas monotone. \mathcal{P}_b diminue d'abord car le rejet à la source devient moins probable, mais, passé un certain seuil, la probabilité de crankback augmente très vite de sorte que

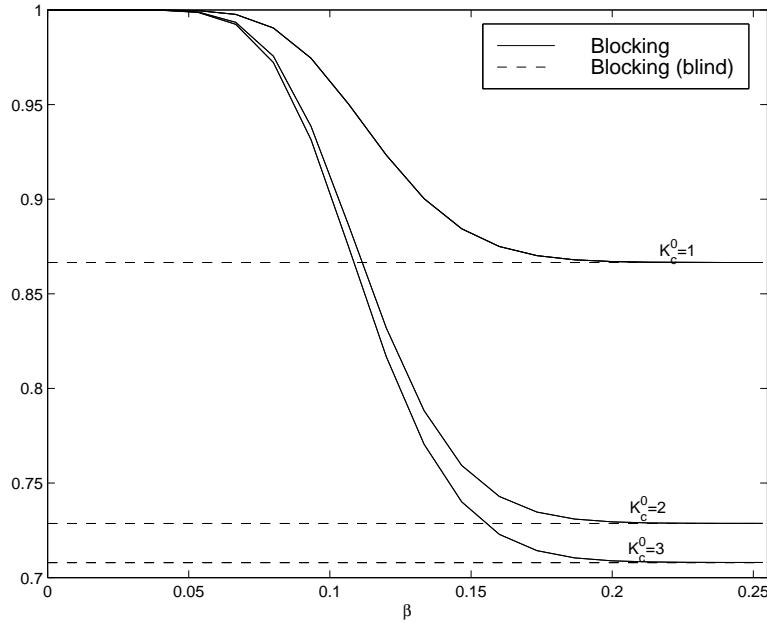


FIG. 7.4 – Influence de β sur la perte d'appels ($K_c^1 = 2$).

\mathcal{P}_b devient croissant lorsque β augmente. Le minimum de $\beta \mapsto \mathcal{P}_b$ correspond donc à une agressivité optimale, c.à.d. à un compromis optimal entre le rejet d'appel et les crankbacks.

Il semble par contre que la valeur de β pour laquelle l'optimum est atteint n'obéisse pas une expression analytique simple. Remarquons également que la notion d'agrégation "optimale" n'a de sens que dans le cas où le reroutage est interdit au niveau hiérarchique supérieur (cf. figure 7.4). De plus, cette "optimalité" est définie par rapport à la perte d'appel. Il ne s'agit certainement pas d'une valeur optimale du point de vue du surcoût protocolaire — le nombre moyen de crankbacks est déjà relativement important pour un tel niveau d'agressivité, comme le montrent les figures suivantes.

Crankbacks et surcoût.

La figure 7.5 montre l'évolution du nombre moyens de crankbacks subis par un appel et la figure 7.6 le surcoût signalétique que ces crankbacks représentent. Les paramètres du réseau sont les mêmes que ceux de la figure 7.2. En effet, lorsque que la politique d'agrégation est conservatrice (c.à.d. $\beta \rightarrow 0$), le nombre moyen de crankbacks est très faible — les crankbacks sont dus uniquement à des problèmes de rafraîchissement des informations de routage mais la valeur $q = 0.95$ choisie conduit à un nombre moyen de crankback quasiment nul. Lorsque β augmente, le nombre moyen de crankbacks augmente car le nombre moyen de routes disponibles diminue : Rappelons nous qu'une proportion β des liens ont une bande passante

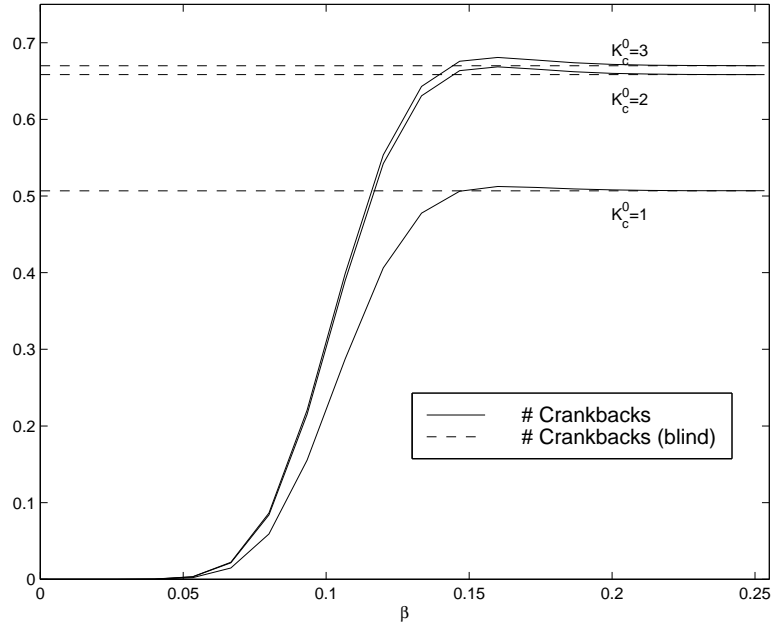


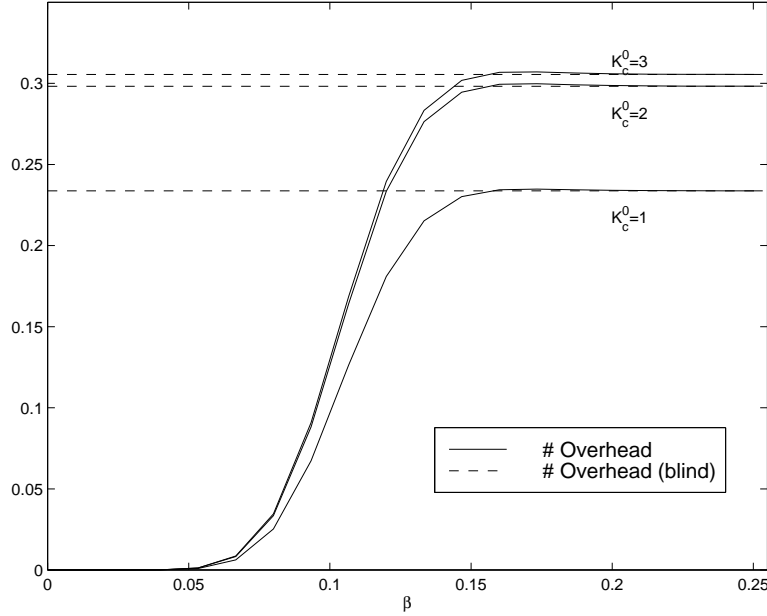
FIG. 7.5 – Influence de β sur le nombre moyen de crankbacks.

inférieure à la valeur agrégée (voir annexe C.1). Lorsque les informations agrégées deviennent très agressives, le nombre moyen de crankbacks atteint rapidement un palier.

Politiques agressives.

Sur les courbes précédentes, nous pouvons remarquer que lorsque $\beta \rightarrow 1$, c'est à dire pour des politiques très agressives, les performances obtenues sont identiques que la fonction GCAC soit utilisée ou non. Intuitivement, ce fait s'explique aisément : Lorsque la politique d'agrégation devient trop agressive, les informations résumées deviennent très grossières. Avec les paramètres choisis (en particulier d'assez grand domaines), les informations agrégées deviennent vite si grossières que leur propagation et leur utilisation par la fonction GCAC devient totalement caduque.

Du point de vue mathématique, cette propriété se retrouve également : Il est possible de montrer que lorsque $\beta \rightarrow 1$, le nombre de routes admissibles pour la traversée d'un domaine, noté R_0 , suit approximativement une loi binomiale — comme c'est le cas pour le routage à l'aveugle. Il résulte que les probabilités de rejet ou de crankbacks subis par une connexion à la traversée d'un peer-group sont identiques au cas du routage aveugle. La connaissance de la valeur agrégée (c.à.d. le conditionnement sur $X^1 > b$) n'offre donc aucune garantie supplémentaire sur la possibilité de traverser ou non un domaine avec succès. Une preuve de cette propriété est donnée en annexe C à la page 225.

FIG. 7.6 – Influence de β sur le surcoût signalétique.

Comportements asymptotiques.

Il est également important de noter que les courbes $\beta \mapsto \mathcal{N}_c(\beta)$ et $\beta \mapsto \eta(\beta)$ évoluent inversement à celle de \mathcal{P}_b . Un compromis assez difficile doit donc être trouvé entre le surcoût protocolaire et l'utilisation du réseau. Ce compromis est d'autant plus difficile à déterminer que les courbes évoluent très rapidement entre deux valeurs extrêmes. Nous pouvons en effet remarquer que toutes les courbes précédentes présentent un seuil très marqué.

- En dessous d'une certaine agressivité⁵ β^* , les divers critères de performances sont quasiment constants. Cette zone peut être dénommée “**zone de rejet**” en raison des très fortes probabilités de rejet observées pour $\beta < \beta^*$.
- Pour $\beta > \beta^*$, les différentes quantités observées atteignent très vite un plateau. Cette zone peut être appelée “**zone de crankbacks**” en raison des forts taux de crankbacks subis par les connexions entrantes pour des valeurs très agressives de β .

Ce comportement en palier (zone de rejet/zone de Crankback) s'explique facilement du point de vue mathématique. Comme le montre la proposition C.2 de annexe C, il existe une valeur seuil \tilde{X}^* pour laquelle, lorsque le nombre de liens du réseau à agréger devient très important (c.à.d. $L_0 \rightarrow \infty$) :

$$\mathbb{P}\{X^1 > b\} \approx \begin{cases} 1 & \text{pour } b < \tilde{X}^* \\ 0 & \text{pour } b > \tilde{X}^* \end{cases} \quad (7.11)$$

⁵ Avec $\beta^* \approx 1 - p_0$ comme montré plus loin dans ce chapitre.

où X^* dépend de l'agressivité de la politique d'agrégation, cette valeur vérifiant l'identité :

$$\mathbb{P}\{\tilde{X} < \tilde{X}^*\} = \beta$$

Autrement dit, lorsque la taille du domaine de routage considéré est très grande, le système est forcément dans un des deux modes de fonctionnement (Zone de rejet / Zone de Crankbacks), suivant la valeur de b par rapport à \tilde{X}^* , c.à.d. suivant la charge du réseau et la bande passante demandée. Les courbes précédentes démontrent bien cette brève transition entre les zones de rejet et de crankbacks car les domaines considérés sont relativement grands (75 liens).

7.3.2 Influence de p_0

Pertes d'appels.

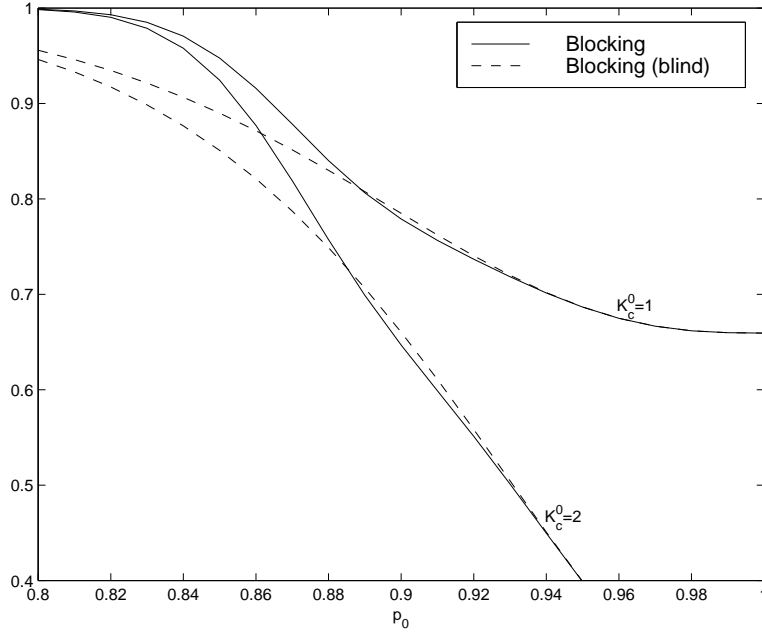
Sur la figure 7.7 est représentée l'évolution de la probabilité de perte d'un appel pour une politique d'agrégation donnée ($\beta = 0.15$) en fonction du paramètre p_0 et pour différentes valeurs de K_c^0 . Les autres paramètres du réseau sont identiques à ceux de la figure 7.2.

Nous pouvons observer très nettement deux phases distinctes : Lorsque $p \approx 1$, les probabilités de blocage avec ou sans GCAC sont très proches. La fonction GCAC permet tout de même d'obtenir de meilleures performances car le nombre de crankback au niveau logique est limité à $K_c^1 = 1$ (cf. figure 7.3). Lorsque p_0 diminue, les performances obtenus par le routage aveugle deviennent meilleures. En effet, la probabilité qu'un lien accepte la connexion (notée p_0) est faible de sorte que le rejet à la source devient très important.

Notion d'agressivité.

La remarque précédente souligne la difficulté que représente la détermination d'une politique d'agrégation adéquate (du point de vue de son agressivité). Nous allons en fait montrer dans la suite que la notion même d'agressivité est un facteur relatif à p_0 . Pour la figure 7.7, nous avons choisi une politique relativement agressive ($\beta = 0.15$) car elle permettait d'obtenir une utilisation optimum des ressources dans le cas de la figure 7.2, c'est à dire pour $p_0 = 0.9$. Lorsque p_0 devient plus petit, les pertes de connexions entrantes augmentent très significativement en raison des rejets à la source. Autrement dit, le choix de $\beta = 0.15$ peut être qualifié de "conservateur" pour $p_0 < 0.85$ environ (nous entrons dans la "zone de rejet") alors qu'il était "agressif" pour $p_0 \approx 0.9$.

Notons que le paramètre $p_0 = \mathbb{P}\{\tilde{X}^0 > b\}$ dépend à la fois de la charge du réseau et de la bande passante demandée b . Une politique d'agrégation peut donc être conservatrice pour certaines classes de connexions (requérant beaucoup de bande passante) et agressive pour d'autres.

FIG. 7.7 – Influence de p_0 sur la perte d'appels.

Pour mieux étudier la notion relative d'agressivité en fonction de p , nous avons tracé sur la figure 7.8 l'évolution de \mathcal{P}_b en fonction de p_0 et de β — la surface sous-jacente correspondant au cas du routage aveugle. Les paramètres du réseau sont identiques à ceux de la figure 7.4 (c.à.d. $K_c^1 = 2$). Les deux modes “zone de perte”/“Zone de crankbacks” apparaissent très nettement. Le seuil d'agressivité séparant ces deux domaines semble évoluer linéairement en fonction de p_0 . La proposition suivante confirme cette intuition.

Proposition 7.9 *La valeur d'agressivité seuil β^* est donnée par l'expression*

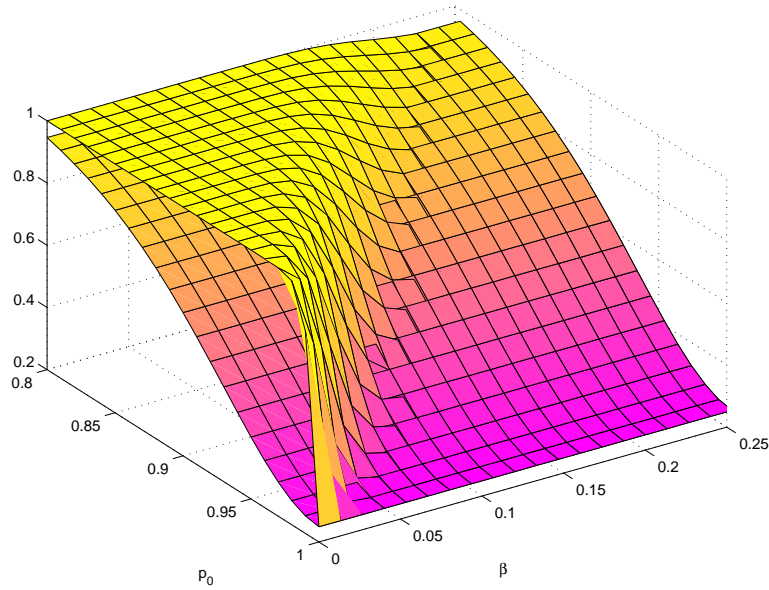
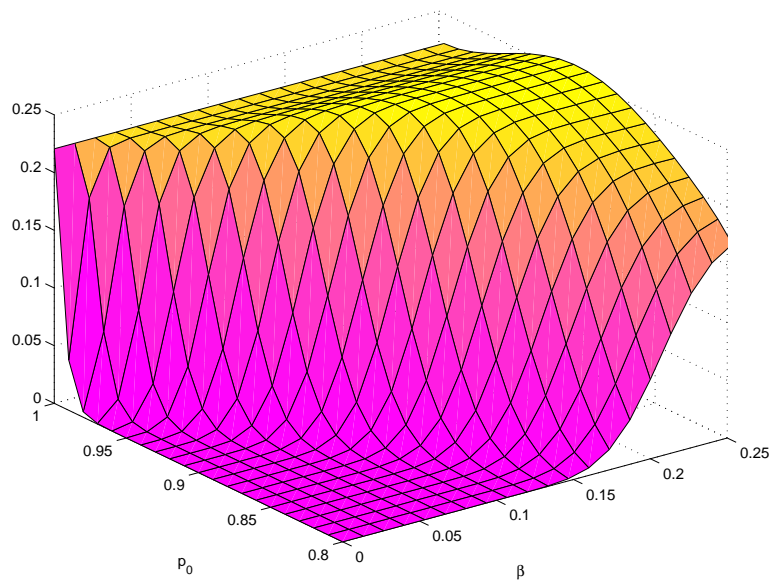
$$\beta^* = 1 - p_0$$

où $p_0 = \mathbb{P}\{\tilde{X}^0 > b\}$ si b désigne la bande passante requise par la connexion.

Preuve. Comme nous l'avons évoqué précédemment (cf. équation (7.11) à la page 155), le seuil est atteint pour une bande passante $b = \tilde{X}^*$ telle que $\mathbb{P}\{\tilde{X}^0 < \tilde{X}^*\} = \beta$. Le seuil est donc atteint pour $\beta = \mathbb{P}\{\tilde{X}^0 < b\} = 1 - p_0$ par définition de p_0 . \square

Surcoût protocolaire.

L'effet de seuil pour $\beta \approx 1 - p_0$ apparaît également clairement sur la figure 7.9 où est tracé l'évolution du surcoût protocolaire en fonction de p_0 et β . Ce surcoût est quasi nul dans la

FIG. 7.8 – Influence de l'agressivité sur la perte d'appels en fonction de p_0 .FIG. 7.9 – Influence de l'agressivité sur le surcoût signalétique en fonction de p_0 .

“zone de rejet” (caractérisé par $\beta < 1 - p_0$) et devient beaucoup plus important une fois la “zone de Crankbacks” franchie (c.à.d. $\beta > 1 - p_0$). L'évolution non monotone de η en fonction de p_0 dans cette zone de crankback est due à la croissance de \mathcal{N}_{PG} et la décroissance de \mathcal{N}_c^1 lorsque p_0 augmente — le surcoût étant à une constante additive près, proportionnel au produit de ces quantités. Cette évolution peut être comparée à celle du surcoût dans un réseau plat (cf. page 108) en fonction de q_0 .

7.3.3 Influence de q_0 .

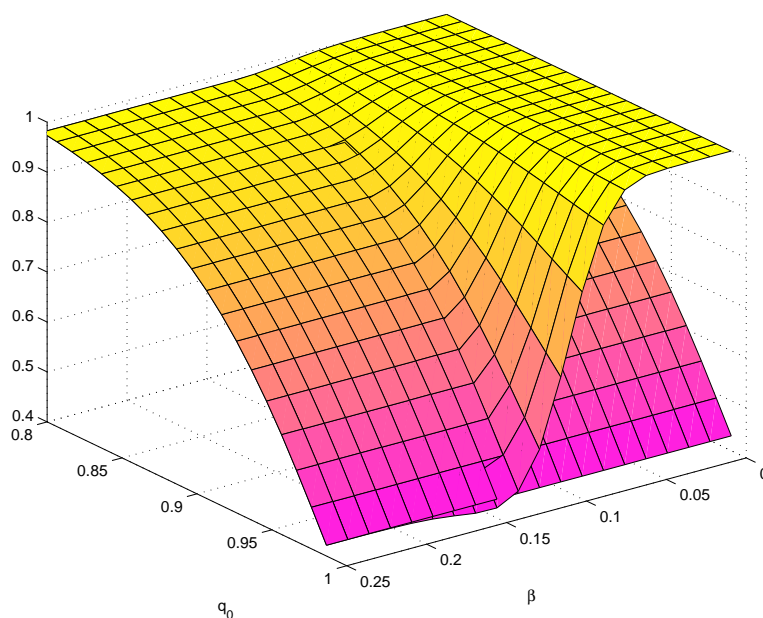


FIG. 7.10 – Influence de l'agressivité sur la perte d'appels en fonction de q_0 .

L'influence de la politique de rafraîchissement des informations de routage sur les performances d'un réseau hiérarchique peut également être appréhendée. Sur la figure 7.10 est représentée l'évolution de la probabilité de perte en fonction de q_0 et de l'agressivité β . Le surcoût protocolaire associé est représenté sur la figure 7.11 — la figure sous-jacente correspond au cas du routage aveugle. Les paramètres du réseau sont identiques à ceux de la figure 7.2.

La politique de rafraîchissement des informations de routage n'influe quasiment pas sur les performances du réseau pour des politiques d'agrégation conservatrice. En effet, la principale cause de perte est due au rejet à la source, et non au crankback. De plus, le surcoût reste très réduit puisque peu d'appels sont effectivement suivis d'une tentative d'ouverture de connexion.

Il est par contre facile à comprendre que le rôle de q_0 est très significatif pour des politiques agressives, en particulier lorsque le nombre de crankbacks est limité. L'évolution non monotone de η en fonction de q_0 dans la zone de crankback est à mettre en relation avec l'évolution de

$q \mapsto \eta$ dans le cas d'un réseau plat, au chapitre 5. Cette évolution découle en effet des mêmes phénomènes — \mathcal{N}_l étant croissant et \mathcal{N}_c^0 décroissant lorsque q_0 augmente.

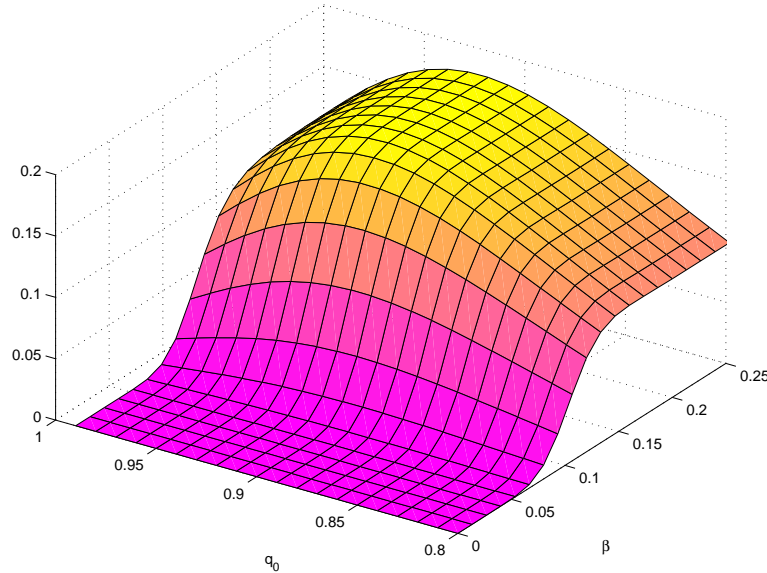


FIG. 7.11 – Influence de l'agressivité sur le surcoût signalétique en fonction de q_0 .

7.3.4 Structure hiérarchique

Jusqu'à présent, les résultats annoncés sont basés sur des variables conditionnelles par rapport à une réalisation des processus de Poisson π_0 et π_1 . Nous proposons dans la suite une méthode approchée permettant d'obtenir des moyennes approximatives. Cette analyse nous permettra d'étudier l'impact de la structure hiérarchique sur les performances globales du réseau.

Déconditionnement.

A la traversée d'un peer-group, les différentes expressions dépendent des deux paramètres L_0 et M_0 . Il est donc nécessaire de connaître la loi conjointe de L_0 et M_0 en fonction des intensités λ_0 et λ_1 . Étant donné le manque de résultats sur les chemins markoviens (et encore plus sur les plus court chemins), nous utiliserons la formule déterministe suivante :

$$M_0 \approx \left\langle \frac{4}{\pi} \sqrt{N_0} \right\rangle$$

La distribution du nombre de noeuds présents dans une cellule "typique"⁶ peut par contre être calculée.

⁶Par typique, il faut entendre un quantité calculée sous la probabilité de Palm, comme expliqué au paragraphe A.4 en annexe.

Proposition 7.10 *Le nombre de noeuds de $\text{supp}(\pi_0)$ présent dans une cellule “typique” générée par π_1 , $N_0 = \pi_0(C_0(\pi_1))$, suit approximativement la distribution suivante :*

$$\mathbb{P}_0^{\pi_1}\{N_0 = i\} = \frac{\Gamma(i + \alpha)}{\Gamma(i + 1)\Gamma(\alpha)} \frac{1}{\left(\frac{\mu}{\alpha} + 1\right)^\alpha \left(\frac{\alpha}{\mu} + 1\right)^i}$$

où $\mu = \frac{\lambda_1}{\lambda_0}$ ($\mu < 1$), $\alpha \approx 3.57$ et où $\mathbb{P}_0^{\pi_1}$ désigne la probabilité de Palm associée au processus π_1 (cf. paragraphe A.4 en annexe).

Preuve. Il suffit de se rappeler que π_0 est un processus de Poisson, de sorte qu’il est possible d’écrire :

$$\mathbb{P}_0^{\pi_1}\{N_0 = i\} = \mathbb{E}_0^{\pi_1} \left[\frac{(\lambda_0 |C_0(\pi_1)|)^i}{i!} e^{-\lambda_0 |C_0(\pi_1)|} \right]$$

Il suffit alors d’utiliser la proposition B.8 de l’annexe B pour achever ce calcul. \square

Bien sûr, nous savons que l’algorithme de routage utilisé peut induire un biais dans le choix des domaines ou des liens (cf. proposition B.12). Par exemple, si l’algorithme de routage markovien est utilisé, la taille moyenne des cellules traversée est légèrement plus grande que la moyenne. Nous négligerons ce phénomène en considérant des grandeurs non biaisées, c’est à dire des grandeurs calculées sous Palm. Il est alors possible de calculer une moyenne approchée des grandeurs précédentes en posant :

$$\begin{aligned} L_0 &= \left\langle \frac{3}{2} N_0 \right\rangle \\ M_0 &= \left\langle \frac{4}{\pi} \sqrt{N_0} \right\rangle \end{aligned}$$

où N_0 est calculé comme montré dans la proposition précédente. Cette approximation, bien que très grossière, permet de dégager quelques aspects qualitatifs intéressants sur l’influence de la structure hiérarchique.

Perte d’appels.

La figure 7.12 représente l’évolution de la probabilité de perte des connexions entrante en fonction de l’intensité λ_1 et pour des politiques d’agrégation plus ou moins agressives. Les autres paramètres sont fixés comme suit : $\lambda_0 = 200$, $p_0 = 0.9$, $q_0 = 0.95$, $K_c^1 = 1$, $K_c^0 = 2$, $K_1 = 2$, $K_0 = 2$ lorsque le domaine est suffisamment grand⁷. Il est important de se rappeler que la taille (en nombre de noeuds) moyenne des domaines de routage sont donnés par les identités suivantes : $m_0 = \frac{\lambda_0}{\lambda_1}$, $m_1 = \lambda_1$. Lorsque λ_1 est faible (à gauche sur notre graphique),

⁷ Il n’est pas possible d’envisager K_0 chemins indépendants comportant M_0 noeuds chacun si $N_0 < 2M_0$ par exemple.

le réseau est constitué d'un faible nombre de domaines, chacun de ses domaines comportant par contre un grand nombre de nœuds logiques. Si au contraire λ_1 est grand, le réseau est constitué d'un grand nombre de peer-groups de petites tailles.

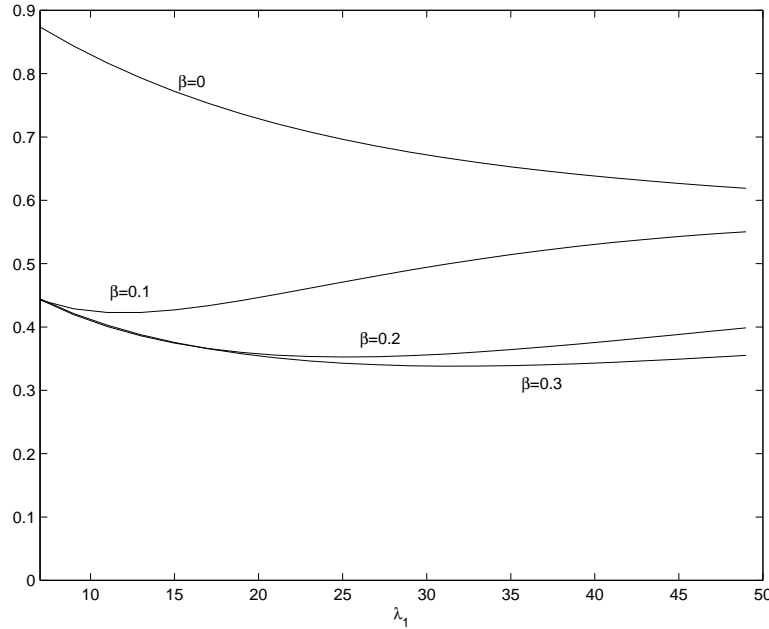


FIG. 7.12 – Influence de la structure hiérarchique sur les pertes d'appels.

Lorsque la politique d'agrégation est très conservatrice ($\beta = 0$ sur la figure 7.12), les pertes sont dues essentiellement aux rejets à la source. Cette probabilité de rejet dépend à la fois de p_1 , qui diminue avec λ_1 ($p_1 = p_0^{L_0} = p_0^{\frac{3}{2}\lambda_1}$), et du nombre de domaines à traverser qui augmente avec λ_1 . Néanmoins, M_1 augmente proportionnellement à $\sqrt{\lambda_1}$ alors que la décroissance de L_0 est de l'ordre de $\frac{1}{\lambda_1}$. On obtient donc bien une courbe décroissante lorsque λ_1 croît.

Pour une politique très agressive ($\beta = 0.3$ sur la figure 7.12), les pertes d'appels sont dues essentiellement au blocages après crankback. La qualité des informations agrégées se dégradant avec la taille des peer-groups de niveau physique, la probabilité de crankback diminue avec λ_1 ce qui cause moins de pertes d'appels. On remarque cependant une légère croissance de \mathcal{P}_b pour de fortes valeurs de λ_1 . Cette incurvation peut être expliquée par la faible taille des domaines dans cette zone ($m_0 = 5$ pour $\lambda_1 = 40$) ce qui réduit considérablement le nombre de chemins possibles (K_0 ne pouvant être égale à deux pour de petits peer-groups).

Pour des valeurs intermédiaires de β , la probabilité de perte évolue entre ces deux comportements. Il faut noter qu'une agressivité $\beta = 0.1$ ne diffère de $\beta = 0$ que pour des domaines comportant plus de 10 liens — Rappelons nous que $X^1 = \tilde{X}_{(m)}$ avec $m = \lfloor \beta L_0 \rfloor + 1$ (cf. paragraphe C.1). On peut donc comprendre que les résultats obtenus pour $\beta = 0.1$, par exemple,

soient assez proche de ceux d'une politique agressive pour de grands peer-groups, mais évolue ensuite vers une conduite conservatrice pour des peer-groups de faible taille.

Crankbacks et Surcoûts associés.

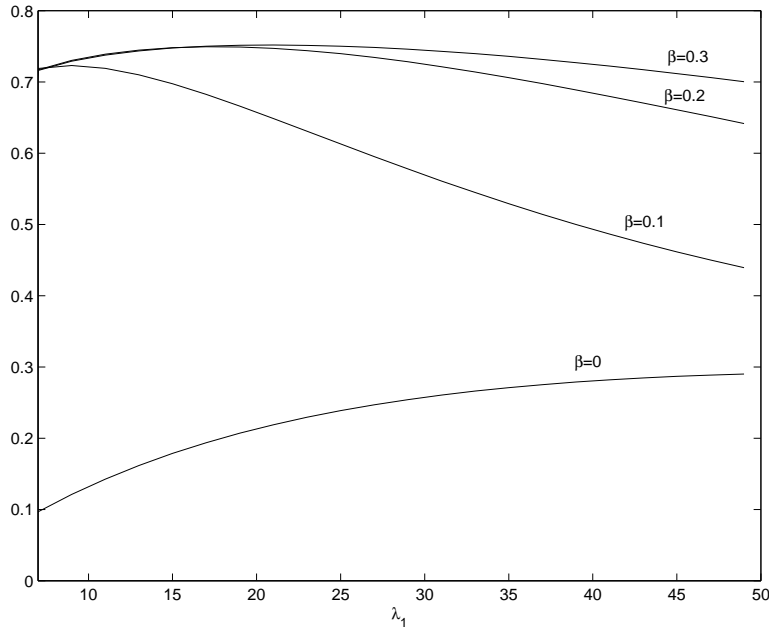


FIG. 7.13 – Influence de la structure hiérarchique sur le nombre moyen de Crankbacks.

Sur la figure 7.13 est représentée l'évolution du nombre de crankbacks en fonction de λ_1 pour plusieurs niveaux d'agressivité de la fonction d'agrégation. La figure 7.14 permet d'observer le surcoût protocolaire associé.

Comme nous l'avons déjà mis en évidence ci-avant, les politiques agressives permettent d'obtenir une meilleure utilisation du réseau au prix d'un plus grand nombre de crankbacks. La figure 7.13 nous permet également de remarquer que l'évolution de \mathcal{N}_c par rapport à λ_1 est inverse de celle de \mathcal{P}_b^1 . Cette constatation n'est guère surprenant car \mathcal{N}_c est proportionnelle à $1 - \mathcal{P}_b^1$ comme le montrent les propositions 7.4 et 7.3. Les autres quantités mises en jeu n'évoluent pas significativement avec les paramètres choisis ($q = 0.95$).

Influence de la hiérarchie.

Il est intéressant de comparer les performances obtenues par un réseau hiérarchiques avec celles d'un réseau plat. La hiérarchie permet de réduire la quantité d'informations de routage à manipuler. En revanche, l'agrégation induit des erreurs dans les informations échangées et peut donc induire négativement sur les performances.

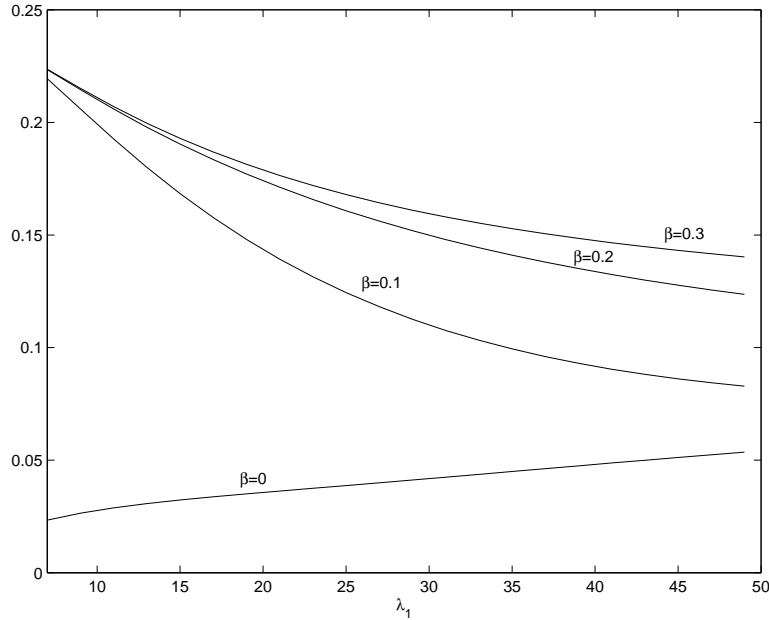


FIG. 7.14 – Influence de la structure hiérarchique sur le surcoût protocolaire.

Remarquons que pour $\beta = 0.1$ et $\lambda_1 = 15$, la probabilité de perte est approximativement égal à 0.45 pour $p = 0.9$, $q = 0.95$ et $\lambda_0 = 200$. Si nous avions à faire à un réseau plat, nous aurions

$$\mathcal{P}_b = (1 - PQ)^K$$

avec $P = p_0^M$ et $Q = q_0^M$. Dans le cas du réseau plat, le nombre de lien à traverser est approximativement $M = \frac{2}{\sqrt{\pi}}\sqrt{\lambda_0}$. Pour obtenir des performances comparables, il n'est pas dur de voir qu'il faudrait avoir $K > 9$. La table de routage contiendrait donc 200 nœuds en moyenne, 300 liens et il faudrait stocker 10 routes pour chaque destination... Soit un total de 2000 routes précalculées (comportant en moyenne 16 liens). Ces résultats sont très optimistes car ils sous-entendent qu'il est possible de trouver 10 chemins disjoints, ce qui n'est certainement pas très réaliste.

Pour des politiques plus conservatrices ($\beta = 0$), les performances sont comparables à celles d'un réseau plat avec $K = 3$ ou 4 chemins disjoints.

Ces calculs simples permettent donc de mettre en évidence l'importance de la hiérarchie sur les performances d'un réseau. Si la politique d'agrégation n'est pas trop conservatrice, la hiérarchie permet d'obtenir une meilleur utilisation du réseau malgré les pertes d'informations dues à l'agrégation. Ceci est dû à la nature hiérarchique du routage considéré. Les tailles des domaines de routage à manipuler étant significativement réduits, il est possible de calculer puis de stocker un grand nombre de routes vers chaque destination – et ce à chaque niveau hiérarchique. Il est donc possible de stocker K_1 routes au niveau logiques. Chacune de ces

routes passe par des domaines réduits que l'on peut traverser selon K_0 routes distinctes. Le nombre total de routes possibles est donc globalement $K_1 K_0^{M_1}$ — même s'il ne s'agit pas de routes disjointes, on comprend que l'algorithme de routage distribué permet de disposer d'un très grand nombre de routes de bout-en-bout tout en stockant une faible quantité d'informations. Ceci explique donc que, malgré les imprécisions et les pertes dues à l'agrégation, les résultats obtenus puisse être meilleurs que dans le cas du réseau plat.

7.4 Conclusion et Remarques

Dans ce chapitre, nous avons étudié les performances d'un réseau utilisant le protocole PNNI (ou tout autre protocole de routage dynamique hiérarchique et basé à la source) à deux niveaux hiérarchiques. L'accent a été mis sur l'étude de la politique d'agrégation et son influence sur l'utilisation du réseau et le surcoût protocolaire généré par les échanges de messages de signalisation.

Nous avons voulu représenter l'algorithme d'agrégation de manière générique et abstraite avec peu de paramètres. Le concept d'agrégation a ainsi été défini. Il apparaît en effet que cette notion a un rôle primordial sur les performances du réseau : Lorsque la politique d'agrégation est trop conservatrice, les connexions entrantes sont rejetées par le nœud source sans qu'aucune tentative d'établissement de connexion ne soit mise en œuvre. Au contraire, une politique trop "agressive" pourrait conduire à de nombreux crankbacks et donc à un surcoût protocolaire inacceptable.

Les différents critères de performances ont pu être exprimés en fonction de l'agressivité de la politique de routage et d'autres paramètres comme l'acuité des informations de routage au niveau physique (cf. chapitre 5) ou la structure hiérarchique. Nous nous sommes heurtés par contre à quelques difficultés lors de l'obtention de quantités moyennes par rapport à la nature aléatoire des réseaux considérés. Ces difficultés sont essentiellement dues au manque de résultats sur les propriétés des chemins (chemins markoviens, plus court chemins) sur les graphes de Delaunay. Nous avons néanmoins pu mettre en évidence les résultats énumérés ci-après grâce à des approximations.

Compromis Utilisation des ressources/Surcoût protocolaire.

Les résultats obtenus permettent de mettre en évidence un dilemme fort difficile à résoudre pour le dimensionnement d'un tel réseau hiérarchique.

- La fonction **GCAC**, associée à une politique d'agrégation des informations de routage conservatrice, permet de réduire considérable la probabilité de crankback et donc le

surcoût signalétique causé pendant la phase d'ouverture de connexions. Les taux de rejet à la source peuvent par contre être inacceptables.

- L'utilisation du réseau peut être très significativement améliorée grâce aux crankbacks et aux reroutages lorsque la politique d'agrégation est agressive. Le surcoût protocolaire peut par contre devenir très important dans ce cas.

Ce compromis apparaît d'ailleurs très nettement dans les propositions 7.4 et 7.3, puisque le nombre moyen de crankbacks subis par un appel est directement proportionnel à la probabilité de succès d'ouverture d'une connexion entrante :

$$\mathcal{N}_c \propto (1 - \mathcal{P}_b)$$

où \mathcal{N}_c est le nombre moyen de crankback et \mathcal{P}_b la probabilité de perte subie par une demande d'établissement de connexion.

Les équations proposées permettent d'obtenir des clés pour dimensionner le réseau : si le surcoût maximal tolérable pour une connexion est connu (en fonction de la capacité des connexions réservées à la signalisation et en fonction des capacités de traitement des commutateurs), il est ensuite possible d'obtenir l'utilisation du réseau associée pour une agressivité donnée — ou de choisir l'agressivité de la politique d'agrégation pour une utilisation optimale du réseau sous les contraintes de surcoût imposées, etc.

Agrégation et concept d'agressivité.

A la lumière de cette étude il apparaît que l'agressivité est un principe extrêmement difficile à manipuler. Tout d'abord, il faut bien comprendre que si nous avons quantifié cette notion (avec le paramètre β), il ne s'agit aucunement d'un paramètre du protocole PNNI. Il faut aussi noter que le concepteur d'algorithmes d'agrégation possède une latitude assez limitée sur cette notion, car des informations topologiques doivent également être prises en compte (ce que nous n'avons pas pu faire dans ce modèle). Cette notion est pourtant primordiale car elle influe de manière très significative sur les performances du réseau, comme nous l'avons souligné précédemment.

Idéalement, il serait très certainement souhaitable de pouvoir régler l'agressivité de l'algorithme d'agrégation pour se situer dans une zone de fonctionnement intermédiaire, entre la "zone de rejet", où les rejets à la source sont trop importants, et la "zone de crankbacks", où le surcoût est trop important. Il serait pourtant tout à fait utopique d'y penser car nous avons pu mettre en évidence que la notion même d'agressivité dépend de la charge du réseau et surtout du type de connexion entrante (bande passante demandée). Il est en effet possible qu'une politique d'agrégation soit agressive pour des connexions à bas débits mais très conservatrice pour des connexions friandes en bande passante...

Plus précisément, nous avons mis en évidence que l'on peut discerner assez nettement (de plus en plus nettement lorsque la taille des peer-groups augmente) deux phases distinctes : Une "zone de rejet" (c.à.d. politiques conservatrices) et une "zone de crankback" (c.à.d. politiques agressives). La transition entre ces deux zones a lieu autour de la valeur $\beta = 1 - p_0$ où $\beta \in [0, 1]$ mesure l'agressivité⁸ et où p_0 désigne la probabilité qu'un lien physique accepte la connexion⁹. La transition entre ces deux phases devient de plus en plus rapide lorsque le nombre de liens présents dans le réseau augmente, ce qui rend encore plus difficile le choix d'une agressivité "optimale".

Nous pouvons tout de même conclure qu'il est indispensable d'utiliser des politiques d'agrégation "agressive" (par rapport aux types d'appels que le réseau sera amené à supporter) pour obtenir une bonne utilisation des ressources. Nos résultats montrent qu'il est préférable de choisir des méthodes d'agrégation agressives, tout en limitant le sur-coût signalétique en imposant de sévères limitations sur le nombre de re-routages autorisés.

Fonction GCAC.

Les courbes présentées montrent que la fonction GCAC est extrêmement sensible à la politique d'agrégation et à la notion d'agressivité. Cette constatation constitue donc un très mauvais point pour cet algorithme car nous avons vu combien la notion d'agressivité est difficile à gérer. Il nous semble falloir conclure que la fonction GCAC se comporte assez mal dans le contexte de réseaux hiérarchiques :

- La fonction GCAC est très efficace pour limiter le nombre de crankbacks (et donc le sur-coût signalétique) lorsque la politique d'agrégation est conservatrice¹⁰. Par contre, cette fonction conduit à refuser un grand nombre d'appels sans même avoir essayé d'ouvrir la connexion. Les probabilités de rejet deviennent vite inacceptables lorsque la taille du peer-group est importante.
- La fonction GCAC se comporte mieux du point de vue du rejet à la source pour des politiques agressives. En revanche, le surcoût protocolaire augmente en contre-partie. Nous avons pu prouver que pour des politiques très agressives ($\beta \rightarrow 1$), les performances obtenues sont quasi-identiques que la fonction GCAC soit utilisée ou non ! Autrement dit, cette fonction devient inutile.

Il semble donc que la fonction GCAC, qui était apparue si efficace dans le contexte d'un réseau plat au chapitre 5, ne permet pas d'obtenir un compromis intéressant entre le contrôle du

⁸Rappelons nous que pour $\beta = 0$, la bande passante annoncée correspond au lien le plus congestionné, c.à.d. à une politique très conservatrice. La valeur $\beta \rightarrow 1$ conduit à des politiques très agressives puisque la valeur annoncée correspond au lien le moins congestionné.

⁹Les liens étant considérés i.i.d. .

¹⁰Pour un type de connexion donné.

surcoût signalétique et l'utilisation du réseau...

Il faut cependant nuancer notre propos, notre modèle pouvant être assez négatif vis à vis de cette fonction. Tout d'abord, nous avons basé notre analyse sur des peer-groups d'assez grandes tailles. Il est évident qu'un algorithme d'agrégation aussi basique que celui que nous avons choisi se comporte donc mal dans ce cas. Un algorithme de type "complex node" serait certainement plus approprié¹¹.

Notons également qu'une fois les routes admissibles sélectionnées, la demande d'ouverture de connexion est envoyée aléatoirement sur l'une de ces routes, indépendamment de leur état de congestion. Comme nous l'avons déjà remarqué au chapitre 5, il serait certainement plus judicieux de choisir le chemin possédant la plus grande bande passante disponible¹². Il serait alors peut-être possible d'être plus agressif pour un surcoût protocolaire donné.

Ces nuances étaient faites, nous pouvons toujours nous interroger sur l'utilité de la fonction GCAC : Une politique de routage différente (type "shortest-widest" comme évoqué précédemment par exemple) n'influerait de toute façon pas sur la probabilité de rejet lorsque la politique d'agrégation est conservatrice. Autrement dit, la fonction GCAC se comporterait tout aussi mal dans ce cas, conduisant encore à choisir des politiques agressives pour obtenir une bonne utilisation des ressources. Or nous avons vu que la fonction GCAC devient vite inutile pour des politiques agressives, les performances du routage "aveugle" étant comparables.

Cette constatation nous conduit à penser que la meilleure solution consiste peut-être à utiliser un algorithme de routage "shortest-widest" sans utiliser la fonction GCAC au niveau logique, le contrôle des crankbacks étant seulement assuré par des limites sévères sur K_c^0 et K_c^1 . On peut notamment penser à poser $K_c^1 = 1$: La probabilité que la connexion soit acceptée sur le second chemin (possédant donc moins de bande passante) étant plus faible. On peut également penser à un contrôle dynamique où un reroutage n'est tenté que si la bande passante disponible sur le second chemin n'est pas trop inférieure à celle du chemin le moins congestionné. Un environnement de simulation serait certainement préférable au modèle proposé dans ce chapitre pour étudier des algorithmes aussi complexes.

Structure hiérarchique

Finalement, nous avons pu donner quelques indications sur l'influence de la structure hiérarchique sur les performances globales du réseau. Un résultat assez peu intuitif a notamment

¹¹Bien que des extensions de ce modèle pour des algorithmes de ce type [1] aient donné des résultats similaires. Le concept de "complex node" (c.à.d. étoile asymétrique) nous permettait de diviser le domaine en plusieurs parties sur lesquelles l'agrégation (par β -quantile) était appliquée isolément. Cette extension permettait donc de réduire la taille des zones d'agrégation et conduisait donc à des transitions plus douces entre la zone de rejet et la zone de crankback...

¹²Cet algorithme de routage serait plus difficile à modéliser.

pu être mis en évidence : La hiérarchie n'est pas forcément négative du point de vue de l'utilisation du réseau et ce malgré les pertes de précisions des informations de routage dues à l'agrégation. En effet, les probabilités de pertes obtenues pour un réseau à deux niveaux hiérarchique peuvent être inférieures à celles d'un réseau similaire mais plat — au prix d'un plus grand nombre de crankbacks. Dans des conditions plus générales, les performances obtenues sont du même ordre de grandeur (malgré l'agrégation) — ce qui est tout à fait en accord avec des résultats de simulations publiés [43].

Ce phénomène est du au fait que nous considérons des routes précalculées et que la nature récursive du routage dans un réseau hiérarchique permet de mettre à la disposition des appels un très grand nombre de chemins différents, tout en stockant une quantité d'information limitée. Il faut bien sûr noter que les performances du réseau hiérarchiques ne peuvent être meilleures pour des politiques trop conservatrices — les rejets à la source étant trop probables dans ce cas.

Conclusion générale

Chapitre 8

Conclusion générale

8.1 Routage Multicast

Nous avons présenté dans le chapitre 3 un modèle générique permettant d'étudier les performances des arbres hiérarchiques centrés. Des structures hiérarchiques minimisant les ressources utilisées ont été mises en évidence. Les résultats obtenus permettent de donner des indications sur le dimensionnement de ces objets. Notre étude montre en particulier qu'un faible nombre de niveaux hiérarchiques suffit à réduire considérablement le "coût" de l'arbre. Nous avons également montré qu'il est préférable de choisir un nombre de niveaux hiérarchiques limité afin de conserver de bon délais de diffusion.

Les résultats mentionnés ont été obtenus pour le cas où les ressources multicast sont rares et où des "tuyaux" (encapsulation IP dans IP) sont utilisés. Des extensions sont en cours d'étude, en collaboration avec F.Baccelli, pour traiter le cas général où les ressources multicast sont déployées dans tout le réseau. L'arbre de concentration de Delaunay, présenté à la fin du chapitre 3, apparaît naturellement pour cette étude.

8.2 Routage QoS

Nos travaux ont porté sur le développement de modèles analytiques permettant d'étudier l'impact des différents paramètres mis en jeu dans les protocoles de routage QoS hiérarchiques.

Rafraîchissement des informations de routage.

Un modèle permettant d'étudier l'impact de l'acuité des informations de routage sur les performances globales du réseau est présenté dans le chapitre 5. Les résultats permettent de donner de premières indications sur les performances attendues en fonction d'une qualité des informations de routage donnée. Le modèle n'est pourtant pas achevé car il ne permet pas, sous sa forme actuelle, de trouver un compromis entre surcoût protocolaire (quantité d'informations échangées) et performances (utilisation des ressources). En effet, le paramètre q , qui mesure l'acuité des informations de routage, doit encore être évalué en fonction des paramètres du protocole de routage (comme la fréquence d'échange des informations par exemple). Des études préliminaires prometteuses nous encouragent à continuer dans ce te voie [8].

Structure hiérarchique.

L'impact de la structure hiérarchique a été étudié dans les chapitres 6 et 7. Dans le chapitre 6, des structures optimales permettant de minimiser les ressources utilisées (mémoire nécessaire ou complexité des calculs de routage) ont été déterminées. Les résultats obtenus montrent qu'un faible nombre de niveaux hiérarchiques permet de réduire considérablement la quantité d'information à manipuler par le protocole de routage, sans que les performances globales du réseau (utilisation des ressources) n'en soient trop affectées (cf. chapitre 7).

Agrégation.

Nous avons également pu appréhender l'impact de l'agressivité de l'algorithme d'agrégation des informations de routage sur les performances du réseau dans le chapitre 7. Il est apparu que les politiques d'agrégation "agressives" sont préférables, afin de réduire les rejets d'appels. Par contre, un contrôle strict du nombre de re-routages autorisés doit être effectué en contre partie pour éviter d'obtenir un trop grand surcoût signalétique.

Nous avons pu montrer que le concept d'agrégation est relatif, dépendant directement de la charge du réseau et de la bande passante requise par l'utilisateur : Un algorithme d'agrégation peut ainsi être "agressif" pour un type de d'appel et au contraire "conservateur" pour d'autres, selon la bande passante demandée. Cette constatation représente une difficulté majeure pour le développement d'algorithmes d'agrégation.

GCAC.

Dans le contexte de PNNI, nous avons également étudié le comportement de la fonction GCAC dans un réseau plat (chapitre 5) ou dans un réseau hiérarchique (chapitre 7). Les résultats obtenus montrent que cet algorithme permet d'obtenir une meilleure utilisation des res-

sources lorsqu'une limite est imposée sur le nombre maximal de re-routages d'une connexion, en sélectionnant les chemins les plus appropriés uniquement. Cette fonction se comporte donc remarquablement bien dans un réseau plat, en permettant de limiter le surcoût protocolaire tout en améliorant les performances globales du réseau.

En revanche, il semble que la fonction GCAC fonctionne assez mal en présence d'informations de routage agrégées trop grossières (avec un "simple node representation" par exemple dans le chapitre 7). Cette fonction est en effet très sensible à l'agressivité de l'algorithme d'agrégation (concept très difficile à maîtriser puisque relatif à la bande passante requise par l'utilisateur par exemple). Les résultats obtenus montrent que l'algorithme GCAC ne parvient pas à contrôler efficacement le surcoût signalétique :

- Pour des politiques d'agrégation conservatrices, le nombre de crankbacks est limité mais les taux de rejets sont inacceptables.
- Pour des politiques agressives, les informations de QoS échangées et la fonction GCAC deviennent inutiles pour de grands domaines de routage, un algorithme de routage "aveugle" fonctionnant tout aussi bien.

Des études sont prévues (voir conclusion du chapitre 7) pour trouver des méthodes alternatives, se comportant mieux en présence d'informations agrégées.

8.3 Géométrie stochastique

La géométrie aléatoire s'est révélée fort adaptée pour l'étude du routage. Cette technique permet de modéliser la topologie d'un réseau avec un nombre très réduit de paramètres. Des expressions explicites des grandeurs étudiées peuvent être déterminées, permettant d'analyser l'impact des différents paramètres mis en jeu.

Cette technique est pourtant encore très jeune¹ de sorte que quelques résultats importants pour nos travaux nous ont parfois manqués, notamment sur le routage dans les graphes de Delaunay. Les structures construites sont également très spécifiques : Il s'agit de graphes planaires fortement connectés. Il est important de comprendre ce qu'impliquent ces topologies spécifiques — des comparaisons par simulations avec d'autres types de graphes sont envisagées.

¹La géométrie stochastique existe depuis de nombreuses années. Mais son application aux réseaux est par contre très récente. Cette nouvelle branche d'applications pose de nouveaux problèmes.

Troisième partie



Annexes

Préambule

Les deux chapitres qui suivent présentent un résumé des principaux résultats sur les processus ponctuels dans le plan et sur les pavages aléatoires du plan. Le but de cette annexe est d'introduire les théorèmes et propriétés, essentiellement sur le calcul de Palm, utilisés dans les chapitres précédents. L'exposé suivant a également fait l'objet d'une publication interne à l'ENST (notes de lecture), après avoir été agrémenté des commentaires et suggestions de A.Ragozini. Qu'il en soit remercié. Nous avons souhaités présenter ces objets aléatoires et leurs propriétés de manière simple, en négligeant certaines considérations quelque peu techniques, tout en restant rigoureux sur le plan mathématique — projet quelque peu difficile à réaliser.

Ce travail à été, au départ, écrit uniquement dans le but de mieux appréhender ces concepts — *Ce qui se conçoit bien, s'énonce aisément*. L'intérêt était aussi de réunir en un même document toutes les propriétés utiles à nos travaux, avec une cohérence dans l'approche et dans les notations. Nous avons en effet constatés que les résultats sur ces objets étaient disséminés dans de nombreux ouvrages, les uns écrits pour des ingénieurs (sans démonstration et pour le moins parcélaire), les autres écrits pour des mathématiciens, les notations et les approches utilisées étant à chaque fois forts différentes. Cette annexe est une synthèse, non exhaustive, des ouvrages suivants : [100, 91, 86, 90, 99, 95, 98, 97, 89].

Une fois la rédaction de ce document achevée, nous avons appris la publication de deux articles de synthèse [93, 94] sur la géométrie aléatoire et leur application aux réseaux de communication, comblant enfin en partie le vide évoqué plus haut. Nous en recommandons vivement la lecture.

Enfin, le dernier chapitre est consacré à diverses rappels de combinatoire utiles à l'étude du routage hiérarchique (chapitre 7).

Annexe A

Processus Ponctuels dans le Plan

A.1 Notations

Dans toute la suite, on s'intéresse aux processus ponctuels, que l'on va définir dans la section suivante, dans le plan. Nous noterons $E = \mathbb{R}^2$ l'espace d'état considéré et \mathcal{B} la σ -algèbre des boréliens de E . L'espace E , muni de sa σ algèbre \mathcal{B} , est appelé un "espace mesurable" et sera noté (E, \mathcal{B}) . De manière générale, le couple (B, \mathcal{F}_B) est une notation commode, qui sert à désigner tout espace espace mesurable B , muni de sa σ -algèbre \mathcal{F}_B .

Pour tout $B \in \mathcal{B}$, \overline{B} dénote l'adhérence de B (le plus petit fermé¹ contenant B) et B^c le complémentaire (ou l'extérieur) de B par rapport à E .

Rappelons qu'une fonction de (E, \mathcal{B}) dans $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ est dite **mesurable** si $f^{-1}(A) \in \mathcal{B} \forall A \in \mathcal{B}(\mathbb{R})$, où $\mathcal{B}(\mathbb{R})$ est la σ -algèbre des boréliens sur la droite réelle \mathbb{R} . On peut aussi définir la mesurabilité de manière équivalente par $\{x : f(x) \leq c\} \in \mathcal{B}, \forall c \in \mathbb{R}$. En effet, les ensemble $[-\infty, c]$ engendrent $\mathcal{B}(\mathbb{R})$. Pour tout fonction mesurable de E dans \mathbb{R} et pour tout mesure μ sur E , on notera

$$\mu(f) \doteq \int_E f(x) d\mu(x)$$

l'intégrale de f par rapport à μ . Lorsque aucune ambiguïté n'est possible sur la variable et l'espace d'intégration, on pourra parfois utiliser

$$\int f d\mu$$

¹Par rapport à l'inclusion (\subset).

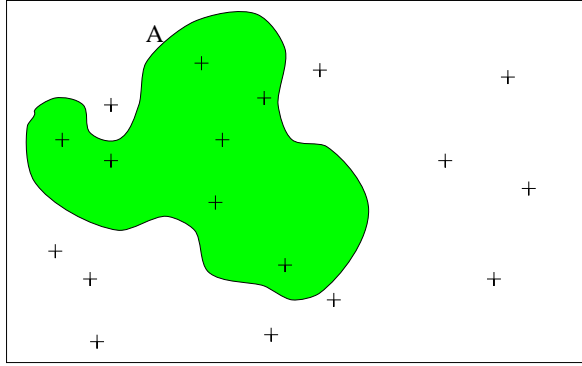


FIG. A.1 – Interprétation d'une mesure ponctuelle.

A.2 Définition des Processus Ponctuels

A.2.1 Mesures ponctuelles

Définition A.1 Une mesure ponctuelle μ est une mesure à valeurs entières, simple et localement finie, c'est à dire vérifiant les conditions suivantes :

- (i) *Finitude locale* : $\mu(A) < \infty$ si A est borné.
- (ii) *Simplicité* : $\forall x \in E, \mu(\{x\}) \in \{0, 1\}$.

Ces mesures peuvent être vues comme la somme dénombrable de mesures de Dirac (comme démontré dans [91, chapitre 7]) :

$$\mu = \sum_i \delta_{x_i} \quad (\text{A.1})$$

avec des $x_i \in E$ deux à deux disjoints (d'après l'hypothèse de simplicité). L'ensemble des points $\{x_i\}_{i \in \mathbb{N}}$ est appelé le **support** de μ et noté $\text{supp}(\mu)$.

Cette propriété nous permet d'interpréter géométriquement cette mesure. Si $A \in \mathcal{B}$, on peut écrire :

$$\mu(A) = \sum_i \mathbf{1}_{\{x_i \in A\}} \quad (\text{A.2})$$

Ainsi, $\mu(A)$ correspond aux nombres de points x_i contenus dans l'ensemble A (Dans le cas de la figure A.1, $\mu(A) = 7$).

Pour toute fonction mesurable $f : (E, \mathcal{B}) \rightarrow (\mathbb{R}^+, \mathcal{B}(\mathbb{R}^+))$, on peut définir l'intégrale de f par rapport à la mesure ponctuelle μ . Cette intégrale peut s'écrire d'une manière fort simple d'après la formule (A.2) :

$$\int f d\mu = \sum_i f(x_i) \quad (\text{A.3})$$

où les $\{x_i\}$ constituent le support de μ .

Preuve. D'après la formule (A.2) la propriété est vérifiée pour toute fonction indicatrice $\mathbf{1}_A$ avec $A \in \mathcal{B}$. Cette propriété se généralise donc à toute fonction mesurable f de E dans \mathbb{R}^+ par des arguments classiques de classes monotones. \square

A.2.2 Processus ponctuels.

De manière informelle, un processus ponctuel est une mesure ponctuelle aléatoire. Autrement dit, il s'agit d'une mesure ponctuelle dont le support (les points $(x_i)_{i \in N}$) sont tirés aléatoirement.

Introduisons un espace abstrait de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$ et notons $M_p(E)$ l'ensemble des mesures ponctuelles de E définies dans la section précédente.

Définition A.2 *Un processus ponctuel N est une application*

$$N : \Omega \rightarrow M_p(E)$$

telle que $N(\cdot, F) = N(\cdot)(F)$ soit une variable aléatoire pour tout borélien F de \mathcal{B} .

Les variables aléatoires $\omega \mapsto N(\omega, F)$ sont appelées les **variables de comptage** du processus ponctuel (voir figure A.1). De la même manière, on peut définir les variables aléatoires $w \mapsto \int f dN(w)$ pour toute fonction mesurable f de E dans \mathbb{R}^+ . D'après la formule A.3 sur les mesures ponctuelles, cette intégrale est en fait une somme dénombrable, soit

$$\int f dN = \sum_{x \in \text{supp}(N)} f(x)$$

L'espérance de cette variable de comptage

$$\Lambda(F) = \mathbb{E}[N(F)] \tag{A.4}$$

définit, pour tout $F \in \mathcal{B}$, une nouvelle mesure aléatoire sur E . Ceci est dû au fait que les propriétés de σ -additivité de N en F se transmettent à la variable Λ , qui est appelée la **mesure d'intensité** du processus N .

L'équation (A.4) peut encore s'écrire $\mathbb{E}[\int \mathbf{1}_F dN] = \int \mathbf{1}_F d\Lambda$, qui peut être étendu à l'équation suivante par des arguments de classes monotones :

$$\mathbb{E}\left[\int f dN\right] = \int f d\Lambda \tag{A.5}$$

pour toute fonction f mesurable de E dans \mathbb{R}^+ .

L'intensité est souvent absolument continue par rapport à la mesure de Lebesgue \mathcal{L} , et admet donc une dérivée de Radon-Nykodym ψ , appelée **densité**.

$$\mathbb{E} \left[\int f(x) dN(x) \right] = \int_E f(x) \psi(x) d\mathcal{L}(x)$$

Si le processus N est **stationnaire**, c'est à dire si $N(A+x) = N(A) \forall x \in E, A \in \mathcal{B}$ où $A+x = \{y+x : y \in A\}$, alors Λ est proportionnelle à la mesure de Lebesgue \mathcal{L} sur E . Dans ce cas, il existe $\lambda \in \mathbb{R}^+$, telle que $\Lambda = \lambda\mathcal{L}$. λ est appelé l'**intensité** du processus ponctuel N .

A.2.3 Processus stationnaires

L'opération d'addition dans l'espace E permet d'introduire la notion de flot et nous permet ainsi d'introduire la notion de stationnarité.

Définition A.3 Un flot $(\theta_x)_{x \in E}$ de l'espace mesurable (M, \mathcal{M}) est une transformation de M vérifiant les trois axiomes suivants :

- (i) $\theta_x \circ \theta_y = \theta_{x+y} \forall (x, y) \in E^2$
- (ii) l'application $(z, x) \mapsto \theta_x(z)$ est mesurable par rapport à $\mathcal{M} \otimes \mathcal{B}$ et \mathcal{M} .
- (iii) θ_x est bijective pour tout $x \in E$.

Le flot le plus utilisé est certainement le "shift", c'est à dire l'opérateur de translation sur l'espace étudié E . Plus précisément, le shift est défini comme suit : $\forall A \in \mathcal{B}$ et $\forall x \in E$,

$$\theta_x A = A + x = \{y + x : y \in A\}$$

A partir de cet opérateur, on peut construire un autre flot ϕ sur $M_p(E)$ en posant $\forall N \in M_p(E), \forall A \in \mathcal{B}$

$$(\phi_x N)(A) = N(\theta_x A)$$

Remarquons que d'après (A.1), $(\phi_x N)$ peut aussi s'écrire sous la forme $(\phi_x N) = \sum_i \delta_{x_i - x}$, où $\{x_i\}_{i \in \mathbb{N}}$ désigne le support de N , de sorte que

$$\text{supp}(\phi_x N) = \theta_{-x} \text{supp}(N)$$

Étant donné un processus ponctuel N , il est possible de définir un flot ϕ' sur l'espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$, en posant : $\forall \omega \in \Omega$,

$$N(\phi'_x \omega) = \phi_x N(\omega)$$

c'est à dire $\forall A \in \mathcal{B}, N(\phi'_x \omega, A) = N(\omega, \theta_x A)$. Dans toute la suite, nous ne distinguerons plus les flux ϕ' et ϕ et les noterons donc tous deux ϕ . Il ne s'agit pas vraiment d'un abus de notation dans la mesure où l'espace canonique de Probabilité (c.à.d. $\Omega = M_p(E)$) sera utilisé dans la suite de ce chapitre.

Proposition A.1 *Si le flot ϕ préserve la probabilité, c'est à dire si $\forall I \in \mathcal{F} \forall x \in \mathbb{R} \mathbb{P}(\phi_x I) = \mathbb{P}(I)$ alors les distributions de N et de son translaté $\phi_x N$ sont identiques.*

Preuve. $\forall A_0, \dots, A_n \in \mathcal{B}, k_0 \dots k_n \in \mathbb{N}$ soit l'événement $I = \{N(A_0) = k_0, \dots, N(A_n) = k_n\}$. D'après la définition de ϕ , $\phi_x^{-1} I = \{\phi_x N(A_0) = k_0, \dots, \phi_x N(A_n) = k_n\}$, d'où

$$\begin{aligned} \mathbb{P}\{\phi_x N(A_0) = k_0, \dots, \phi_x N(A_n) = k_n\} &= \\ \mathbb{P}(\phi_x^{-1} I) &= \\ \mathbb{P}(I) &= \mathbb{P}\{N(A_0) = k_0, \dots, N(A_n) = k_n\} \end{aligned}$$

□

Cette remarque justifie la définition suivante.

Définition A.4 *Soit N est un processus ponctuel sur E et ϕ un flot sur $(\Omega, \mathcal{F}, \mathbb{P})$, vérifiant les propriétés suivantes :*

- (i) *Compatibilité avec le Shift : $\forall \omega \in \Omega, N(\phi_x \omega) = \theta_x N(\omega)$*
- (ii) *Préservation de la probabilité : $\mathbb{P} \circ \phi_x = \mathbb{P}$*

(N, ϕ) est appelé processus ponctuel stationnaire.

A.3 Processus de Poisson

A.3.1 Définition

Définition A.5 *Un processus ponctuel N sur E est appelé processus de Poisson de mesure d'intensité μ (pour une mesure positive donnée μ sur E , localement finie), si*

- (i) *$N(F_1), N(F_2), \dots, N(F_k)$ sont indépendants, pour tous $F_1, \dots, F_k \in \mathcal{B}$ deux à deux disjoints.*
- (ii) *$N(F)$ suit une loi de Poisson d'espérance $\mu(F)$, pour tout $F \in \mathcal{B}$.*

La proposition suivante permet de définir un moyen de simuler de tels processus sur des espaces bornés.

Proposition A.2 *Soit θ un réel positif et μ une mesure de probabilité sur E , finie sur les bornés. Soient X_1, X_2, \dots est une suite infinie de variables aléatoires à valeurs dans E , de même loi μ , et soit ν une v.a. de Poisson de paramètre θ indépendante des $(X_i)_{i \in \mathbb{N}}$. Alors le processus ponctuel N défini par*

$$N = \sum_{n=1}^{\nu} \delta_{X_n} \quad (= 0 \text{ si } \nu = 0)$$

est un processus de Poisson de mesure d'intensité $\theta\mu$.

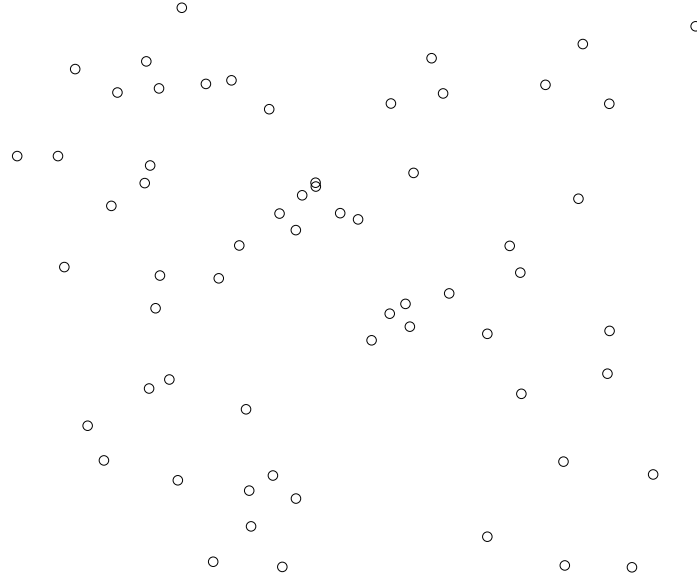


FIG. A.2 – Une réalisation d'un processus de Poisson homogène dans un carré d'aire unitaire ($\lambda = 50$).

Preuve. D'après [95]. Cette proposition peut être démontrée aisément en utilisant les fonctions génératrices, et en vérifiant donc que

$$\mathbb{E} \left[\prod_{j=1}^k u_j^{N(F_j)} \right] = \exp \left[\sum_{j=1}^k (u_j - 1) \theta \mu(F_j) \right]$$

Soit $v : E \rightarrow [0, 1]$ la fonction étagée égale à u_j sur les F_j et 1 partout ailleurs. On identifie alors

$$\prod_{j=1}^k u_j^{N(F_j)} = \prod_{n=1}^{\nu} v(X_n)$$

Les propriétés de X et ν entraînent de plus que

$$\begin{aligned} \mathbb{E} \left[\prod_{n=1}^{\nu} v(X_n) \right] &= \sum_{n=0}^{\infty} \mathbb{P}\{v = n\} (\mathbb{E}[v(X_1)])^n \\ &= \exp \left[\theta \left(\int_E v(x) \mu(dx) - 1 \right) \right] \\ &= \exp \left[\sum_{j=1}^k (u_j - 1) \theta \mu(F_j) \right] \end{aligned}$$

□

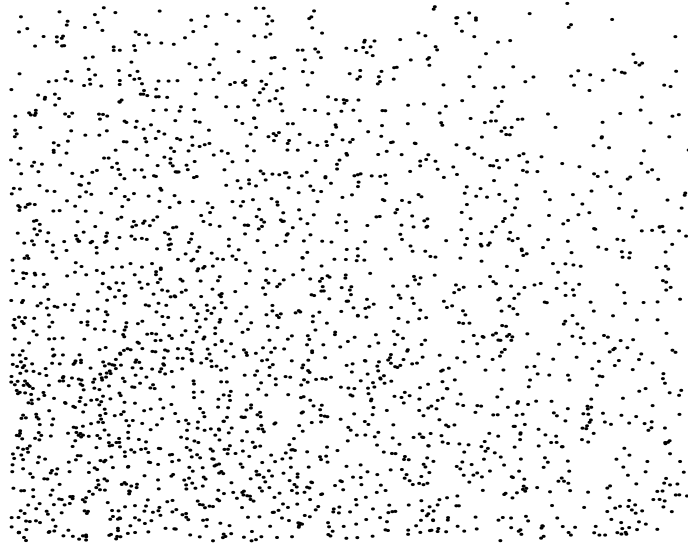


FIG. A.3 – Une réalisation d’un processus de Poisson de densité $\lambda(x, y) = \lambda^* \cdot \left(1 - \frac{\sqrt{x^2 + y^2}}{\sqrt{2}}\right)$ sur $[0, 1]^2$ ($\lambda^* = 5000$).

A.3.2 Simulation

Pour un processus de Poisson homogène, la mesure d’intensité μ est proportionnelle à la mesure de Lebesgue $\mu = \lambda \mathcal{L}$ et λ est appelée l’intensité de ce processus. Dans ce cas, il n’est pas difficile de construire une représentation d’un tel processus. Les variables aléatoires $(X_i)_{i \in \mathbb{N}}$ du lemme A.2 sont des variables aléatoires uniformes sur E . Autrement dit, sur une surface A de E (par exemple, le carré unitaire $[0, 1] \times [0, 1]$ de \mathbb{R}^2), il suffit de tirer ν le nombre de points, selon une loi de poisson de paramètre $\lambda \mathcal{L}(A)$ puis de tirer ν points uniformément sur A . C’est ce qui a été fait pour obtenir la figure A.2.

Pour des processus de Poisson généraux, on peut utiliser un procédé d’amincissement (ou *thinning* en anglais). Soit μ la mesure d’intensité du processus de Poisson que l’on désire simuler, et λ sa densité. Si $\exists \lambda^*$ tel que $\lambda(x) < \lambda^* \forall x \in E$, alors on peut tirer un processus de Poisson stationnaire d’intensité λ^* , puis éliminer certains points (indépendamment les uns des autres) avec la probabilité $p(x) = \frac{\lambda(x)}{\lambda^*}$. Il est facile de voir que le processus obtenu est bien un processus de Poisson, de mesure d’intensité μ donnée par :

$$\mu(A) = \int_A p(x) \lambda^* dx = \int_A \lambda(x) dx$$

La figure A.3 a été construite de cette manière.

A.3.3 Transformée de Laplace

Définition A.6 La transformée de Laplace d'un processus ponctuel N de E est la fonctionnelle

$$L_N[f] = \mathbb{E} \left[e^{-N(f)} \right]$$

définie sur l'ensemble des applications boréliennes f de E dans \mathbb{R}^+ .

Théorème A.1 Si N est un processus de Poisson de mesure d'intensité Λ , sa transformée de Laplace prend la forme suivante :

$$L_N[f] = \exp \left\{ - \int_E [1 - \exp(-f(x))] \Lambda(dx) \right\} \quad (\text{A.6})$$

Preuve. D'après [95]. Cette propriété se démontre facilement pour les fonctions étagées du type $f = \sum_{i=1}^k \alpha_i \mathbf{1}_{F_i} \forall \alpha_i \in \mathbb{R}^+$ et où les $(F_i)_{i=1, \dots, k}$ sont des pavés bornés de E deux à deux disjoints. Dans ce cas en effet,

$$\exp[-N(f)] = \prod_{i=1}^k [\exp(-\alpha_i)]^{N(F_i)}$$

de sorte que

$$\begin{aligned} L_N[f] &= \prod_{i=1}^k \exp[-(1 - \exp(-\alpha_i)) \Lambda(F_i)] \\ &= \exp \left\{ - \int_E [1 - \exp(-f(x))] \Lambda(dx) \right\} \end{aligned}$$

Par des arguments de continuité, il est ainsi possible de généraliser cette propriété à toute fonction borélienne positive f . \square

Remarquons que d'après la formule (A.3) sur les mesures ponctuelles, la transformée de Laplace $L_N(f)$ peut aussi s'écrire

$$L_N(f) = \mathbb{E} \left[\exp \left(- \sum_{x_i \in \text{supp}(N)} f(x_i) \right) \right] = \mathbb{E} \left[\prod_i \exp(-f(x_i)) \right]$$

Posant de plus $f = \log\left(\frac{1}{g}\right)$, la proposition (A.6) prend la forme suivante, très utile pour les applications.

Théorème A.2 Pour toute fonction borélienne positive g de E dans \mathbb{R}^+ , et pour tout processus de Poisson N sur E de mesure d'intensité Λ ,

$$\mathbb{E} \left[\prod_{x_i \in \text{supp}(N)} g(x_i) \right] = \exp \left\{ - \int_E [1 - g(x)] \Lambda(dx) \right\}$$

A.3.4 Perturbation

Nous terminons cette sections consacrée aux processus de Poisson par une propriété particulière dite de perturbation. Pour une étude plus étendue sur le sujet, le lecteur peut se référer à [92].

Perturbation d'un processus de Poisson

Soit N un processus de Poisson stationnaire à valeur dans un ensemble I de mesure finie (par rapport à la mesure de Lebesgue \mathcal{L}). Soit λ son intensité, et soit F une fonctionnelle quelconque définie sur $M_p(I)$, l'ensemble des mesures ponctuelles sur I . Le théorème suivant est un résultat bien connu de l'analyse des perturbations, permettant d'exprimer la dérivée de $\mathbb{E}[F(N)]$ par rapport à λ , l'intensité du processus N .

Théorème A.3 *Avec les notations précédentes, l'égalité suivante est vérifiée :*

$$\frac{\partial}{\partial \lambda} \mathbb{E}[F(N)] = \mathcal{L}(I) \mathbb{E}[F(N + \delta_U) - F(N)] \quad (\text{A.7})$$

où U est une variable aléatoire uniformément distribué sur I et indépendant de N .

Preuve. Il existe de nombreuses preuves de cette propriété. Nous utilisons l'approche simple imaginée par F.Baccelli et P.Brémaud, dont nous avons eu connaissance dans par L.Decreusefond [92].

Par définition, la dérivée de $\mathbb{E}[F(N)]$ en λ_0 est donnée par la limite :

$$\left. \frac{\partial}{\partial \lambda} \mathbb{E}[F(N)] \right|_{\lambda=\lambda_0} = \lim_{\lambda \rightarrow \lambda_0} \frac{1}{\lambda - \lambda_0} \mathbb{E}[F(N_\lambda) - F(N_{\lambda_0})]$$

où N_λ et N_{λ_0} sont deux processus de Poisson sur I aux intensités respectives de λ et λ_0 . Si la limite existe, on peut se contenter de calculer :

$$\frac{1}{\lambda_0} \lim_{\epsilon \rightarrow 1^+} \frac{1}{\epsilon - 1} \mathbb{E}[F(N_{\lambda_0 \epsilon}) - F(N_{\lambda_0})]$$

où le processus $N_{\lambda_0 \epsilon}$ est obtenu par amincissement à partir de N_{λ_0} avec la probabilité $1-\epsilon$. Il est bien connu qu'un tel processus est également un processus de Poisson, et que son intensité est $\lambda_0 \epsilon$. Introduisons un processus auxiliaire $\Delta = N_{\lambda_0 \epsilon} - N_{\lambda_0}$. On peut remarquer qu'il s'agit d'un processus ponctuel de Poisson, dont l'intensité est $(\lambda_0 \epsilon - 1)$ et dont il n'est pas dur de montrer qu'il est indépendant de N_{λ_0} . En conditionnant par rapport à Δ , $\mathbb{E}[F(N_{\lambda_0 \epsilon}) - F(N_{\lambda_0})]$ peut alors être exprimé de cette manière :

$$\sum_{j=0}^{\infty} \mathbb{E}[F(N_{\lambda_0} + \Delta) - F(N_{\lambda_0}) | \Delta(I) = j] \mathbb{P}(\Delta(I) = j)$$

On observe que, lorsque $\mathbb{E}[F(N_{\lambda_0\epsilon}) - F(N_{\lambda_0})|\Delta(I) = 0]$ vaut zéro, $\mathbb{P}(\Delta(I) = j)$ vaut $o(\epsilon - 1)$ pour $\epsilon \rightarrow 1^+$ avec $\forall j > 1$. En passant à la limite $\epsilon \rightarrow 1$, seul le terme $j = 1$ est non nul dans l'expression précédente. Étant donné $\Delta(I) = 1$, il est alors facile de montrer que, $\forall \epsilon$, $\text{supp}(\Delta)$ est un point uniformément distribué sur I , indépendant de N_{λ_0} . Dénotant U ce point, on a finalement :

$$\mathbb{E}[F(N_{\lambda_0} + \Delta) - F(N_{\lambda_0})/\Delta(I) = 1] = \mathbb{E}[F(N_{\lambda_0} + \delta_U) - F(N_{\lambda_0})]$$

De plus, comme $\mathbb{P}(\Delta(I) = 1) = \lambda_0(\epsilon - 1)\mathcal{L}(I)e^{-(\lambda_0(\epsilon-1)\mathcal{L}(I))}$, on conclue que

$$\frac{1}{\lambda_0} \lim_{\epsilon \rightarrow 1^+} \frac{1}{\epsilon - 1} \mathbb{P}(\Delta(I) = 1) = \mathcal{L}(I)$$

Les deux expressions précédentes permettent alors d'arriver à l'identité annoncée :

$$\left. \frac{\partial}{\partial \lambda} \mathbb{E}[F(N)] \right|_{\lambda=\lambda_0} = \mathcal{L}(I) \mathbb{E}[F(N_{\lambda_0} + U) - F(N_{\lambda_0})]$$

□

Remarque A.1 Autrement dit, la dérivée de certaines grandeurs moyennes par rapport à l'intensité du processus peut être analysée en observant l'effet de l'ajout d'un point supplémentaire U — la “perturbation” — sur cette grandeur. D'où cette dénomination.

A.4 Calcul de Palm

Dans toute la suite, nous utilisons l'espace de Probabilité canonique, c'est à dire $\Omega \equiv M_p(E)$ (et pour leur σ -algèbre, $\mathcal{F} = \mathcal{B}(M_p(E))$). Un processus ponctuel est ainsi caractérisé par une distribution \mathcal{P} sur l'espace des mesures ponctuelles sur E . En particulier, les moyennes des processus sont ainsi définies de la sorte :

$$\begin{aligned}\mathbb{E}_{\mathcal{P}}[N(F)] &= \int_{M_p(E)} \xi(F) \mathcal{P}(d\xi) \\ \mathbb{E}_{\mathcal{P}}[f(N)] &= \int_{M_p(E)} f(\xi) \mathcal{P}(d\xi)\end{aligned}$$

pour tout borélien F de E et pour toute fonction mesurable f de $M_p(E)$ vers \mathbb{R}^+ .

Nous ne considérons que le cas des processus stationnaires, sauf mention explicite.

A.4.1 Formule de Palm

La probabilité de Palm peut être définie par le théorème suivant. D'autres définitions, bien plus théoriques mais plus générales ² sont données dans [91, 95] par exemple.

Définition A.7 (Formule de Palm) *Pour tout processus stationnaire (N, ϕ) de E , d'intensité finie λ , soit \mathcal{P}_0 la probabilité de Palm associée. Les relations suivantes sont vérifiées pour toute fonction h mesurable de $M_p(E)$ dans \mathbb{R}^+ et pour tout événement U de $\mathcal{F} = \mathcal{B}(M_p(E))$, et sont indépendantes du choix de $B \in E$*

$$\mathbb{E}_{\mathcal{P}_0}[h(N)] = \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\int_B h(\phi_x N) N(dx) \right]$$

$$\mathcal{P}_0(U) = \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\int_B \mathbf{1}_U(\phi_x N) N(dx) \right]$$

Interprétation locale

Les deux propositions suivantes permettent de donner une interprétation à la probabilité de Palm (en un mot, comme la probabilité "sachant que N a un point situé à l'origine", d'où la notation).

Proposition A.3 *Sous Palm, le processus ponctuel dispose presque sûrement d'un point à l'origine. Autrement dit, si $\Omega_0 = \{\xi \in M_p(E) : 0 \in \text{supp}(\xi)\}$, alors $\mathcal{P}_0(\Omega_0) = 1$.*

²Permettant la définition de probabilités de Palm pour des processus non stationnaires.

Preuve. Par construction, $\forall B \in \mathcal{B}$,

$$\begin{aligned} \mathcal{P}_0(\Omega_0) &= \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\int_B \mathbf{1}_U(\phi_x N) N(dx) \right] \\ &= \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\int_B \mathbf{1}_{\{\theta_x N(0) > 0\}} N(dx) \right] \\ &= \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\sum_{x_i \in \text{supp}(N)} \mathbf{1}_{\{N(x_i) > 0\}} \right] = 1. \end{aligned}$$

□

La proposition suivante donne une interprétation locale de la mesure de Palm, et est donnée sans démonstration (voir [91, page 481]).

Proposition A.4 *Soit la suite décroissante $(A_n)_{n=1,2,\dots}$ de sphères de E centrées à l'origine telle que $\mathcal{L}(A_n) \rightarrow 0$ lorsque $n \rightarrow \infty$ alors pour toute fonction borélienne positive, continue et bornée f de $M_p(E)$ dans \mathbb{R}^+ ,*

$$\mathbb{E}_{\mathcal{P}}[f(N) | N(A_n) > 0] \rightarrow \mathbb{E}_{\mathcal{P}_0}[f(N_0)] \quad (\text{A.8})$$

où N_0 est défini par $\text{supp}(N_0) = \text{supp}(N) + \{0\}$

Notion de Caractéristique “typique”

L'interprétation précédente découle en fait directement de la construction de la probabilité de Palm. En revanche, cette façon de voir les choses apporte peu quant à l'appréhension de cet outil, et n'a que de rares applications pratiques. Nous essayons dans cette section de donner une interprétation plus intuitive (mais peu être moins rigoureuse...) de la probabilité de Palm.

Comme on va le voir dans cette section, on s'intéresse grâce à cet outil à déterminer la probabilité d'occurrence de certains événements, au niveau des points de $\text{supp}(N)$ uniquement (et non au niveau de tous les points de l'espace). Le processus ponctuel considéré étant stationnaire, il n'est pas dur de comprendre que ces probabilités ne dépendent pas du point considéré — mais du fait qu'on est situé au niveau d'un point de $\text{supp}(N)$. Cette idée est explicitée ci dessous.

Définition A.8 *Nous appellerons fonction compatible d'un processus ponctuel stationnaire (N, ϕ) , toute fonction f mesurable de $M_p(E) \times E$ vers \mathbb{R}^+ qui est compatible avec le shift, c'est à dire vérifiant*

$$f(\xi, x) = f(\phi_x \xi, 0)$$

pour tout $\xi \in M_p(E)$ et $x \in E$.

De nombreux exemples de fonctions compatibles sont utilisées dans l'étude des pavages du plan. On peut prendre aussi par exemple $f(x, \xi) = \mathcal{L}(C_x(\xi))$ où $C_x(\xi)$ est la cellule engendrée par $\text{supp}(\xi)$ et contenant x , définie un peu plus loin (définition A.9 page 196).

Proposition A.5 *Pour toute fonction f compatible avec le processus stationnaire (N, ϕ) d'intensité λ , on peut écrire, par une application directe de la définition de la probabilité de Palm :*

$$\begin{aligned} \mathbb{E}_{\mathcal{P}_0}[f(N, 0)] &= \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\int_B h(\phi_x N, 0) N(dx) \right] \\ &= \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E}_{\mathcal{P}} \left[\sum_{x \in \text{supp}(N) \cap B} h(N, x) N(dx) \right] \end{aligned}$$

Corollaire A.1 *Si de plus, $(\Omega, \phi, \mathcal{P})$ est ergodique³, alors on peut écrire :*

$$\mathbb{E}_{\mathcal{P}_0}[f(N, 0)] = \lim_{\mathcal{L}(A_n) \rightarrow \infty} \frac{1}{\lambda \mathcal{L}(A_n)} \sum_{x \in \text{supp}(N) \cap A_n} h(\xi, x)$$

où ξ est une réalisation quelconque de N ($\exists \omega : \xi = N(\omega)$).

La moyenne sous Palm correspond dans ce cas à une moyenne empirique sur les points de $\text{supp}(N)$, par opposition à la moyenne stationnaire qui fait intervenir tous les points de E :

$$\mathbb{E}_{\mathcal{P}}[f(N, x)] = \lim_{\mathcal{L}(A_n) \rightarrow \infty} \frac{1}{\mathcal{L}(A_n)} \int_{A_n} f(\xi, x) dx$$

où ξ est une réalisation de N et pour une espace de probabilité ergodique $(\Omega, \phi, \mathcal{P})$.

Remarque A.2 Lorsqu'on calcule des expressions du type $\mathbb{E}_{\mathcal{P}_0}[f(N, 0)]$, on parle de caractéristiques "typique" de N . Cette attribution désigne bien le fait que le choix de l'origine n'induit aucun "biais" mais désigne bien la probabilité d'occurrence de l'événement au niveau de tous les points du processus stationnaire N . Le corollaire 127 illustre bien cette remarque.

La définition A.7, grâce à quelques manipulations techniques sur le "shift", permet de tout ramener au niveau de l'origine. L'intérêt est que, sous Palm, nous sommes sûr qu'il existe un point à l'origine. Il suffit alors de s'intéresser, sous la probabilité de Palm, aux caractéristiques étudiées au niveau de l'origine. Les caractéristiques étudiées par ce procédé seront valables au niveau de tous les points de $\text{supp}(N)$ — et non pas spécifiques à l'origine.

Les caractéristiques "typiques" et la probabilité de Palm sont très largement utilisés dans le chapitre suivant qui traite des pavages du plan et offre un exemple particulièrement élégant de ces notions.

³Voir [91] par exemple pour une étude de l'ergodicité dans des espaces de type \mathbb{R}^d .

Le cas du processus de Poisson

La probabilité de Palm d'un processus de Poisson prend une forme particulière, comme le montre le théorème de Slivnyak.

Théorème A.4 (Slivnyak) *Pour tout processus de Poisson N , on peut écrire :*

$$\mathbb{E}_{\mathcal{P}_0}[f(N)] = \mathbb{E}_{\mathcal{P}}[f(N_0)]$$

où N_0 est défini par $\text{supp}(N_0) = \text{supp}(N) + \{0\}$.

Cette identité est très intéressante pour l'évaluation de probabilité de Palm à partir de simulation : Il suffit d'y ajouter un point à l'origine. Cette technique est traitée plus exhaustivement dans [93].

A.4.2 Formules élémentaires du Calcul de Palm

Formule de Campbell

On obtient la formule de Campbell pour les processus stationnaires (plus exactement *refined Campbell formula* dans la littérature anglophone) :

Théorème A.5 (Formule de Campbell) *Pour tout processus stationnaire (N, ϕ) sur E d'intensité λ , soit \mathcal{P} la probabilité sur $(M_p(E), \mathcal{B}(M_p(E)))$ associée à ce processus et soit \mathcal{P}_0 sa mesure de Palm. Alors pour toute fonction mesurable g de $E \times M_p(E) \rightarrow \mathbb{R}^+$,*

$$\mathbb{E}_{\mathcal{P}} \left[\int_E g(x, \phi_x \xi) \xi(dx) \right] = \lambda \int_E \mathbb{E}_{\mathcal{P}_0}[g(x, \xi)] \mathcal{L}(dx) \quad (\text{A.9})$$

Preuve. Cette formule est vérifiée de manière triviale pour les fonctions indicatrices $g = \mathbf{1}_{B \times U}$ $\forall B \in \mathcal{B}$ et $\forall U \in \mathcal{B}(M_p(E))$ — il s'agit alors de la définition de la probabilité de Palm (définition A.7 page 191). On généralise ensuite cette propriété à toute fonction mesurable positive g par des arguments classiques de classes monotones. \square

Cette équation est à l'origine de nombreuses formules, sous des formes dérivées en choisissant des fonctions g appropriées.

Symétrie

La mesure de Palm possède une propriété de symétrie particulière, comme le montre le théorème suivant :

Théorème A.6 (Mecke) *Pour toute fonction mesurable g de $E \times M_p(E) \rightarrow \mathbb{R}^+$, l'égalité suivante est vérifiée :*

$$\mathbb{E}_{\mathcal{P}_0} \left[\int_E g(x, \xi) \xi(dx) \right] = \mathbb{E}_{\mathcal{P}_0} \left[\int_E g(-x, \phi_x \xi) \xi(dx) \right] \quad (\text{A.10})$$

Preuve. Soit h une fonction mesurable de \mathbb{R}^2 dans \mathbb{R}^+ unitaire, c'est à dire vérifiant $\int_E h(y) \Lambda(dy) \doteq \lambda \int_E h(y) \mathcal{L}(dy) = 1$. On a donc, en vertu de la formule de Campbell,

$$\begin{aligned} \mathbb{E}_{\mathcal{P}_0} \left[\int_E g(x, \xi) \xi(dx) \right] &= \mathbb{E}_{\mathcal{P}_0} \left[\int_E \int_E h(y) g(x, \xi) \xi(dx) \Lambda(dy) \right] \\ &= \mathbb{E}_{\mathcal{P}} \left[\int_E \int_E h(y) g(x, \phi_y \xi) \phi_y \xi(dx) \xi(dy) \right] \end{aligned}$$

Posant $z = \theta_y x$, en en remarquant que $x \in \text{supp}(\phi_y \xi) \equiv \theta_y x \in \text{supp}(\xi)$, on arrive donc à l'égalité suivante :

$$\mathbb{E}_{\mathcal{P}_0} \left[\int_E g(x, \xi) \xi(dx) \right] = \mathbb{E}_{\mathcal{P}} \left[\int_E \int_E h(y) g(z - y, \phi_y \xi) \xi(dz) \xi(dy) \right]$$

On peut alors poser, de manière symétrique, $y = \theta_z u$, ce qui veut dire que pour un z donné $y \in \text{supp}(\xi) \equiv u \in \text{supp}(\phi_z \xi)$. On arrive alors à l'expression désirée :

$$\begin{aligned} \mathbb{E}_{\mathcal{P}_0} \left[\int_E g(x, \xi) \xi(dx) \right] &= \mathbb{E}_{\mathcal{P}} \left[\int_E \int_E h(u + z) g(-u, \phi_{u+z} \xi) \phi_z \xi(du) \xi(dz) \right] \\ &= \mathbb{E}_{\mathcal{P}_0} \left[\int_E \int_E h(u + z) g(-u, \phi_u \xi) \xi(du) \Lambda(dz) \right] \\ &= \mathbb{E}_{\mathcal{P}_0} \left[\int_E g(-u, \phi_u \xi) \xi(du) \right] \end{aligned}$$

□

Formule de Ryll-Nardzewski

Étant donné une fonction k de $E \times M_p(E)$ vers \mathbb{R}^+ vérifiant

$$\int_E k(x, \xi) \xi(dx) = 1 \quad (\text{A.11})$$

pour tout processus $\xi \in M_p(E)$ non nul, posons $g(x, \xi) = k(x, \xi) f(\xi)$, la formule de Campbell devient

$$\begin{aligned} \mathbb{E}_{\mathcal{P}} \left[\int_E k(x, N) f(N) N(dx) \right] &= \lambda \int_E \mathbb{E}_{\mathcal{P}_0} [k(x, \phi_{-x} N) f(\phi_{-x} N)] \mathcal{L}(dx) \\ \mathbb{E}_{\mathcal{P}} [f(N)] &= \lambda \mathbb{E}_{\mathcal{P}_0} \left[\int_E k(x, \phi_{-x} N) f(\phi_{-x} N) \mathcal{L}(dx) \right] \end{aligned}$$

Cette équation permet donc de passer de la probabilité stationnaire à la probabilité de Palm. Néanmoins, cette formule est assez peu parlante sous cette forme. Elle peut être réécrite en utilisant la notion de cellule :

Définition A.9 *Pour tout processus ponctuel N sur E et pour tout $x \in \mathbb{R}$, soit $C_x(N)$ un ensemble de E vérifiant les propriétés suivantes :*

- (i) $x \in C_x(N)$
- (ii) $N(C_0(N)) = 1$
- (iii) $C_0(\phi_x N) = \theta_{-x} C_x(N)$ (compatibilité avec le shift)

$C_x(N)$ sera appelée la **cellule** contenant x .

Cette définition générique est valable pour tout type de “cellule” d’un pavage du plan, étudié dans le chapitre suivant.

Théorème A.7 (Ryll-Nardzewski) *Pour tout processus stationnaire (N, ϕ) de E , d’intensité finie λ , soit \mathcal{P}_0 la probabilité de Palm associée. Les relations suivantes sont vérifiées pour toute fonction f mesurable de $M_p(E)$ dans \mathbb{R}^+ ,*

$$\mathbb{E}_{\mathcal{P}}[f(N)] = \lambda \mathbb{E}_{\mathcal{P}_0} \left[\int_{C_0(N)} f(\phi_x N_0) \mathcal{L}(dx) \right]$$

Preuve. Prenant $\forall x \in \mathbb{R}$ et $\xi \in M_p(E)$, posons $k(x, \xi) = \mathbf{1}_{\{x \in C_0(\xi)\}}$. Il n’est pas dur de voir que cette fonction répond bien au critère (A.11) évoqué précédemment. On arrive donc à

$$\begin{aligned} \mathbb{E}_{\mathcal{P}}[f(N)] &= \lambda \mathbb{E}_{\mathcal{P}_0} \left[\int_E \mathbf{1}_{\{x \in C_0(\phi_{-x} N)\}} f(\phi_{-x} N) \mathcal{L}(dx) \right] \\ &= \lambda \mathbb{E}_{\mathcal{P}_0} \left[\int_E \mathbf{1}_{\{0 \in C_{-x}(N)\}} f(\phi_{-x} N) \mathcal{L}(dx) \right] \end{aligned}$$

□

Corollaire A.2 *Remarquons au passage que si l’on pose $f(N) \equiv 1$ et $f(N) \equiv (\mathcal{L}(C_0(N)))^{-1}$, on obtient*

$$\mathbb{E}_{\mathcal{P}_0}[\mathcal{L}(C_0(N))] = \frac{1}{\lambda} \tag{A.12}$$

$$\mathbb{E}_{\mathcal{P}} \left[\frac{1}{\mathcal{L}(C_0(N))} \right] = \lambda \tag{A.13}$$

A.5 Processus Conjointement Stationnaires

A.5.1 Préliminaires

Cette section reprend l'analyse précédente sur le calcul de Palm pour le cas de plusieurs processus corrélés. Cette nouvelle analyse a été motivée par l'étude des pavages du plan où de nombreux phénomènes de corrélation apparaissent. L'intérêt principal est de définir proprement les probabilité de Palm par rapport à l'un ou l'autre des processus et d'obtenir la formule d'échange de Neveu sous une forme compatible avec les processus corrélés⁴. Cette section est basée en grande partie sur l'analyse de J.Mecke dans [97] pour les pavages du plan.

Pour plus de clarté nous nous restreindrons au cas de deux processus⁵ N_1 et N_2 . On définit la probabilité conjointe $\mathcal{P}_{N_1N_2}$, de sorte que pour toute fonction mesurable de $M_p(E) \times M_p(E)$ dans \mathbb{R} , on écrira :

$$\mathbb{E}[f(N_1, N_2)] \doteq \mathbb{E}_{N_1N_2}[f(N_1, N_2)] \doteq \int_{M_p(E)} \int_{M_p(E)} f(\xi, \zeta) \mathcal{P}_{N_1N_2}(d\xi \times d\zeta)$$

Nous supposons que N_1 et N_2 sont conjointement stationnaires par rapport à ϕ , c'est à dire que

- (i) N_1 et N_2 sont compatibles avec le shift (cf. définition A.4 page 185).
- (ii) ϕ préserve $\mathcal{P}_{N_1N_2}$, c.à.d. que pour tout $U, V \in \mathcal{B}(M_p(E))$,

$$\mathcal{P}_{N_1N_2}(U \times V) = \mathcal{P}_{N_1N_2}(\phi_x U \times \phi_x V)$$

De plus, nous supposons qu'il est possible de définir le processus fusionné $N = N_1 + N_2$. Ceci implique quelques contraintes sur (N_1, N_2) pour que N soit bien un processus ponctuel — c'est à dire qu'en particulier $N(\{x\}) \in \{0, 1\}$ pour presque tout $x \in \mathbb{R}$. Soient λ, λ_1 et λ_2 les intensités respectives de N, N_1 et N_2 . On peut bien sûr écrire $\lambda = \lambda_1 + \lambda_2$.

A.5.2 Calcul de Palm

Processus Somme.

Définition A.10 *On définit la probabilité de Palm par rapport au processus somme N , introduit précédemment, de cette manière :*

$$\mathbb{E}_0[h(N_1, N_2)] = \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E} \left[\int_B h(\phi_x N_1, \phi_x N_2) N(dx) \right]$$

pour tout $B \in \mathcal{B}$.

⁴L'expression donnée usuellement étant en fait un cas particulier de la formule de neveu qui n'est pas toujours vérifiée lorsque les processus prennent des formes de corrélation bien spécifiques.

⁵Les résultats donnés dans cette section sont néanmoins généralisables à un nombre arbitraire de processus.

Les résultats obtenus sur le calcul de Palm dans la section précédente peuvent alors être généralisés sans aucune difficulté au cas du processus somme N .

Proposition A.6 *Pour toute fonction mesurable h de $M_p(E) \times M_p(E)$ dans \mathbb{R}^+ , et pour toute fonction mesurables g de $E \times M_p(E) \times M_p(E)$ dans \mathbb{R}^+ ,*

$$\mathbb{E} \left[\int_E g(x, \phi_x N_1, \phi_x N_2) N(dx) \right] = \lambda \int_E \mathbb{E}_{\mathcal{P}} [g(x, N_1, N_2)] \mathcal{L}(dx) \quad (\text{A.14})$$

$$\mathbb{E}_0 \left[\int_E g(x, N_1, N_2) N(dx) \right] = \mathbb{E}_0 \left[\int_E g(-x, \phi_x N_1, \phi_x N_2) N(dx) \right] \quad (\text{A.15})$$

$$\mathbb{E}[f(N_1, N_2)] = \lambda \mathbb{E}_0 \left[\int_{C_0(N)} f(\phi_x N_1, \phi_x N_2) \mathcal{L}(dx) \right] \quad (\text{A.16})$$

où la cellule $C_0(N)$ répond à la définition A.9.

Processus conjointement stationnaires.

Nous nous intéressons à présent à l'obtention des propriétés équivalentes pour les probabilité de Palm par rapport à N_1 et N_2 , notées respectivement $\mathcal{P}_{N_1^0 N_2}$ et $\mathcal{P}_{N_1 N_2^0}$.

Définition A.11 (Formule de Palm) *Pour toute fonction mesurable h de $M_p(E) \times M_p(E)$ dans \mathbb{R}^+ , et pour tout $B \in E$, les probabilités de Palm $\mathcal{P}_{N_1^0 N_2}$ et $\mathcal{P}_{N_1 N_2^0}$ sont définies comme suit :*

$$\begin{aligned} \mathbb{E}_{N_1^0 N_2} [f(N_1, N_2)] &= \frac{1}{\lambda_1 \mathcal{L}(B)} \mathbb{E} \left[\int_B f(\phi_x N_1, \phi_x N_2) N_1(dx) \right] \\ \mathbb{E}_{N_1 N_2^0} [f(N_1, N_2)] &= \frac{1}{\lambda_2 \mathcal{L}(B)} \mathbb{E} \left[\int_B f(\phi_x N_1, \phi_x N_2) N_2(dx) \right] \end{aligned}$$

Les propriété de Campbell, de symétrie, et de Ryll-Nardzewski sont ensuite aisément obtenus en suivant les démonstrations données dans la section précédente.

Proposition A.7 (Campbell) *Pour toute fonction mesurable g de $E \times M_p(E) \times M_p(E)$ dans \mathbb{R}^+ ,*

$$\begin{aligned} \mathbb{E} \left[\int_E g(x, \phi_x N_1, \phi_x N_2) N_1(dx) \right] &= \lambda_1 \mathbb{E}_{N_1^0 N_2} \left[\int_E g(x, N_1, N_2) \mathcal{L}(dx) \right] \\ \mathbb{E} \left[\int_E g(x, \phi_x N_1, \phi_x N_2) N_2(dx) \right] &= \lambda_2 \mathbb{E}_{N_1 N_2^0} \left[\int_E g(x, N_1, N_2) \mathcal{L}(dx) \right] \end{aligned}$$

Proposition A.8 (Ryll-Nardzewski) *Pour toute fonction mesurable h de $M_p(E) \times M_p(E)$ dans \mathbb{R}^+ , l'égalité suivante est vérifiée :*

$$\begin{aligned} \mathbb{E}[f(N_1, N_2)] &= \lambda_1 \mathbb{E}_{N_1^0 N_2} \left[\int_{C_0(N_1)} f(\phi_x(N_1, N_2)) \mathcal{L}(dx) \right] \\ &= \lambda_2 \mathbb{E}_{N_1 N_2^0} \left[\int_{C_0(N_2)} f(\phi_x(N_1, N_2)) \mathcal{L}(dx) \right] \end{aligned}$$

où la cellule $C_0(\xi)$ répond à la définition A.9 ($\forall \xi \in M_p(E)$).

A.5.3 Relation entre ces probabilités.

On s'intéresse à présent aux relations reliant les probabilités de Palm par rapport à N , N_1 et N_2 . Nous utilisons l'approche de J.Mecke [97].

Lemme A.1 *Soient Ω_1 et Ω_2 les événements de $\mathcal{B}(M_p(E))$ définis par $\Omega_i = \{0 \in \text{supp}(N_i)\}$ pour $i = 1, 2$. Pour toute fonction mesurable f de $M_p(E) \times M_p(E)$ dans \mathbb{R}^+ , on a alors :*

$$\lambda \mathbb{E}_0[\mathbf{1}_{\Omega_1} f(N_1, N_2)] = \lambda_1 \mathbb{E}_{N_1^0 N_2} [f(N_1, N_2)] \quad (\text{A.17})$$

$$\lambda \mathbb{E}_0[\mathbf{1}_{\Omega_2} f(N_1, N_2)] = \lambda_2 \mathbb{E}_{N_1 N_2^0} [f(N_1, N_2)] \quad (\text{A.18})$$

Preuve. D'après la définition A.10, on a, pour tout $B \in \mathcal{B}$,

$$\lambda \mathbb{E}_0[\mathbf{1}_{\Omega_1} f(N_1, N_2)] = \frac{1}{\mathcal{L}(B)} \mathbb{E} \left[\int_B \mathbf{1}_{\Omega_1}(\phi_x N_1, \phi_x N_2) f(\phi_x N_1, \phi_x N_2) N(dx) \right]$$

Remarquant que $\{0 \in \text{supp}(\phi_x N_1)\}$ équivaut à $\{x \in \text{supp}(N_1)\}$, on arrive alors à

$$\lambda \mathbb{E}_0[\mathbf{1}_{\Omega_1} f(N_1, N_2)] = \frac{1}{\mathcal{L}(B)} \mathbb{E} \left[\int_B f(\phi_x N_1, \phi_x N_2) N_1(dx) \right]$$

ce qui permet de conclure, en utilisant la définition A.11. □

La relation entre les mesures de Palm sous N_1 et N_2 est donnée par la formule d'échange de Neveu décrite ci-après. Nous donnons cette formule sous une forme spécifique, adaptée à l'étude des pavages du plan. L'énoncé original⁶ de cette identité peut être trouvée dans [96].

⁶L'énoncé original de cette formule requiert néanmoins une approche plus abstraite de la probabilité de Palm.

Théorème A.8 (Formule de Neveu — forme générique) *Pour toute fonction mesurable h de $E \times M_p(E) \times M_p(E)$ dans \mathbb{R}^+ ,*

$$\begin{aligned} & \lambda_1 \mathbb{E}_{N_1^0 N_2} \left[\int_E h(x, N_1, N_2) \mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}} N_2(dx) \right] \\ &= \lambda_2 \mathbb{E}_{N_1 N_2^0} \left[\int_E h(x, \phi_x N_1, \phi_x N_2) \mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}} N_1(dx) \right] \end{aligned}$$

Preuve. Posons pour tout $\xi, \zeta \in M_p(E)$,

$$f(x, \xi, \zeta) = \mathbf{1}_{\Omega_1}(\xi) h(x, \xi, \zeta) \mathbf{1}_{\{x \in \text{supp}(\zeta)\}} \mathbf{1}_{\{0 \in \overline{C_0(\zeta)}\}}$$

On s'intéresse alors à $\mathcal{E} = \lambda \mathbb{E}_0 \left[\int_E f(x, N_1, N_2) N(dx) \right]$.

$$\begin{aligned} \mathcal{E} &= \lambda \mathbb{E}_0 \left[\mathbf{1}_{\Omega_1} \int_E h(x, N_1, N_2) \mathbf{1}_{\{0 \in \overline{C_0(N_2)}\}} N_2(dx) \right] \\ &= \lambda_1 \mathbb{E}_{N_1^0 N_2} \left[\int_E h(x, N_1, N_2) \mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}} N_2(dx) \right] \end{aligned}$$

On a utilisé l'équation (A.17) du lemme A.1. Pour obtenir le terme de droite, on utilise la propriété de symétrie de \mathcal{P}_0 (équation A.15) :

$$\begin{aligned} \mathcal{E} &= \lambda \mathbb{E}_0 \left[\int_E f(-x, \phi_x(N_1, N_2)) N(dx) \right] \\ &= \lambda \mathbb{E}_0 \left[\int_E \mathbf{1}_{\{0 \in \text{supp}(\phi_x N_1)\}} h(-x, \phi_x(N_1, N_2)) \right. \\ &\quad \left. \mathbf{1}_{\{-x \in \text{supp}(\phi_x N_2)\}} \mathbf{1}_{\{0 \in \overline{C_{-x}(\phi_x(N_2))}\}} N(dx) \right] \end{aligned}$$

Remarquons que

$$\begin{aligned} 0 \in \text{supp}(\phi_x N_1) &\Leftrightarrow x \in \text{supp}(N_1) \\ -x \in \text{supp}(\phi_x N_2) &\Leftrightarrow 0 \in \text{supp}(N_2) \\ 0 \in \overline{C_{-x}(\phi_x(N_2))} &\Leftrightarrow x \in \overline{C_0(N_2)} \end{aligned}$$

Une nouvelle application de (A.18) achève cette démonstration :

$$\begin{aligned} \mathcal{E} &= \lambda \mathbb{E}_0 \left[\mathbf{1}_{\Omega_2} \int_E h(-x, \phi_x(N_1, N_2)) \mathbf{1}_{\{x \in \overline{C_0(N_2)}\}} N_1(dx) \right] \\ &= \lambda_2 \mathbb{E}_{N_1 N_2^0} \left[\int_E h(-x, \phi_x(N_1, N_2)) \mathbf{1}_{\{x \in \overline{C_0(N_2)}\}} N_1(dx) \right] \end{aligned}$$

□

Corollaire A.3 (Formule de Neveu — forme simplifiée) *Si les deux processus N_1 et N_2 sont indépendants, alors, pour toute fonction mesurable g de $M_p(E) \times M_p(E)$ dans \mathbb{R}^+ ,*

$$\lambda_1 \mathbb{E}_{N_1^0 N_2} [g(N_1, N_2)] = \lambda_2 \mathbb{E}_{N_1 N_2^0} \left[\int_{C_0(N_2)} g(\phi_x(N_1, N_2)) N_1(dx) \right]$$

Preuve. L'indépendance de N_1 et N_2 implique que, presque sûrement,

$$\int_E \mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}} N_2(dx) = 1 \quad \text{p.s. sous } \mathcal{P}_{N_1^0 N_2}$$

De plus, $C_0(N_2)$ est presque sûrement la seule cellule vérifiant $\mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}}$ pour $x \in \text{supp}(N_1)$. \square

Remarque A.3 L'analyse effectuée dans cette section est généralisable au cas de k processus N_1, N_2, \dots, N_k conjointement stationnaires tels que $N = \sum_{i=1}^k N_i$ soit (p.s.) un processus ponctuel. Le cas de trois processus conjointement stationnaires est particulièrement intéressant pour l'étude des pavages aléatoire du plan, comme montré dans le chapitre suivant.

A.6 Bréviaire sur le calcul de Palm

Pour plus de rigueur dans la définition de la probabilité de Palm, il a été souhaitable d'utiliser l'espace canonique de Probabilité, c'est à dire $\Omega = M_p(E)$. Il est parfois plus commode dans les applications de garder les notations classiques sur un espace de probabilité générique $(\Omega, \mathcal{F}, \mathbb{P}, \phi)$. Les formules précédentes peuvent être réécrite comme montré ci-après dans le récapitulatif. Par exemple, une fonctionnelle mesurable de $M_p(E)$ dans \mathbb{R} devient une fonction mesurable de Ω dans \mathbb{R} , c'est à dire une variable aléatoire. On identifie :

$$\mathbb{E}_{\mathcal{P}}[f(N)] \equiv \int_{M_p(E)} f(\xi) \mathcal{P}(d\xi) = \int_{\Omega} f(\omega) \mathbb{P}(d\omega) \equiv \mathbb{E}[f]$$

De même, on note usuellement \mathbb{E}_0 , au lieu de $\mathbb{E}_{\mathcal{P}_0}$, la moyenne par rapport à la probabilité de Palm et \mathbb{E} la moyenne par rapport à la probabilité stationnaire. Lorsque plusieurs processus sont en jeux, on note aussi souvent \mathbb{E}_0^N la moyenne par rapport à la mesure de Palm relatif au processus N . Dans le tableau A.1 sont rappelées les formules les plus usuelles du calcul de Palm, pour toute variable aléatoire f à valeurs dans \mathbb{R}^+ et pour toute fonction mesurable g de $E \times \Omega$ vers \mathbb{R}^+ .

$\mathbb{E}_0[f] = \frac{1}{\lambda \mathcal{L}(B)} \mathbb{E} \left[\int_B f \circ \phi_x N(dx) \right]$	Palm (Déf.A.7)
$\mathbb{E} \left[\int_E g(x) N(dx) \right] = \lambda \int_E \mathbb{E}_0[g(x) \circ \phi_{-x}] \mathcal{L}(dx)$	Campbell (Th.A.5)
$\mathbb{E}_0 \left[\int g(x) N(dx) \right] = \mathbb{E}_0 \left[\int g(-x) \circ \phi_x N(dx) \right]$	Symétrie (Mecke) (Th.A.6)
$\mathbb{E}[f] = \lambda \mathbb{E}_0 \left[\int_{C_0(N_0)} f \circ \phi_x \mathcal{L}(dx) \right]$	Ryll-Nardzewski (Th. A.7)
$\lambda_1 \mathbb{E}_0^{N_1}[f] = \lambda_2 \mathbb{E}_0^{N_2} \left[\int_{C_0(N_2)} f \circ \phi_x N_1(dx) \right]$	Neveu (Cor.A.3)

TAB. A.1 – Calcul de Palm : Formules usuelles.

Annexe B

Pavages du plan

Dans ce chapitre sont brièvement présentés les pavages aléatoires du plan et leurs principales propriétés sont démontrées. Nous utilisons les processus ponctuels et le calcul de Palm pour cette étude — approche initiée par J.Mecke dans [97]. Il est donc nécessaire de bien connaître ces notions, présentés en annexe A, avant d’entreprendre la lecture de ce chapitre.

B.1 Introduction

Un pavage du plan ou Mosaïque (*Tesselation* en anglais) est une division du plan en polygones. La figure B.4 représente un exemple de mosaïque. Plus formellement, on peut définir cet objet suivant l’approche de J.Mecke [100, chapitre 10].

Définition B.1 Soit Π l’ensemble des polygones convexes, ouverts, bornés et non-vides de E . Un sous-ensemble Θ de Π est un pavage du plan (ou mosaïque) si et seulement si :

- (i) $P_1 \cap P_2 = \emptyset \forall (P_1, P_2) \in \Theta^2$ avec $P_1 \neq P_2$
- (ii) $\bigcup_{P \in \Theta} \overline{P} = E$
- (iii) $\{P \in \Theta : P \cap B \neq \emptyset\}$ est fini pour tout ensemble $B \subset E$ borné.

Les polygones $P \in \Theta$ sont appelées les **cellules** de la mosaïque Θ .

Soit \mathcal{T} l’ensemble de tous les pavages du plan et $\mathcal{B}(\mathcal{T})$ la σ -algèbre associée. Soit enfin $(\Omega, \mathcal{F}, \mathbb{P})$ un espace de probabilité.

Définition B.2 Une mosaïque aléatoire (ou pavage du plan aléatoire) est une variable aléatoire Θ de (Ω, \mathcal{F}) vers $(\mathcal{T}, \mathcal{B}(\mathcal{T}))$.

B.2 Résultats généraux

B.2.1 Notations

Dorénavant, on s'intéresse uniquement aux mosaïques stationnaires. Il est possible d'associer plusieurs processus stochastiques à un pavage du plan Θ . Par exemple, le processus de lignes que constituent les polygones de Θ peut être étudié (voir [100] pour une définition des processus de lignes). Nous nous intéresserons plutôt aux processus ponctuels associés à Θ , en utilisant les notations et l'approche de Mecke [100].

- Soit N_0 le processus ponctuel (stationnaire) formé par les sommets des polygones de Θ .
- Soit N_1 le processus ponctuel (stationnaire) formé par les milieux des arêtes de Θ .
- Soit N_2 les centres des cellules de Θ (i.e. les centres des polygones de Θ).

Remarque : La définition d'un "centre" est ambiguë ; de quel "centre" veut-on parler ? Centre de gravité, centre de symétrie ? Il n'existe pas vraiment de choix "canonique" pour l'ensemble des mosaïques. J.Møller donne plus d'indication sur la question dans [98]. En fait, la définition est volontairement ambiguë, car on peut choisir n'importe quel type de "centre" du moment que ce choix soit compatible avec le "shift". Autrement dit, si $c_0(\Theta)$ est le centre de la cellule contenant l'origine, il faut et il suffit que $\forall x \in E, c_0(\phi_x \Theta) = \theta_{-x} c_x(\Theta)$. Il en est de même pour la définition des "centres" des arêtes (c.à.d. N_1).

Soient $\lambda_0, \lambda_1, \lambda_2$ leur intensité respective, et $\mathbb{P}_0^{N_0}, \mathbb{P}_0^{N_1}, \mathbb{P}_0^{N_2}$ les probabilités de Palm associées¹. Soient $\alpha_0, \alpha_1, \alpha_2$ les supports respectifs de ces processus ponctuels.

- On peut aussi définir plusieurs variables aléatoires caractéristiques de Θ .
- $S(\Theta)$ représente la surface de la cellule de Θ contenant l'origine. On définit aussi la grandeur $\bar{S} = \mathbb{E}_0^{N_2}[S] = \mathbb{E}_0^{N_2}[\mathcal{L}(C_0(\Theta))]$
 - De même $U(\Theta)$ correspond au périmètre de la cellule contenant 0, et sa moyenne sous P_0^2 est notée \bar{U}
 - $s(\Theta)$ est le nombre de sommets de la cellule contenant l'origine. Θ . Sa moyenne $\mathbb{E}_0^{N_2}[s]$ sera notée \bar{s} .
 - $A_x(\Theta)$ est le nombre d'arêtes issues du point x pour $x \in \alpha_0(\Theta)$. On définit aussi $\bar{A} = \mathbb{E}_0^{N_0}[A_0]$.
 - $L_x(\Theta)$ est la longueur d'une arête dont le centre est x , pour $x \in \alpha_1(\Theta)$. La grandeur $\mathbb{E}_0^{N_1}[L_0(\Theta)]$ sera notée \bar{L} .

¹On pourra remarquer qu'il s'agit de processus corrélés. Nous faisons donc référence aux définitions de la probabilité de Palm de la section A.5.

B.2.2 Formules

Les théorèmes énoncés dans le chapitre précédent sur le calcul de Palm peuvent être directement appliqués dans ce contexte. Il suffit d'identifier $C_0(\Theta)$ à $C_0(N_2)$, $f(\Theta)$ à $f(N_0, N_1, N_2)$, \mathbb{P} à $\mathbb{P}_{N_0 N_1 N_2}$ et $\mathbb{P}_0^{N_2}$ à $\mathbb{P}_{N_0 N_1 N_2^0}$ etc.

D'après la formule de Ryll-Nardzewski (A.7), on obtient l'équation

$$\mathbb{E}[f(\Theta)] = \lambda_2 \mathbb{E}_0^{N_2} \left[\int_{C_0(\Theta)} f \circ \phi_x \mathcal{L}(dx) \right] \quad (\text{B.1})$$

valable pour toute variable aléatoire mesurable f et pour toute mosaïque stationnaire Θ .

Posant $f \equiv 1$, on obtient la relation suivante :

$$\bar{S} = \mathbb{E}_0^{N_2}[\mathcal{L}(C_0(\Theta))] = \frac{1}{\lambda_2} \quad (\text{B.2})$$

Posant $f(\Theta) = \mathcal{L}(C_0(\Theta))$, on arrive à l'équation

$$\begin{aligned} \mathbb{E}[\mathcal{L}(C_0(\Theta))] &= \lambda_2 \mathbb{E}_0^{N_2} \left[[\mathcal{L}(C_0(\Theta))]^2 \right] \\ &\geq \lambda_2 \left(\mathbb{E}_0^{N_2}[\mathcal{L}(C_0(\Theta))] \right)^2 \end{aligned}$$

En utilisant alors l'équation B.2, on arrive finalement à l'inégalité suivante :

$$\mathbb{E}[\mathcal{L}(C_0(\Theta))] \geq \mathbb{E}_0^{N_2}[\mathcal{L}(C_0(\Theta))] \quad (\text{B.3})$$

Les formules suivantes dérivent directement de la formule d'échange de Neveu.

Proposition B.1 (Mecke [97]) *Pour toute fonction f caractéristique du pavage, c.à.d. $f : \Theta \rightarrow \mathbb{R}^+$,*

$$\lambda_2 \mathbb{E}_0^{N_2} \left[\int_{C_0(\Theta)} f(\phi_x \Theta) N_1(dx) \right] = 2\lambda_1 \mathbb{E}_0^{N_1}[f(\Theta)]$$

Preuve. Par la formule d'échange de Neveu (théorème A.8 à la page 200). Il suffit de remarquer que

$$\mathbb{E}_0^{N_1} \left[f(\Theta) \int_E \mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}} N_2(dx) \right] = 2\mathbb{E}_0^{N_1}[f(\Theta)]$$

car il existe deux points pour lesquels $x \in \alpha_2$ et $x \in C_0(N_2)$ sous $\mathbb{P}_0^{N_1}$, comme le montre la figure B.1. \square

Corollaire B.1 *En posant $f \equiv 1$, on obtient*

$$2\lambda_1 = \lambda_1 \bar{S} \quad (\text{B.4})$$

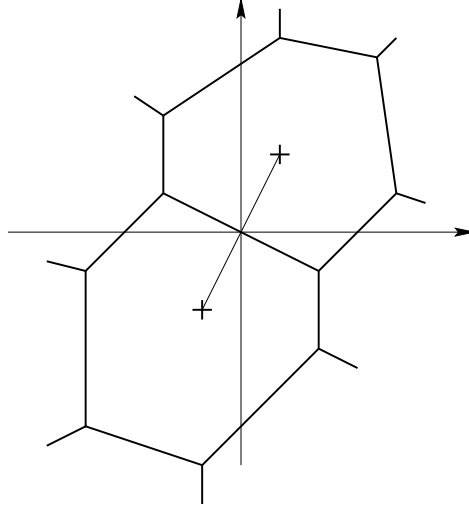


FIG. B.1 – Cas où $0 \in \alpha_1(\Theta) \cap C_x(\Theta)^*$ pour $x \in \alpha_2(\Theta)$.

De même, en posant $f(\Theta) = L_0(\Theta)$, on obtient :

$$2\lambda_1 \bar{L} = \lambda_2 \mathbb{E}_0^{N_2} \left[\int_{C_0(\Theta)} L_x(\Theta) N_1(dx) \right] = \lambda_2 \bar{U} \quad (\text{B.5})$$

Proposition B.2 (Mecke [97]) Pour chaque $x \in N_0(\Theta)$ soit $S_x(\Theta)$ l'ensemble des centres des cellules se joignant en x . La relation suivante est vérifiée pour tout fonction mesurable h de $E \times \mathcal{T}$ dans \mathbb{R}^+ :

$$\lambda_2 \mathbb{E}_0^{N_2} \left[\int_{C_0(\Theta)} h(x, \Theta) N_0(dx) \right] = \lambda_0 \mathbb{E}_0^{N_0} \left[\sum_{x \in S_0(\Theta)} h(-x, \phi_x \Theta) \right]$$

Preuve. Cette équation est aussi une application directe de la formule de Neveu (théorème A.8 dans le chapitre précédent). Il suffit d'identifier

$$\mathbb{E}_0^{N_0} \left[\int_E h(-x, \phi_x \Theta) \mathbf{1}_{\{0 \in \overline{C_x(N_2)}\}} N_2(dx) \right] = \mathbb{E}_0^{N_0} \left[\sum_{x \in S_0(\Theta)} h(-x, \phi_x \Theta) \right]$$

comme illustré par la figure B.2. □

Corollaire B.2 Prenant $h \equiv 1$ on obtient l'identité suivante :

$$\lambda_2 \bar{s} = \lambda_0 \bar{A} \quad (\text{B.6})$$

Corollaire B.3

$$\lambda_2 \pi (\bar{s} - 2) = 2\pi \lambda_0 \quad (\text{B.7})$$

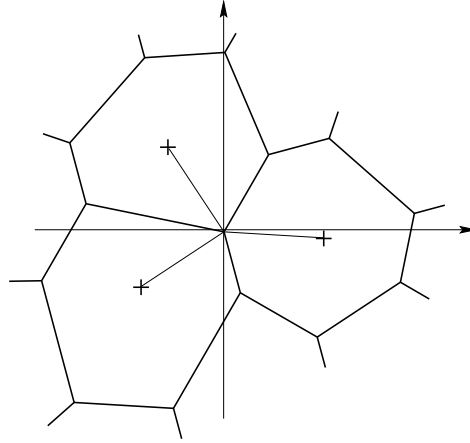


FIG. B.2 – Cas où $0 \in \alpha_0(\Theta) \cap C_x(\Theta)^*$ pour $x \in \alpha_2(\Theta)$.

Preuve. En posant $h(x, \Theta) = w(x, C_0(\Theta))$, où $w(x, C)$ est l'angle intérieur de la cellule C au point x , et en appliquant le théorème précédent, on arrive à l'identité annoncée. En effet, en remarquant que $w(-x, C_0(\phi_x \Theta)) = w(0, C_x(\Theta))$, l'expression de droite dans le théorème précédent correspond à la somme des angles $w(o, C_{x_1}) + w(o, C_{x_2}) + w(o, C_{x_3})$ qui vaut évidemment 2π (où x_1, x_2 et x_3 sont les trois centres de $S_0(\Theta)$) — voir figure B.2.

Il est aussi facile de voir que l'expression de gauche

$$\int_{C_0(\Theta)^*} w(x, C_0(\Theta)) N_0(dx)$$

correspond à la somme des angles intérieurs pour chaque sommet de la cellule C_0 . En utilisant le lemme B.1, on voit donc que l'expression de gauche vaut $\mathbb{E}_0^{N_2}[\pi(s - 2)]$. \square

Lemme B.1 *Pour une figure géométrique quelconque de S sommets, la somme des angles intérieurs pris à chaque sommet est $2\pi(S - 2)$.*

Preuve. Ce résultat est bien connu pour les triangles, dont la somme des angles fait bien π . Pour généraliser, il suffit de voir qu'une figure comportant s sommets peut se décomposer en s triangles, comme montré sur la figure B.3. La somme des angles peut donc s'écrire sous la forme $\sum_{i=1}^s (\pi - \alpha_i) = \pi(s - 2)$. \square

Proposition B.3 *Les formules (B.4) (B.6) et (B.7) peuvent aussi s'écrire :*

$$\lambda_1 - \lambda_2 + \lambda_3 = 0 \tag{B.8}$$

$$\bar{s} = 2 + 2 \frac{\lambda_0}{\lambda_2} \tag{B.9}$$

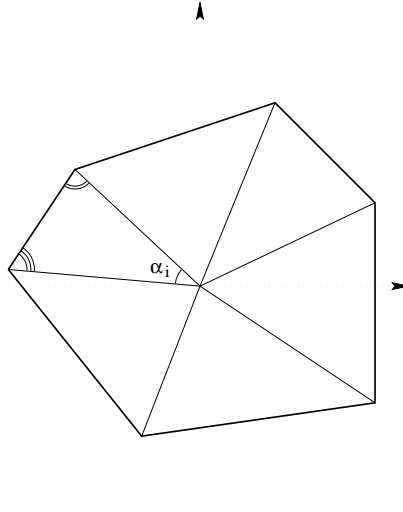


FIG. B.3 – Un résultat élémentaire de géométrie...

$$\bar{A} = 2 + 2 \frac{\lambda_2}{\lambda_0} \quad (\text{B.10})$$

B.3 Pavages de Poisson-Voronoi

B.3.1 Définition

Définition B.3 Pour tout $N \in M_p(E)$ et $x \in \text{supp}(N)$, soit $C_x(N)$ l'ensemble de \mathcal{B} défini par

$$C_x(N) = \{y \in E : \|y - x\| < \|y - x_i\| \forall x_i \in \text{supp}(N), x_i \neq x\}$$

Cette ensemble convexe est la **Cellule de Voronoi** centrée en x .

Note. Par abus de notation, on note également $C_x(N)$ la cellule de Voronoi contenant x (même si $x \notin \text{supp}(N)$). Il existe p.s. une seule cellule répondant à cette définition.

Il est très facile de construire un pavage du plan à partir d'un processus ponctuel donné N_2 , comme la réunion des cellules de Voronoi associées aux points du support du processus N_2 . Cette mosaïque est appelée **pavage de Voronoi** ou **pavage de Dirichlet** (dans le cas $E = \mathbb{R}^2$) — voir figure B.4.

Nous nous intéressons dans la suite au cas où le processus sous-jacent est un processus de Poisson stationnaire. On parlera dans ce cas de **pavage de Poisson-Voronoi**. Il est très facile de voir que dans ces conditions, le pavage est "normal", c'est à dire que $A(\Theta) = 3$ presque sûrement.

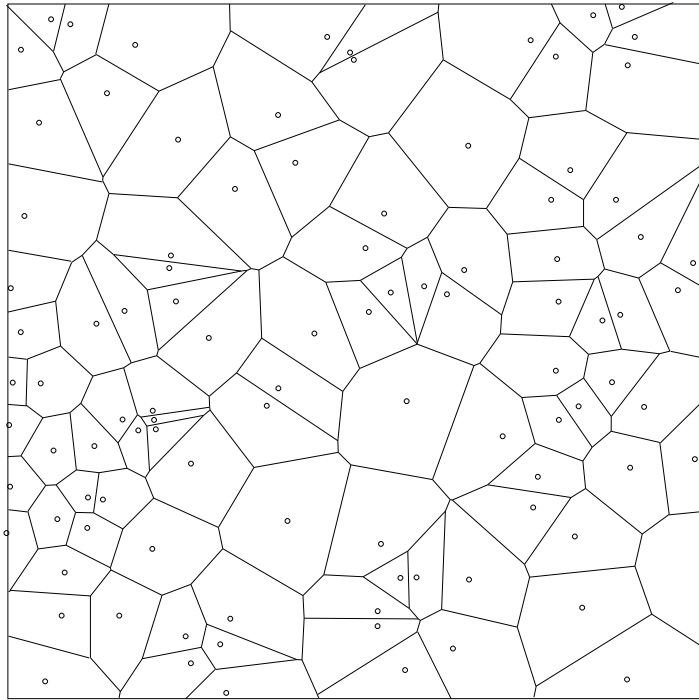


FIG. B.4 – Réalisation d'un pavage du plan de Voronoi

B.3.2 Propriétés génériques

Proposition B.4 *Le pavage de Poisson-Voronoi est normal.*

Preuve. Par construction, le centre d'une cellule correspond à l'intersection des bissectrices de ces arêtes. Inversement, les sommets (points de N_0) correspondent aux intersections des bissectrices du **pavage de Delaunay** (voir figure B.5). Pour que quatre arêtes se croisent en un point de N_0 , il faut donc que les sommets des cellules avoisinantes soient disposées en rectangle. Cette configuration a, bien sûr, une probabilité nulle d'être réalisée. Il en est de même pour les cas $A = 5, 6, \dots$, qui demandent des configurations tout aussi spécifiques et dont la probabilité d'occurrence est nulle. \square

Corollaire B.4 *D'après les formules (B.9), (B.10) et (B.8), les pavages de Poisson-Voronoi vérifient les propriétés suivantes :*

$$\bar{A} = 3$$

$$\bar{s} = 6$$

$$\lambda_0 = 2 \lambda_2$$

$$\lambda_1 = 3 \lambda_2$$

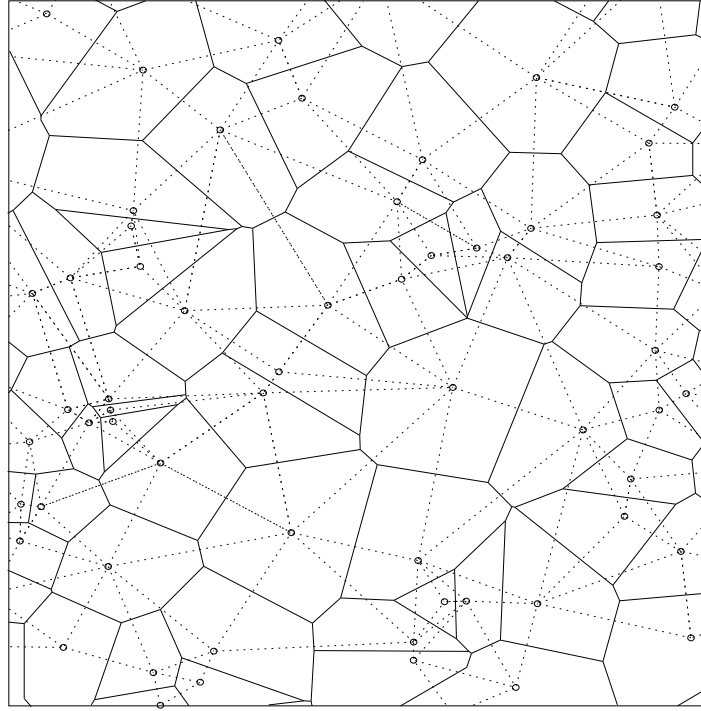


FIG. B.5 – Réalisation d'un pavage du plan de Voronoi et du pavage de Delaunay associé

B.3.3 Tailles des Cellules.

Sous Palm.

Proposition B.5

$$\mathbb{E}_0^{N_2}[\mathcal{L}(C_0(\Theta))] = \frac{1}{\lambda_2} \quad (\text{B.11})$$

$$\mathbb{E}_0^{N_2}[(\mathcal{L}(C_0(\Theta)))^2] = \frac{\gamma}{\lambda_2^2} \quad \text{où } \gamma = 1.280 \quad (\text{B.12})$$

$$(\text{B.13})$$

Preuve. Nous avons déjà démontré la première expression (équation (B.4)) par une application directe du théorème de Ryll-Nardzewski. Nous admettrons la seconde expression, due à J.L.Meijering en 1953 (voir [100] pour plus de renseignement). \square

Sous la probabilité stationnaire.

Proposition B.6

$$\mathbb{E}[\mathcal{L}(C_0(\Theta))] = \frac{\gamma}{\lambda_2} \quad (\text{B.14})$$

Preuve. La moyenne $\mathbb{E}[\mathcal{L}(C_0(\Theta))]$ peut être déterminé par la formule de Ryll-Narzewski (équation (B.1))

$$\begin{aligned} \mathbb{E}[\mathcal{L}(C_0(\Theta))] &= \lambda_2 \mathbb{E}_0^{N_2} \left[\int_{C_0(\Theta)} \mathcal{L}(C_0(\phi_x(\Theta))) N_2(dx) \right] \\ &= \lambda_2 \mathbb{E}_0^{N_2} [(\mathcal{L}(C_0(\Theta)))^2] \\ &= \frac{\gamma}{\lambda_2} \end{aligned}$$

□

Remarque B.1 Remarquons que l'inégalité (B.3) est bien vérifiée... Cette différence est due à la sélection des cellules (celles contenant l'origine) qui introduit un biais dans les statistiques considérées : Une cellule de grande taille a en effet plus de chance de contenir l'origine qu'une petite cellule.

Nous admettrons également la proposition suivante :

Proposition B.7

$$\mathbb{E}[(\mathcal{L}(C_0(\Theta)))^2] = \delta \left(\frac{1}{\lambda_2} \right)^2 \quad (\text{B.15})$$

avec $\delta \approx 2$.

Preuve. Par des arguments d'échelles, il est facile de voir que cette quantité est proportionnelle à $(1/\lambda_2)^2$. Le terme δ a été déterminée par simulation. □

Distribution.

La distribution de la taille d'une cellule typique n'est pas connue de manière explicite. En revanche, des simulations ont permis de déterminer des règles approchées (par "fitting"). Les principales distributions approchées sont disponibles dans [99]. Nous donnons dans la proposition suivante la distribution qui nous semble la plus adaptée à nos travaux.

Proposition B.8 *Sous Palm, $|C_0(\pi)|$ admet une densité proche de*

$$f(x) = \frac{b^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-bx),$$

avec $b \approx \alpha\lambda_2$, si λ_2 est l'intensité du processus de Poisson sous-jacent, et $\alpha \approx 3.57$. Dans l'expression précédente, Γ représente la fonction d'Euler.

Corollaire B.5 *Remarquons qu'avec la distribution donnée dans la proposition précédente, $\forall k = 1, 2, \dots$,*

$$\begin{aligned} \mu_k &= \mathbb{E}_0^{N_2} \left[(\mathcal{L}(C_0(\pi)))^k \right] \\ &= \frac{\Gamma(\alpha + k)}{\Gamma(\alpha)} (\alpha\lambda_2)^{-k} \\ &= \frac{(\alpha + k - 1) \dots (\alpha + 2)(\alpha + 1)}{\alpha^{k-1}} \lambda_2^{-k} \end{aligned}$$

En particulier, on trouve bien $\mu_1 = \frac{1}{\lambda_2}$ et $\mu_2 \approx \frac{1.28}{\lambda_2^2}$.

Appartenance à une Cellule.

Proposition B.9 *Soit Θ un pavage de Poisson-Voronoi stationnaire. La propriété suivante est vérifiée :*

$$\mathbb{P}_0^{N_2} \{x \in C_0(\Theta)\} = \exp\left(-\lambda_2 \pi \|x\|^2\right) \quad (\text{B.16})$$

Preuve.

$$\begin{aligned} \{x \in C_0(\Theta)\} &= \{\forall x_i \in \text{supp}(N_2), x_i \neq 0, \|x\| \leq \|x - x_i\|\} \\ &= \{\forall x_i \in \text{supp}(N_2), x_i \neq 0, 2x \cdot x_i \leq \|x_i\|^2\} \\ &= \{\forall x_i \in \text{supp}(N_2), x_i \neq 0, x_i \notin \mathcal{C}(x, \|x\|)\} \end{aligned}$$

où $\mathcal{C}(u, r)$ est le cercle de centre u et de rayon r . On utilise alors le théorème A.2 sur les processus de Poisson (page 188) pour achever ce calcul.

$$\begin{aligned} \mathbb{P}_0^{N_2} \{x \in C_0(\Theta)\} &= \mathbb{E}_0^{N_2} \left[\prod_{x_i \in \text{supp}(N_2)} \mathbf{1}_{\mathcal{C}(x, \|x\|)^c}(x_i) \mathbf{1}_{\{x_i \neq 0\}} \right] \\ &= \mathbb{E} \left[\prod_{x_i \in \text{supp}(N_2)} \mathbf{1}_{\mathcal{C}(x, \|x\|)^c}(x_i) \right] \quad (\text{d'après Slivnyak}) \\ &= \exp \left(- \int_E [1 - \mathbf{1}_{\mathcal{C}(x, \|x\|)^c}(u)] \lambda_2 \mathcal{L}(du) \right) \\ &= \exp(-\lambda_2 \mathcal{L}(\mathcal{C}(x, \|x\|))) \end{aligned}$$

□

La proposition suivante est aussi très utile pour l'étude des réseaux hiérarchiques (voir [87] ou les chapitres 3 et 6) :

Proposition B.10 *Deux points quelconques distants de ρ appartiennent à la même cellule d'un pavage de Poisson-Voronoi Θ avec la probabilité $Q_{\lambda_2}(\rho)$ avec*

$$Q_{\lambda_2}(\rho) = Q_1(\rho\sqrt{\lambda_2}) \doteq Q(\rho\sqrt{\lambda_2})$$

où

$$Q(\rho) = 2\rho^2 \int_0^\pi d\phi \int_0^\infty \alpha \exp(-\rho^2 A(\alpha, \phi)) d\alpha$$

$$A(\alpha, \phi) = \alpha \sin \phi + \alpha^2(\pi - \phi) + (\alpha^2 + 1 - 2\alpha \cos \phi) \left(\pi - \arccos \frac{1 - \alpha \cos \phi}{\sqrt{\alpha^2 + 1 - 2\alpha \cos \phi}} \right)$$

Preuve. Voir [87] par exemple.

□

B.4 Pavages de Delaunay

B.4.1 Caractéristiques

Le pavage de Delaunay peut être construit simplement à partir du pavage de Voronoi — il s’agit de son dual comme le montre la figure B.5. Il est donc facile de voir que

$$\lambda_i^D = \lambda_{2-i}^V$$

si λ_i^D est l’intensité du processus $N_i(\Theta_D)$ d’un pavage de Delaunay Θ_D et où λ_i^V est l’intensité du processus $N_i(\Theta_V)$ d’un pavage de Voronoi Θ_V , pour $i = 1, 2, 3$ — Les processus $N_1(\Theta), N_2(\Theta), N_3(\Theta)$ d’un pavage du plan quelconque Θ étant définis au début de ce chapitre.

Les principales caractéristiques de ces pavages sont énumérés ci-dessous.

Proposition B.11

$$\begin{aligned} \lambda_1 &= 3\lambda_0 & \lambda_2 &= 2\lambda_0 \\ \bar{s} &= 3 & \bar{A} &= 6 \\ \bar{S} &= \frac{1}{\lambda_2} \\ \bar{L} &= \frac{32}{9\pi} \frac{1}{\sqrt{\lambda_2}} \end{aligned}$$

Preuve. Les relations génériques des pavages du plan permettent de retrouver les valeurs de \bar{s}, \bar{A} et \bar{S} . Le dernier résultat sur \bar{L} sera admis. \square

Nous rappelons dans la suite quelques caractéristiques des chemins sur le graphe de Delaunay.

B.4.2 Routage sur le pavage de Delaunay

Introduction

Soit π un processus de Poisson homogène, d’intensité λ . Considérons tout d’abord que l’on cherche à joindre deux points de π , s et d (pour “source” et “destination”), à travers le graphe de Delaunay engendré par π . Plusieurs algorithmes sont possible (algorithme de routage dans le vocabulaire des réseaux). Nous noterons $\text{path}(s, d, \pi)$ l’ensemble des noeuds de π qui permettent de relier les points s et d . Certaines caractéristiques de ces chemins, sont indispensables à l’étude du routage. Tout particulièrement, on s’intéresse aux quantités suivantes :

- Nombre de sauts, noté $|\text{path}(s, d, \pi)|$. Le nombre de liens utilisés pour joindre s à d .
- Longueur du chemin, noté $\|\text{path}(s, d, \pi)\|$. Il s’agit de la somme des longueurs des arêtes utilisées par le chemin.
- Longueur moyenne des arêtes sur le chemin, etc.

Toutes ces caractéristiques dépendent fortement de l’algorithme considéré. D’une part par le nombre de sauts utilisé, mais aussi par les longueurs des arêtes. En effet, on remarque de manière générale que la longueur moyenne d’une arête sur les chemins est inférieure à la longueur typique d’une arête du graphe de Delaunay. Ceci est dû au fait que l’algorithme de routage introduit généralement un biais, en privilégiant les arêtes courtes...

Chemin Markovien.

Définition B.4 *Étant donné les deux points quelconques du plan s et d , définissons Δ le segment de droite $[s, d]$. Le chemin passant par les centres des cellules rencontrées par Δ est appelé “chemin markovien” sur le graphe de Delaunay. Le premier point correspond au centre de la cellule contenant s et le dernier point est le centre de la cellule contenant d .*

Cette dénomination en effet due à des propriétés markoviennes de ce chemin : Si $X_0 = s, X_1, \dots, X_n = d$ sont les noeuds traversés, alors le couple de points (X_i, X_{i+1}) ne dépend que de (X_{i-1}, X_i) (et pas des segments précédents (X_j, X_{j+1}) pour $j < i - 1$). Voir [89] pour plus de précisions. Une représentation visuelle d’un tel chemin est donnée dans la figure B.6.

Le nombre moyen de sauts de cet algorithme de routage est donné par le célèbre théorème de Møller (donné sans démonstration) :

Théorème B.1 (Møller [98]) *Soit Θ un pavage de Poisson-Voronoi. Soit $\pi(\Theta)$ le processus des centres des cellules, d’intensité λ . Soit $N_\Delta(\Theta)$ le processus ponctuel sur \mathbb{R} défini comme l’intersection de Θ avec une droite arbitraire Δ . $N_\Delta(\Theta)$ hérite des propriétés de stationnarité de Θ — En revanche, le processus N_Δ n’est pas un processus de Poisson sur \mathbb{R} . Son intensité λ_Δ peut de plus s’écrire :*

$$\lambda_\Delta = \frac{4}{\pi} \sqrt{\lambda}$$

Corollaire B.6 *Le nombre moyen de sauts du chemin markovien reliant les points s et d est donnée par la formule $\mathbb{E}[\|\text{path}(s, d, \pi)\|] = \frac{4}{\pi} \sqrt{\lambda} \|d - s\|$.*

Théorème B.2 (Baccelli, Tchoumatchenko, Zuyev [89]) *La longueur moyenne du chemin markovien reliant les points s et d obéit à la relation suivante :*

$$\frac{\mathbb{E}[\|\text{path}(s, d, \pi)\|]}{\|d - s\|} \sim \frac{4}{\pi} \text{ pour } \|d - s\| \rightarrow \infty$$

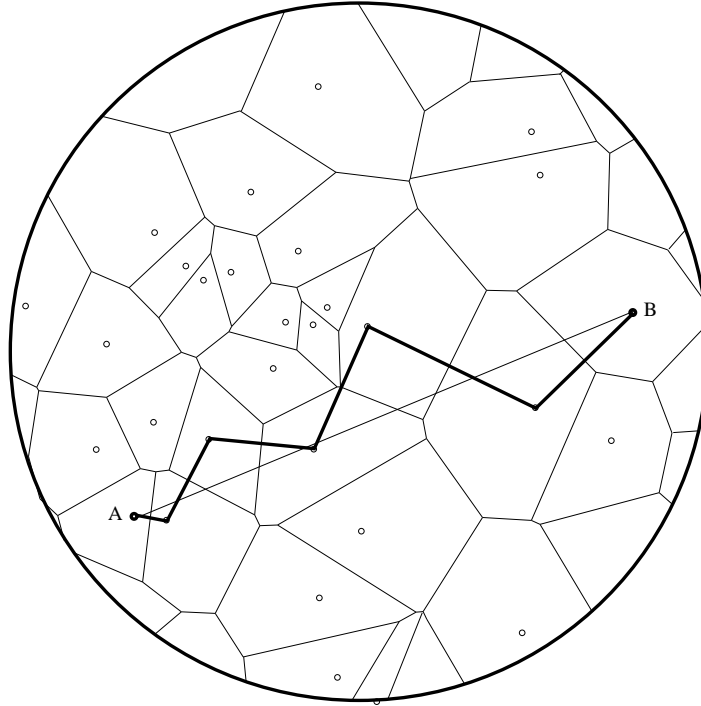


FIG. B.6 – Exemple de chemin markovien sur le graphe de Delaunay.

Caractéristiques typiques sur les chemins markoviens.

Définition B.5 *Par analogie avec la probabilité de Palm définie dans le chapitre précédent, on peut définir une moyenne sur les points du chemin markovien $\text{path}(s, d, \pi)$ de cette manière :*

$$\mathbb{E}_0^\Delta[f] = \frac{1}{\lambda_\Delta \|d - s\|} \mathbb{E} \left[\sum_{x \in \text{path}(s, d, \pi)} f \circ \phi_x \right]$$

pour tout fonction variable aléatoire f .

Remarque B.2 Si l'espace de probabilité est ergodique, et si $f : \mathbb{R}^2 \times \Omega \rightarrow \mathbb{R}^+$ est compatible avec le shift, alors il est possible d'écrire :

$$\mathbb{E}_0^\Delta[f] = \lim_{\|d-s\| \rightarrow \infty} \frac{1}{|\text{path}(s, t, \pi)|} \sum_{x \in \text{path}(s, d, \pi)} f(x)$$

La façon dont sont choisies les cellules (c.à.d. celles qui sont coupées par la droite Δ) introduit un biais statistique – de manière similaire au biais constaté dans la remarque B.1. Par exemple, la taille moyenne des cellules traversées est plus grande (avec une variance plus faible) que celle d'une cellule "typique", comme le montre la proposition suivante, que nous admettrons.

Proposition B.12 *Les identités suivantes ont été constatées par simulation :*

$$\mathbb{E}_0^\Delta[\mathcal{L}(C_0(\pi))] = \delta_1 \frac{1}{\lambda} \quad (\text{B.17})$$

$$\mathbb{E}_0^\Delta[(\mathcal{L}(C_0(\pi)))^2] = \delta_2 \frac{1}{\lambda^2} \quad (\text{B.18})$$

avec $\delta_1 \approx 1.12$ et $\delta_2 \approx 1.62$.

* *
*

Annexe C

Quelques distributions

Ce chapitre présente brièvement quelques distributions utilisées dans ce rapport, ainsi que quelques unes de leurs propriétés. La plupart des propositions annoncées sont des résultats élémentaires issus de la statistique ou de la combinatoire.

C.1 Statistique d'Ordre

C.1.1 Introduction

Soit X une variable aléatoire continue sur un espace arbitraire E et soit $P(x) = \mathbb{P}\{X < x\}$ pour tout $x \in E$.

Définition C.1 (Quantile) *Si $x \mapsto P(x)$ est strictement croissante, alors l'équation $P(x) = p$ admet une unique solution X_p^* , qui est appelée le quantile d'ordre p de la variable aléatoire X .*

Définition C.2 (Statistique d'ordre) *Soit $\{X_i\}_{i \in \mathcal{I}}$ une famille de variables aléatoires (ou "échantillon") sur un espace arbitraire E de cardinal $N = |\mathcal{I}|$. Ces variables peuvent être ordonnées dans le sens croissant :*

$$X_{(1)} \leq \dots \leq X_{(i)} \leq \dots \leq X_{(N)}$$

où $X_{(i)}$ est appelé $i^{\text{ème}}$ statistique d'ordre de l'ensemble considéré.

Associé à cet échantillon, on peut définir une distribution empirique $P_N(x)$ en posant pour tout $x \in E$,

$$P_N(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{X_i < x\}}$$

Définition C.3 (Quantile d'un échantillon) *Étant donné $\beta \in [0, 1[$, posons $m = \lfloor \beta N \rfloor + 1$. La variable aléatoire $X_{(m)}$ est appelée le β -quantile de l'échantillon.*

Remarque C.1 Ce nom de *quantile* est justifiée car $X_{(m)}$ correspond asymptotiquement (lorsque $N \rightarrow \infty$) au quantile d'ordre β de l'échantillon par rapport à la distribution empirique P_N . En effet,

$$P_N(X_{(m)}) = \frac{\lfloor \beta N \rfloor + 1}{N} \rightarrow \beta \text{ pour } N \rightarrow \infty$$

Cette remarque met également en évidence l'interprétation physique de cette grandeur : Une fraction proche de β des variables $\{X_i\}$ sont inférieures à $X_{(m)}$.

C.1.2 Propriétés

Distribution.

Proposition C.1 *Si les variables $\{X_i\}_{i \in \mathcal{I}}$ sont indépendantes et identiquement distribuées (i.i.d.), la propriété suivante est vérifiée :*

$$p_1 = \sum_{s=0}^{m-1} \binom{N}{s} p_0^{N-s} (1-p_0)^s$$

si $N = |\mathcal{I}|$, $p_0 = \mathbb{P}\{X_i > b\}$ (pour tout $i \in \mathcal{I}$) et $p_1 = \mathbb{P}\{X_{(m)} > b\}$ pour un b donné.

Preuve. D'après les définitions précédentes,

$$\{X_{(m)} > b\} = \bigcup_{s=0}^{m-1} \Omega_s$$

où $\Omega_s = \{X_{(s)}^0 \leq b < X_{(s+1)}^0\}$ pour tout $0 < s < m$ et $\Omega_0 = \{b < X_{(1)}^0\}$. Les variables étant supposées i.i.d., Ω_s suit une loi binomiale :

$$\mathbb{P}(\Omega_s) = \binom{N}{s} p_0^{N-s} (1-p_0)^s$$

Il suffit alors de remarquer que $p_1 = \sum_{s=0}^{m-1} \mathbb{P}(\Omega_s)$ pour arriver au résultat annoncé. \square

Asymptote.

Soit X^* le quantile d'ordre β des $\{X_i\}_{i \in \mathcal{I}}$ par rapport à leur loi P — c.à.d. vérifiant $P(X^*) = \mathbb{P}\{X < X^*\} = \beta$. Cette quantité apparaît dans la distribution asymptotique de $X_{(m)}$, comme le montre la proposition suivante :

Proposition C.2 Soit U la variable aléatoire définie par

$$U = \sqrt{N} \frac{X_{(m)} - X^*}{\sqrt{\beta(1-\beta)}} p(X^*)$$

où p est la distribution des variables X_i . Lorsque $N \rightarrow \infty$, $U \sim N(0, 1)$.

Preuve. Il s'agit en fait d'une conséquence (non triviale) du théorème centrale limite, $P_N(X^*)$ pouvant être vu comme une somme de N variables de Bernouilli de moyenne β . Voir [101] par exemple pour une démonstration. \square

Remarque C.2 Ce résultat asymptotique permet de mettre en évidence deux propriétés intéressantes des quantiles d'échantillons :

- La variance de $X_{(m)}$ diminue (en $\frac{1}{\sqrt{N}}$) avec la taille de l'ensemble considéré, lorsque N est grand.
- En reprenant les notations de la proposition C.1,

$$p_1 \doteq \mathbb{P}\{X_{(m)} > b\} \approx \begin{cases} 0 & \text{pour } X^* < b \\ 1 & \text{pour } X^* > b \end{cases}$$

lorsque $N \rightarrow \infty$.

C.2 Quelques Lois élémentaires

C.2.1 Expression Multinomiale

Définition C.4 Soit $N \in \mathbb{N}$ le nombre total d'objets considérés — par exemple, les liens du réseau dans le chapitre 7. Considérons l ensembles contenant respectivement k_0, k_1, \dots, k_{l-1} objets, avec $k_0 + k_1 + \dots + k_{l-1} < N$. Le nombre total de combinaisons possibles est donnée par :

$$\binom{N}{k_0 \dots k_{l-1}}_l = \frac{N!}{k_0! \dots k_{l-1}! (N - k_0 - \dots - k_{l-1})!}$$

Lorsqu'aucune confusion n'est possible, nous pourrions omettre l'index l , dont le rôle est de rappeler le nombre d'ensembles considérés. De plus, nous poserons $\binom{N}{k_0 \dots k_{l-1}}_l = 0$ lorsque $\sum_{i=0}^{l-1} k_i > N$.

Les l ensembles peuvent être partitions en deux; on obtient alors la propriété suivante :

Proposition C.3 Pour tout $m = 1, \dots, l-1$,

$$\binom{N}{k_0 \dots k_{l-1}} = \binom{N}{k_0 \dots k_{m-1}} \binom{N - \sum_{i=0}^{m-1} k_i}{k_m \dots k_{l-1}}$$

C.2.2 Loi Hypergéométrique

Définition.

Définition C.5 Soit N le nombre total d'objets considérés — par exemple, les liens d'un réseau dans le chapitre 7. Imaginons un scénario quelque peu manichéenne où, parmi tous ces objets, certains d'entre eux sont "mauvais" (par exemple, un lien défectueux) et certains autres sont "bons". Soit s le nombre de "mauvais" objets.

Considérons à présent un ensemble de M objets. La probabilité que k objets (parmi M) soient "mauvais" est donnée par l'expression hypergéométrique suivante :

$$\frac{\binom{N-s}{M-k} \binom{s}{k}}{\binom{N}{M}}$$

Cette distribution peut être généralisée pour l ensembles distincts de taille M_0, \dots, M_{l-1} .

Définition C.6 La probabilité qu'il y ait k_0 "mauvais" objets parmi le premier ensemble, k_1 dans le second, \dots , k_{l-1} dans le dernier, est donnée par l'expression :

$$\frac{\binom{N-s}{M_0-k_0 \dots M_{l-1}-k_{l-1}} \binom{s}{k_0 \dots k_{l-1}}}{\binom{N}{M_0 \dots M_{l-1}}}$$

Propriétés.

Proposition C.4 Pour tout $m = 1, \dots, l$,

$$\frac{\binom{N-s}{M_0-k_0 \dots M_{l-1}-k_{l-1}} \binom{s}{k_0 \dots k_{l-1}}}{\binom{N}{M_0 \dots M_{l-1}}} = \frac{\binom{N-s}{M_0-k_0 \dots M_{m-1}-k_{m-1}} \binom{s}{k_0 \dots k_{m-1}}}{\binom{N}{M_0 \dots M_{m-1}}_m} \frac{\binom{N-s-\sum_{i=0}^{m-1} (M_i-k_i)}{M_m-k_m \dots M_{l-1}-k_{l-1}} \binom{s-\sum_{i=0}^{m-1} k_i}{k_m \dots k_{l-1}}}{\binom{N-\sum_{i=0}^{m-1} M_i}{M_m \dots M_{l-1}}}$$

Preuve. Il s'agit simplement d'une réécriture de la proposition C.3. □

Proposition C.5

$$\sum_{k_0=0}^{M_0} \frac{\binom{N-s}{M_0-k_0 \dots M_{l-1}-k_{l-1}} \binom{s}{k_0 \dots k_{l-1}}}{\binom{N}{M_0 \dots M_{l-1}}} = \frac{\binom{N-s}{M_0-k_1 \dots M_{l-1}-k_{l-1}} \binom{s}{k_1 \dots k_{l-1}}}{\binom{N}{M_1 \dots M_{l-1}}}$$

Preuve. Cette identité est une application directe de la proposition C.4 pour $m = l - 2$. \square

C.3 Blocage dans un domaine de routage

Dans cette section, nous utilisons les propriétés sur les lois élémentaire décrites ci-avant, pour déterminer quelques grandeurs utiles à l'analyse des performances d'un réseau hiérarchique, dans le chapitre 7.

Dans la suite, nous considérerons un domaine de routage arbitraire, représenté par la cellule de Voronoi $C_u(\pi_1)$ (voir paragraphe 7.1 pour une description précise du modèle utilisé). Soit L_0 le nombre de liens présents dans la cellule et soit M_0 le nombre de "sauts" nécessaires à la traversée du domaine considéré. Nous noterons également par K_0 le nombre de routes précalculées permettant de traverser l'aire de routage (représentée par la cellule de Voronoi) dans une direction appropriée.

C.3.1 Sélection d'un domaine

Comme expliqué dans le chapitre 7, une aire de routage n'est traversée que si la bande passante agrégée qui la représente est suffisante (fonction GCAC dans le contexte PNNI). Plus précisément, un domaine u n'est sélectionné que sous la condition : $X_u^1 > b$ où b est la bande passante requise par l'utilisateur et X_u^1 représente la bande passante agrégée pour le domaine u . Cette valeur X^1 est choisie comme le β -quantile (cf. paragraphe C.1) sur les bandes passantes \tilde{X}_i^0 des liens i du domaine (supposés i.i.d).

Théorème C.1 *Si b correspond à la bande passante requise et notant $m_0 = \lfloor \beta L_0 \rfloor + 1$, la probabilité que la cellule $C_u(\pi_1)$ soit sélectionnée est donné par l'expression suivante :*

$$p_1 = \sum_{s=0}^{m_0-1} \binom{L_0}{s} p_0^{L_0-s} (1-p_0)^s$$

avec $p_1 = \mathbb{P}\{X^1 > b\}$ et $p_0 = \mathbb{P}\{\tilde{X}^0 > b\}$.

Preuve. Cette identité découle directement de la proposition C.1 de la section C.1 sur l'ensemble des bandes passante estimées $\{\tilde{X}_i^0\}_{i \in C_u(\pi_1)}$. \square

C.3.2 Traversée d'un domaine sélectionné

Nous considérons à présent que la cellule étudiée a été sélectionnée, c'est à dire que $X^1 > b$. Ceci veut dire que, a priori (en fonction des informations agrégées dont dispose le noeud

source), le domaine devrait disposer d'assez de ressources pour supporter le flux entrant (resp. la nouvelle connexion). Nous analysons dans la suite si il est effectivement possible de traverser le domaine.

Routes admissibles.

Tout d'abord, le noeud d'entrée dans le domaine doit déterminer si les routes précalculées menant au prochain domaine sont admissibles, c.à.d. si ces routes disposent à priori d'assez de bande passante. K_0 désigne dans la suite le nombre de routes précalculées.

Notons par R_0 le nombre de routes admissibles pour traverser le domaine et Υ_r l'événement conditionnel $\{R_0 = r \mid X^1 > b\}$.

Lemme C.1 *Connaissant Ω_s (introduit dans la proposition C.1), l'identité suivante est vérifiée pour tout $r = 0, \dots, K_0$:*

$$\mathbb{P}(R_0 = r \mid \Omega_s) = \sum_{i=0}^{K_0-r} (-1)^i \frac{\binom{K_0}{i,r} \binom{L-s}{iM,rM}}{\binom{L}{iM,rM}}$$

Preuve. Par définition, l'événement $\{R_0 = r\}$ correspond au cas où il n'y a aucun lien bloquant sur r routes and où il y a au moins un lien bloquant sur les $K_0 - r$ routes restantes. D'où l'expression :

$$\mathbb{P}(R_0 = K_0 \mid \Omega_s) = \frac{\binom{L-s}{M,\dots,M}_{K_0}}{\binom{L}{M,\dots,M}_{K_0}} = \frac{\binom{L-s}{K_0 M}}{\binom{L}{K_0 M}} \quad (\text{C.1})$$

$$\mathbb{P}(R_0 = r \mid \Omega_s) = \binom{K_0}{r} \sum_{h_0 \geq 1} \dots \sum_{h_{K_0-r-1} \geq 1} \frac{\binom{L-s}{M-h_0,\dots,M-h_{K_0-r-1},M,\dots,M} \binom{s}{h_0,\dots,h_{K_0-r-1},0,\dots,0}}{\binom{L}{M,\dots,M}} \quad (\text{C.2})$$

En utilisant la proposition C.4, la dernière expression peut être réécrite :

$$\mathbb{P}(R_0 = r \mid \Omega_s) = \binom{K_0}{r} \frac{\binom{L-s}{rM}}{\binom{L}{rM}} \sum_{h_0=1}^M \dots \sum_{h_{K_0-r-1}=1}^M \Xi_{K_0-r}(L-rM)$$

avec

$$\Xi_m(N) = \frac{\binom{N-s}{M-h_0,\dots,M-h_{m-1}} \binom{s}{h_0,\dots,h_{m-1}}}{\binom{N}{M,\dots,M}}$$

pour tout $m = 0, \dots, K_0$. Une application directe des propositions C.5 et C.4 (à la section précédente) permet alors d'arriver à la relation suivante :

$$\sum_{h_{m-1}=1}^M \Xi_m(N) = \Xi_{m-1}(N) - \frac{\binom{N-s}{M}}{\binom{N}{M}} \Xi_{m-1}(N-M)$$

qui, par récurrence, devient :

$$\sum_{h_0=1}^M \dots \sum_{h_{m-1}=1}^M \Xi_m(N) = \sum_{j=0}^m \binom{N}{i} (-1)^i \frac{\binom{N-s}{iM}}{\binom{N}{iM}}$$

Le résultat annoncé en découle alors directement. \square

Remarque C.3 Les simplifications des expressions (C.1) et (C.2) ne sont pas nécessairement d'une grande utilité à cause des problèmes numériques que peuvent causer ces séries alternées. Néanmoins, l'ensemble des routes précalculées est de taille très limitée ($K = 3$ en général dans nos analyses) de sorte que nous n'avons jamais rencontrés de problèmes. Il faut noter que de tout façon, K doit être faible par soucis de réalisme (les noeuds ne peuvent stocker trop de routes vers chaque destination) et par cohérence vis à vis de notre modèle : Nous avons supposés que les K chemins étaient disjoints.

Théorème C.2 *En utilisant la proposition C.1 et le lemme C.1, il est possible d'obtenir la distribution de R :*

$$\mathbb{P}(\Upsilon_r) = \frac{1}{p_1} \sum_{i=0}^{K_0-r} (-1)^i \binom{K_0}{i, r} \sum_{s=0}^{m-1} \binom{L - (i+r)M_0}{s} p_0^{L_0-s} (1-p_0)^s$$

Remarque C.4 Pour $\beta \rightarrow 1$, c.à.d. pour une politique d'agrégation très "agressive" R suit une loi proche de la loi binomiale binomiale :

$$\begin{aligned} \mathbb{P}(\Upsilon_r) &= \frac{1}{p_1} \binom{K_0}{r} P^r (1-P)^{K_0-r} \quad \text{pour } r > 0 \\ \mathbb{P}(\Upsilon_0) &= \frac{1}{p_1} \left[(1-P)^{K_0} - 1 \right] \end{aligned}$$

avec $P = p_0^{M_0}$.

Lorsque $p \approx 1$, où lorsque $p > 0.5$ et $L_0 \gg 1$, $p_1 = 1 - (1-p_0)^{L_0} \rightarrow 1$ et R_0 suit alors une loi binomiale, comme c'est le cas pour l'algorithme de routage "aveugle" (voir 7.2.6).

Preuve. Pour $m = L_0 - 1$, nous pouvons écrire pour tout $r > 0$:

$$\begin{aligned} \mathbb{P}(\Upsilon_r) &= \sum_{i=0}^{K_0-r} (-1)^i \binom{K_0}{i, r} \frac{1}{p_1} \sum_{s=0}^{L_0-(i+r)M_0} \binom{L_0 - (i+r)M_0}{s} p_0^{L_0-s} (1-p_0)^s \\ &= \frac{1}{p_1} \sum_{i=0}^{K_0-r} (-1)^i \binom{K_0}{i, r} p_0^{(i+r)M_0} \\ &= \frac{1}{p_1} \binom{K_0}{r} p_0^{rM_0} \sum_{i=0}^{K_0-r} (-1)^i \binom{K_0-r}{i} p_0^{iM_0} \\ &= \frac{1}{p_1} \binom{K_0}{r} p_0^{rM_0} (1-p_0^{M_0})^{K_0-r} \end{aligned}$$

Pour le cas particulier de Υ_0 , le raisonnement est le même, mais il faut prendre en compte le fait que m_0 ne peut dépasser $L_0 - 1$. On arrive alors à l'expression annoncée. \square

Corollaire C.1 *La grandeur $\mathcal{N}_c^b = \frac{1}{1-p_{0|1}} \sum_{r=0}^{K_0} (r \wedge K_c^0) (1-q_0)^{r \wedge K_c^0} \mathbb{P}(\Upsilon_r)$ est aisément calculées d'après les deux théorèmes précédents et le théorème C.3. Cette grandeur correspond au nombre moyen de crankbacks subis avant que la connexion ne soit finalement rejetée (voir chapitre 7).*

Corollaire C.2 *La connexion est directement rejetée à l'entrée dans le domaine sélectionné avec la probabilité $p_{0|1}^* = \frac{p_{0,1}^*}{p_1}$ où p_1 est donnée dans le théorème C.1 et où*

$$p_{0,1}^* = \mathbb{P}(\Upsilon_0) + \sum_{r>0} \mathbb{P}(\Upsilon_r) (1-q_0)^r$$

peut être aisément calculé avec les expressions précédentes.

Preuve. La connexion est rejetée par le noeud d'entrée si

- Aucune route admissible n'a pu être trouvée, avec la probabilité $\mathbb{P}(\Upsilon_0)$.
- Il existe bien au moins une route admissible, mais la fonction CAC bloque la connexion car le premier lien de ces routes n'a pas assez de ressources.

\square

Traversée d'un domaine.

Théorème C.3 *Soit K_c^0 le nombre maximum de crankbacks tolérés à la traversée d'un domaine (avec bien sûr $K_c^0 \leq K_0$). La probabilité de passer à travers le domaine est, conditionnellement au fait que ce domaine ait été sélectionné,*

$$p_{0|1} = 1 - \sum_{r=0}^{K_0} (1 - q_0^{M_0})^{r \wedge K_c^0} \mathbb{P}(\Upsilon_r)$$

où nous avons utilisés les notations du théorème C.2 et $q_0 = \mathbb{P}\{X_0 > b \mid \tilde{X}^0 > b\}$.

Preuve. Il suffit de remarquer que la connexion est acceptée sur une route admissible avec la probabilité $q_0^{M_0}$ et utiliser ensuite les théorèmes précédents sur le nombre de routes admissibles pour conclure. \square

Bibliographie

Publications personnelles

- [1] J.L.Rougier, A.Ragozini, A.Gravey, D.Kofman. *A model for evaluating the impact of aggregated routing information on network performance*. Actes de IFIP BC'99, Honk-Hong, Novembre 1999.
- [2] A.Ragozini, J.L.Rougier, A.Gravey, D.Kofman. *Analysis of the Performance of a Hierarchical PNNI Network*. Actes de IEEE ICATM'99, Colmar (France), Juin 1999.
- [3] F.Baccelli, D.Kofman, J.L.Rougier. *Self-Organizing Hierarchical Multicast Trees and their Optimization*. Actes de IEEE Infocom'99, New-York (USA), Mars 1999.
- [4] J.L.Rougier, D.Kofman, A.Gravey. *Optimization of Hierarchical Routing Protocols*. Actes de IFIP ATM'98, Ilkley (GB), Juillet 1998. à paraître dans la revue Telecommunication Systems.
- [5] Sh.Barakat, J.L.Rougier, D.Kofman. *Optimization of Hierarchical Multicast trees in ATM Networks*. Actes de IFIP ATM'98, Ilkley (GB), Juillet 1998.
- [6] J.L.Rougier, D.Kofman, A.Gravey. *Traffic Management for Connectionless Traffic over ATM*. Actes de IFIP WATM'97, Montreal (Canada), Octobre 1997. à paraître dans la revue Telecommunication Systems.

Rapports de Stages

- [7] Shadi Barakat. *Le routage Multicast Hiérarchique*. Rapport de stage de DEA. DEA Réseau et Système Distribué de l'université de Nice Sophia-Antipolis. Juillet 1998.
- [8] Yann Vasseur. *Optimisation du routage PNNI dans les réseaux ATM*. Rapport de stage de fin d'étude de l'école Polytechnique (Promotion X-96). Juillet 1999.

Routage Multicast

- [9] T.Ballardie, P.Francis, J.Crowcroft. *Core Based Trees*. Actes de ACM Sigcomm'93, San Francisco, Septembre 1993.
- [10] S.Deering et al, *The PIM Architecture for Wide Area Multicast Routing*, IEEE/ACM Transactions on Networking, Vol.4, No.2, Avril 1996.
- [11] M.Doar, I.Leslie. *How bad is naive multicast routing ?*. Actes de IEEE Infocom'93. 1993.
- [12] M.J.Donahoo, E.W.Zegura. *Core Migration for Dynamic Multicast Routing*. Actes de IEEE ICCCN'96, 1996.
- [13] D.Estrin, M.Handley, A.Helmy, P.Huang, D.Thaler. *A Dynamic Bootstrap Mechanism for Rendez-vous-based Multicast Routing*. Actes de IEEE Infocom 1999. New-York, Mars 1999.
- [14] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1) :1-29, Janvier 1968.
- [15] M.Handley, J.Crowcroft, I. Wakeman. *Hierarchical Protocol Independent Multicast (HPIM)*. non publié. University College London, Novembre 1995.
- [16] L.Kou, G.Markowsky, and L.Berman. *A fast algorithm for Steiner trees*. Acta Informatica, 15 :141-145, 1981.
- [17] S.Kumar et al. *The MASC/BGMP Architecture for Inter-domain Multicast Routing*. Actes de ACM Sigcom 98, Vancouver, Septembre 1998.
- [18] A.Meddeb, A.Girard, C.Rosenberg. *The impact of tree selection on the design of networks with multipoint connections*. Proc. IEEE GLOBECOM'97, Phoenix, Novembre 1997.
- [19] S.Pejhan, A.Eleftheriadis, D.Anastassiou. *Distributed Multicast Address Management in the Global Internet*. IEEE JSAC, Octobre 1995.
- [20] G.N. Rouskas, I.Baldine. *Multicast routing with end-to-end delay and delay variation constraints*. IEEE JSAC, 15(3) :346-356, Avril 1997.
- [21] C.Shields,J.J. Garcia-Luna-Aceves. *The Ordered Core Based Tree Protocol*. Actes de IEEE INFOCOM'97, Avril 1997.
- [22] C. Shields, J.J. Garcia-Luna-Aceves, *The HIP Protocol for Hierarchical Multicast Routing*, Actes de ACM PODC'98, Mexico, Juillet 1998.

- [23] R.Venkateswaran, C.S.Raghavendra, X.Chen, V.Kumar. *Hierarchical Multicast Routing in ATM Networks*. Actes de IEEE ICC'96, Juin 1996.
- [24] R.Venkateswaran, C.S.Raghavendra, X.Chen, V.Kumar. *A scalable, Dynamic Multicast Routing Algorithm in ATM Networks*. Actes ICC'97. 1997.
- [25] B.M.Waxman. *Routing of Multipoint connections*. IEEE J.S.A.C., Vol. 6, No. 9, Décembre 1988.
- [26] L.Weï, D.Estrin. *The trade-offs of multicast trees and algorithms*. Actes de IEEE ICCCN'94, Aout 1994.
- [27] L.Weï, D.Estrin. *Multicast Routing in Dense and Sparse Modes : Simulation Study of Tradeoffs and Dynamics*. Actes de IEEE ICCCN'95. 1995.
- [28] E.W.Zegura, K.Calvert, M.J.Donahoo. *A Quantitative Comparison of Graph-based Models for Internet Topology*. IEEE/ACM Transactions on Networking, Vol.5, No.6, Décembre 1997.
- [29] J.C.Konig. *Arbres de Steiner*. Présentation invitée à Algotel'99 (non publiée), Roscoff, Juin 1999.

Routage Dynamique

- [30] G.Apostolopoulos, S.K.Tripathi, *On the Effectiveness of Path Pre-computation in Reducing the Processing Cost of On-demand QoS Path Computation*. Actes de IEEE SCN'98, Athènes, Juin 1998.
- [31] G.Apostolopoulos, R.Guerin, S.Kamat, S.Tripathi. *Quality of Service Based Routing : A performance Perspective*. Actes de ACM SIGCOMM'98, Vancouver, Septembre 1998.
- [32] G.Apostolopoulos, R.Guerin, S.Kamat, S.K.Tripathi. *Improving QoS Routing Performance Under Inaccurate Link State Information*. Actes de ITC-16, Endimbourg. 1999.
- [33] J.Chen, P.Druschel, D.Subramanian. *A New Approach to Routing With Dynamic Metrics.*, Actes de IEEE INFOCOM'99, New-York, Mars 1999.
- [34] D.Bertsekas, R.Gallager. *Data Networks*. Prentice-Hall Englewood Cliffs (New Jersey), 1987.
- [35] J.J.Garcia-Luna-Aceves. *A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States*. Actes de ACM Sigcomm'89, Septembre 1989.

- [36] A. Girard. *Routing and Dimensioning in Circuit-switched Networks*. Addison-Wesley, 1990.
- [37] R.Guérin, A.Orda. *QoS-based Routing in Networks with Inaccurate Information : theory and Algorithms*. Actes de IEEE Infocom'97, 1997.
- [38] R.Guérin, A.Orda, D.Williams. *QoS Routing Mechanisms and OSPF Extensions*. Actes de IEEE 2nd Global Internet mini-conference, Phoenix, Novembre 1997.
- [39] R.Guérin, S.Kamat, S.Herzog. *QoS path management with RSVP*. Actes de IEEE 2ng Global Internet mini-conference, Phoenix, Novembre 1997.
- [40] F.Hao, E.W.Zegura, S.Bhatt. *Performance of the PNNI Protocol in Large Networks*. Actes de IEEE ATM'98, 1998.
- [41] F. Hao¹ and E. W. Zegura, *On Scalable QoS Routing : Performance Evaluation of Topology Aggregation*. Rapport de Recherche GIT-CC-99-04, College of Computing, Georgia Tech., 1999.
- [42] C.Huitema. *Routing in the Internet*². Prentice Hall Englewood Cliffs (New Jersey), 1995.
- [43] A.Iwata, R.Izmailov et al. *QoS Aggregation Algorithms in Hierarchical ATM Networks*. Actes de IEEE ICC'98, Atlanta, Juin 1998.
- [44] A.Iwata, R.Izmailov et al. *Routing Algorithms for ATM Networks with multiple QoS requirements*. Actes de IFIP WATM'95, Paris, Décembre 1995.
- [45] A.Iwata, R.Izmailov, B.Sengupta. *Alternative Routing Methods for PNNI Networks with Partially Disjoint Paths*. Actes de IEEE Globecom'98, Sidney. 1998.
- [46] L.Kleinrock, F.Kamoun. *Hierarchical Routing for Large Networks*. Computer Networks, vol.1. 1977.
- [47] W.C.Lee. *Topology Aggregation for Hierarchical Routing in ATM Networks*. ACM Computer Communication Review, vol.25, Avril 1995.
- [48] W.C.Lee. *Minimum Equivalent Subspanner Algorithms for Topology Aggregation in ATM Networks*. Actes IEEE ICATM'99, Colmar, Juin 1999.
- [49] Q.Ma. *Quality of Service Routing in Integrated Service Networks*. Thèse de doctorat, Carnegie Mellon University, Pittsburg. Janvier 1998.
- [50] Q.Ma, P.Stenkiste. *On path selection for traffic with bandwidth guarantees*. Actes de IEEE ICNP'97, Atlanta, Octobre 1997.

¹Rapports de recherches et publications disponibles sur la page oueb <http://www.cc.gatech.edu/people/home/fang/>.

²Une version française, nommée "Le routage dans l'Internet" est également disponible.

- [51] Q.Ma, P.Stenkiste. *Supporting Dynamic Inter-Class Resource Sharing : A Multi-Class QoS Routing Algorithm* . Actes de IEEE INFOCOM'99, New York, Mars 1999.
- [52] A.Magi, A.Szenteri, B.Szviatowski. *analysis of link cost functions for PNNI routing* . Actes de IFIP ATM'98, Juillet 1998.
- [53] P.Van Mieghem. *Estimation of an Optimal PNNI Topology*. Actes de IEEE ATM'97, Lisbonne. 1997.
- [54] P.Van Mieghem. *Routing in a Hierarchical Structure* . Actes de IEEE ICATM'98, Colmar. Juin 1998.
- [55] P.Van Mieghem. *Topology Information Condensation in hierarchical Networks*. à paraître dans Computer Networks and ISDN Systems. 1999.
- [56] E.Oubagha, A.Simonian. *Etudes de performance du protocole PNNI*. Rapport de recherche France Telecom CNET/DAC/GTR 16.98. (diffusion restreinte). Juin 1998.
- [57] A.Shaikh, J.Rexford, K.G.Shin. *Efficient Precomputation of Quality-of-Service Routes* . Actes de IEEE NOSSDAV'98, Juillet 1998.
- [58] A.Shaikh, J.Rexford, K.G.Shin. *Evaluation of Overheads of Source Directed Quality of Service Routing*. Actes de IEEE ICNP98, Octobre 1998.
- [59] A.Shaikh, J.Rexford, K.G.Shin. *Load-Sensitive Routing of Long-Lived IP Flows* . Actes de ACM SIGCOMM'99, Septembre 1999.
- [60] Z.Wang, J.Crowcroft. *Quality of Service Routing for supporting Multimedia Applications*. IEEE JSAC 14(7) :1228-1234. 1996.
- [61] E.W.Zegura, K.Calvert, M.J.Donahoo. *A Quantitative Comparison of Graph-based Models for Internet Topology*. IEEE/ACM Transactions on Networking, Vol.5, No.6, Décembre 1997.

Normes, RFCs³ et documents de travail.

- [62] G.Apostolopoulos, R.Guerin, S.Kamat, A.Orda, T.Przygienda, D.Williams. *QoS Routing Mechanisms and OSPF Extensions*. Travaux en cours (draft-guerin-qos-routing-ospf-05). Avril 1998.
- [63] The ATM Forum⁴. *Private Network-Network Interface Specification, version 1* . document : af-pnni-0055.000. Mars 1996.

³RFC : Request for Comment.

⁴Les spécifications de cette institutions sont disponibles à l'adresse internet <http://www.atmforum.com/> .

- [64] The ATM Forum. *Private Network-Network Interface Specification version 2*. Travaux en cours. Septembre 1997.
- [65] The ATM Forum. *User Network Interface, version 4.0*. document : af-sig-0061.000. Juillet 1996.
- [66] T.Ballardie, B.Cain, and Z.Zhang. *Core Based Trees (CBT version 3) Multicast Routing Protocol Specification*. TRavaux en cours (Groupe IDMR de l'IETF), Mars 1998,
- [67] T.Bates, R.Chandra, D.Katz, Y.Rekhter. *Multiprotocol Extensions for BGP-4*. Travaux en cours (groupe IDR de l'IETF). Septembre 1999.
- [68] B.Braden et. al. *Resource Reservation Protocol (RSVP) - Version 1 Functional Specification*, RFC 2205, Septembre 1997.
- [69] R.Braudes, S.Zabele. *Requirements for Multicast Protocols*. RFC-1458, Mai 1993.
- [70] Cisco Systems. *Multicast BGP*⁵. Juin 1999.
- [71] D.Estrin, M.Handley, S.Kumar, D.Thaler. *The multicast Address Set Claim (MASC) protocol*. travaux en cours (groupe IDMR de l'IETF). Novembre 1997.
- [72] D.Estrin et al. *Protocol Independent multicast (PIM), Dense Mode Protocol Specification*. Travaux en cours (groupe IDMR de l'IETF). 1999.
- [73] D.Estrin et al. *Protocol Independant Multicast — Sparse Mode Protocol Specification*. RFC 2362. 1998.
- [74] D.Estrin, T.Li, Y.Rekhter, K.Varadhan, D.Zappala. *Source Demand Routing : Packet Format and Forwarding Specification (Version 1)*. RFC 1940, Mai 1996.
- [75] W.Fenner. *Internet Group Management Protocol, Version 2*. RFC 2236. Novembre 1997.
- [76] M.Handley, V.Jacobson. *SDP : Session Description Protocol*. RFC 2327. Avril 1998.
- [77] M.Handley, H.Schulzrinne, E.Schooler, J.Rosenberg. *Session Initiation Protocol*. RFC 2543. Mars 1999.
- [78] M.Handley, C.Perkins, E.Whelan. *Session Announcement Protocol*. travaux en cours à l'IETF (groupe MMUSIC⁶). Aout 1999.
- [79] G. Malkin. *RIP Version 2 — Carrying Additional Information*. RFC 1723. Novembre 1994.
- [80] J.Moy. *Multicast Extensions to OSPF*. RFC 1584. Mars 1994.
- [81] J.Moy. *Open Shortest Path First, version 2*. RFC 2328. Avril 1998.

⁵URL : <http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/mbgp.htm>

⁶MMUSIC : <http://www.ietf.org/html.charters/mmusic-charter.html>.

- [82] Y.Rekhter, T.Li. *A Border Gateway Protocol 4 (BGP-4)*, RFC 1771, Mars 1995.
- [83] D.Thaler, D.Estrin, D.Meyer. *Border Gateway Multicast Protocol (BGMP)*. Travaux en cours (groupe IDMR de l'IETF). Mars 1998.
- [84] D.Thaler, B.Cain. *BGP attributes for multicast tree construction*. Travaux en cours (groupe IDMR de l'IETF). Février 1999.
- [85] D.Waitzman, C.Partridge, S.Deering, *Distance Vector Multicast Routing Protocol*, RFC 1075, Novembre 1988.

Processus Ponctuels et Géométrie Aléatoire

- [86] F.Baccelli, P.Bremaud. *Elements of Queueing Theory. Palm-Martingale Calculus and Stochastic Recurrences*. Springer Series in Applications of Mathematics, Springer-Verlag, 1994.
- [87] F. Baccelli, M. Klein, M. Lebourges and S. Zuyev, *Stochastic Geometry and Architecture of Communication Networks*⁷, Telecommunication Systems, 7, pp. 209-227, 1997.
- [88] F.Baccelli, M.Klein, S.Zuyev. *Perturbation Analysis of Functionals of Random Measures*. Rapport de Recherche INRIA RR2225, Mars 1994.
- [89] F.Baccelli, K.Tchoumatchenko, S.Zuyev, *Markov Paths on the Poisson-Delaunay Graph*. Rapport de Recherche INRIA RR3420. Mai 1998, à paraître dans *Advances in Applied Probability*.
- [90] F.Baccelli, S.Zuyev. *Poisson-Voronoi Spanning Trees with applications to the Optimization of Communication Networks*. Rapport de recherche INRIA RR3040. Novembre 1996.
- [91] D.Daley, D.Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer Series in Statistics, Springer-Verlag New-York, 1988.
- [92] L. Decreasefond. *Perturbation Analysis and Malliavin Calculus*. In *Annals of Applied Probability*, volume 8, number 2, 1997.
- [93] A.Frey, V.Schmidt, *Marked Point Processes in the Plane — Part I. A survey with applications to Spatial Modeling of Communication Networks*. *Adv. Perf. Anal.* 1, pp. 65-110, 1998.

⁷Une version française de cet article a été publiée dans la revue interne du CNET. Cette version est également disponible à l'INRIA (rapport de recherche RR2542).

- [94] A.Frey, V.Schmidt, *Marked Point Processes in the Plane — Part II. A survey with applications to Spatial Modeling of Communication Networks*. Adv. Perf. Anal. 2, pp. 171-214, 1998.
- [95] J.Neveu. *Systèmes Discrets*. Cours de DEA. (non publié).
- [96] J.Neveu. *Sur les mesures de Palm de deux processus ponctuels stationnaires*. Z. Wahrsch. verw. Gebiete, 34 :199-203, 1976.
- [97] J.Mecke. *Palm Methods for stationary random mosaics*. Combinatorial Principles in stochastic Geometry, Armenian Academy of Sciences Publ., Erevan, 1980.
- [98] J.Møller. *Random Tessellation in \mathbb{R}^d* . Adv. Appl. Prob. 21, 1989.
- [99] A.Okabe, B.Boots, K.Sugihara. *Spatial Tessellations. Concepts and Applications of Voronoi Diagrams*. J.Wiley & Sons, Chichester, 1992.
- [100] D.Stoyan, W.Kendall, J.Mecke. *Stochastic Geometry and its Applications*. 2nd Edition, J.Wiley & Sons Chichester, 1995.

Ouvrages et articles probabilistes divers

- [101] H.A.David. *Order Statistics*. Wiley Interscience, Second Edition, 1981.
- [102] R.J. Gibbens, F.P. Kelly, and P.B. Key. *Dynamic Alternative Routing*. Dans *Routing in Communication Networks* (Editeur : Martha Steenstrup). Prentice Hall, Englewood Cliffs, 1995.
- [103] Serge Halberstadt, D.Kofman *A dynamic bandwidth allocation mechanism for connectionless traffic over ATM networks*. Actes de ITC-15, Washington, 1997.
- [104] Serge Halberstadt. *Gestion de trafic pour des services en mode connecté sur un réseau ATM*. Thèse de doctorat de l'école nationale supérieure des Télécommunications, référence : ENST96 E039. 1996.
- [105] F.P.Kelly, S.Zachary, I.B.Ziedins. *Notes on effective bandwidths*. Stochastic Networks : Theory and Applications, Oxford University Press, 1996.

Index

– A –

Adressage	
Multicast	12–15
Agrégation	84, 86–89, 115–135, 139, 174
Agressivité	139, 151–156, 166
Anycast	14
Arbre	voir Multicast, voir aussi Multicast/Arbre
de concentration de Delaunay	65–67
AS	84
ATM	11, 13, 36

– B –

BGMP	35, 36
BGP	33, 84

– C –

Calcul de Palm	voir Palm/Calcul
CBT	27, 29, 35
CBT	voir aussi Multicast/Arbre
Cellule	203
Typique	42, 116, 129, 192
Voronoi	41, 116, 138, 208, 223
Centre d'un arbre	voir RP
Chemin Markovien	voir Routage/Markovien
CIDR	84
Cranckback	83, 90
Crankback	95–96, 100, 112, 141, 143, 153, 163

– D –

Dense voir aussi PIM, voir Mode/Dense
 Distance Vector voir Routage/Protocole
 DTL 89
 DVMRP 25–27

– E –

ECMP 77
 Epars voir aussi PIM, voir Mode/Epars
 Ethernet 13

– F –

Flot 184
 Frame-Relay 11, 13

– G –

GCAC 82, 95, 98, 108–110, 112, 139, 167, 174
 Graphe
 de Delaunay 44, 94, 117, 126, 138, 214–217

– H –

Hypergéométrique voir Loi

– I –

IGMP 15

– L –

Laplace voir Transformée de Laplace
 LIJ 36
 Link State voir Routage/Protocol
 Loi
 Hypergéométrique 222
 Multinomiale 221

– M –

M-Bone 29–32
 MASC 35, 36
 Mesure
 Ponctuelle 182

-
- Mode
- Dense 28, 50
 - Epars 28, 49
- Møller voir Théorème
- Mosaïque voir Pavage du Plan
- MOSPF 29
- MPLS 92
- Multicast 1, 9
- Adressage voir Adressage/Multicast
 - Arbre 10, 16–24
 - CBT 22–24
 - Hiérarchique 33–37, 39–64, 68
 - Partagé 17
 - Spécifique 17
 - SPT 18–19, 24
 - Steiner 19–22, 24
 - Routage voir Routage/Multicast
- Multimédia 1
- Multinomiale voir Loi
- N –
- Neveu voir Théorème
- O –
- OSPF 77
- OSPF
- Extensions pour la QoS voir QOSPF
 - Extensions pour le Multicast voir MOSPF
- P –
- Palm
- Calcul 191–199
 - Probabilité 41, 116, 191–194, 197
- Pavage
- Delaunay 214–217
 - du Plan 203–217
 - Poisson-Voronoi 41, 208–213

Perturbation	189
PG	85, 126
PIM	27–29
PIM	
Mode épars	27
Mode dense	27
PNNI	78, 79, 84, 93, 115, 126, 137
Poids Administratif	voir Routage/Paramètre
Point de Rendez-Vous	voir RP
Poisson	voir Processus/Poisson
Policy Routing	voir Routage/Politique
Processus	
Associé à un Pavage	204
Conjointement Stationnaires	196–201
Poisson	40, 116, 137, 185–190
Ponctuel	181–201
Somme	197
Stationnaire	184
– Q –	
QOSPF	79
Quantile	219
– R –	
Rafraîchissement	voir Routage/Paramètre
RIP	75, 77
Routage	
Algorithme	77, 95, 139
Bellman-Ford	77
Compétitif	82
Complexité	81, 125–135
Dijkstra	76, 77
Dijkstra	127
Hiérarchique	89–90, 125–135, 139
Sensible à la QoS	80–82
Dynamique	73–77
Hiérarchique	83–90, 115–135, 137–169, 174

Markovien	44, 215
Multicast	4, 9–37, 173
Paramètre	78, 94
Attribut	78
Métrique	78
Poids Administratif	78
Rafraîchissement	79, 94, 138, 174
Protocol	
Distance Vector	27
Protocole	
Distance Vector	74
Link State	75
Surcoût	100, 145, 153, 157, 163, 165
Rafraîchissement des informations	voir Routage/Paramètre
Sensible à la QoS	4, 77–83, 110, 173
Surcoût protocolaire	voir Routage/Protocole/Surcoût
Table	75, 118–125
Précalculée	81, 98, 135, 139
RP	22, 34, 40, 65
RPB	voir RPF
RPF	25, 27, 33
RSVP	18
Ryll-Nardzewski	voir Théorème

– S –

SAP	30
SDP	30
SDR	92
Shift	voir Flot
Shortest-Widest	voir Routage/Algorithme
Signalisation	11, 12, 15, voir aussi Crankback
SIP	31
Slivnyak	voir Théorème
SPT	27
SPT	voir Multicast/Arbre
Statistique d'Ordre	219–221

Symétrie voir Théorème

– T –

Théorème

Campbell 194, 198, 202
 Mecke 194, 202
 Møller 215
 Neveu 199, 202
 Palm 202
 Ryll-Nardzewski 195, 198, 202
 Slivnyak 194
 Symétrie voir Mecke
 Transformée de Laplace 188
 TTL 31
 Tunnel voir Tuyau
 Tuyau 27, 30, 32
 Tuyaux 44

– U –

Unicast 10

– V –

Voronoi

Cellule voir Cellule/Voronoi
 Pavage voir Pavage/Voronoi

Glossaire

– A –

API..... Application Part Interface
AS..... Autonomous System
ATM..... Asynchronous Transfer Mode

– B –

BGMP..... Border Gateway Multicast Protocol
BGP..... Border Gateway Protocol
BMA..... Broadcast Multiple Access

– C –

CBT..... Core Based Tree, Center Based Tree
CIDR..... Classless Inter Domain Routing
CNET à présent France Telecom R&D, Centre Nationale d'Etude des Télécommunications

– D –

DTL..... Designated Transit List
DVMRP..... Distance Vector Multicast Routing Protocol

– E –

ECMP..... Equal Cost Multi-Path
ENST..... Ecole Nationale Supérieure des Télécommunications

– G –

GCAC..... Generic Connexion Admission Control
GRE..... Generic Routing Encapsulation

– I –

IETF..... Internet Engineering Task Force
IGMP..... Internet Group Membership Protocol

INRIA Institut Nationale de Recherche en Informatique et en Automatique
IP Internet Protocol
ISP Internet Service Provider

– L –

LAN Local Area Network
LIJ Leaf Initiated Join

– M –

M-Bone Multicast Backbone (réseau expérimental Multicast IP)
MAC Medium Access Control
MASC Multicast Address Set Claim
MBGP Multicast Border Gateway Protocol
MOSPF Multicast Extensions to OSPF
MPLS Multi Protocol Label Switching
MSDP Multicast Source Discovery Protocol

– N –

NBMA Nom Broadcast Multiple Access

– O –

OSPF Open Shortest Path First

– P –

PG Peer Group
PGL Peer Group Leader
PIM Protocol Independent Multicast
PIM-DM PIM — Dense Mode
PIM-SM PIM — Sparse Mode
PNNI Private Network Network Interface

– Q –

QoS Qualité de Service
QOSPF QoS extensions to OSPF

– R –

RIP Routing Information Protocol
RP Rendez-Vous Point
RPB Reverse Path Broadcasting

RPF..... Reverse Path Forwarding
RSVP..... Resource reSerVation Protocol

- S -

SAP..... Session Announcement Protocol
SDP..... Session Directory Protocol
SDR..... Source Demand Routing
SIP..... Session Initiation Protocol
SPT..... Shortest Path Tree

- T -

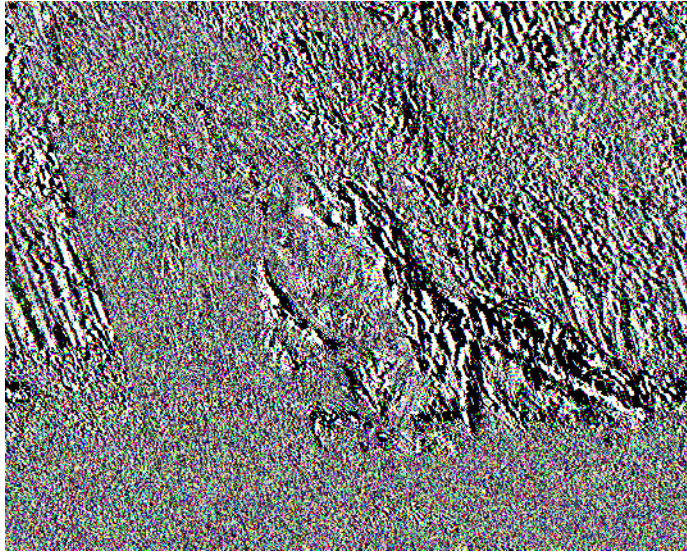
TCP..... Transmission Control Protocol
TTL..... Time To Live, Time To Live

- U -

UCL..... University College London

- W -

WAN..... Wide Area Network



Newton.

