



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

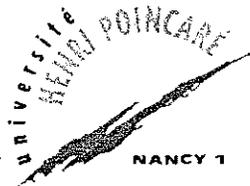
LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



S.C.D. - U.H.P. NANCY 1
BIBLIOTHÈQUE DES SCIENCES
Rue du Jardin Botanique - BP 11
54601 VILLERS-LES-NANCY Cédex

CENTRE DE RECHERCHE
EN AUTOMATIQUE DE
NANCY

FACULTE DES SCIENCES & TECHNIQUES

U.F.R. Sciences & Techniques : Mathématiques, Informatique et Automatique

Ecole Doctorale : IAEM

Département de Formation Doctorale : Automatique et Production Automatisée



CNRS UMR 7039

Thèse

présentée pour l'obtention du titre de

Docteur de l'université Henri Poincaré, Nancy 1

en Génie Informatique

par **Nicolas KROMMENACKER**

Heuristiques de conception de topologies réseaux : application aux réseaux locaux industriels

Soutenue publiquement le 5 Décembre 2002

Membres du jury :

| | | |
|----------------------|------------------------|--|
| Président : | M. Francis LEPAGE | Professeur, Université Henri Poincaré, Nancy I |
| Rapporteurs : | M. Laurent FOULLOY | Professeur, Université de Savoie, Annecy |
| | M. Michel MISSON | Professeur, Université Blaise Pascal, Clermont-Ferrand |
| | M. Jean-Pierre RICHARD | Professeur, Ecole Centrale, Lille |
| Examineurs : | M. Thierry DIVOUX | Professeur, Université Henri Poincaré, Nancy I (<i>directeur de thèse</i>) |
| | M. Eric RONDEAU | Professeur, Université Henri Poincaré, Nancy I (<i>directeur associé</i>) |
| | M. Anatoly MOLDOVANSKY | Sr. Project Engineer, Rockwell Automation, Mayfield Hts. USA |

Université Henri Poincaré, Centre de Recherche en Automatique de Nancy

Faculté des Sciences B.P 239 - 54506 Vandœuvre-lès-Nancy

S.C.D. - I.N.P. NANCY 1
BIBLIOTHÈQUE DES SCIENCES
Rue du Jardin Botanique - BP 11
54601 VILLERS-LES-NANCY Cédex

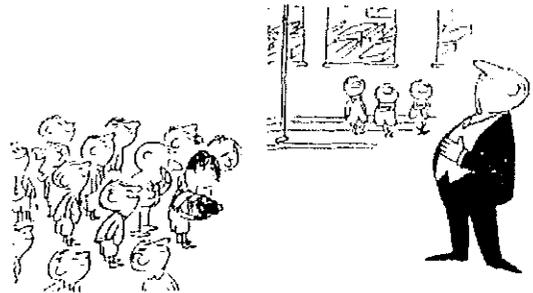
A mon épouse, Catherine.
A mes enfants, Théo et Zoé.

Remerciements

Le petit Nicolas en thèse¹

Le Directeur de Thèse

Pour commencer une thèse, il faut avoir un patron.
 Un patron, c'est un monsieur très, très fort qui me
 pose un problème et qui va m'aider à le résoudre.



Mes remerciements iront en premier lieu à mon directeur de thèse, le Professeur Thierry Divoux, et à mon directeur associé, le Professeur Eric Rondeau, sans qui, aujourd'hui, je ne serais certainement pas en train d'écrire ces lignes. Je les remercie donc pour leurs précieux conseils, leur collaboration et leur grande convivialité, qu'ils ont su m'apporter durant ces trois dernières années. Je tiens également à remercier le Professeur Francis Lepage, de m'avoir accueilli au CRAN.

La Recherche

Au début, c'est tellement compliqué, on n'y comprend rien.



THEOREME 1 - On a l'inégalité

$$\sum_{n=1}^K \frac{K+1(K-1)}{K^2} \log \left(\frac{K+1}{K} \right) + \frac{K-1}{K^2} \sum_{n=1}^K \sum_{d|n} \log |a_n|$$

$$\leq \left(1 - \frac{1}{K}\right) \frac{2D}{K} \sum_{n=1}^K h(n) + \frac{D}{K} \left(1 + \frac{K-1}{2D}\right) + \log \frac{K}{2}$$

¹ Extrait d'une parodie du petit Nicolas en Thèse :

(Dessins : J.J. Sempé, Formules : Y. Bugeaud - M. Mignotte - F. Normandin, Texte : G. Taviot, A. Maes)

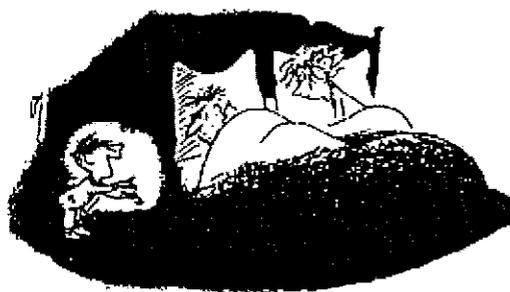
On peut passer des heures et des heures à chercher sans rien trouver. Dans ces moments-là, mon papa et ma maman sont drôlement inquiets et quand ma maman demande si c'était une bonne idée de faire faire une thèse au petit (c'est moi), mon papa ouvre la bouche sans parler, il agite les bras, et il s'en va lire le journal dans le salon.



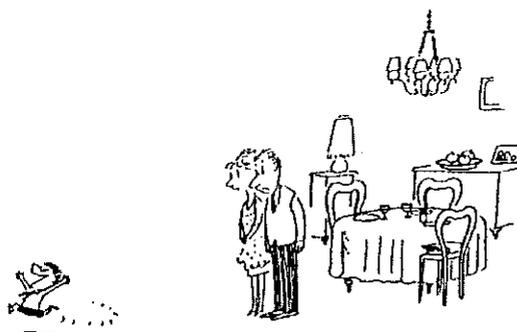
Dans ces moments-là, rien ne vaut le *Brain Storming Room* pour échanger avec les autres. Je tiens donc à remercier, pour leur aide et leurs conseils, toutes les personnes avec qui j'ai pu partager lors de longues discussions, et plus particulièrement Houssein, Jean-Marie, Jean-Philippe et Vincent. Je ne saurais oublier les personnes que l'on côtoie tous les jours et à qui on ne dit jamais assez merci. Alors, merci Anne, merci Christelle!!! Je tiens également à remercier toutes les personnes du laboratoire pour le climat de travail, et les bons moments de vie.

La Découverte

Des fois, c'est super, parce que je découvre des trucs que mon patron m'avait demandés. Evidemment, ça peut arriver à n'importe quelle heure, et mes parents ne sont pas toujours ravis.



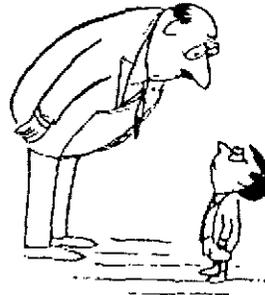
Ils se demandent si je ne deviens pas complètement fou, mais ma maman sait que mon papa n'aime pas qu'elle le lui dise.



Moi, je trouve ça plutôt normal d'être content. D'ailleurs, quand mon patron trouve un théorème, il est super fier et ses copains (qui sont aussi des gens très, très forts) sont super contents de lui. Mais ça mes parents, ils ne le savent pas.



Des fois aussi, ça se passe mal, parce que je me trompe. Et quand je me trompe, avec mon patron, ça ne rigole pas, mais alors pas du tout. "Regardez-moi dans les yeux, Nicolas", il me dit, pas content du tout. "Vous appelez ça du travail, peut-être?" qu'il me demande. Eh ben, là, ça a l'air d'une question, mais il ne faut surtout pas répondre, parce que sinon, il se fâche tout rouge!

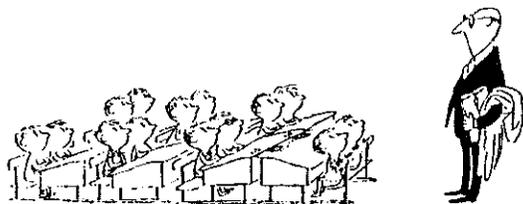


Ce travail n'est en aucun cas le fruit d'une recherche menée en solitaire et je tiens à remercier Thierry et Eric pour leur aide et la grande confiance qu'ils m'ont accordées tout au long de cette thèse. Je ne saurais oublier Jean-Philippe pour son aide précieuse, sa grande disponibilité et ses talents de programmeur que j'envie.

Enfin, j'adresse une tendre pensée à mon épouse, Catherine, ainsi qu'à mes enfants, Théo et Zoé, pour leur grande patience et leurs nombreux encouragements qui m'ont permis de terminer ce travail dans de bonnes conditions.

Les Séminaires

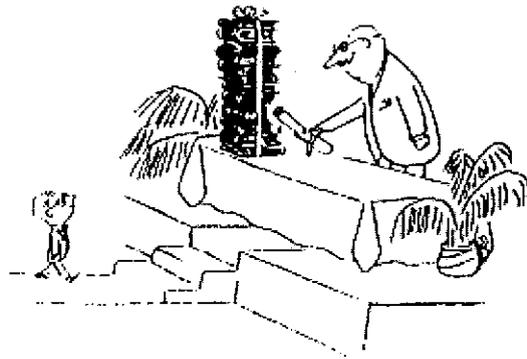
De temps en temps, un monsieur très, très important et vachement fort (mais pas aussi fort que mon patron, quand même) vient nous parler de trucs super-complicés. Ça s'appelle un séminaire, et pendant un séminaire, ça ne rigole pas non plus. Quand le monsieur a fini de parler, mon patron lui pose des tas de questions très compliquées, et il ne sait pas toujours répondre. Et là c'est pas juste, parce que lui, il ne se fait pas disputer!



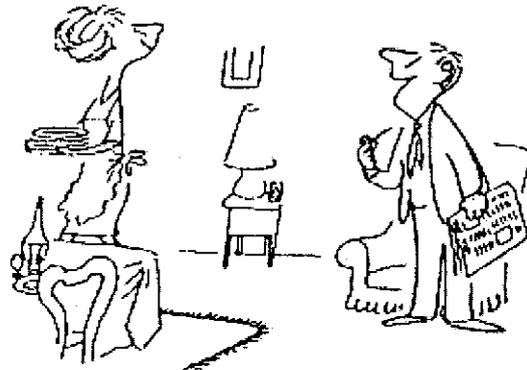
Je remercie l'ensemble des personnes qui ont participé, de près ou de loin, à ma formation scientifique au travers les différents cours, conférences, ou encore séminaires que j'ai suivis durant mes études universitaires.

La Soutenance

Quand j'aurai fini, il y aura une grande cérémonie avec plein de gens très, très forts (il y aura même d'autres patrons, c'est dire) et il y aura un vieux monsieur très, très important qui me dira que c'est très bien, mon petit, les chemins de la Recherche me sont glorieusement ouverts et je suis l'honneur de mes parents et l'orgueil de mon pays, et tout le baratin. Et après, il y aura un super goûter avec tous mes amis. Génial!



Et quand il lira tout cela dans le journal, mon papa sera très fier et ma maman sera tellement contente qu'elle me servira deux fois de la crème renversée, mon dessert préféré. C'est vraiment super, une thèse, à la fin!



J'adresse mes sincères remerciements aux Professeurs Jean-Pierre Richard de l'Ecole Centrale de Lille, Laurent Foulloy de l'Université de Savoie, et Michel Misson de l'Université Blaise Pascal de Clermont-Ferrand, pour avoir accepté de rapporter sur ma thèse et pour leurs suggestions et remarques pertinentes qui m'ont permis de mettre un point final à ce travail.

Je remercie également les autres membres du jury et plus particulièrement Monsieur Anatoly Moldovansky de la société Rockwell Automation, pour avoir accepté de participer au jury en qualité d'invité, et de faire le déplacement depuis les Etats-Unis.

La Gloire

D'ailleurs les filles, ça les impressionne drôlement de savoir qu'on a fait une thèse de mathématiques et qu'on a trouvé des tas de théorèmes compliqués et tout, et tout. Même la maman de Marie-Edwige, elle me fait des grands sourires maintenant, alors qu'elle trouvait que j'étais un petit garçon très turbulent.

$$h(X/\mathbb{Z}) \leq H^{m+1} \exp\{c_{32} p^{m+1} (\log^2 p)^{m+1} |D_K|^{m+1} / |N_{K/\mathbb{Q}}(\Delta_p)|^{m+1} (\log |A D_K N_{K/\mathbb{Q}}(\Delta_p)|)^{m+1}\}.$$



Pour finir, je remercie mes parents et toute ma famille pour leur soutien durant mes années d'étude. Ca y est Jérôme, tu peux le dire, ton petit frère a enfin fini d'aller à l'école!!!

Table des matières

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Le contexte industriel | 2 |
| 1.2 | Des réseaux propriétaires à <i>Ethernet Industriel</i> | 4 |
| 1.3 | La problématique Ethernet industriel | 8 |
| 1.4 | Objectifs de la thèse | 11 |
| 1.5 | Organisation du document | 12 |
| | | |
| 2 | Conception de réseaux locaux | 13 |
| 2.1 | Problématique | 13 |
| 2.2 | Optimisation combinatoire | 15 |
| 2.2.1 | Définition | 15 |
| 2.2.2 | Éléments de complexité | 17 |
| 2.2.3 | Panorama des méthodes de résolution | 18 |
| 2.2.3.1 | Approches constructives | 19 |
| 2.2.3.2 | Approches amélioratives | 20 |
| 2.2.3.3 | Approches évolutives | 21 |
| 2.2.3.4 | Approches hybrides | 23 |
| 2.3 | Etat de l'art en conception de réseaux | 23 |
| 2.3.1 | Interconnexions de réseaux locaux et métropolitains (LAN-MAN) | 24 |
| 2.3.2 | Bi-segmentation de réseaux locaux | 25 |
| 2.3.3 | Réseaux locaux commutés | 26 |
| 2.3.4 | Synthèse des travaux antérieurs | 28 |
| 2.3.5 | Positionnement de notre contribution | 29 |
| 2.4 | Partitionnement de graphes et conception de réseaux | 30 |
| 2.4.1 | Le partitionnement de graphes | 30 |

| | | |
|----------|--|-----------|
| 2.4.1.1 | Le mécanisme de partitionnement | 31 |
| 2.4.1.2 | Complexité | 32 |
| 2.4.2 | Les méthodes de résolution pour le partitionnement | 32 |
| 2.4.2.1 | Les méthodes globales | 33 |
| 2.4.2.2 | Les méthodes locales | 36 |
| 2.4.2.3 | Les processus multi-niveaux | 40 |
| 2.5 | Conclusion | 41 |
| 3 | Heuristiques de conception de RLI | 43 |
| 3.1 | Modélisation des données du problème | 43 |
| 3.1.1 | Les communications | 43 |
| 3.1.2 | La topologie | 44 |
| 3.1.2.1 | Les architectures commutées | 46 |
| 3.1.2.2 | Les architectures fiabilisées | 48 |
| 3.2 | Formulation du problème d'optimisation | 50 |
| 3.3 | Résolution à l'aide des algorithmes spectraux | 55 |
| 3.3.1 | Fondements des méthodes spectrales | 55 |
| 3.3.2 | Application à la conception de réseaux | 58 |
| 3.3.2.1 | Répartition des équipements | 58 |
| 3.3.2.2 | Modifications apportées à l'algorithme RSB | 60 |
| 3.3.2.3 | Architecture fiabilisée : Problème de sérialisation | 64 |
| 3.4 | Résolution par algorithmes génétiques | 66 |
| 3.4.1 | Fondements des algorithmes génétiques | 66 |
| 3.4.1.1 | De l'évolution des espèces à la théorie de l'évolution | 66 |
| 3.4.1.2 | Les algorithmes génétiques | 67 |
| 3.4.1.3 | La théorie des schémas | 68 |
| 3.4.2 | Application à la conception de RLI | 69 |
| 3.4.2.1 | Codage des solutions | 69 |
| 3.4.2.2 | Fonction coût (ou d'adaptation) | 71 |
| 3.4.2.3 | Les opérateurs génétiques | 72 |
| 3.4.2.4 | Contrôle et mise en oeuvre de l'algorithme | 74 |
| 3.5 | Conclusion | 75 |

| | |
|---|------------|
| 4 Résultats et évaluation de performances | 77 |
| 4.1 Etude des paramètres des AGs | 78 |
| 4.1.1 Influence de la taille de la population t_{pop} | 78 |
| 4.1.2 Influence de la probabilité de croisement p_c | 79 |
| 4.1.3 Influence de la probabilité de mutation p_m | 81 |
| 4.2 Performances intrinsèques des algorithmes | 84 |
| 4.2.1 Influence de la taille des graphes | 85 |
| 4.2.2 Influence de la densité des graphes | 86 |
| 4.2.3 Synthèse | 87 |
| 4.3 Evaluation du délai moyen | 88 |
| 4.3.1 Etude de la charge des commutateurs | 88 |
| 4.3.1.1 Méthode analytique | 88 |
| 4.3.1.2 Exemple numérique | 91 |
| 4.3.2 Calcul du délai de bout en bout | 94 |
| 4.3.2.1 Méthode analytique | 94 |
| 4.3.2.2 Exemple numérique | 95 |
| 4.4 Détermination de la borne maximale du délai | 97 |
| 4.4.1 Concepts du calcul réseau | 97 |
| 4.4.2 Modélisation du trafic | 99 |
| 4.4.3 Modélisation du réseau Ethernet commuté | 99 |
| 4.4.4 Calcul du délai de bout en bout | 101 |
| 4.4.4.1 Méthode analytique | 101 |
| 4.4.4.2 Exemple numérique | 101 |
| 4.5 Evaluation de performance des solutions | 104 |
| 4.5.1 Evaluation selon le critère de partitionnement | 104 |
| 4.5.2 Evaluation temporelle | 108 |
| 4.6 Conclusion | 111 |
| Conclusion générale et perspectives | 113 |
| Bibliographie | 125 |

Table des figures

| | | |
|------|--|----|
| 1.1 | Situation du réseau industriel dans l'entreprise | 2 |
| 1.2 | Equipements traditionnels de pilotage d'un processus industriel | 3 |
| 1.3 | Positionnement des principaux réseaux actuels | 5 |
| 1.4 | Evolution d'Ethernet à travers les différents niveaux de communication | 6 |
| 1.5 | Notion de domaine de collision | 10 |
| 2.1 | Représentation de l'espace de recherche | 16 |
| 2.2 | Approche constructive : réduction de l'espace de recherche | 19 |
| 2.3 | Approche séquentielle : amélioration par modifications élémentaires | 21 |
| 2.4 | Approche évolutive : exploration de régions de l'espace de recherche | 22 |
| 2.5 | Topologie du réseau | 26 |
| 2.6 | Un graphe et une partition de ce graphe. | 31 |
| 2.7 | Une maille partitionnée selon les centres de masse | 34 |
| 2.8 | Une maille bipartitionnée par rapport aux axes X,Y | 35 |
| 2.9 | Partitionnement avec l'algorithme LND | 37 |
| 2.10 | Raffinement par l'algorithme de KL | 38 |
| 2.11 | Processus de partitionnement multi-niveaux de graphes. | 40 |
| 3.1 | Représentation des communications | 44 |
| 3.2 | Bus de communication | 45 |
| 3.3 | Les différentes topologies | 46 |
| 3.4 | Temps de cycle / topologie | 48 |
| 3.5 | Architecture à redondance totale | 49 |
| 3.6 | Architecture à redondance de liens. | 49 |
| 3.7 | Architecture hybride | 50 |
| 3.8 | Distribution des équipements sur les commutateurs de terminaison | 52 |

| | | |
|------|---|-----|
| 3.9 | Analyse des charges et des communications | 53 |
| 3.10 | Démarche de conception | 54 |
| 3.11 | Décomposition suivant RSB | 59 |
| 3.12 | Décomposition suivant RSB modifié | 63 |
| 3.13 | Architecture hiérarchique déduite de la décomposition suivant RSB modifié | 65 |
| 3.14 | Sérialisation des commutateurs | 66 |
| 3.15 | Codage d'une solution | 70 |
| 3.16 | Exemple de sélection proportionnelle | 73 |
| 3.17 | Opérateur de croisement évolué | 73 |
| 3.18 | Opérateurs de mutation | 74 |
| 4.1 | Interface graphique de l'outil de conception " <i>CERISE</i> " | 78 |
| 4.2 | Influence de la taille de la population ($t_{pop.}$) | 80 |
| 4.3 | Influence de la probabilité de croisement (p_c) | 81 |
| 4.4 | Choix de la probabilité de croisement (p_c) | 81 |
| 4.5 | Influence de la probabilité de croisement (p_m) | 82 |
| 4.6 | Choix de la probabilité de croisement (p_m) | 83 |
| 4.7 | Evaluation du partitionnement | 85 |
| 4.8 | Influence de la taille des graphes | 86 |
| 4.9 | Influence de la densité des graphes | 87 |
| 4.10 | Modélisation d'un commutateur et d'un réseau commuté | 89 |
| 4.11 | Graphes d'arborescence représentant la topologie hiérarchique | 92 |
| 4.12 | Graphes d'arborescence représentant la topologie linéaire | 93 |
| 4.13 | Charges des commutateurs | 94 |
| 4.14 | Arriéré de traitement d'un système | 98 |
| 4.15 | Le concept du seau percé | 99 |
| 4.16 | Modèle de commutation | 100 |
| 4.17 | Exemple de réseau <i>Ethernet Industriel</i> | 102 |
| 4.18 | Scénario d'échanges avec 90 noeuds : solution non optimisée (zones grisées : échanges dans les commutateurs ; cellules grisées : échanges sur le réseau) | 105 |
| 4.19 | Apport de la modification de l'algorithme RSB | 106 |
| 4.20 | Scénario d'échanges avec 90 noeuds : solution optimisée | 107 |
| 4.21 | Outil de simulation de réseaux | 108 |
| 4.22 | Influence de la taille de coupure et de l'équilibre sur le délai max | 109 |

TABLE DES FIGURES

ix

| | |
|--|-----|
| 4.23 Influence de la valeur de coupure | 110 |
| 4.24 Influence de l'équilibre | 110 |

Liste des tableaux

| | | |
|-----|--|-----|
| 3.1 | Types de données | 44 |
| 3.2 | Critères réseaux / partitionnement | 53 |
| 4.1 | Résultats obtenus à partir du graphe Add20 | 84 |
| 4.2 | Comparatif des méthodes | 106 |
| 4.3 | Evaluation temporelle des solutions | 109 |

Chapitre 1

Introduction

Le besoin croissant de communication rend actuellement la notion de réseau incontournable. Dans le marché des réseaux industriels, ce sont les solutions réseaux dites *propriétaires* qui dominent. Cependant, la combinaison des nouveaux besoins de communication dans l'industrie et l'émergence des technologies liées à l'Intranet/Internet, montrent les limites des réseaux actuels. Une refonte radicale des architectures de réseaux industriels est alors engagée avec les besoins suivants : performance, garantie de qualité de service pour les applications temps réel, faible coût de déploiement et compatibilité avec les solutions propriétaires existantes.

Dans ce contexte, l'avènement d'Ethernet¹ dans les environnements industriels bouleverse le schéma classique des réseaux industriels. Néanmoins, cette migration se justifie par la volonté des utilisateurs de disposer d'un réseau unique, stable, ouvert, peu onéreux et indépendant des constructeurs. Au demeurant, "*Ethernet Industriel*" reste un sujet très controversé (cf. journées-débats : [AUT 01], [CER 01], [EUR 01]), principalement à cause de ses caractéristiques intrinsèques et en particulier de son non-déterminisme. Aussi, de nombreux travaux de recherche portent sur l'amélioration des capacités d'Ethernet pour qu'il offre les performances déterministes requises par les applications critiques.

Dans cette introduction, nous allons dans un premier temps définir les concepts de base qui permettent d'élaborer la problématique des réseaux industriels, puis mentionner les objectifs visés en matière d'amélioration de performance, pour déboucher sur le plan du mémoire.

¹Ethernet était alors principalement utilisé dans les réseaux informatiques.

1.1 Le contexte industriel

Il est usuel de distinguer deux secteurs d'application privilégiés pour les réseaux locaux : le secteur tertiaire (gestion, bureautique, ...) et le secteur secondaire (production, supervision, ...). Si le terme de "réseaux locaux d'entreprise" est généralement employé dans le premier cas, nous parlerons de "réseaux locaux industriels" dans le second. Les différences entre ces deux types de réseaux sont franches et variées. Ceci est principalement dû à une différence de besoins. Pour les premiers, il s'agit d'interconnecter au moindre coût un grand nombre d'éléments s'échangeant des informations volumineuses mais non contraintes par le temps. Pour les réseaux industriels, l'interconnexion a pour souci le respect de contraintes telles que le fonctionnement, la robustesse ou la disponibilité, imposés par les entités raccordées. Les deux secteurs se distinguent donc davantage par les contraintes de temps et de sécurité qui pèsent sur les échanges, que par les structures des informations échangées.

Comme le montre la figure 1.1, l'environnement du réseau industriel se caractérise par sa localisation dans les ateliers des entreprises. L'entreprise s'est dotée de plusieurs réseaux : Un réseau de communication classique type Ethernet TCP/IP, et séparément, un réseau industriel spécifique. Dès lors, une passerelle est installée entre ces deux réseaux, de manière à séparer la gestion du processus industriel des applications de niveau supérieur. L'environnement industriel diffère également par son hostilité (bruit, température, humidité, ...).

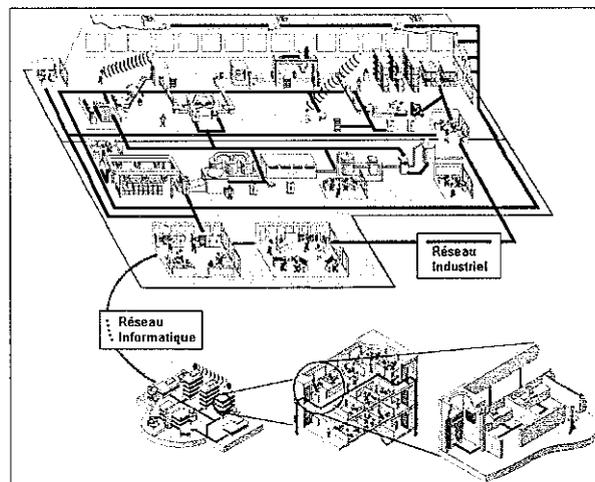


FIG. 1.1 – Situation du réseau industriel dans l'entreprise

L'objectif d'un réseau industriel consiste en l'établissement du support de communication entre les équipements d'un processus industriel. Ces équipements sont de deux types. Le premier concerne les équipements traditionnels de pilotage des parties opératives industrielles (Fig. 1.2) tels que les automates, les capteurs, les actionneurs, les modules d'Entrées/Sorties déportés ainsi que des Interfaces Homme-Machine (ou H.M.I.) et des Directeurs de Commande Numérique (ou D.N.C.).

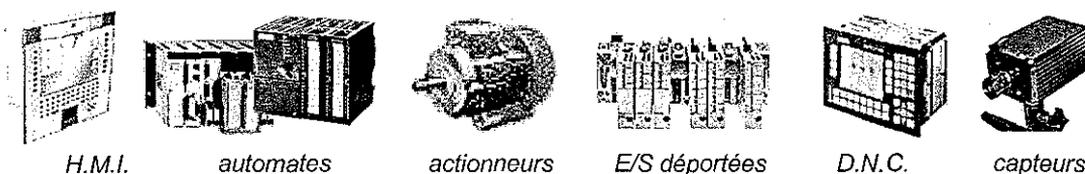


FIG. 1.2 – Équipements traditionnels de pilotage d'un processus industriel

Avec l'apparition des nouvelles technologies, de nouveaux équipements sont apparus dans les ateliers des entreprises tels que des équipements de télécommunications (téléphone, Pocket PC) ainsi que des fournisseurs d'applications multimédias type caméra et poste d'accès à Internet. Si ces technologies assurent un accès aux données de l'automatisme en tout lieu, elles contribuent également à augmenter la charge du réseau avec des volumes de données de plus en plus importants et des exigences en QoS (Qualité de Service) de plus en plus fortes.

En plus des contraintes temporelles liées à l'application industrielle, celles appliquées aux systèmes de traitement de ces informations concernent aussi les moyens de communication qui les relient. Elles peuvent être résumées selon trois mots clés définissant la QoS requise en milieu industriel :

Promptitude : Elle s'exprime par le temps de réponse qui doit être fini, borné supérieurement, aussi faible que nécessaire. Il peut parfois descendre à quelques millisecondes. Il est toujours dicté par l'application.

Robustesse : Elle regroupe les notions de fiabilité, de disponibilité et de sécurité. Le système doit "survivre" à des erreurs ou des dysfonctionnements consécutifs à la présence de défauts permanents ou transitoires. Elle doit aussi assurer la sûreté des hommes et du matériel.

Flexibilité : Les matériels à interconnecter sont très hétérogènes. Ils sont disposés selon des topologies variées et génèrent des trafics périodiques ou aperiodiques qui peuvent coexister. Les moyens de communication doivent satisfaire à ces exigences et s'accommoder des besoins des utilisateurs.

En résumé, la gestion du processus industriel s'apparente le plus souvent à une application de type temps réel. Cette notion de temps réel s'explique par le besoin de réactivité du processus face aux événements engendrés par l'application ou l'environnement. [MES 02] spécifie qu'un système temps réel doit être capable de réagir en toutes circonstances et dans une durée bornée.

1.2 Des réseaux propriétaires à *Ethernet Industriel*

Durant les dernières décennies, différents protocoles réseaux ont été développés et implémentés pour supporter ce besoin de communications temps réel, notamment au niveau des effecteurs du processus industriel. Les réseaux normalisés *WorldFip*, *Profibus*, et *CAN* sont à ce jour les plus populaires. Conçus pour prendre en compte les problèmes de déterminisme de transmission, ces réseaux sont fortement implantés et font toujours l'objet de travaux (comme l'utilisation de la technologie sans-fil). Néanmoins, ces réseaux ne sont pas sans inconvénients. Ils sont généralement propriétaires, coûteux² et incompatibles. Ils présentent également des débits faibles (inadaptés à la transmission de vidéos par exemple). De plus, une certaine passivité du réseau peut être mise en avant face à des événements de type réorganisation topologique ou coupure de lien.

Aujourd'hui, les utilisateurs souhaitent disposer d'un réseau standardisé, peu onéreux, ouvert et homogène (c'est à dire, le même pour l'ensemble des *niveaux de communication* définis par le CIM³). La figure 1.3 inspirée de [BEL 02, HIR 00], illustre les principaux réseaux d'entreprise et industriels, en fonction des trois niveaux de communication considérés : Information, Contrôle-Commande et Instrumentation.

²à titre d'exemple, une carte Fip pour PC coûte entre 450 et 600 euros, contre 30 euros pour une carte Ethernet standard.

³organisation du processus de production reposant sur une hiérarchisation des communications selon plusieurs niveaux d'abstraction dépendant de la perception du système et du degré de finesse de décomposition (Computer Integrated Manufacturing).

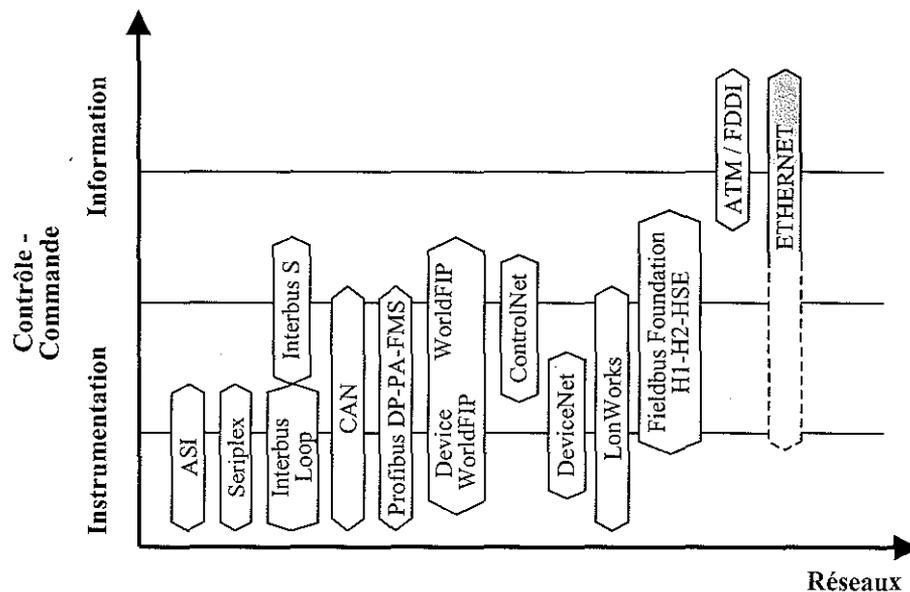


FIG. 1.3 – Positionnement des principaux réseaux actuels

Développée dans les années 1970 au Centre de Recherche XEROX de Palo Alto, la technologie Ethernet est désormais la plus utilisée dans les réseaux locaux (LAN⁴). Normalisée en 1985 par l'IEEE puis par l'ISO, elle bénéficie aujourd'hui d'une grande maturité qui lui confère de nombreux intérêts comme son faible coût, sa flexibilité ou encore son interopérabilité. Toutefois, son principal inconvénient réside dans le non-déterminisme qui ne permet pas de garantir le transport de messages en temps réel. Dès lors, sa migration vers le monde des réseaux industriels fut très controversée. Quoi qu'il en soit, les évolutions technologiques de ces dernières années comme la microsegmentation en mode full-duplex ou encore la priorisation du trafic, ont permis d'utiliser Ethernet dans des applications industrielles (cf section suivante). L'idée d'utiliser Ethernet dans un contexte industriel n'est d'ailleurs pas nouvelle puisque des solutions technologiques telles que le réseau FACTOR [BOU 86], avaient été proposées dans les années 1980. A l'époque, peu de produits avaient été développés et maintenus, contrairement à aujourd'hui où la revue *The Industrial Ethernet Book* n'en référence pas moins de 680.

La figure 1.4 illustre l'évolution d'Ethernet au cours des dernières années au sein des différents niveaux de communication. Au niveau "contrôle-commande", Ethernet TCP-

⁴Local Area Network en anglais.

IP est aujourd'hui parfaitement adapté pour assurer la communication entre les unités de traitement et remplace déjà des solutions propriétaires. La plupart des constructeurs d'automates programmables, d'interfaces homme-machine (ou HMI), ..., proposent aujourd'hui une interface de connexion de type Ethernet. Au niveau "terrain", il faut distinguer deux secteurs manufacturiers en fonction des contraintes temporelles requises et de la criticité :

- Le premier secteur est le secteur manufacturier à faibles contraintes temporelles pour lequel Ethernet semble répondre aux besoins de manière satisfaisante. C'est le cas par exemple dans l'agro-alimentaire⁵ ou dans l'industrie pharmaceutique⁶ qui utilisent déjà du "tout Ethernet".
- Le second secteur est le secteur manufacturier à fortes contraintes temporelles pour lequel les contraintes "temps critique" sont importantes. Les solutions reposent encore sur des réseaux propriétaires qui sont couplés avec des solutions Ethernet. C'est le cas par exemple dans le secteur de l'automobile, où le site de tôlerie du constructeur Jaguar est contrôlé via un réseau Ethernet commuté, transportant les informations de soudage et de contrôle de la zone de collision des robots. Un réseau de type Interbus complète l'installation au niveau des entrées/sorties des robots et des alarmes [HOP 01].

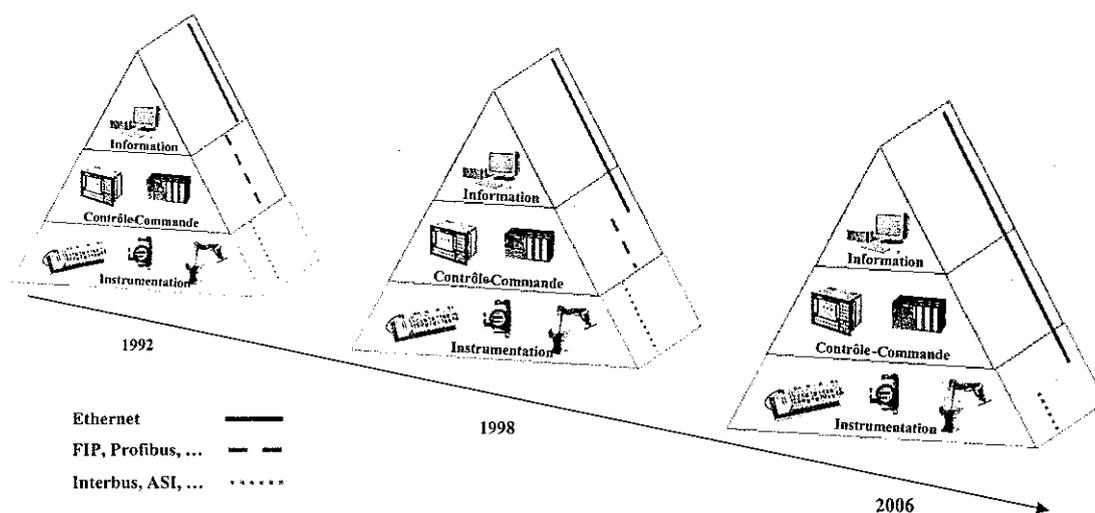


FIG. 1.4 – Evolution d'Ethernet à travers les différents niveaux de communication

⁵ c'est le cas des fromageries Milleret [JUN 01].

⁶ notamment, au sein de Boehringer Ingelheim [BAS 01].

La technologie Ethernet est donc en train de descendre vers le niveau instrumentation. La tendance actuelle consiste également à inclure de plus en plus de traitements au niveau capteurs et actionneurs. Munis de serveurs web intégrés, ils deviennent alors de véritables unités de traitement intelligentes. Il est alors possible de se connecter à tout environnement d'automatisme depuis des systèmes MES (Manufacturing Execution System), ERP (Enterprise Resource Planning) ou encore de supervision. Les outils tels que les logiciels de navigation du Web sont utilisés pour le diagnostic, le contrôle local et distant, etc. Le concept de "*Transparent Factory*" initié par *Schneider* [SCH 02b], s'appuie sur toutes ces nouvelles technologies pour définir une structure permettant de mettre à disposition en temps réel l'information quel que soit l'endroit où se trouve l'utilisateur qui peut ainsi agir en conséquence.

L'implantation d'Ethernet dans les réseaux industriels est soutenue par différents constructeurs et intégrateurs tels que *Rockwell* avec *EtherNet/IP* [ROC 98, MOL 02, BRO 01], *Siemens* [SIE 99], *Schneider* avec *Modbus TCP/IP* [SCH 02a] ou encore *Hirschmann* [HIR 99]. Ces acteurs sont d'ailleurs regroupés dans des associations, comme l'association IEA (Industrial Ethernet Association), et surtout l'association IAONA (Industrial Automation Open Networking Alliance), pour promouvoir Ethernet comme le standard dans les environnements industriels. Cette dernière (IAONA), implantée aux Etats-Unis et depuis 1999 en Europe, est constituée de six groupes de travail traitant aussi bien les problèmes de temps-réel, que ceux de connexion et câblage, de sécurité, de commercialisation, etc. Divers projets tels que CIDER (Communication Infrastructure for Dependable Evolvable Real-time Systems) [ALV 00], soutiennent et étudient notamment les besoins des futurs systèmes temps réel en termes d'adaptabilité et de reconfiguration. Finalement, l'adoption de standards communs par les deux mondes de l'automatique et de l'informatique tend à faire disparaître les frontières issues du concept CIM et à développer des alliances. Ainsi, une récente alliance entre *GE Industrial Systems* et *Cisco Systems* au sein d'une entreprise commune *GE Cisco Industrial Networks*, a permis à chacun d'apporter ses connaissances (respectivement industrielles et réseaux) afin de promouvoir Ethernet dans le monde industriel.

Ethernet Industriel est déjà intégré dans des projets de grande envergure comme la construction du nouvel Airbus A380 et le développement de l'*avionique nouvelle*⁷ ou Avionique

⁷ par opposition à l'*avionique classique* qui repose sur aucun partage des ressources de calcul et de communication permettant ainsi une maîtrise totale du comportement temps réel du système avionique.

Modulaire Intégrée (IMA), favorisant le partage des ressources [DUR 01]. L'architecture de communication basée sur des commutateurs AFDX (Avionics Full Duplex Ethernet switch) permet de connecter les systèmes avioniques tels que les écrans, les radios et les capteurs de navigation. La flexibilité d'Ethernet doit permettre de supporter d'autres applications dans le futur [GMB 01].

1.3 La problématique Ethernet industriel

La norme Ethernet IEEE 802.3 [IEE 98] spécifie un mode d'accès aléatoire au canal de transmission selon la méthode CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Elle spécifie également un algorithme de résolution de collision bien connu sous le nom de *Binary Exponential Backoff* (BEB), et d'autres paramètres comme le support physique. Nous ne détaillerons pas ici la méthode ni les algorithmes. Toutefois, le lecteur intéressé est invité à consulter les références suivantes [IEE 98, MET 76] pour avoir plus de détails.

Dès lors, un des arguments le plus souvent avancé contre l'utilisation d'Ethernet dans une application "*temps réel*" est le suivant : Ethernet dispose d'un délai d'accès au médium non déterministe, menant à un comportement temporel imprévisible [DEC 01]. En effet, le protocole CSMA/CD est à l'origine une méthode d'accès au support de communication physique, lui-même partagé⁸ entre plusieurs équipements. Des collisions peuvent donc se produire. Néanmoins, l'intérêt porté aux réseaux Ethernet pour supporter des applications temps réel, conduit la communauté scientifique à développer des techniques et des méthodologies pour palier le non déterminisme d'Ethernet. Depuis la fin des années 80, plusieurs travaux ont été amorcés ayant comme objectif de réaliser un réseau Ethernet avec un comportement temporel prévisible. Les solutions apportées pour traiter le problème peuvent être classées suivant trois approches :

- La première consiste à éliminer totalement les collisions en modifiant par exemple la méthode d'accès au support. L'approche la plus basique, TDMA (Time Division Multiple Access) arbitre l'accès au canal en réalisant un fenêtrage temporel autorisant le noeud du réseau à émettre pendant un intervalle de temps alloué. De la même façon, P-CSMA (ou Prioritised-CSMA) réalise un fenêtrage axé sur des politiques

⁸le terme "*Shared Ethernet*" est couramment employé.

de priorité des messages. D'autres méthodes telles que TEMPRA (TimEd Mechanism Packet Release Access)[PRI 95] ou RETHER (Realtime ETHERnet) [CHI 94] utilisent des mécanismes utilisés dans les réseaux industriels propriétaires comme le mécanisme de passage de jeton.

- Une autre approche consiste à adapter le protocole CSMA/CD, et plus exactement l'algorithme de résolution des collisions, afin de résoudre les collisions de façon déterministe. Pour cela, les méthodes mettent en œuvre généralement un mécanisme de temporisation permettant de retarder l'émission d'un paquet selon une politique pré-établie (Packet Release Delay). Cette dernière peut être basée sur les adresses (CSMA/DCR ou CSMA with Deterministic Collisions Resolution) [LAN 93], sur des priorités (CSMA/PDCR ou CSMA with Priority Deterministic Collisions Resolution) [TUR 96] ou sur une planification temporelle liée aux messages (DODCSMA/CD ou Deadline Oriented Deterministic CSMA/CD) [LAN 93].
- La dernière méthode enfin, consiste à réaliser le lissage du trafic (ou en anglais traffic smoothing) pour s'affranchir des fluctuations d'arrivée des messages générant des collisions. L'intégration d'un régulateur de trafic est proposé dans [KWE 99, CAP 02].

Les inconvénients majeurs de toutes ces méthodes résident dans le développement matériel, logiciel et la perte de flexibilité inhérente. Des solutions d'ordre technologique ont également été proposées comme le surdimensionnement du réseau. En effet les évolutions vers Fast Ethernet (Ethernet à 100 Mbit/s), puis Gigabit Ethernet (Ethernet à 1 Gbit/s), réduisent les temps de transfert, et donc la probabilité de collision par rapport à un réseau classique à 10 Mbit/s. Comme la plupart des applications industrielles consomment et produisent de petites quantités de données (encapsulées dans des trames de 64 octets, taille minimale d'une trame supportée par Ethernet), il est donc peu probable qu'elles utilisent entièrement 100 Mbit/s. De plus, la disponibilité de hauts débits incite à faire supporter aux réseaux de nouveaux médias gourmands en bande passante comme la voix et la vidéo, au détriment des flux temps-réel. Cet exemple montre que l'augmentation du taux de transfert ne permet pas seul, de surmonter le problème du non déterminisme.

Parallèlement aux trois approches cités ci-dessus, les dernières évolutions technologiques telles que Ethernet commuté (ou "*Switched Ethernet*"), permettent également d'éliminer les collisions. Contrairement à "*Shared Ethernet*", les architectures commutées permettent de dédier le médium et de réduire les domaines de collisions [THO 00]. Avec un hub, une

collision entre plusieurs paquets simultanément émis, se propage à tout le réseau, rendant indisponible celui-ci pour un certain temps. En revanche, les commutateurs séparent les équipements en petits groupes (ou segments), voire même en unités simples, et réduisent ou éliminent ainsi le nombre de collisions (Fig. 1.5). Nous parlerons alors de segmentation du réseau.

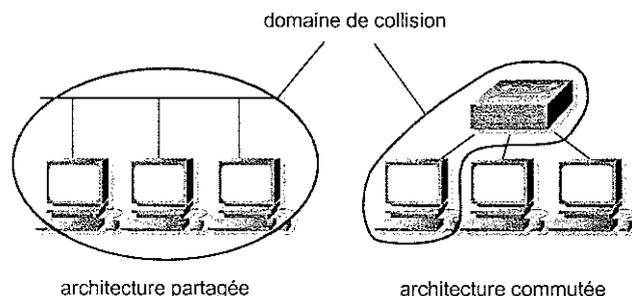


FIG. 1.5 – Notion de domaine de collision

Si la segmentation réalisée par le commutateur est poussée à l'extrême, chaque équipement est connecté à un port du commutateur. Isolé, il dispose alors de son propre segment avec le débit maximal. Cette technique appelée "*microsegmentation*" élimine les collisions entre les équipements disposant chacun d'un segment dédié. De plus, ce type d'architecture génère uniquement des communications deux à deux entre le commutateur et l'équipement; un mode de communication *Full Duplex*⁹ permet d'éliminer les contentions entre ces deux équipements.

Après avoir éliminé toutes les contentions possibles, il n'y a plus de réel besoin de détection de collision, ni d'algorithme de résolution des collisions. Dans ces conditions, un réseau Ethernet construit à partir de commutateurs avec les techniques de *microsegmentation* et d'opérations *Full Duplex*, semble approprié à une utilisation en milieu industriel. Cependant, les phénomènes de collisions se transforment en problème de congestion au niveau des commutateurs. En effet, dans le cas où de nombreux équipements tentent de communiquer avec un même destinataire, la convergence du trafic peut générer de la

⁹ Ethernet partagé utilise un mode opératoire *half duplex* inhérent au partage du médium. A un instant donné, un équipement est soit en transmission, soit en réception. Le mode *full duplex* permet d'émettre et de recevoir en même temps.

congestion. Il devient alors difficile de garantir le délai de bout en bout. Nos travaux se positionnent dans ce contexte et les objectifs sont présentés dans la section suivante.

1.4 Objectifs de la thèse

Comme nous l'avons vu précédemment, les contraintes de temps et de fiabilité distinguent les réseaux industriels, des réseaux locaux d'information. Si Ethernet Industriel est d'ores et déjà utilisé dans des environnements industriels tels que l'agro-alimentaire et l'industrie pharmaceutique, il reste difficile à implanter dans des secteurs manufacturiers à fortes contraintes temporelles. La garantie du délai de bout en bout est essentielle pour ce type d'application. Les solutions actuelles s'orientent vers l'utilisation d'architectures Ethernet Commuté Full-Segmenté¹⁰ permettant d'éliminer totalement les collisions. Cependant, des phénomènes de congestion subsistent et conduisent à augmenter les délais de bout en bout. Ce problème est traité actuellement à travers différentes propositions comme :

- la priorisation du trafic (norme 802.1p) définissant des niveaux de priorité en fonction de la criticité des paquets,
- le contrôle d'admission utilisant des mécanismes de régulation de trafic (*back-pressure*, *pause-command*, *traffic smoothing*, ...).

L'ensemble de ces propositions est d'ordre technologique et ne prend pas en compte les aspects applications qui génèrent le trafic sur ces réseaux. L'originalité de nos travaux est donc d'étudier les échanges entre les différents noeuds du réseau et de proposer une organisation d'architecture physique de communication la mieux adaptée, sans remettre en cause les concepts de base d'Ethernet, ni le trafic. L'optimisation de l'architecture topologique permet de rationaliser les échanges, et ainsi de limiter les congestions. Cette affirmation est démontrée dans la thèse où nous proposons un ensemble de méthodes de conception d'architectures réseaux. Dès lors, l'apparition de congestion est prise en compte en amont de façon à en éliminer les effets. Ainsi les solutions proposées conduisent au respect des contraintes temporelles imposées par l'application.

En résumé, l'objectif visé est de concevoir des architectures répondant aux besoins de l'application, parmi lesquels la garantie du délai de bout en bout. D'autre part, nous

¹⁰contraction de Full duplex et micro Segmenté.

nous attacherons à en assurer la fiabilité. Notons que pour parvenir à maîtriser ces deux attributs de Qualité de Service, nous serons amenés à définir un contexte de modélisation pour décliner des heuristiques optimisant les topologies des réseaux concernés.

1.5 Organisation du document

Le deuxième chapitre présente un panorama des activités de recherche menées dans le domaine de la conception d'architectures de communication à travers les différentes méthodes de résolution de problèmes d'optimisation combinatoire. Il permet de situer notre approche qui traite le problème de conception de réseaux industriels comme un problème de partitionnement des graphes. Les méthodes de résolution de cette sorte de problème sont ensuite étudiées et leur adéquation au problème de conception est discutée.

Le troisième chapitre est consacré à notre approche de conception de réseaux *Ethernet Industriel*. Dans un premier temps, nous présentons la modélisation des communications sous forme de graphes ainsi qu'une étude des différentes topologies de réseaux commutés. Nous proposons alors une topologie redondante permettant de répondre aux besoins de fiabilité et de disponibilité des réseaux industriels. Dans la suite, nous proposons d'optimiser les délais d'acheminement des messages sur cette architecture, à partir du partitionnement du graphe associé. Pour cela, nous utilisons deux heuristiques de résolution : les Algorithmes Spectraux et les Algorithmes Génétiques. Les fondements de chaque technique de partitionnement sont alors étudiés. Ils permettent d'adapter chaque algorithme au problème de conception.

Dans le dernier chapitre, nous présentons l'ensemble des résultats obtenus. Tout d'abord, une étude des paramètres des Algorithmes Génétiques est menée afin d'améliorer leurs performances. Nous comparons alors les deux heuristiques hors contexte d'application, c'est à dire, sur la qualité du partitionnement. Ensuite, les architectures optimisées sont évaluées du point de vue temporel. Nous utilisons alors des méthodes d'évaluation du délai moyen d'acheminement de bout en bout des messages, ainsi que des outils de simulation de réseaux. Pour finir, une étude de détermination de la borne maximale du délai permet de vérifier que l'architecture respecte les contraintes temporelles fixées par l'application.

Chapitre 2

Conception de réseaux locaux

2.1 Problématique

Selon Nussbaumer [NUS 87], le problème de conception d'une architecture réseau peut être formulé de la façon suivante :

Définition *Etant donnés les équipements informatiques à interconnecter, leur localisation, le trafic entre les stations, l'objectif est de minimiser le coût des liaisons et des équipements d'interconnexion. Cette optimisation doit être effectuée en tenant compte d'un certain nombre de contraintes dont les principales sont généralement d'assurer une fiabilité minimum au réseau et de garantir que le temps de traversée du réseau par les paquets ne dépasse pas une limite donnée. Pour effectuer cette optimisation, le concepteur peut faire varier la topologie, c'est à dire essentiellement décider des équipements de commutation et de multiplexage à utiliser, ainsi que les lignes à établir entre les équipements des abonnés et entre les équipements du réseau et adopter une politique de son choix en ce qui concerne le routage des paquets.*

La conception des réseaux de communication pose des problèmes complexes, notamment en raison de l'essor qu'a connu le domaine des télécommunications dans les vingt dernières années et de l'évolution constante qui le caractérise. Ces problèmes touchent notamment au dimensionnement (i.e. à la nature et au nombre de câbles à installer entre les noeuds du réseau) et au routage (i.e. aux chemins choisis pour satisfaire la demande des clients) dans les réseaux. Si les problèmes de conception de réseaux sont la plupart du temps relativement simples à énoncer, il ne faut en aucun cas sous-estimer la difficulté

à trouver une solution. Ces problèmes font souvent partie des problèmes d'*optimisation combinatoire* pour lesquels, dans la majorité des cas, il est très difficile de trouver la solution optimale. En effet, à quelques exceptions près, la seule méthode connue pour résoudre ce type de problème de manière exacte est de faire une énumération complète de toutes les solutions possibles. Cependant, si nous devons explorer de façon exhaustive toutes les configurations possibles, nous déboucherions rapidement sur un problème d'explosion combinatoire où même les ordinateurs les plus rapides prendraient un temps de calcul excessif. En considérant des caractéristiques indépendantes telles que la topologie, la capacité ou encore le trafic, et en ne travaillant que sur une seule caractéristique à la fois, le problème reste tout de même complexe. À elle seule, la configuration topologique nécessite de nombreux calculs. Par exemple, il y a $n(n - 1)/2$ liaisons possibles entre n noeuds, et $n(n - 1)$ possibilités de configuration topologique.

Dans la conception d'un réseau de télécommunications [SCH 77], on doit considérer :

- la configuration topologique, qui se réfère à l'emplacement géographique des liaisons et des noeuds,
- le trafic, qui correspond au nombre de paquets par seconde échangé entre chaque paire de noeuds,
- l'allocation de capacité aux liaisons, qui consiste à déterminer une borne supérieure sur la quantité d'informations que peut acheminer ces liaisons,
- la procédure de routage, qui sélectionne le meilleur chemin parmi les multiples chemins reliant chaque paire de noeuds,
- les procédures de contrôle de flot, par lesquelles on s'assure que la quantité d'informations émise puisse être absorbée ou traitée par le récepteur.

On doit également répondre à des critères de performance de moindre coût relatifs au délai et à la fiabilité comme :

- le délai moyen de réponse du réseau, c'est à dire le temps moyen que prend un paquet pour aller d'un noeud source à un noeud destination,
- la fiabilité en terme de connexité : le réseau doit demeurer fonctionnel malgré la panne d'une liaison ou d'un noeud.

Ce qui rend la conception complexe, c'est le fait que plusieurs facteurs soient parfois intimement liés : la topologie du réseau est directement reliée non seulement à la fiabilité, mais également au trafic et aux capacités des liaisons. Le développement de modèles et de

méthodes de résolution efficaces de ce type de problèmes constitue un aspect essentiel de la recherche en conception des réseaux de télécommunications.

Dans ce chapitre, nous rappelons les éléments de l'*optimisation combinatoire* et ses méthodes de résolution. Puis nous présentons un panorama des activités de recherche menées dans le domaine de la conception de réseaux locaux. Il nous permettra de situer notre approche qui traite le problème comme un problème de partitionnement de graphes. Pour finir, nous étudierons les différentes techniques de partitionnement et leur adéquation au problème de conception de réseau.

2.2 Optimisation combinatoire

2.2.1 Définition

Un problème d'*optimisation* est défini par un ensemble X de solutions et une fonction objectif f (ou fonction de coût) qui assigne à chaque candidat x de X une valeur $f(x)$. Résoudre un tel problème consiste à trouver une solution optimale x_{opt} de X optimisant la fonction objectif.

Un problème *combinatoire* est défini par un ensemble de solutions S (ou espace de recherche) et un ensemble de contraintes C formalisant le problème. L'ensemble de solutions est un ensemble fini (ou dénombrable) de solutions candidates. Un candidat qui satisfait toutes les contraintes C du problème est dit solution réalisable (ou admissible) ou tout simplement une solution. L'objectif, quand on résout un problème combinatoire est de trouver une solution réalisable.

Un problème d'*optimisation combinatoire* est donc implicitement défini par un ensemble discret de solutions S , un sous-ensemble X de S représentant les solutions réalisables suivant les contraintes $c \in C$ et une fonction objectif f . Résoudre un tel problème consiste à trouver une solution $x_{opt} \in X$ optimisant la valeur de la fonction de coût f . En d'autres termes, c'est équivalent à trouver l'ensemble de toutes les solutions réalisables qui satisfont C et à trouver la solution optimale x_{opt} selon la fonction objectif f . Une telle solution est parfois dite solution optimale globale.

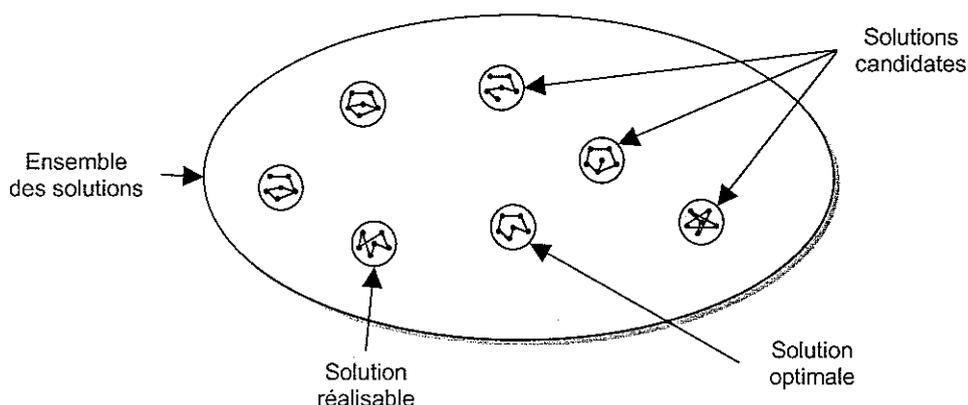


FIG. 2.1 – Représentation de l'espace de recherche

La figure 2.1 est une représentation de l'espace de recherche dans un problème d'optimisation combinatoire. Dans le cas d'un problème de minimisation, nous obtenons la définition suivante :

Définition *Un problème de minimisation est formalisé à l'aide du triplet (X, C, f) où $X \in S$ est un ensemble fini de solutions admissibles vérifiant l'ensemble des contraintes $c \in C$, et f une fonction de coût à minimiser $f : X \rightarrow \mathbb{R}$. Le problème est de trouver $x_{opt} \in X$ tel que $f(x_{opt}) \leq f(x)$ pour tout élément $s \in X$.*

L'optimisation combinatoire trouve des applications dans des domaines aussi variés que la gestion, l'ingénierie, la conception, la production, les télécommunications, les transports, l'énergie, les sciences sociales et l'informatique elle-même.

Toutes ces notions peuvent être illustrées à travers le problème bien connu du voyageur de commerce¹ dans lequel il s'agit de trouver le chemin le plus court entre n villes, qui passe une seule fois par chaque ville. L'ensemble des chemins existants représente alors l'espace de recherche. Les solutions réalisables doivent satisfaire la contrainte d'unicité de visite des villes. Un chemin comportant au minimum deux fois la même ville représente une solution non réalisable. La fonction objectif est quant à elle donnée par la notion de coût du chemin, ou plus exactement de distance dans ce cas. La solution optimale est le chemin le plus court qui traverse une seule fois l'ensemble des villes. Les possibilités (nombre de trajets possibles) augmentent exponentiellement avec le nombre de villes. Si n est le nombre de villes, le nombre de possibilités est de $(n - 1)!/2$.

¹ou Traveling Salesman Problem (TSP) en anglais.

2.2.2 Éléments de complexité

La complexité d'un algorithme permet de quantifier mathématiquement le niveau de difficulté lors de la résolution de problèmes, selon des critères de temps d'exécution, de nombre d'itérations (ou d'opérations arithmétiques) ou encore d'espace mémoire nécessaire. Ces grandeurs sont indiquées en fonction de la taille des données du problème à résoudre. La théorie de la complexité établit une classification des problèmes en fonction des algorithmes susceptibles de les résoudre, les machines de Turing constituant un étalon fréquemment utilisé. Sans entrer dans les détails, nous distinguons deux grandes classes² P et NP des problèmes. Une description plus complète des différentes classes des problèmes d'optimisation combinatoire peut être trouvée dans [GAR 79][PAP 82].

Un problème est dit de classe P s'il peut être résolu en un temps polynomial par une machine de Turing déterministe dont le temps d'exécution est de complexité polynomiale (P signifiant alors *Polynomial time*). Les problèmes de classe P sont considérés comme des problèmes faciles. C'est le cas notamment du problème de la connectivité d'un graphe.

La classe NP est une classe plus riche de problèmes pour lesquels on peut construire une machine de Turing non déterministe dont le temps d'exécution est de complexité polynomiale (NP signifiant *Nondeterministic Polynomial time*).

Un problème NP est dit *NP-complet* si tout problème de la classe NP peut lui être réduit : c'est le cas de nombreux problèmes considérés comme intraitables en pratique (pour lesquels les algorithmes déterministes connus sont exponentiels). Les problèmes *NP-complets* sont les problèmes NP les plus difficiles, et un algorithme qui peut les résoudre requiert un temps exponentiel. Dans le pire des cas, un tel algorithme aurait besoin d'énumérer toutes les solutions candidates de l'espace de recherche. Par conséquent, seule une très petite instance du problème peut être résolue à l'intérieur d'un temps raisonnable et les grandes instances sont impraticables.

Une heuristique est un algorithme ne garantissant pas de trouver une solution optimale mais qui a une bonne chance de trouver une bonne solution en un temps raisonnable. De tels algorithmes sont nécessaires pour traiter de grandes instances des problèmes d'optimisation combinatoire.

²Les notions de classes ne s'appliquent qu'à des problèmes de décision (ou de reconnaissance), c'est à dire à des énoncés pour lesquels la réponse est *oui* ou *non*, et non pas à des problèmes d'optimisation [CHA 96]. Cependant, on peut toujours reformuler un problème d'optimisation en : "existe t'il une solution meilleure que ...".

2.2.3 Panorama des méthodes de résolution

Il existe plusieurs méthodes possibles, de natures radicalement différentes, pour résoudre un problème *NP-complet*. On peut cependant distinguer trois grandes catégories parmi les méthodes d'optimisation :

- les méthodes exactes : le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Ces méthodes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante. Dans ces conditions, il est nécessaire de trouver un mode de résolution qui fournisse une solution de bonne qualité dans un laps de temps raisonnable, ce que font les méthodes heuristiques,
- les heuristiques : il s'agit d'algorithmes spécifiques à un problème donné et utilisant les connaissances du domaine. Elles ne garantissent pas de trouver de manière exacte l'optimal et donnent un résultat approché. Ces méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale,
- les métaheuristiques³ : ce sont des heuristiques générales applicables à une grande variété de problèmes.

Face à la complexité et à la taille des problèmes de conception de réseau, peu de travaux exploitent les méthodes exactes et par conséquent, elles ne seront pas abordées par la suite. De même, nous ne ferons pas de distinction entre *heuristiques* et *métaheuristiques* et utiliserons le terme heuristique pour désigner de manière générale les méthodes approchées.

On distingue trois classes principales d'heuristiques, à savoir les méthodes constructives, celles dites amélioratives ou séquentielles et celles considérées comme évolutionnaires [COS 95]. Cette classification, non exhaustive mais plus ou moins commune, des heuristiques est réalisée selon la manière dont la solution est créée.

³Le terme « métaheuristique » a été initialement utilisé par F. Glover pour distinguer la méthode tabou des heuristiques spécifiques [GLO 86].

2.2.3.1 Approches constructives

L'objectif de ces méthodes, comme indiqué dans le titre, est de construire une bonne solution réalisable en partant d'une solution initiale vide et en réalisant à chaque étape, le choix d'une composante à insérer dans la solution partielle courante. Ce processus est réitéré jusqu'à ce qu'une solution admissible soit construite.

Pour illustrer ce type de recherche, reprenons le problème du voyageur de commerce abordé dans la partie 2.2.1. Le voyageur de commerce doit rendre visite à un ensemble de n clients situés dans n villes. Il peut démarrer sa tournée en visitant le client le plus proche. Puis, il poursuit en visitant à chaque fois le client situé dans la ville la plus proche de la ville courante et qu'il n'a pas encore rencontré. De cette manière, il aura parcouru toutes les villes et visité tous ses clients lors de sa tournée.

L'idée générale de ce type d'approche consiste à diminuer la taille du problème à chaque étape, ce qui revient à se restreindre à un sous-ensemble $X_k \subseteq X$ toujours plus petit (Figure 2.2). Une méthode constructive trouve une solution optimale lorsque chacun des sous-ensembles considérés contient au moins une solution optimale $x_{opt} \in X$. Malheureusement, rares sont les cas où une telle condition est remplie avec certitude.

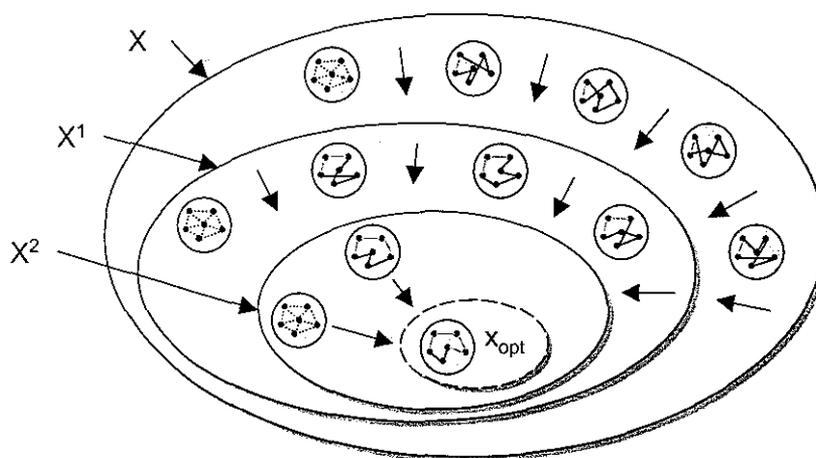


FIG. 2.2 – Approche constructive : réduction de l'espace de recherche

La majorité des méthodes constructives sont de type glouton⁴. A chaque étape, la solution courante est complétée de la meilleure façon possible sans tenir compte de toutes les

⁴ou *greedy algorithm* en anglais.

conséquences que cela entraîne au niveau de la solution finale. Dans ce sens, les méthodes de type glouton sont souvent considérées comme "*myopes*".

Les méthodes constructives se distinguent par leur rapidité et leur grande simplicité. On obtient en effet très rapidement une solution admissible pour un problème donné sans avoir recours à des techniques hautement sophistiquées. Le principal défaut de ces méthodes réside malheureusement dans la qualité des solutions obtenues. Le fait de vouloir opérer à tout prix le meilleur choix à chaque étape est une stratégie dont les effets peuvent être catastrophiques à long terme. Il est donc judicieux, dans le cas général, de mettre au point des procédures anticipant les effets secondaires et les conséquences futures occasionnées par les décisions prises lors de la construction d'une solution admissible.

2.2.3.2 Approches amélioratives

Contrairement aux méthodes constructives dont l'objectif est de construire une solution, les méthodes amélioratives (également appelées séquentielles ou itératives) modifient une solution initiale, en vue d'améliorer sa valeur. Cette solution initiale peut être, soit choisie de manière aléatoire, soit obtenue à l'aide d'un pré-traitement via une méthode constructive. A chaque étape du processus, la solution courante est transformée en une autre solution à l'aide de modifications élémentaires (Figure 2.3). Dans le cas du voyageur de commerce, une modification élémentaire peut être représentée par la permutation de deux villes dans sa tournée. Le processus s'arrête lorsque le critère de terminaison prédéfini est atteint. Il peut s'agir d'un nombre de modifications (ou d'étapes), du dépassement d'un seuil fixé par la fonction de coût, ou encore le temps d'exécution.

La plupart de ces méthodes utilise la notion de voisinage. Il s'agit de trouver, à chaque itération, une bonne solution parmi l'ensemble des solutions qui définit un voisinage d'une solution courante. Parmi ces méthodes, on distingue, celles dites *méthodes de descente*, qui améliorent à chaque itération la valeur de la fonction économique (comme la *méthode tabou*), de celles qui permettent de choisir une solution qui n'améliore pas forcément la valeur de la fonction économique (c'est le cas du *recuit simulé*).

D'une manière générale, le principal inconvénient des méthodes de descente est qu'elles donnent souvent des solutions correspondant à des optima locaux qui ne sont pas de très bonne qualité. Aussi, une phase dite de *backtracking* peut être ajoutée pour élargir

la recherche. Elle consiste, à partir d'une solution partielle, à revenir en arrière vers la solution précédente et à poursuivre le processus de recherche vers une autre solution.

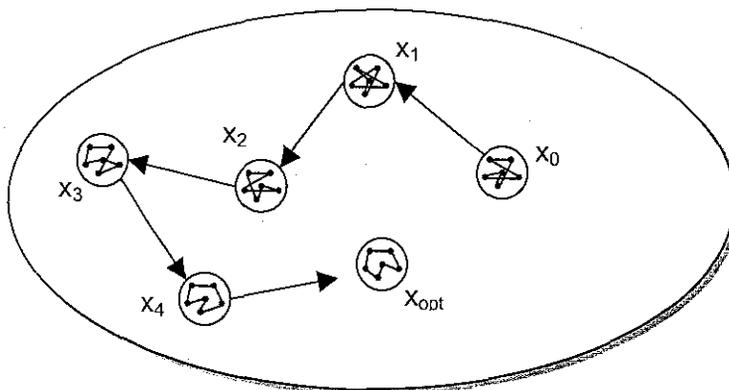


FIG. 2.3 - Approche séquentielle : amélioration par modifications élémentaires

2.2.3.3 Approches évolutives

Les algorithmes évolutionnaires sont inspirés des mécanismes d'évolution biologique et naturelle⁵. C'est pourquoi, un vocabulaire spécifique à la biologie a été emprunté et est communément admis dans la communauté. Par exemple, nous parlerons d'individu pour désigner une solution du problème ou bien encore de population afin de désigner un ensemble de solutions.

Contrairement aux méthodes constructives et amélioratives qui font intervenir une solution unique, les méthodes évolutives manipulent un ensemble de solutions admissibles (ou population) à chacune des étapes du processus de recherche. L'idée centrale consiste à utiliser régulièrement les propriétés collectives d'un ensemble de solutions différentiables, appelé population, dans le but de guider efficacement la recherche vers de bonnes solutions dans l'espace X . Ainsi, après avoir généré une population initiale de solutions, aléatoirement ou par l'intermédiaire d'une méthode constructive, une méthode évolutive tente d'améliorer la qualité moyenne de la population courante en ayant recours à des principes d'évolution naturelle.

⁵Charles Darwin évoqua, en 1859 dans "On The Origin of Species by Means of Natural Selection", les premiers principes de la théorie de l'évolution.

Le processus cyclique (ou génération) à la base de la plupart des méthodes évolutives est composé d'une phase de coopération et d'une phase d'adaptation individuelle qui se succèdent à tour de rôle :

- Dans la phase de coopération, les solutions de la population courante sont comparées puis combinées entre elles dans le but de produire des solutions inédites et de bonne qualité à long terme. L'échange d'information qui en résulte se traduit par l'apparition de nouvelles solutions admissibles qui héritent des caractéristiques prédominantes contenues dans les solutions de la population courante.
- Dans la phase d'adaptation individuelle, les solutions évoluent de manière indépendante en respectant un ensemble de règles prédéfini. Les modifications subies par chacune d'entre elles se font sans aucune interaction avec les autres solutions de la population. Une nouvelle génération de solutions est créée au terme de chaque phase d'adaptation individuelle.

La figure 2.4 illustre le principe de base d'une approche évolutive. Le but est de repérer des solutions aussi bonnes que possible en manipulant à chaque étape un ensemble de solutions localisées (Se) dans différentes régions prometteuses de l'espace X .

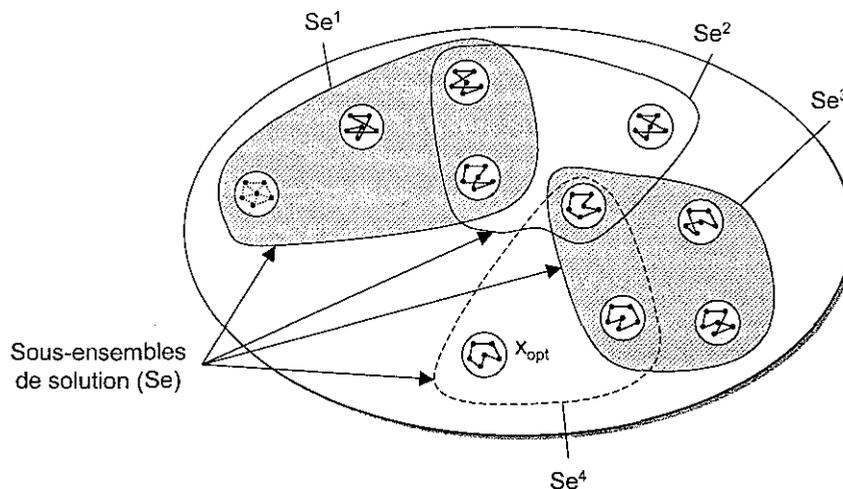


FIG. 2.4 – Approche évolutive : exploration de régions de l'espace de recherche

La caractéristique des approches évolutionnaires est leur capacité à explorer simultanément les différentes régions de l'espace de recherche. Des stratégies de diversification sont souvent utilisées dans le but de favoriser l'exploration, et d'éviter la convergence trop

rapide dans une même région. Le premier algorithme répertorié comme un algorithme évolutionnaire, est l'algorithme génétique introduit par Holland en 1975 [HOL 75].

2.2.3.4 Approches hybrides

Une dernière approche a été développée, et consiste à exploiter pleinement la puissance de recherche de méthodes itératives et de recombinaison des algorithmes évolutifs. Cette approche qualifiée d'*hybride* a permis de produire d'excellents, voire les meilleurs résultats sur des problèmes comme celui du voyageur de commerce. Il est également possible "d'*hybrider*" d'autres types d'approches, par exemple une heuristique gloutonne et une méthode de voisinage, ou un algorithme évolutif. Les algorithmes hybrides⁶ sont sans doute parmi les méthodes les plus puissantes. Malheureusement, les temps de calcul nécessaires peuvent devenir prohibitifs à cause du nombre d'individus manipulés dans la population. Une voie pour résoudre ce problème est la parallélisation de ces algorithmes.

Après avoir décrit les différentes approches de résolution des problèmes d'*optimisation combinatoire*, nous présentons les différents travaux déjà réalisés en conception de réseaux.

2.3 Etat de l'art en conception de réseaux

Dans les années 1990, une infrastructure de communication LAN se composait de plusieurs segments (Ethernet, Token Ring) interconnectés par des ponts. Aussi quelques recherches se sont focalisées sur la conception de réseau LAN et de l'organisation de leur interconnexion. Dans [ZAM 90], Zambre montre alors que les performances du réseau dépendent essentiellement des trois facteurs suivants : le nombre de noeuds actifs sur le réseau, la distance entre les noeuds (longueur du câble) et de l'application réseau (charge du réseau). Bien que la notion de segmentation améliore implicitement les performances du réseau (réduction du domaine de collision), elle est appliquée de façon empirique. C'est pourquoi, l'auteur suggère de prendre en compte plusieurs éléments tels que le trafic ou encore la topologie, afin de placer les éléments d'interconnexions de façon optimale. De la même manière, Khalil [KHA 91] montre que dans les grands réseaux locaux, le trafic est souvent concentré de manière locale. Ainsi, il est possible de trouver des groupes d'utilisateurs qui communiquent la plupart du temps exclusivement à l'intérieur de leur groupe. C'est à partir de ce constat qu'il propose de trouver la meilleure architecture

⁶aussi appelés «memetic algorithms».

d'interconnexion optimisant la performance et le faible coût à partir de la connaissance des besoins de communication des utilisateurs. Contrairement aux travaux précédents, Khalil propose un algorithme de résolution. Le problème étant complexe, il le divise alors en deux sous-problèmes : un problème de *clustering* (division des utilisateurs en cluster⁷) et un problème d'assignement (affectation de chaque cluster à un segment distinct). Il utilise un indice de "voisinage de trafic" (2.1) afin de mesurer la qualité de sa solution :

$$I = \frac{\sum_{1 \leq k \leq K} x_{k,k}}{\sum_{\substack{1 \leq i \leq K \\ 1 \leq j \leq K}} x_{i,j}} \quad (2.1)$$

avec K le nombre de cluster et $x_{i,j}$ le trafic allant du cluster i vers le cluster j .

Ces premiers travaux ont fait émerger un problème d'optimisation dans le domaine de la conception de réseaux locaux. Les recherches qui ont suivi se sont orientées naturellement vers l'application d'outils et de méthodes d'optimisation combinatoire. Elles considèrent le problème de manière plus formelle en le décomposant en deux étapes :

- Définition d'une fonction coût représentant les paramètres du système de communication,
- Utilisation d'heuristiques de résolution pour optimiser la fonction coût.

Nous avons recensé les principaux travaux de la littérature et nous les présentons succinctement dans la suite en fonction de leur évolution.

2.3.1 Interconnexions de réseaux locaux et métropolitains (LAN-MAN)

Fetterolf [FET 91] puis Ersoy [ERS 93] proposent de concevoir des architectures d'interconnexion LAN-MAN (Metropolitan Area Network). Le problème est alors de trouver une topologie de coût minimal pour l'interconnexion de réseaux locaux à travers un réseau métropolitain. Tous deux utilisent l'approche itérative du recuit simulé et comparent les résultats obtenus sur deux problèmes. Le premier correspond à une topologie de petite taille. Ils comparent alors leurs résultats avec une méthode exacte (énumération de toutes les solutions) et montrent qu'ils obtiennent l'optimum global. Le deuxième problème concerne une topologie de taille plus importante pour laquelle une méthode exacte n'est pas utilisable. Dans ce cas, les auteurs utilisent les méthodes statistiques afin de déterminer

⁷Un cluster est défini par un ensemble d'objets partageant les mêmes propriétés.

la probabilité d'obtenir une meilleure solution et prouvent ainsi que leurs algorithmes donnent de bons résultats.

2.3.2 Bi-segmentation de réseaux locaux

Dans [FAR 94b, FAR 94a, SON 94], le problème de conception d'un réseau local est appelé *Network Division Problem* (ou NDP), et s'attache à diviser un réseau en sous-réseaux de telle sorte que le trafic à l'intérieur de chaque segment soit maximisé et que le trafic entre les segments soit minimisé. Dans [FAR 94b], Farrell et al. présente un outil de conception de réseau "*Analyser*" couplant un analyseur réseau avec un algorithme génétique. Cet outil permet de diviser un réseau en deux sous-réseaux à partir d'une matrice de trafic obtenue par le biais de l'analyse réseau. La fonction coût de ce problème est alors énoncée comme suit :

$$f = t_1 + t_2 + c * t_{1 \rightarrow 2} \quad (2.2)$$

avec :

t_1 : trafic du sous-réseau 1

t_2 : trafic du sous-réseau 2

$t_{1 \rightarrow 2}$: trafic entre les deux sous-réseaux

c : facteur pénalisant les communications entre les deux segments

Songerwala [SON 94] complète ces travaux en ajoutant tout d'abord un facteur d'équilibre de charge entre les deux sous-réseaux. La fonction devient alors :

$$f = t_1 + t_2 + c * t_{1 \rightarrow 2} + |t_1 - t_2|^2 \quad (2.3)$$

Puis il propose d'utiliser d'autres heuristiques (recuit simulé et algorithme du *K-means*) et compare les résultats obtenus par ces trois approches suivant des critères quantitatifs (temps d'exécution et nombre d'itérations de l'algorithme) et qualitatifs (facilité d'implantation de la solution et respect des contraintes). Il montre alors que les trois méthodes fournissent en général de bons résultats avec cependant un avantage à l'algorithme du *K-means* pour la rapidité. L'algorithme génétique ainsi que celui du recuit simulé, explorent mieux l'espace des solutions et fournissent des résultats un peu plus performants sur le plan qualitatif.

2.3.3 Réseaux locaux commutés

Avec l'évolution des technologies réseaux, de nouveaux travaux ont émergé avec pour objectif de concevoir un réseau à base de commutateurs (ou pont multi-ports). Ainsi Elbaum [ELB 96] propose un nouvel algorithme génétique pour concevoir une architecture de communication locale de type arborescente (Fig. 2.5). Il décrit alors une solution à l'aide de trois représentations (ou codes) :

- codage des groupes : il représente la répartition des utilisateurs (ou stations) dans les différents segments du réseau. L'encodage est de la forme suivante :

$C_{groupe} = 1\ 2\ 3\ 4\ 5\ 0\ 1\ 2\ 3\ 4\ 5\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 3\ 3\ 3\ 3\ 3$ où le premier élément représente l'utilisateur 0 et le chiffre correspondant au numéro du segment auquel il est affecté. Ainsi l'ensemble des utilisateurs appartenant au segment 1 est : $\{0, 6, 12 - 19\}$

- un codage de l'arbre : il décrit la configuration en arbre en utilisant les arbres de Huffman. Par exemple, $C_{config} = 000\ 100\ 150\ 300\ 310\ 500$ où 000 représente le sommet racine de l'arbre, les éléments 100, 300, 500 les sommets de premier niveau et les éléments 150, 310 les sommets de deuxième niveau.

- un codage d'ordonnancement des groupes : il permet d'associer les sommets de l'arbre à des numéros de groupe.

$C_{ordre} = 2\ 5\ 3\ 4\ 1\ 0$ où le groupe 2 (codage 000 dans l'arbre) correspond à la racine de l'arbre, le groupe 5 (codage 100 dans l'arbre) est connecté au groupe 2, le groupe 3 (150) au groupe 5, le groupe 4 (300) au groupe 2 et ainsi de suite (Fig. 2.5).

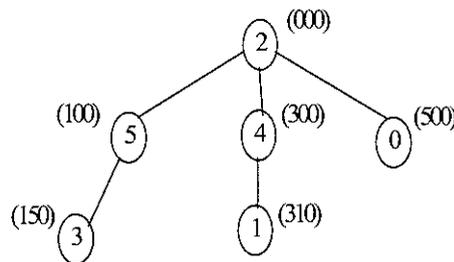


FIG. 2.5 – Topologie du réseau

Ensuite la fonction d'évaluation repose sur le délai moyen d'acheminement des messages. Pour cela, un segment LAN est modélisé par une file d'attente $M/M/1$ ce qui

permet d'évaluer le temps d'attente dans chaque segment traversé. De plus, il faut ajouter les temps d'attente dans les ponts qui sont supposés ici fixes. On pose alors :

$B_{i,j}$: temps d'attente dans le pont entre le segment i et j

$F_{i,j}$: trafic total passant par un pont reliant les segments i et j

E : trafic total

L_i : trafic total dans le segment i

C_i : capacité du segment i

P : nombre de groupes

Le temps moyen d'acheminement des messages est donc :

$$D = D_{segment} + D_{pont} = \frac{1}{E} \left[\sum_{k=1}^P \frac{L_K}{C_K - L_K} + \sum_{i=1}^P \sum_{j=1}^P F_{i,j} \cdot B_{i,j} \right] \quad (2.4)$$

L'algorithme génétique utilise ensuite des opérateurs de croisement et de mutation de façon à minimiser D . Un ensemble de résultats est donné dans [ELB 96] montrant les performances de l'algorithme.

Dans [NUS 97], Nusekabel utilise également une modélisation du réseau par files d'attente de type $M/M/1$ et modifie la fonction 2.4 afin de concevoir un réseau commuté. La fonction objectif est alors relative au délai d'acheminement des différents segments et des éléments d'interconnexion traversés par les données. Elle prend également en compte un critère d'équilibrage de charge entre les segments. L'expression à minimiser cherche alors le meilleur rapport entre délai et équilibrage des charges. Elle est donnée par la formule suivante :

$$D = \frac{\sum_{i=1}^S L_{Si} B_{Si}}{S} + \frac{\sum_{i=1}^G \frac{L_{Gi}}{C_{Gi} + \beta L_{Gi}}}{G} + \frac{1}{K} \sqrt{\frac{\sum_{i=1}^G \left[\left(\frac{N_{Gi}}{C_{Gi}} \right) - \left(\frac{\sum_{i=1}^G \frac{N_{Gi}}{C_{Gi}}}{G} \right) \right]^2}{G}} \quad (2.5)$$

avec :

B_{Si} : temps d'attente dans le commutateur i

L_{Si} : trafic dans le commutateur i

C_{Gi} : capacité du segment i

L_{Gi} : trafic dans le segment i

S : nombre de commutateurs

N_{Gi} : nombre de machines sur le segment i

G : nombre de segments

β : facteur d'équilibrage

K : constante qui minimise l'impact du dernier terme comparé aux deux autres

Parallèlement au développement de méthodes évolutives, l'auteur utilise l'heuristique tabou pour minimiser la fonction 2.5. Il évalue alors son algorithme avec d'autres (Algorithme génétique, recuit simulé, etc.). Il obtient de meilleurs résultats pour des durées de traitements identiques.

Pour finir, une autre technique a été développée par Youssef et al. [YOU 00]. Elle est basée sur une approche hybride dans laquelle un algorithme itératif est couplé avec des règles de logique floue qui guident l'algorithme lors de la phase de recherche.

2.3.4 Synthèse des travaux antérieurs

L'ensemble des travaux répertoriés en conception de réseaux locaux, montre une évolution vers une approche méthodologique commune de résolution du problème. Cette approche est constituée de trois phases :

1. **Modélisation** : La théorie des files d'attente est utilisée afin de modéliser le réseau. Cette modélisation permet d'approximer le comportement temporel du réseau et des équipements d'interconnexion. Ainsi, le délai moyen d'acheminement des données dans le réseau peut être calculé et est utilisé comme fonction coût.
2. **Élaboration d'heuristiques de résolution** : Les méthodes mathématiques utilisées sont généralement des approches amélioratives, évolutives, voire hybrides. Nous n'avons repertorié aucune proposition utilisant une approche constructive pour la conception de réseau. De manière intuitive, nous pouvons déduire que cette technique n'est pas adaptée à ce type de problème. La construction progressive d'une solution consisterait par exemple à placer un premier noeud sur un segment. Ensuite, le noeud échangeant le plus d'informations avec le noeud précédent serait ajouté au segment (sans remettre en cause les choix antérieurs) et ainsi de suite. La solution finale dépend alors fortement de l'initialisation du processus et la vision du problème est trop réduite.
3. **Evaluation de performance** : L'évaluation de la solution se fait implicitement au travers de la fonction coût. L'évaluation de performance de l'algorithme est réalisée

en proposant parallèlement d'autres méthodes afin de les comparer entre elles selon des critères tels que le temps d'exécution ou le nombre d'itérations.

Ces travaux sont appliqués à la conception de réseaux locaux d'informations et ne sont pas toujours adaptés au contexte industriel, notamment au niveau des caractéristiques temps-réel du trafic ou de la fiabilité des architectures. De plus, l'utilisation d'une fonction coût représentant le délai d'acheminement moyen des messages, peut se révéler longue et fastidieuse avec la taille du problème. C'est le cas des solutions basées sur une approche évolutive comme les Algorithmes Génétiques, qui manipulent un ensemble de solutions. Dès lors, l'algorithme évalue chaque solution au détriment du temps consacré à la recherche d'une solution optimale. Pour finir, il est à noter qu'il n'existe pas d'architectures de réseaux locaux servant de référentiel et permettant d'évaluer les algorithmes entre eux. Par conséquent, tous les travaux précédents proposent à chaque fois plusieurs méthodes de résolution.

2.3.5 Positionnement de notre contribution

Les réseaux de communication sont souvent modélisés par des graphes ou leurs généralisations (hypergraphes⁸, graphes valués ou pondérés). Rappelons qu'un graphe (respectivement graphe orienté) $G = (V, E)$ est défini par un ensemble V de sommets et un ensemble E d'arêtes (resp. arcs) formé de paires (ou couples) de sommets. Dans notre cas, un sommet représentera alors un équipement de communication (automate, E/S déportées, capteurs, IHM, stations « industrielles ») ou d'interconnexion (commutateur, hub, routeur). Une arête représentera une liaison physique entre les éléments représentés par les sommets. Des informations supplémentaires pourront être ajoutées sur les sommets ou les arêtes telles que des valuations (ou poids) qui correspondront à des capacités, des largeurs de bande ou des coûts. L'ensemble de ces objets permettent de modéliser des entités du monde réel au delà des problèmes d'interconnexion et de communication. L'étude des propriétés et des comportements de ces entités est alors effectuée en profitant des nombreux résultats existant dans la littérature en Mathématiques Discrètes, soit sur des propriétés structurelles telles que la connectivité, soit sur des aspects algorithmiques non spécifiques à la théorie des graphes comme le partitionnement.

⁸Un hypergraphe $H = (V, E')$ est une généralisation d'un graphe où la cardinalité des éléments de E' peut être plus grande que 2. Une arête reliant plus de deux sommets est alors qualifiée d'hyper-arête.

En conception de réseau, une telle modélisation est systématiquement simplificatrice, mais elle retient les paramètres critiques entrant en jeu. Nous cherchons alors :

- soit à construire le meilleur objet possible (satisfaisant les contraintes et optimisant une fonction de coût); on parle alors de problème de conception («design»); l’approche est en général duale puisque la tâche de construction s’accompagne de la preuve de la qualité de celle-ci;
- soit à déterminer les propriétés de certains objets combinatoires.

Nous sommes alors conduits à utiliser les outils développés en théorie des graphes ou à déterminer des algorithmes «efficaces» afin de résoudre ces questions. L’état de l’art précédent montre que la majorité des travaux repose sur la théorie des files d’attente et non sur la théorie des graphes. Nous proposons de les coupler afin d’étudier dans un premier temps les évaluateurs pertinents pour les applications réseau issus de la théorie des files d’attente (délai d’acheminement d’un message, charge d’un réseau, ...) et de les traduire en contraintes utilisables dans la théorie des graphes. Dans un deuxième temps, nous générons l’architecture réseau à partir de ces contraintes, puis la solution obtenue est ensuite évaluée par la théorie des files d’attente. Dans la section suivante, nous présentons le problème de partitionnement de graphes, et ses méthodes de résolution afin d’observer leur adéquation au problème de conception de réseaux industriels.

2.4 Partitionnement de graphes appliqué à la conception de réseaux

2.4.1 Le partitionnement de graphes

Les données de l’application sont utilisées pour générer un graphe où chaque entité élémentaire est représentée par un nœud du graphe et les dépendances entre ces entités par des arcs. Si les entités et les dépendances ne sont pas uniformes, alors le graphe peut être pondéré par le biais de poids sur chaque nœud et chaque arc. Le problème est de partitionner un graphe (généralement de grande taille) en un nombre de parties ou d’agrégats non strictement fixé à l’avance, en cherchant à optimiser un critère mixte :

Critère 1 *Minimiser la valeur totale des arêtes inter-agrégats,*

Critère 2 *Equilibrer approximativement les cardinalités des parties.*

La figure 2.6 illustre une partition d'un graphe en deux sous ensembles de même cardinalité.

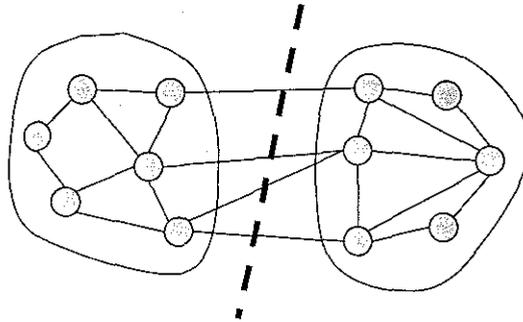


FIG. 2.6 – Un graphe et une partition de ce graphe.

Les méthodes de partitionnement de graphes sont encore aujourd'hui très étudiées et visent des domaines d'applications très larges tels que :

- le placement de tâches sur des architectures parallèles : affectation optimale des tâches sur plusieurs processeurs dans le but de diminuer le volume de communication et de profiter de la localité des données,
- la conception de circuits VLSI (Very Large Scale Integration) avec la disposition optimale des composants sur un circuit intégré,
- le calcul scientifique parallèle : la distribution équilibrée, le partitionnement des structures de données irrégulières de type "grande matrice creuse", le recouvrement calcul/communication,
- la décomposition de domaine : processus de partitionnement d'un domaine de données d'une application en sous-domaines qui peuvent travailler individuellement en parallèle pour accélérer la solution d'un grand problème. Cette technique est généralement utilisée pour le partitionnement d'applications maillées non structurées pour paralléliser l'exécution. De nombreux algorithmes de décomposition de domaine sont disponibles,
- la conception de réseaux et de systèmes d'information.

2.4.1.1 Le mécanisme de partitionnement

Il existe diverses formulations mathématiques du problème de partitionnement de graphes. Nous présentons ci-après la *p* arête-partition, c'est-à-dire une partition du point

de vue des arêtes, et nous supposerons les graphes non orientés dans la suite.

Considérons le problème classique, où il s'agit de partitionner l'ensemble des sommets en un nombre de parties fixé en minimisant le critère "nombre d'arêtes coupées". Le problème s'énonce très synthétiquement :

Soit $G = (V, E)$ un graphe non orienté où V est l'ensemble des sommets et E est l'ensemble des arêtes. Les poids d'une arête e et d'un sommet v sont respectivement désignés par $W(e)$ et $W(v)$. La p *arête-partition* (pour $p \geq 2$) est une partition de V en p sous-ensembles disjoints V_1, V_2, \dots, V_p tels que :

- $V = \bigcup_{k=1}^{k=p} V_k$
- $V_i \cap V_j = \emptyset \quad \forall i \neq j$
- $|V_k| \approx |V_l| \quad \text{pour } k \neq l$
- $W(V_i) \approx \frac{W}{p} \quad \text{pour } i = 1, 2, \dots, p$ où W et $W(V_i)$ sont les sommes W des poids des noeuds de V et de V_i
- $E_0 = \{e(v_i, v_j) \mid v_i \in P_k, v_j \in P_l \quad \forall k \neq l\}$

En d'autres termes, E_0 est l'ensemble des arêtes "coupées" par la partition de V et l'objectif est de minimiser $W(E_0)$, la somme des poids des arêtes de cet ensemble (i.e. la *taille de coupure*).

2.4.1.2 Complexité

Le problème de partitionnement est un problème *NP-complet* [GAR 76, BUI 92]. Certains algorithmes de résolution garantissent de trouver une solution optimale sur des graphes de faible taille [KAR 97], mais sont trop lents sur des graphes plus larges. C'est pourquoi, les chercheurs poursuivent le développement d'heuristiques permettant de résoudre de manière approchée le problème de partitionnement de graphes. Dans la suite, nous présentons une synthèse non exhaustive des différentes techniques de partitionnement, réalisée à partir de différents travaux [LAS 93, CIA 94, ALP 95, POT 97, FJA 98].

2.4.2 Les méthodes de résolution pour le partitionnement

Il existe deux types d'heuristiques, les unes reposant sur l'exploitation de la globalité de l'information disponible sur le graphe (méthodes "globales") et les autres sur un algorithme de déplacement de proche en proche dans le graphe (méthodes "locales").

Parmi les principales méthodes locales, on retrouve essentiellement des méthodes combinatoires séquentielles (ou amélioratives) telles que les algorithmes de Kernighan-Lin, Fiduccia-Mattheyses, le recuit simulé ou encore la méthode tabou, mais aussi évolutives comme les algorithmes génétiques. Parallèlement aux approches purement combinatoires, se sont développées des méthodes dites globales. Elles consistent à plonger préalablement le graphe abstrait dans un espace métrique, souvent euclidien, et à appliquer ensuite des algorithmes de classification afin d'améliorer les performances des heuristiques classiques basées bien souvent sur des améliorations successives. Cette approche conceptuellement très différente a connu un grand développement cette dernière décennie. Parmi ces méthodes globales, on retrouve les méthodes dites géométriques telles que les algorithmes CND (Coordinate Nested Dissection) et RCB (Recursive Coordinate Bisection) mais aussi des méthodes basées sur la décomposition du Laplacien discret du graphe. Ces méthodes sont dites *spectrales*.

2.4.2.1 Les méthodes globales

L'idée sous-jacente consiste à transformer le problème combinatoire initial en un problème de classification en définissant une bijection entre les sommets du graphe et des points d'un espace vectoriel. En plongeant G dans un espace vectoriel, l'objectif est de transférer les relations d'interdépendance entre les composantes de G dans la représentation qui sert de support au partitionnement. La plupart de ces méthodes sont récursives et permettent d'obtenir deux partitions à chaque itération. On parle alors de bisection du graphe. Une k -partition du graphe peut être réalisée en appliquant récursivement la bisection jusqu'à obtention du nombre de partitions souhaité. Les méthodes globales sont souvent utilisées en combinaison avec des méthodes d'amélioration locale.

Les techniques géométriques

Les techniques de partitionnement géométriques [BER 87, GIL 98, PAT 98] se réfèrent uniquement aux coordonnées des nœuds maillés, et non à leur connectivité. Aussi il n'existe pas pour ces méthodes de concept de taille de coupure, mais la qualité de la partition est relative, par exemple, à la longueur de la frontière entre les sous-graphes. Habituellement, ces techniques divisent directement les éléments maillés, et non le graphe modélisant la structure de calculs. C'est la raison pour laquelle ces techniques sont souvent référencées comme techniques de partitionnement maillées. En général, les techniques géométriques

sont très rapides mais de faible qualité par rapport à d'autres algorithmes basés sur la connectivité des éléments maillés. Il existe différentes variantes :

- *L'algorithme RCB* : Recursive Coordinate Bisection connu aussi comme Coordinate Nested Dissection CND [BER 87, HEA 95]

Le principe de fonctionnement de l'algorithme RCB est le suivant : les centres de masse des éléments de la maille sont calculés et ceux-ci sont projetés sur l'axe qui correspond à la plus longue dimension de la maille. Ceci crée une liste ordonnée d'éléments de la maille. Ensuite la liste est coupée à la moitié pour produire une bisection. Chacun des sous-domaines peut alors être subdivisé par la même technique [BER 87]. La figure 2.7 illustre une division en 8-partitions calculée par cette méthode. Elle montre les différents centres de masse des éléments de la maille et les bisections récursives calculées. La ligne en trait plein divise la maille entière. Puis, les lignes en pointillé divisent les deux sous-domaines. Ensuite, les lignes en tiret et pointillé divisent les sous-graphes résultants. La figure 2.7.b montre les éléments de la maille colorés selon leurs sous-domaines. Les techniques RBC sont extrêmement rapides, consomment peu de mémoire et sont faciles à paralléliser.

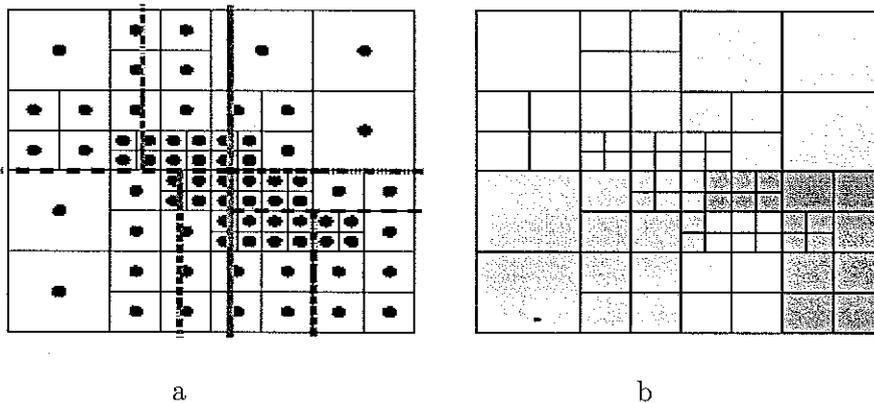


FIG. 2.7 – Une maille partitionnée selon les centres de masse

- *L'algorithme RIB* : Recursive Inertial Bisection [POT 97]

Ce type de technique de partitionnement est souvent utilisé pour partitionner des mailles (Fig. 2.8). Il tente de minimiser les communications entre les frontières des sous-domaines en coupant la maille au milieu de sa dimension la plus longue. Par exemple la figure 2.8.a montre une maille bipartitionnée par rapport à l'axe x. La figure 2.8.b montre la même maille bipartitionnée selon l'axe y. Il est bien évident

que la coupe choisie est celle de la figure 2.8.a, car c'est celle qui a la plus petite taille.

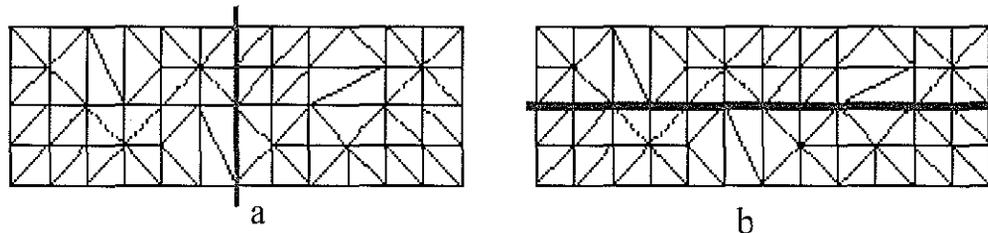


FIG. 2.8 – Une maille bipartitionnée par rapport aux axes X,Y

Il existe également d'autres techniques telles que l'algorithme SFC (ou Space-Filling Curve) et SC (ou Sphere-Cutting)[FJA 98]

Les Méthodes spectrales

Une autre approche pour résoudre un problème de bisection, est de le formuler comme une optimisation d'une fonction quadratique discrète. Aussi des méthodes de partitionnement de graphes appelées méthodes spectrales "*relâchent*" ce problème d'optimisation discrète en le transformant en problème continu.

- *Bisection Spectrale* : Les coordonnées des sommets du graphe sont fonction des vecteurs propres de la matrice Laplacienne $Q = D - A$ où A est la matrice d'adjacence de G et D la matrice diagonale des degrés des sommets. Chacune de ses coordonnées associée à l'un des sommets du graphe donne une mesure de la dépendance des sommets entre eux. Ainsi, la mise en ordre (croissante ou décroissante) des composantes des vecteurs propres, induit l'ordonnement des sommets du graphe, assurant que leur dichotomie selon cet ordre permet d'obtenir deux groupes les plus indépendants possible. Le travail consiste ensuite à déterminer le meilleur séparateur permettant de minimiser les dépendances.
- *Bisection Spectrale Réursive*⁹ : L'approche réursive a été développée par Pothén et Simon [POT 90, SIM 91]. Elle est réursive comme la plupart des méthodes globales et permet un partitionnement récurrent de chaque sous-graphe obtenu à partir d'un graphe initial. Ainsi, une k-partition peut être calculée en utilisant l'algorithme RSB.

⁹ou Recursive Spectral Bisection (RSB) en anglais.

L'algorithme RSB comparé aux approches géométriques, permet d'obtenir des partitions de meilleure qualité, mais est handicapé par le coût associé à la détermination des vecteurs propres de la matrice laplacienne du graphe considéré. Pour cela, de nombreux travaux sont menés pour réduire le temps d'exécution de l'algorithme RSB. Usuellement, on utilise une méthode de diagonalisation de type Lanczos mais il existe d'autres méthodes comme la méthode de Davidson. Dans [HOL 98], Holzrichter compare les performances obtenues par les deux algorithmes et montre l'efficacité de ce dernier. Parallèlement, [BAR 93] et [HEN 93b] ont proposé un autre algorithme reposant sur la génération d'une suite de graphes "grossiers" ayant de moins en moins de sommets afin de réduire le coût de la méthode par rapport à l'algorithme initialement proposé. Le calcul est alors effectué sur le graphe le plus grossier, puis propagé aux graphes précédents par expansion. Ces approches qualifiées de multi-niveaux seront abordées dans la partie 2.4.2.3.

Pour conclure sur ces méthodes spectrales, mentionnons également les travaux de Hendrickson et Leland [HEN 93a] qui proposent un algorithme permettant de réaliser directement le découpage des sommets d'un graphe en quatre ou en huit sous-ensembles à partir du calcul de plusieurs valeurs propres et vecteurs propres associés de la matrice Laplacienne. Ceci étant, le coût de ces algorithmes s'avère prohibitif [CIA 94].

2.4.2.2 Les méthodes locales

Contrairement aux techniques géométriques, les méthodes locales tentent de regrouper les sommets fortement connectés sans prendre en compte leur position géographique. Ces méthodes reposent essentiellement sur des techniques purement combinatoires. On retrouve ainsi les différentes approches détaillées dans la partie 2.2.3.

Algorithme LND (Levelized Nested Dissection) [GEO 73]

L'algorithme LND est basé sur une approche constructive durant laquelle une solution est élaborée à partir des adjacences des sommets du graphe. Il débute avec un sous-graphe contenant seulement un sommet auquel on ajoute successivement des sommets adjacents. Pour cela, l'algorithme utilise un marquage de chaque sommet du graphe en fonction de son niveau de profondeur lors de la phase de recherche. Les sommets marqués constituent le premier sous-graphe, et les sommets non marqués constituent l'autre sous-graphe. La figure 2.9 illustre cette technique. Un premier sommet est marqué "0" puis ses sommets adjacents sont numérotés "1". Ensuite tous les sommets adjacents aux sommets marqués

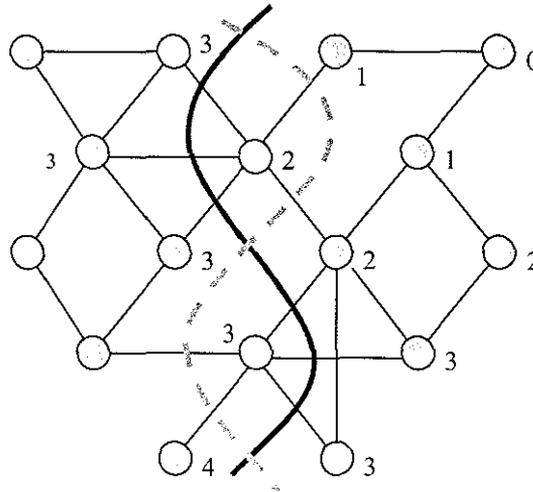


FIG. 2.9 – Partitionnement avec l’algorithme LND

"1" sont marqués "2" et ainsi de suite jusqu'à ce que la moitié des sommets soit marquée. La ligne pleine indique la frontière de la partition, dont la taille de la coupe est égale à 6. La ligne en pointillé montre la meilleure solution avec une taille de coupe égale à 4 qui ne peut pas être trouvée avec ce sommet initial. Le choix du premier sommet (niveau 0) est très important et influe considérablement sur la qualité de la partition. En général, il est conseillé de choisir un sommet périphérique au graphe pour obtenir une bonne solution. Dans certains cas, la qualité du partitionnement est pauvre, aussi dans la pratique on exécute plusieurs fois l'algorithme LND en commençant avec des sommets différents et on choisit la meilleure partition.

Algorithme Kernighan-Lin / Fiduccia-Mattheyses

Kernighan et Lin ont proposé en 1970 une des premières méthodes de partitionnement de graphes [KER 70]. Spécialement conçue pour le problème de bipartition de graphes, cette méthode améliorative utilise une partition initiale, et tente de l'améliorer tout en conservant le même nombre de sommets dans chacun des sous-graphes. Cette technique est souvent employée lors de phases dites de *raffinement* dans d'autres algorithmes de recherche. L'idée est la suivante :

Soit une bissection $\{V_1, V_2\}$ d'un graphe $G = \{V, E\}$, le traitement par l'algorithme de K-L consiste à trouver deux sous-ensembles V'_1 et V'_2 (avec $|V'_1| = |V'_2|$) contenus respectivement dans V_1 et V_2 puis de les permuter entre les partitions de façon à réduire

au maximum la taille de la coupe (Figure 2.10). Ces permutations se font jusqu'à ce que l'on ne puisse plus améliorer la qualité de la partition. L'identification des ensembles V'_1 et V'_2 est basée sur un algorithme glouton recherchant de manière itérative une paire de sommets (chacun appartenant à des sous-graphes différents), puis à les permuter si cela diminue la taille de la coupe. La première paire sélectionnée est celle qui apporte la plus forte amélioration et ainsi de suite, sachant qu'un sommet ne peut être déplacé qu'une seule fois durant une même procédure. Lorsque toutes les paires ont été trouvées, une nouvelle procédure peut alors commencer.

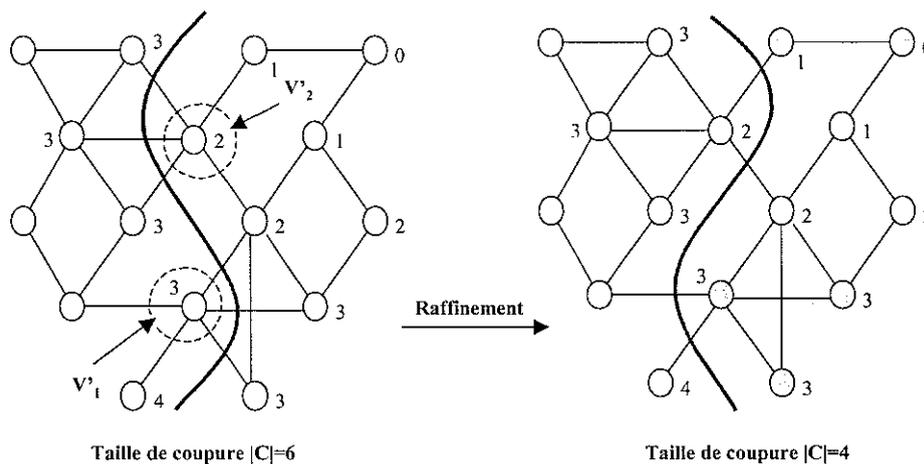


FIG. 2.10 – Raffinement par l'algorithme de KL

Cet algorithme a été appliqué avec succès au problème de bipartition de graphe [KER 70]. En général l'algorithme KL converge assez rapidement, limitant ainsi le nombre de procédures à réitérer. Cependant, de nombreuses variantes ont été, et sont encore développées à l'heure actuelle afin de réduire le temps d'exécution. Une des variantes les plus connues est celle de Fiduccia et Mattheyses [FID 82] qui ont proposé une modification de l'algorithme de KL, appelée algorithme de Fiduccia-Mattheyses (ou FM). Il diffère du précédent uniquement par le fait qu'il déplace les sommets un par un et non pas par couple. Cela réduit le temps de traitement sans altérer son efficacité.

Le recuit simulé (ou simulated annealing)

Le recuit simulé a été développé par S. Kirkpatrick et al. [KIR 83]. Il est basé sur un algorithme de simulation de recuit de métaux et s'inspire de modèles de la physique

statistique [MET 53]. L'idée principale consiste à autoriser, à chaque itération, le choix d'une solution dont la qualité décroît. Ainsi, on considère aléatoirement une solution S' dans le voisinage de la solution courante S . La solution considérée est choisie comme solution courante de la prochaine itération, si la fonction économique décroît. Sinon, on lui affecte une probabilité de sélection donnée par l'expression $e(-\delta/T)$ dérivée de la physique statistique, avec $\delta = q(S) - q(S')$, où $q(S)$ est la qualité de la solution S , et T un nombre positif qui représente une température initiale. Si aucun critère d'arrêt n'est satisfait après un certain nombre d'itérations, le processus se poursuit par une décroissance régulière de la température jusqu'à obtention d'une solution satisfaisante.

Johnson et al. [JOH 89] ont adapté le recuit simulé au partitionnement de graphes. En comparant leurs résultats avec ceux obtenus par l'algorithme de KL, ils ont alors montré que la méthode du recuit ne fournit pas toujours une meilleure solution (selon la structure du graphe) bien qu'elle soit une alternative aux approches gloutonnes, telles que les algorithmes KL ou FM.

La méthode tabou

La méthode tabou est une autre technique élaborée par Glover [GLO 89, GLO 90]. Elle est basée sur la notion de mouvements interdits (ou tabou). Chaque itération consiste à trouver le mouvement qui nous donne la meilleure solution dans le voisinage de la solution courante, sachant que certains mouvements sont interdits. Parfois, on choisit une solution qui détériore légèrement la solution courante pour sortir des minima locaux. L'inverse du mouvement effectué à chaque itération est rajouté dans une liste, appelée liste tabou, qui contient les mouvements interdits. Initialement, la liste tabou est vide.

Rolland et al. [ROL 96] ont utilisé cette méthode pour le partitionnement de graphes et ont obtenu de bons résultats, en terme de qualité de la solution et de temps d'exécution, par rapport à des approches de type KL et recuit simulé.

Les algorithmes génétiques

Ces algorithmes sont basés sur une analogie avec l'évolution des espèces [GOL 89]. Ils partent d'une population initiale (ensemble de solutions) à partir de laquelle ils effectuent des croisements pour engendrer de nouvelles configurations. Ils gardent les meilleures populations résultantes pour effectuer d'autres croisements. Afin d'éviter les minima locaux, de temps à autre, les espèces sont soumises à des mutations. Des travaux de recherche sur l'utilisation des algorithmes génétiques pour le partitionnement de graphe [MAI 94, BUI 96]

ont permis de montrer l'intérêt d'une approche évolutive. Dans [BUI 96], Bui et al. ont proposé un algorithme génétique qu'ils comparent avec les algorithmes de KL et de recuit simulé. Ils obtiennent expérimentalement des partitions de qualité comparable voire supérieure.

2.4.2.3 Les processus multi-niveaux

Récemment, une nouvelle classe d'algorithmes de partitionnement, basés sur le paradigme multi-niveaux, a été développée [HEN 95, KAR 95, GUP 96]. Ce paradigme se décompose en trois phases (Fig 2.11) :

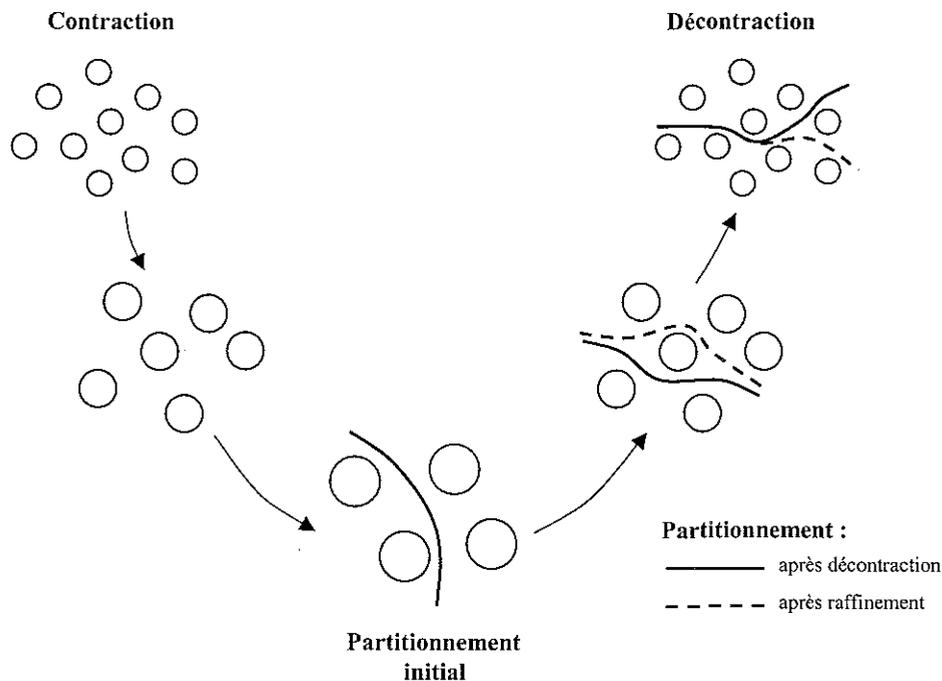


FIG. 2.11 – Processus de partitionnement multi-niveaux de graphes.

La phase de *contraction* (ou coarsening) :

Elle consiste à réduire la taille du graphe initial en le contractant tout en conservant les propriétés du graphe d'origine. A chaque étape, des sommets sont appariés et des arêtes sommées. Lorsque le graphe est suffisamment petit, la phase de partitionnement est alors appliquée sur le graphe contracté.

La phase de *partitionnement* :

Le graphe contracté est de taille réduite et tous les algorithmes classiques de partitionnement sont capables de donner rapidement une bonne partition du graphe. Dans [BAR 94], Barnard et al. proposent un algorithme multi-niveaux MRB (Multi-level Recursive Bisection) utilisant la technique spectrale RSB. Cependant d'autres méthodes, telles que l'algorithme LND [KAR 98], peuvent être employées.

La phase de *décontraction* (ou uncoarsening) :

Après avoir obtenu une partition du graphe contracté, celle-ci est reportée sur les graphes intermédiaires jusqu'à ce que l'on obtienne une partition du graphe initial. Cette phase de décontraction est souvent accompagnée d'une étape de raffinement. Ainsi, la méthode "multilevel-KL" utilise l'algorithme de KL lors de la phase de décontraction et permet d'obtenir des résultats comparables à ceux de KL. Cependant la méthode est beaucoup plus rapide notamment pour de grands graphes [HEN 95].

2.5 Conclusion

Nous avons présenté dans ce chapitre le problème de conception de réseaux locaux et son évolution au cours des dernières années. Ce problème d'optimisation a été traité avec différentes approches combinatoires permettant d'optimiser une fonction coût obtenue en général par le biais d'une modélisation par file d'attente. A partir de l'état de l'art en conception de réseaux, nous proposons de modéliser le réseau de communication à l'aide de la théorie des graphes. Le problème de conception de réseaux devient alors un problème de partitionnement de graphes pour lequel nous avons repertorié les méthodes de résolution les plus connues. Avec un domaine d'applications plus large, les travaux sur le problème du partitionnement sont nombreux et d'autres méthodes, autres que purement combinatoires, sont développées. Nous avons choisi d'étudier et d'utiliser une approche globale et une approche locale à des fins de conception de réseaux de communication. Il s'agit notamment des algorithmes spectraux et génétiques. Ces deux méthodes ont été retenues pour plusieurs raisons :

- Les algorithmes génétiques présentent des caractéristiques intéressantes les rendant "populaires". Leur pouvoir de représentation leur permet de s'adapter facilement au problème traité. Leurs points forts sont une certaine robustesse par rapport aux différentes instances du problème.

- Les techniques spectrales sont très utilisées et ont montré leur efficacité dans de nombreux problèmes d'optimisation combinatoire. Elles ont également permis de résoudre de manière exacte certains problèmes tels que la sérialisation, ce qui selon [ATK 99] positionne cette technique entre les méthodes exactes et les heuristiques.

Le chapitre 3 est consacré au développement de notre approche en conception de réseaux *Ethernet Industriel*. Dans un premier temps, nous présentons la modélisation des communications sous forme de graphes, ainsi qu'une étude des différentes topologies de réseaux commutés. Nous proposons alors une topologie redondante permettant de répondre aux besoins de fiabilité et de disponibilité des réseaux industriels. Ensuite, nous optimisons les délais d'acheminement des messages sur cette architecture en réalisant le partitionnement du graphe associé. Ainsi, une partition du graphe représente la répartition des équipements industriels (noeuds du graphe) sur les différents commutateurs. Pour cela, nous utilisons deux heuristiques de résolution : les Algorithmes Spectraux et les Algorithmes Génétiques. Les fondements de chaque technique de partitionnement sont alors étudiés. Ils permettent d'adapter chaque algorithme au problème de conception.

Chapitre 3

Heuristiques de conception de réseaux *Ethernet Industriel* commutés

3.1 Modélisation des données du problème

3.1.1 Les communications

Le problème de conception de réseaux locaux peut intervenir soit dans une phase d'installation d'un réseau (pré-étude), soit dans une phase d'optimisation d'un réseau existant (*reengineering*). Dans le premier cas, les besoins en communications doivent être connus à l'avance afin de réaliser l'étude. Dans le deuxième cas, ces communications peuvent être observées sur le réseau afin d'en déduire le scénario d'échanges. Cette observation peut être réalisée à l'aide d'un logiciel "*analyseur de trafic*" (ou sniffer). C'est notamment le cas dans la majorité des travaux de conception présentés dans la partie 2.3, qui concernent la conception de réseaux locaux d'entreprises ou universitaires¹. Cependant, il est à noter qu'il est difficile de prendre en compte un trafic dynamique dans de telles conditions.

Dans le cas des réseaux locaux industriels, les échanges entre les différents équipements et leur volume sont dans la plupart des cas parfaitement connus. Les données échangées sur ce type de réseau sont diverses. Le tableau 3.1 présente les différents types de données

¹Farell et al. [FAR 94b] ont réalisé un outil appelé *Analyzer*, combinant un analyseur réseau et un algorithme d'optimisation. Ils ont utilisé leur outil sur le réseau du département Computer Science de l'Université CURTIN.

avec la taille des messages, les délais de transmission recherchés et la nature périodique ou apériodique des échanges.

| | Volume (en bits) | Nature | Délais (en ms) |
|------------------------|------------------|-------------|-----------------|
| Capteurs / Actionneurs | 1 à 100 | périodique | 1 à 50 |
| Alarmes | 1 à 100 | apériodique | ~ 0 |
| Robots (vision) | 10^6 à 10^7 | périodique | 50 |
| Terminaux / Automates | 10^2 à 10^3 | apériodique | 10^2 à 10^3 |
| Ordinateurs (fichiers) | 10^3 à 10^6 | apériodique | 10^3 à 10^4 |

TAB. 3.1 – Types de données

Nous pouvons alors établir un graphe d'échanges entre les différents équipements. Chaque noeud représente un équipement, et chaque arc représente les données échangées entre les noeuds (Fig. 3.1). Par la suite, la forme matricielle (ou graphe d'adjacence) est utilisée pour des raisons de facilité de traitement. La pondération du graphe permet de représenter les différents volumes échangés sur une même période. Cependant, les paquets échangés sur les réseaux industriels sont souvent inférieurs à 46 octets (taille minimale du champ "données" dans une trame Ethernet). Par conséquent, les matrices présentées par la suite seront pour la plupart unitaires.

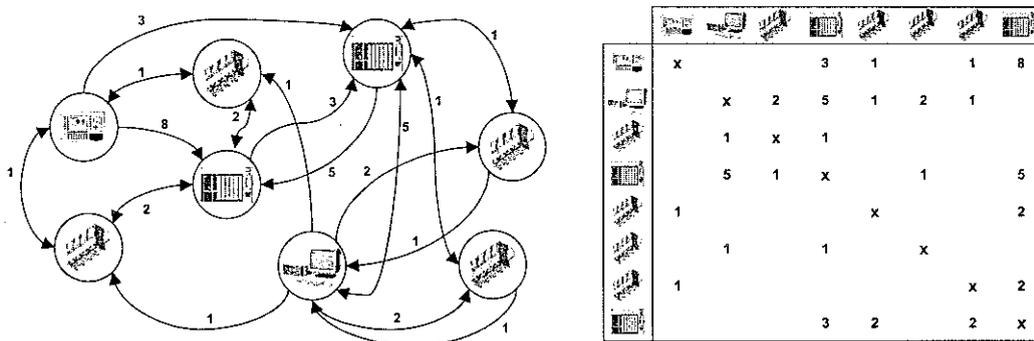


FIG. 3.1 – Représentation des communications

3.1.2 La topologie

Avant les années 80, la communication entre deux noeuds s'effectuait sur une ligne point-à-point. Dans ce cas, des liaisons propriétaires, souvent basées sur *RS232* ou *RS422*,

étaient utilisées. Ces liaisons étaient difficiles à installer, maintenir et surtout très coûteuses. Puis, le concept des réseaux (ou bus) de communication a permis d'avoir plus de souplesse dans l'installation, et de réduire les coûts d'installation et de maintenance (Fig. 3.2).

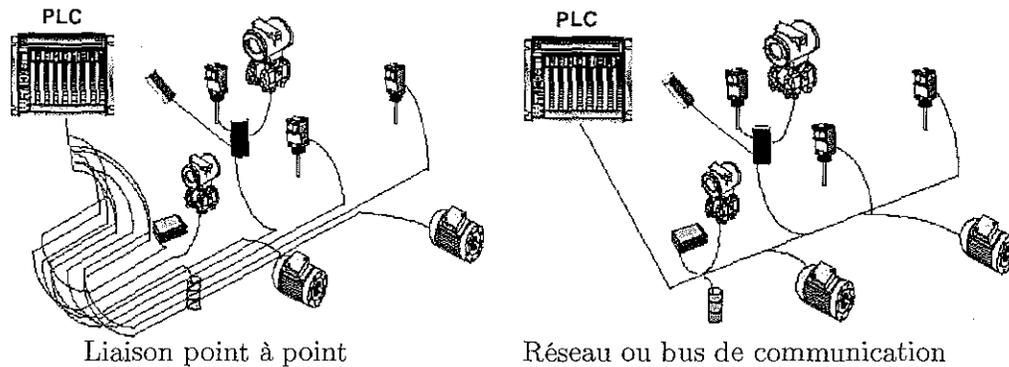


FIG. 3.2 – Bus de communication

La topologie n'est rien d'autre que la forme que prend le réseau, sur la base des éléments de communication et des noeuds mis en service, mais aussi sur la base du type de connexions effectuées. Il existe plusieurs modèles de topologie :

- *la structure linéaire* (ou bus), qui interconnecte chaque noeud en parallèle sur le médium. L'ajout de répéteurs permet d'étendre le réseau sur de plus grandes distances.
- *la structure en anneau*, qui permet de relier en cascade tous les noeuds du réseau formant ainsi une boucle. Entre un noeud et le suivant, la communication est unidirectionnelle. Le noeud suivant s'occupera de répéter le signal reçu en entrée, ou d'ajouter ses propres informations au moment approprié. Cette structure présente certains avantages, comme la régénération du signal à chaque noeud permettant de couvrir des distances plus importantes. Néanmoins, il existe aussi des désavantages non négligeables, comme l'impossibilité d'étendre le réseau sans l'interrompre.
- *La structure en arbre* (ou hiérarchique), qui, contrairement à la structure en boucle, permet d'ajouter un nouveau segment de réseau sans interrompre le réseau.

L'utilisation d'Ethernet comme réseau industriel, nécessite l'utilisation de commutateurs (cf. chapitre 1). Ainsi, nous présentons dans le paragraphe suivant une étude comparative des différentes topologies commutées.

3.1.2.1 Les architectures commutées

Dans [RÜP 99], Rüping et al. présentent une étude comparative des différentes topologies appliquées dans les réseaux industriels Ethernet commutés. Pour cela, une application type est proposée, regroupant un nœud maître qui contrôle N équipements de terrain. Chaque équipement de terrain représente une carte d'entrées-sorties de b bits de données. Le système est rafraîchi cycliquement. Un cycle commence par l'émission par chaque carte d'entrées-sorties de l'état de ses capteurs vers le maître. Celui-ci les traite (en un temps nul) et détermine les nouvelles consignes des actionneurs qui sont ensuite envoyées successivement à tous les équipements de terrain. Les trois topologies étudiées sont celles présentées dans la figure 3.3. Chaque connexion entre deux nœuds est configurée en full-duplex. Tous les calculs sont réalisés en utilisant des liaisons Fast Ethernet (100 *Mbits/s*) et des commutateurs en mode "Store-and-Forward"² présentant une latence de 2 μ s.

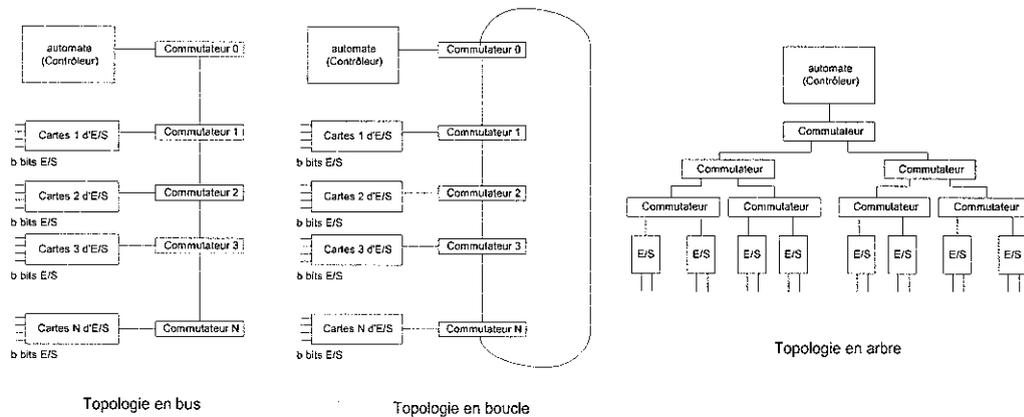


FIG. 3.3 – Les différentes topologies

Le temps nécessaire pour transmettre une trame T_f est :

$$T_f = \max\{h + b; 512\}.t_b + t_g \quad (3.1)$$

avec :

h : le nombre de bits d'entête

²Dans ce mode de commutation, la trame entière doit être reçue avant de pouvoir être acheminée. Par opposition, le mode "Cut-through" achemine la trame avant de l'avoir reçue entièrement.

b : le nombre de bits de données

t_b : le temps bit

t_g : le temps inter-trame

$\max\{h + b; 512\}$: la taille minimale d'une trame sur Ethernet étant de 64 octets soit 512 bits, on choisit la valeur maximale entre $h + b$ et 512.

Pour déterminer le temps de cycle minimum t_c des tâches qui sont exécutées par le nœud maître, il suffit de calculer ces expressions :

en ligne : $t_c = 2(N + 1).t_s + 2(N + 2).t_f$

en boucle : $t_c = (N + 3).t_s + 2(N + 2).t_f$

en arbre : $t_c = 2.E.t_s + (2.E + N + 1).t_f$
avec :

N : le nombre d'équipements de terrain

t_s : le temps de latence dans les commutateurs

E : le nombre de niveaux de commutateurs (dans la topologie en arbre uniquement)

La figure 3.4 donne les temps de cycle minima pour les trois topologies, en fonction du nombre d'équipements de terrain à raccorder. Pour chaque topologie, on prend deux longueurs de données (128 octets et 34 octets). Pour la topologie en arbre, on choisit trois niveaux d'arborescence. Les résultats montrent un très large avantage à la topologie en arbre. Puis la topologie en boucle est d'environ 10% plus rapide que l'organisation en bus, mais il faut prévoir dans ce cas un mécanisme pour diriger les messages de manière optimale.

Il est à noter que le cadre de cette étude reste très spécifique. Elle porte sur une organisation centralisée, où toutes les cartes d'E/S communiquent avec un seul automate. Cette approche semble assez réductrice par rapport aux nouvelles exigences fonctionnelles des applications d'automatisme. En effet, on déporte de plus en plus de traitements directement sur les capteurs et les actionneurs pour les rendre plus autonomes. Les échanges sont alors plus transversaux, car ces effecteurs dits « intelligents » doivent en permanence coopérer avec d'autres pour réagir au plus vite, évoluer et se synchroniser avec leur environnement.

De plus, la notion de fiabilité est également très importante dans les réseaux industriels. Des liens redondants doivent être mis en place et permettre de préserver le réseau des défaillances. C'est dans ce contexte que nous proposons de résoudre le problème de conception de réseaux locaux industriels fiabilisés.

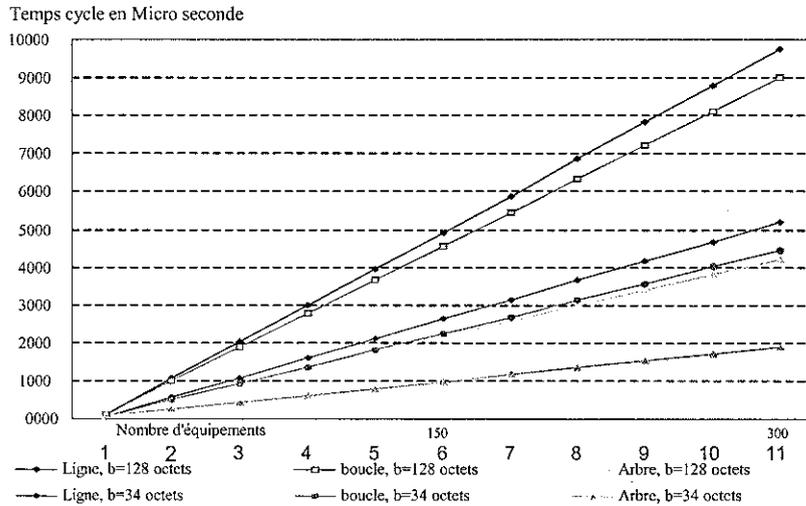


FIG. 3.4 – Temps de cycle / topologie

3.1.2.2 Les architectures fiabilisées

Pour améliorer la fiabilité du support de communication, il est nécessaire de mettre en place des chemins de secours afin de pallier la défaillance d'un lien. Toutefois, il est important de remarquer que la méthode d'accès au médium implantée dans Ethernet, ne permet pas d'avoir plusieurs chemins opérationnels simultanément entre une paire de noeuds. Aussi, un algorithme de gestion des chemins redondants tel que le protocole *spanning tree* (IEEE 802.1d), doit être utilisé. Il permet de détecter et désactiver les chemins multiples. Ainsi, lorsqu'une défaillance survient sur un lien, un chemin alternatif est créé. Il est important de noter que les architectures en boucle ne peuvent pas être utilisées et seront systématiquement réduites à une architecture linéaire. Nous allons maintenant observer les différentes techniques pour fiabiliser un système de communication.

La première méthode consiste à mettre en œuvre une redondance dite "totale", à partir d'une architecture hiérarchisée sur plusieurs niveaux (niveau 1, niveau 2, niveau 3, ...). Le niveau 1 correspond à l'épine dorsale du réseau (ou backbone). Puis, chaque commutateur du niveau inférieur est relié à deux commutateurs du niveau supérieur. La figure 3.5 illustre ce principe de redondance totale. Chaque noeud de terminaison possède deux interfaces réseaux reliées chacune à un commutateur différent de même niveau. Cependant, cette solution est très coûteuse, et reste dans la plupart des cas difficile à réaliser. En effet, les noeuds de terminaison de type industriel, sont rarement équipés de deux interfaces pour

des raisons de coût et de technologie (emplacement, gestion, mémoire, ...). La redondance au niveau des équipements industriels ne sera d'ailleurs pas prise en compte dans la suite des travaux.

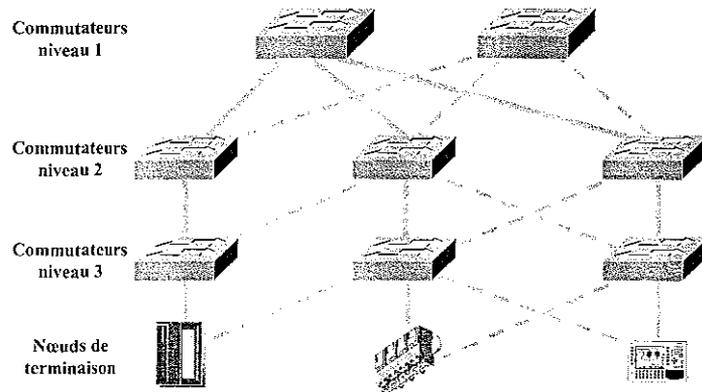


FIG. 3.5 – Architecture à redondance totale

Une deuxième solution consiste à réaliser de la redondance sur chaque lien. La figure 3.6 illustre cette méthode sur les architectures linéaire et hiérarchique. Dans les deux cas, la mise en place de cette redondance double le coût de l'architecture au niveau des liens, mais aussi des ports des commutateurs. De plus, contrairement à la méthode précédente, nous pouvons remarquer la présence de commutateurs dits "critiques". En cas de panne de ces commutateurs, les communications entre les autres équipements seraient également perturbées. Même si la fiabilité en terme de connectivité (panne d'un lien) est assurée, les architectures ne sont pas tolérantes aux pannes des équipements.

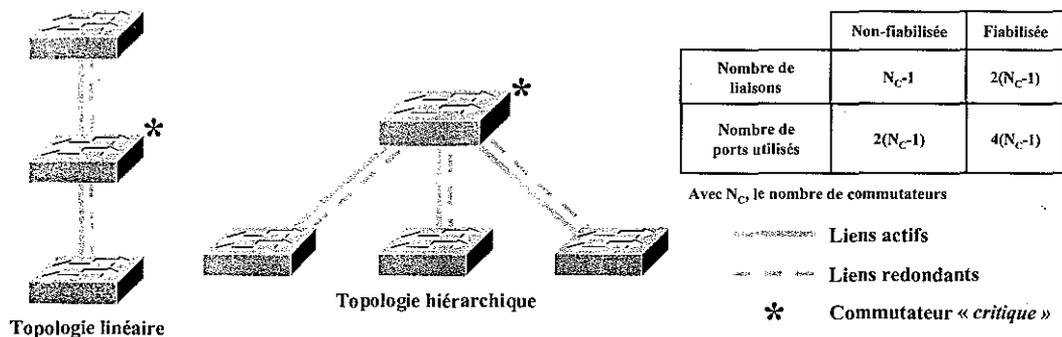


FIG. 3.6 – Architecture à redondance de liens.

Nous proposons d'établir nos travaux de conception sur une architecture hybride permettant de conserver une bonne fiabilité et à moindre coût. Cette architecture présentée sur la figure 3.7 est une combinaison entre une topologie hiérarchique et une topologie linéaire. Elle est constituée de deux niveaux de commutateurs. Le premier représente le backbone du réseau (commutateur fédérateur). Le deuxième niveau permet de connecter les équipements (commutateur de terminaison). La mise en place de cette architecture avec un algorithme de gestion des chemins redondants, permet, en cas de défaillance, de basculer de la topologie active (ou hiérarchique) sur la topologie linéaire, assurant la connectivité du réseau de communication. Nous pouvons constater l'absence de commutateurs "critiques" ce qui rend cette architecture plus tolérante aux pannes. Le coût, en terme de liaisons et d'interfaces d'interconnexion, est également réduit, puisque nous obtenons pour une architecture fiabilisée et un nombre de commutateurs (N_C) équivalent : $Nb_{liaisons} = 2.N_C - 3$ et $Nb_{ports} = 2(2.N_C - 3)$ (cf figure 3.6).

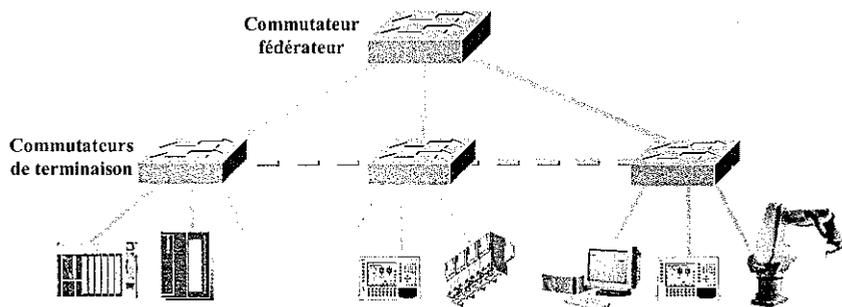


FIG. 3.7 – Architecture hybride

3.2 Formulation du problème d'optimisation

L'objectif est de minimiser le délai d'acheminement des messages. Ce délai dépend de deux critères :

- le nombre de commutateurs traversés d'une part. Le délai de commutation d'un paquet est important par rapport au délai de propagation. Le délai d'acheminement d'un paquet s'accroît avec le nombre de commutateurs traversés. Il faut alors limiter les échanges entre les commutateurs,
- l'équilibre de la charge dans chaque commutateur d'autre part. Le délai d'attente

(ou de latence) d'un message dans un commutateur dépend de la quantité de ses arriérés. Moins le nombre d'arriérés est grand, plus le message sera traité rapidement. Ce critère ne doit pas être considéré au niveau de chaque message mais globalement, c'est à dire sur l'ensemble du trafic.

Pour illustrer les critères de conception, nous avons réalisé à partir du scénario d'échanges de la figure 3.1, quatre répartitions différentes des équipements sur une architecture hiérarchique. La figure 3.8 présente, pour chaque organisation, la répartition des équipements sur le réseau ainsi que la matrice d'échanges associée [KRO 02a].

Nous avons calculé pour chaque organisation les charges des commutateurs ainsi que le pourcentage de communications inter-groupe et intra-groupe dans les matrices d'échanges correspondantes. Les résultats obtenus sont donnés sur la figure 3.9.

Reprenons les deux critères énoncés ci-dessus. Nous souhaitons minimiser le nombre de commutateurs à traverser. Dans ce cas, il faut limiter les échanges à travers le commutateur fédérateur, c'est à dire minimiser sa charge. Parmi l'ensemble des solutions proposées, l'organisation 2 est celle qui répond le mieux à ce critère. En terme de partitionnement de graphes, cela se traduit par une réduction des communications inter-groupe, c'est à dire une minimisation de la *taille de coupure*. Ce critère correspond au premier critère énoncé dans la définition du partitionnement, qui est donnée dans la section 2.4.1 : *Minimiser la valeur totale des arêtes inter-agrégats*.

Le deuxième critère concerne l'équilibre. La solution 4 présente un déséquilibre de charge important entre ses commutateurs de terminaison. Les équipements connectés sur le commutateur $n^{\circ}2$ ne génèrent aucun trafic local au commutateur et chargent les autres commutateurs, dont le commutateur fédérateur. Ceci est d'ailleurs contraire au premier critère identifié. De plus, la surcharge provoquée sur le commutateur $n^{\circ}1$, génère des arriérés de traitement susceptibles d'augmenter la latence pour chaque message traversant ce commutateur (soit 100% des communications dans ce cas). Il faut alors minimiser les communications inter-groupes tout en équilibrant les charges (communications intra groupe). Ce critère correspond au deuxième critère énoncé dans la définition du partitionnement, qui est donnée dans la section 2.4.1 : *Équilibrer approximativement les cardinalités des parties*.

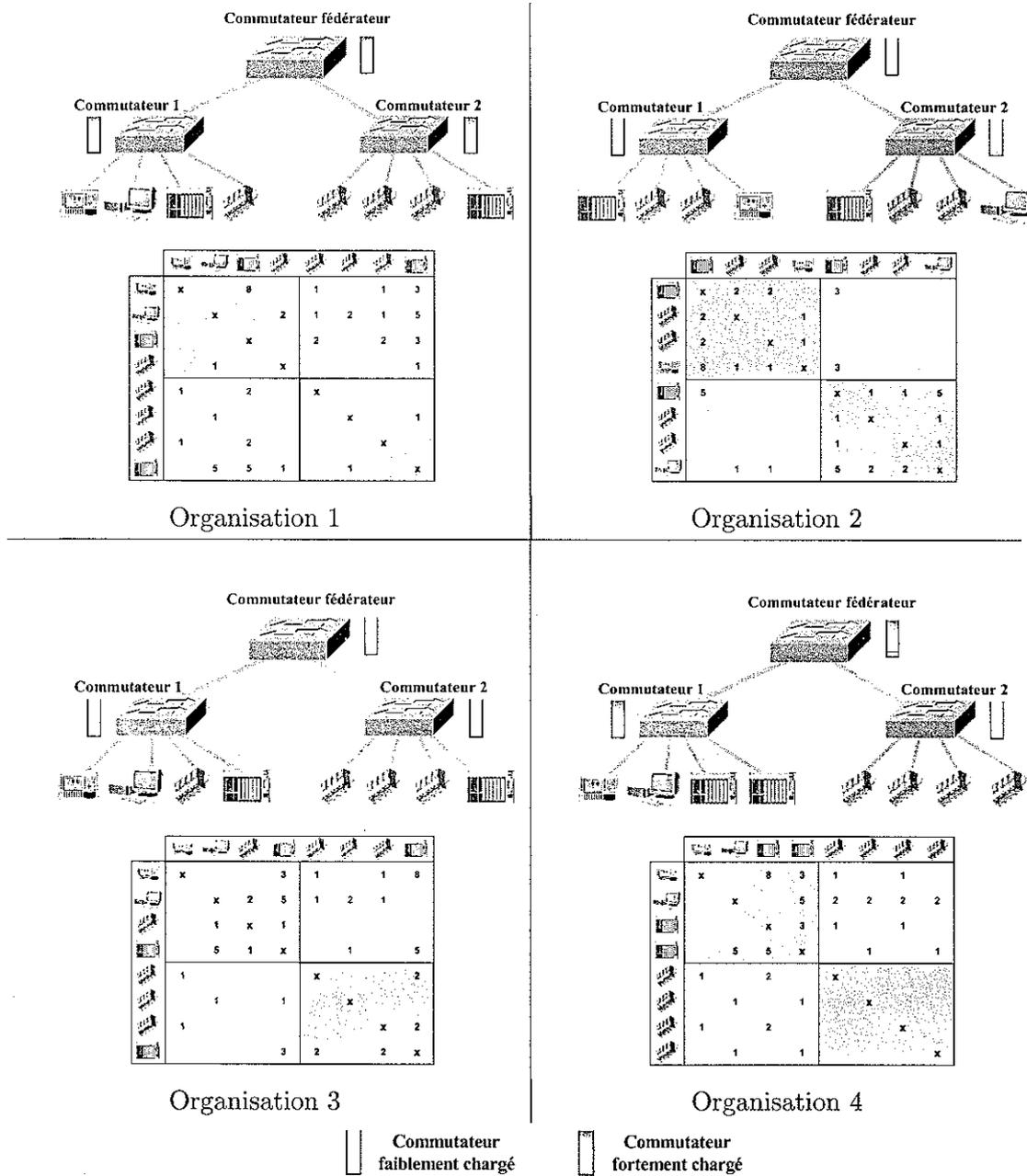


FIG. 3.8 – Distribution des équipements sur les commutateurs de terminaison

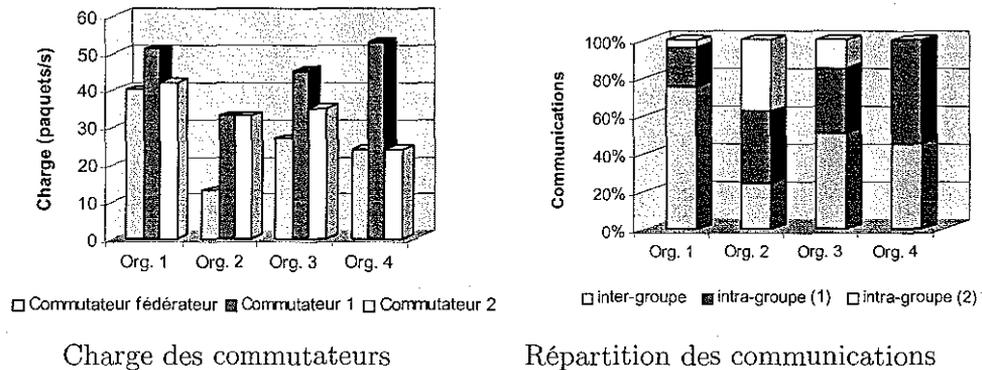


FIG. 3.9 – Analyse des charges et des communications

L'architecture réseau $n^{\circ}2$ respecte les deux critères "réseaux", et la partition associée correspond au partitionnement optimal. Afin de vérifier la concordance entre les critères d'optimisation du délai et ceux du partitionnement de graphes, nous avons simulé chaque architecture avec les paramètres suivants : réseau à 10Mbits/s , temps de cycle de 10ms et des paquets de 64 octets. Les résultats sont présentés dans le tableau 3.2. La solution présentant le délai le plus faible, correspond à l'architecture $n^{\circ}2$, pour laquelle la valeur de coupure est minimisée. Nous pouvons également vérifier l'importance de l'équilibre en comparant les solutions 3 et 4. Bien que l'organisation $n^{\circ}4$ présente une valeur de coupure plus petite que celle de l'organisation $n^{\circ}3$, son délai d'acheminement est plus important en raison du fort déséquilibre de charge. En conclusion, les critères de minimisation du délai d'acheminement des messages dans le réseau correspondent aux critères de partitionnement de graphes. C'est pourquoi, nous proposons de traiter le problème de conception de réseaux *Ethernet Industriel* comme un problème de partitionnement de graphes.

| | délai (ms) | valeur de coupure | déséquilibre (écart type) |
|----------------|------------|-------------------|---------------------------|
| Organisation 1 | 0,239 | 40 | 9 |
| Organisation 2 | 0,089 | 13 | 0 |
| Organisation 3 | 0,179 | 27 | 10 |
| Organisation 4 | 0,222 | 24 | 29 |

TAB. 3.2 – Critères réseaux / partitionnement

Deux autres aspects sont pris en compte dans nos travaux : la redondance de liens et le coût. Le premier permet de fiabiliser le système de communication. Pour cela, nous proposons d'utiliser l'architecture définie précédemment (Fig. 3.7), dans laquelle une topologie linéaire permet d'assurer la connectivité. Cette topologie doit aussi être analysée au même titre que la topologie en étoile, pour éviter que certains commutateurs deviennent des goulots d'étranglement. Ainsi, une fois la répartition des équipements industriels réalisée sur la topologie active, nous pouvons optimiser le placement des commutateurs sur l'architecture linéaire. En effet, contrairement à la topologie hiérarchique, la longueur des chemins peut varier. De ce fait, il faut regrouper les commutateurs ayant le plus d'interactions deux à deux.

Le deuxième concerne le coût de l'architecture. Celle-ci étant imposée, nous pouvons ainsi déterminer le nombre minimal de commutateur en fonction du nombre d'équipements à répartir, et de la capacité des commutateurs. De même, nous considérons les commutateurs de terminaisons identiques. Ainsi, au coût minimum correspond une architecture utilisant un minimum de commutateurs. Ce critère sera donc par la suite exprimé comme une contrainte technologique.

En résumé, nous utilisons la démarche de conception de réseaux *Ethernet Industriel* présentée sur la figure 3.10. Après avoir fixé le nombre minimal de commutateurs pour la topologie, nous réalisons le partitionnement de la matrice d'échanges pour optimiser la valeur de coupure et l'équilibre. La solution obtenue est alors évaluée et doit satisfaire la contrainte temporelle fixée par l'application. Dans le cas contraire, l'algorithme d'optimisation peut être réitéré en augmentant le nombre de commutateurs.

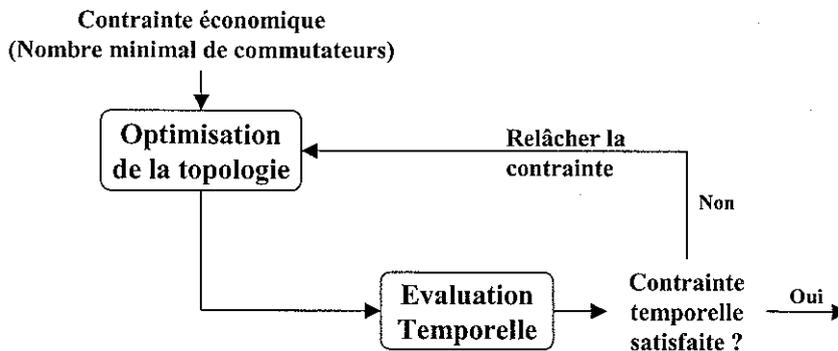


FIG. 3.10 – Démarche de conception

3.3 Résolution à l'aide des algorithmes spectraux

L'utilisation des propriétés spectrales de la matrice Laplacienne $L(G)$ du graphe G (graphe d'échanges dans notre cas) est à l'origine du nom de la méthode. Cette méthode fait partie des heuristiques globales de partitionnement qui sont pour la plupart récursives. Nous présentons dans un premier temps les méthodes spectrales, puis nous verrons comment ces méthodes se comportent sur le problème de conception topologique et les modifications apportées, afin de satisfaire les critères précédents.

3.3.1 Fondements des méthodes spectrales

Dans la partie 2.4.2.1, nous avons présenté succinctement les méthodes spectrales. Elles sont basées sur une approximation continue avec des contraintes du problème initial discret. Afin de donner un exemple d'une telle approximation, considérons le problème de bisection d'un graphe qui consiste à séparer en deux sous-ensembles, l'ensemble des sommets du graphe. Soit x un n -vecteur ($n = |V|$) avec les composantes $x_i = \pm 1$. Nous pouvons associer à chaque sommet v_i une composante du vecteur x . Nous pouvons constater que la fonction $f(x) = \frac{1}{4} \sum_{(i,j) \in E} (x_i - x_j)^2$ comptabilise exactement le nombre d'arêtes ($|E_0|$) reliant les deux sous-ensembles. En effet, nous avons :

$$(x_i - x_j)^2 = \begin{cases} 0 & \text{si } x_i = x_j \text{ (les sommets } v_i \text{ et } v_j \text{ sont dans le même sous-ensemble)} \\ 4 & \text{si } x_i \neq x_j \text{ (les sommets } v_i \text{ et } v_j \text{ sont dans des sous-ensembles distincts)} \end{cases}$$

Le problème de partition qui consiste à minimiser le nombre d'arêtes entre les sous-ensembles, revient alors à minimiser la fonction quadratique :

$$\min |E_0| = \min f(x) = \frac{1}{4} \sum_{(i,j) \in E} (x_i - x_j)^2 \quad (3.2)$$

Les partitions devant être distinctes, nous devons respecter la contrainte $x_i = \pm 1$ avec $i = 1, \dots, n$. De même, les sous-ensembles doivent contenir le même nombre de sommets (contrainte d'équilibre). Ceci peut se traduire par le fait qu'il y ait autant de composantes négatives que positives, c'est à dire par l'expression $\sum_{i=1}^n x_i = 0$ ou encore par $(x, \vec{1}) = 0$ où $\vec{1}$ est le n -vecteur unitaire et (\cdot, \cdot) est le produit scalaire. En revanche, nous pouvons d'ores et déjà constater que l'équilibre des charges n'apparaît pas comme un critère à minimiser.

Il est possible de réécrire la fonction quadratique 3.2 sous une forme matricielle à l'aide de la matrice Laplacienne $L = L(G)$ du graphe G . Soit $d(v_i)$, le degré d'un sommet v_i ,

c'est à dire le nombre de sommets adjacents à v_i . La matrice de Laplace $L(G)$ du graphe G est définie par les éléments suivants :

$$l_{ij} = \begin{cases} -1 & \text{si } i \neq j \text{ et } (v_i, v_j) \in E \\ 0 & \text{si } i \neq j \text{ et } (v_i, v_j) \notin E \\ d(v_i) & \text{si } i = j \end{cases}$$

La matrice Laplacienne L possède des propriétés intéressantes. Elle est symétrique, semi-définie positive et la somme de chaque ligne et colonne est nulle. L'expression $L = D - A$ permet également de représenter la matrice Laplacienne à l'aide de deux autres matrices associées au graphe : la matrice diagonale D (matrice $n \times n$ des degrés $d(v_i)$ des sommets du graphe) et la matrice d'adjacence A du graphe. De plus, la multiplicité de la valeur propre 0 équivaut au nombre de composantes connexes du graphe. En effet, dans le cas d'un graphe non connexe, L peut se décomposer sous forme de blocs, chacun représentant un graphe.

En développant l'équation 3.2, nous obtenons :

$$\frac{1}{4} \sum_{(i,j) \in E} (x_i - x_j)^2 = \frac{1}{4} \left(\sum_{(i,j) \in E} (x_i^2 + x_j^2) - 2 \sum_{(i,j) \in E} x_i \cdot x_j \right)$$

$$\text{or, } \sum_{(i,j) \in E} (x_i^2 + x_j^2) = x^T D x \text{ et } 2 \sum_{(i,j) \in E} x_i \cdot x_j = x^T A x$$

Le problème discret peut donc s'écrire :

$$\left\{ \begin{array}{l} \text{minimiser} \quad \frac{1}{4}(x^T D x - x^T A x) = \frac{1}{4} [x^T (D - A)x] = \frac{1}{4} x^T L x \\ \text{avec les contraintes} \quad x_i = \pm 1, i = 1, \dots, n \\ \quad \quad \quad (x, \vec{1}) = 0 \end{array} \right.$$

La minimisation de celle ci demeurant un problème NP-complet, le problème peut être approximée en relaxant la contrainte discrète $x_i = \pm 1$ et en la remplaçant par $(x, x) = n$ ou encore $\sum_{i=1}^n x_i^2 = n$. Nous obtenons alors le problème relaxé suivant :

$$\min_{\substack{x_i = \pm 1, \\ \sum_{i=1}^n x_i = 0}} \frac{1}{4} x^T L x \geq \min_{\substack{\sum_{i=1}^n x_i^2 = n, \\ \sum_{i=1}^n x_i = 0}} \frac{1}{4} x^T L x \quad (3.3)$$

Il est alors possible de montrer que la solution du problème est en fait un vecteur propre de la matrice L . Supposons que G soit un graphe connexe, nous pouvons alors définir l'ensemble des vecteurs propres u_1, u_2, \dots, u_n normalisés de L avec comme valeur

propre correspondante $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n$. Le problème relaxé 3.3 avec la contrainte $\sum_{i=1}^n u_i^2 = n$, devient alors :

$$\min_{\substack{\sum_{i=1}^n x_i^2 = n, \\ \sum_{i=1}^n x_i = 0}} \frac{1}{4} x^T L x = \frac{1}{4} u_i^T L u_i = \frac{\lambda_i(L)}{4} u_i^T u_i = \frac{n}{4} \lambda_i(L) \quad (3.4)$$

La plus petite valeur propre non nulle λ_i et u_i le vecteur propre associé, de norme \sqrt{n} est solution du problème de minimisation. La matrice étant symétrique, toutes les valeurs propres sont réelles. En les ordonnant, le vecteur propre solution est associé à la seconde valeur propre λ_2 puisque le graphe est considéré comme connexe. Ainsi, $x = \sqrt{n} \cdot u_2$ minimise la fonction $f(x)$ (équation 3.2). Cette solution x^F , encore appelée vecteur de Fiedler [FIE 75] doit être ensuite projetée sur un espace discret, afin d'obtenir une partition ne contenant que les valeurs -1 et 1 . Il faut alors déterminer une méthode de coupure du vecteur solution permettant de répartir les sommets dans chaque sous-ensemble. Par exemple, la méthode de la *coupure médiane* permet d'assigner chaque composante de x^F à partir d'une valeur médiane x_m^F en posant :

$$x_i^F = \begin{cases} -1 & \text{si } x_i^F < x_m^F \\ +1 & \text{si } x_i^F > x_m^F \\ \pm 1 & \text{si } x_i^F = x_m^F \quad (\text{pour respecter l'équilibre}) \end{cases} \quad (3.5)$$

Cette méthode de coupure possède les propriétés suivantes, mises en évidence par Fiedler [FIE 75] :

- Si G est connexe, un des deux sous-graphes $G_i = (V_i, (V_i \times V_i) \cap E)$ l'est aussi,
- Si de plus x^F possède exactement $n/2$ composantes positives et $n/2$ composantes négatives, les deux sous-graphes sont connexes.

La bisection se résume à :

- calculer le vecteur x^F ,
- déterminer la valeur médiane x_m^F , puis assigner les sommets selon 3.5.

Ceci étant, x^F minimise le problème relaxé, et nous ne connaissons pas la qualité de la partition induite par la *coupure médiane*. Des travaux ont été menés dans le but de prouver la qualité de la bisection induite par x_m^F [POT 94, CHA 97, GUA 98] et également pour améliorer les performances des méthodes spectrales en proposant de nouvelles méthodes de coupure [HAG 92].

Dans la suite, nous appliquons les algorithmes spectraux pour résoudre le problème de conception de réseaux *Ethernet Industriel*.

3.3.2 Application à la conception de réseaux

Dans un premier temps, nous utilisons les algorithmes spectraux afin de déterminer la répartition optimale des équipements industriels sur le réseau. L'architecture obtenue doit alors respecter la contrainte de coût minimal. Ensuite, nous devons fiabiliser l'architecture en couplant deux à deux les commutateurs de terminaison ayant de fortes corrélations. Ce problème s'apparente à un problème de sérialisation.

3.3.2.1 Répartition des équipements

Les notations utilisées pour résoudre ce problème sont les suivantes :

- Soit T , la matrice de trafic et $G = (V, E)$ le graphe non orienté (obtenu par symétrisation)
- Soit V , l'ensemble des sommets du graphe (représentant les nœuds de terminaison)
- Soit E , l'ensemble des arcs pondérés reliant les sommets (représentant les volumes d'informations échangés entre deux nœuds du graphe)
- Soit n , le nombre de ports disponibles (ou disponibilité) des commutateurs permettant de relier des nœuds de terminaison,
- Soit N_s , le nombre de commutateurs de terminaison à mettre en œuvre pour interconnecter la totalité des nœuds de terminaison. Le nombre optimal de commutateurs de terminaison peut être déterminé par :

$$N_s = \left\lceil \frac{|V|}{n} \right\rceil \quad (3.6)$$

Notre problème de conception de réseaux ne s'arrête pas à un simple problème de bisection. L'application traitée et la technologie des commutateurs nécessitent de réaliser une k -partition. De la même façon que précédemment, il est possible de montrer que d'autres vecteurs sont solution du problème de k -partition [HEN 93a]. La complexité croît avec le nombre de partitions. Une approche itérative dite bisection spectrale récursive (ou RSB), a alors été développée dans [POT 90, SIM 91]. Elle permet de réaliser le k -partitionnement d'un graphe G à partir d'une succession de bisections. L'algorithme est le suivant :

1. Calculer le vecteur de Fiedler x^F du graphe G associé à la plus petite valeur propre non nulle.
2. Ordonner les sommets selon les composantes du vecteur de Fiedler.
3. Affecter la moitié des sommets au sous-graphe G_1 et l'autre moitié au sous graphe G_2 .
4. Appliquer récursivement la procédure aux sous-graphes G_1 et G_2 jusqu'à ce que le critère d'arrêt soit atteint.

La figure 3.11 illustre le résultat obtenu, en exécutant l'algorithme RSB sur un système composé de 20 nœuds de terminaison à répartir sur des commutateurs, avec une disponibilité de 4 ports. La première ligne représente l'ordre des sommets du graphe relativement au vecteur de Fiedler, et les lignes inférieures représentent les itérations successives. La méthode de coupure utilisée est celle de la coupure médiane. La solution finale conduit à la répartition des 20 nœuds de terminaison sur 8 commutateurs au lieu de 5 dans le cas d'une solution optimale définie par l'équation 3.6. Le taux de raccordement des ports des commutateurs de terminaison est de 62%. Nous proposons dans la suite de modifier l'algorithme de façon à générer des groupes faiblement dépendants tout en s'approchant d'un taux de raccordement de 100% [KRO 01b].

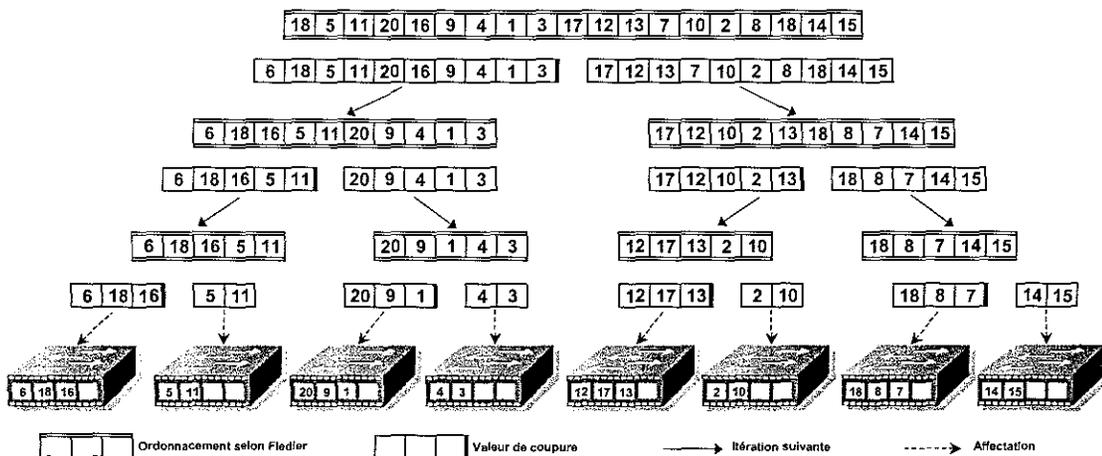


FIG. 3.11 - Décomposition suivant RSB

3.3.2.2 Modifications apportées à l'algorithme RSB

Afin d'optimiser le taux de raccordement, c'est à dire minimiser le nombre de commutateurs de terminaison, nous proposons de modifier le séparateur (ou valeur de coupure) s_k permettant de définir les deux partitions. Les notations seront les suivantes :

- Soit $L(L_1, \dots, L_{|V|})$, la liste des sommets ordonnés suivant les valeurs des éléments du vecteur de fiedler
- Soit P , l'ensemble des partitions tel que :

$$\begin{cases} \cup_{i=1}^k P_i = V & \text{avec } k \text{ le nombre de partition} \\ P_i \cap P_j = \{\emptyset\} & \forall i \neq j \end{cases}$$
- Soit s_k , la valeur de coupure correspondant à la bissection spectrale $P_1 = \{L_1, \dots, L_{s_k}\}$ et $P_2 = \{L_{s_k+1}, \dots, L_{|V|}\}$
- Soit $w(v_i, v_j)$ le poids de l'arête entre le sommet v_i appartenant à la partition P_1 et v_j appartenant à la partition P_2

Dans l'algorithme RSB initial, la valeur de coupure correspond à $s = \left\lceil \frac{|V|}{2} \right\rceil$. Notre objectif est de garantir un remplissage optimal des commutateurs à partir de l'ordonnement des sommets du graphe (obtenu suivant les valeurs des éléments du vecteur de Fiedler). Nous recherchons l'ensemble des valeurs de coupure du graphe, de telle sorte que la partition P_1 soit constituée de s_k éléments (s_k multiple de n et proche de la valeur médiane $\frac{|V|}{2}$).

Cela revient à chercher l'ensemble des séparateurs s_k tel que :

$$\frac{|V|}{2} - n < s_k < \frac{|V|}{2} + n \quad (3.7)$$

Lemme 1 *Existence de la valeur de coupure $s_1 = n \left\lceil \frac{Ns}{2} \right\rceil$*

Soit le séparateur $s_1 = n \left\lceil \frac{Ns}{2} \right\rceil$ tel que $Ns = \left\lceil \frac{|V|}{n} \right\rceil$ et $|V| \geq 2n$

Alors $\frac{|V|}{2} - n < s_1 < \frac{|V|}{2} + n$

Démonstration. $\frac{Ns}{2} \geq \left\lceil \frac{Ns}{2} \right\rceil \implies n \frac{Ns}{2} \geq s_1 \quad \forall n > 0$

$$\frac{|V|}{n} + 1 = \left\lceil \frac{|V|}{n} \right\rceil = Ns \implies n \frac{|V|+1}{2} > s_1 \implies \frac{|V|}{2} + n > s_1$$

De même, on aura :

$$\left\lceil \frac{Ns}{2} \right\rceil > \frac{Ns}{2} - 1 \implies s_1 > n \left(\frac{Ns}{2} - 1 \right) \quad \forall n > 0$$

$$\left\lceil \frac{|V|}{n} \right\rceil \geq \frac{|V|}{n} \implies s_1 > n \left(\frac{|V|}{2} - 1 \right) \implies s_1 > \frac{|V|}{2} - n$$

Lemme 2 *Existence de la valeur de coupure s_2 dérivée de s_1*

Soit le séparateur $s_1 = n \lfloor \frac{Ns}{2} \rfloor$ tel que $Ns = \lfloor \frac{|V|}{n} \rfloor$ et $|V| \geq 2n$, il est alors démontrable que :

- si Ns est pair, alors il existe s_2 tel que $s_2 = n (\lfloor \frac{Ns}{2} \rfloor - 1)$ et $\frac{|V|}{2} - n < s_2 < \frac{|V|}{2} + n$
- si Ns est impair, alors il existe s_2 tel que $s_2 = n (\lfloor \frac{Ns}{2} \rfloor + 1)$ et $\frac{|V|}{2} - n < s_2 < \frac{|V|}{2} + n$

Démonstration. - Si Ns est pair et si $\frac{|V|}{2} < s_1 < \frac{|V|}{2} + n$ alors :

$$\frac{|V|}{2} - n < s_1 - n < \frac{|V|}{2} \implies \frac{|V|}{2} - n < n (\lfloor \frac{Ns}{2} \rfloor - 1) < \frac{|V|}{2} \implies \frac{|V|}{2} - n < s_2 < \frac{|V|}{2} + n$$

- Si Ns est pair et si $s_1 = \frac{|V|}{2}$ (c'est à dire $Ns = \lfloor \frac{|V|}{n} \rfloor = \frac{|V|}{n}$) alors :

$$s_2 = n (\lfloor \frac{Ns}{2} \rfloor - 1) \implies s_2 = n (\frac{Ns}{2} - 1) - n \implies s_2 = s_1 - n \implies s_2 = \frac{|V|}{2} - n$$

- Si Ns est impair alors : $\frac{|V|}{2} - n < s_1 < \frac{|V|}{2}$

$$\frac{|V|}{2} < s_1 + n < \frac{|V|}{2} + n \implies \frac{|V|}{2} < n (\lfloor \frac{Ns}{2} \rfloor + 1) < \frac{|V|}{2} + n \implies \frac{|V|}{2} - n < s_2 < \frac{|V|}{2} + n$$

En conservant la même démarche, nous pouvons également rechercher l'ensemble des valeurs de coupure du graphe de telle sorte que la partition P_2 soit constituée de s_k éléments (s_k multiple de n et proche de la valeur médiane $\frac{|V|}{2}$). Ainsi nous pouvons définir deux autres valeurs de coupure s_3 et s_4 telles que :

$$s_3 = |V| - n \lfloor \frac{Ns}{2} \rfloor \text{ et } s_4 = |V| - n (\lfloor \frac{Ns}{2} \rfloor \pm 1) \text{ selon la parité de } Ns$$

A partir des lemmes précédents, nous pouvons définir les séparateurs à considérer selon la valeur de $|V|$ et de Ns :

Théorème. Soit $\frac{|V|}{2} - n < s_k < \frac{|V|}{2} + n$

Si $|V| \geq 2n$ et si :

- $Ns \neq \frac{|V|}{n}$, alors les séparateurs s_1, s_2, s_3 et s_4 seront considérés avec s_2 et s_4 définis suivant la parité de Ns
- $Ns = \frac{|V|}{n}$ et Ns impair, alors seuls les séparateurs s_1 et s_3 seront à retenir ($s_1 = s_4$ et $s_3 = s_2$)
- $Ns = \frac{|V|}{n}$ et Ns pair, alors seul s_1 sera à retenir ($s_1 = s_3$)

Si $n < |V| < 2n$ alors :

seuls les séparateurs s_1 et s_3 appartiennent à l'ensemble S et seront considérés

Démonstration. - si $Ns = \frac{|V|}{n}$ et Ns impair, alors :

$$s_3 = |V| - n \left\lceil \frac{Ns}{2} \right\rceil \implies s_3 = |V| - n \left\lfloor \frac{|V|}{2n} \right\rfloor = n \left(\frac{|V|}{n} - \left\lfloor \frac{|V|}{2n} \right\rfloor \right) = n \left\lfloor \frac{|V|}{2n} \right\rfloor = n \left(\left\lfloor \frac{|V|}{2n} \right\rfloor + 1 \right) = n \left(\left\lceil \frac{Ns}{2} \right\rceil + 1 \right) = s_2$$

de même, $s_1 = s_4$

- si $Ns = \frac{|V|}{n}$ et Ns pair, alors :

$$s_1 = n \left\lceil \frac{Ns}{2} \right\rceil = n \left\lfloor \frac{|V|}{2n} \right\rfloor = \frac{|V|}{2} \quad \text{et} \quad s_3 = |V| - n \left\lceil \frac{Ns}{2} \right\rceil = |V| - s_1 = |V| - \frac{|V|}{2} = s_1$$

Après avoir défini l'ensemble S des valeurs de coupure à prendre en considération, l'algorithme doit évaluer chaque coupure afin de déterminer celle qui minimise le nombre de communications inter commutateurs c'est à dire celle qui minimise le nombre d'arêtes reliant les sommets appartenant à deux sous-graphes différents. La fonction minimisant le nombre d'arêtes peut alors s'écrire :

$$\min \sum_{i=1}^{s_k} \sum_{j=s_k+1}^{|V|} w(e_{ij}) \quad \text{avec} \quad w(e_{ij}) = \{w(v_i, v_j) | v_i \in P_1, v_j \in P_2\} \quad \text{et} \quad s_k \in S \quad (3.8)$$

L'algorithme modifié nous permet de réaliser le partitionnement du graphe initial et définir ainsi la topologie étoile (ou hiérarchique) minimisant les coûts en terme d'équipement et de délais d'acheminement.

Algorithme

1. Calculer le vecteur de Fiedler du graphe
2. Ordonner les sommets selon les valeurs des éléments du vecteur de Fiedler dans une liste L où L_1 représente le premier élément de la liste et $L_{|V|}$ le dernier
3. Calculer Ns pour le domaine à partitionner
4. Si $n < |V| < 2n$ alors $S = \{s_1, s_3\}$ avec $s_1 = n \left\lceil \frac{Ns}{2} \right\rceil$ et $s_3 = |V| - s_1$

Sinon ($|V| \geq 2n$)

si $Ns = \left\lfloor \frac{|V|}{n} \right\rfloor = \frac{|V|}{n}$ et si Ns est pair alors : $S = \{s_1\}$ avec $s_1 = n \left\lceil \frac{Ns}{2} \right\rceil$

sinon $S = \{s_1, s_3\}$

sinon ($Ns \neq \left\lfloor \frac{|V|}{n} \right\rfloor$) et si Ns est pair alors : $S = \{s_1, s_2, s_3, s_4\}$ avec $s_2 = n \left(\left\lceil \frac{Ns}{2} \right\rceil - 1 \right)$

et $s_4 = |V| - s_2$

sinon $S = \{s_1, s_2, s_3, s_4\}$ avec $s_2 = n \left(\left\lceil \frac{Ns}{2} \right\rceil + 1 \right)$ et $s_4 = |V| - s_2$

5. Choisir s_k tel que :

$$s_k \text{ minimise } \sum_{i=1}^{s_k} \sum_{j=s_k+1}^{|V|} w(e_{ij}) \text{ avec } w(e_{ij}) = \{w(v_i, v_j) | v_i \in P_1, v_j \in P_2\} \text{ et } s_k \in S$$

6. Définir les partitions P_1 et P_2 telles que :
$$\begin{cases} P_1 = \{L_1, \dots, L_{s_k}\} \\ P_2 = \{L_{s_k+1}, \dots, L_{|V|}\} \end{cases}$$

7. Appliquer récursivement les étapes 1 à 6 pour chaque partition P tant que $|V| > n$

Appliquons notre algorithme *RSB modifié* sur l'exemple précédent (Fig. 3.11). Nous disposons de 20 noeuds ($|V| = 20$) à répartir sur des commutateurs disposant de 4 ports ($n = 4$) permettant de connecter des noeuds de terminaison.

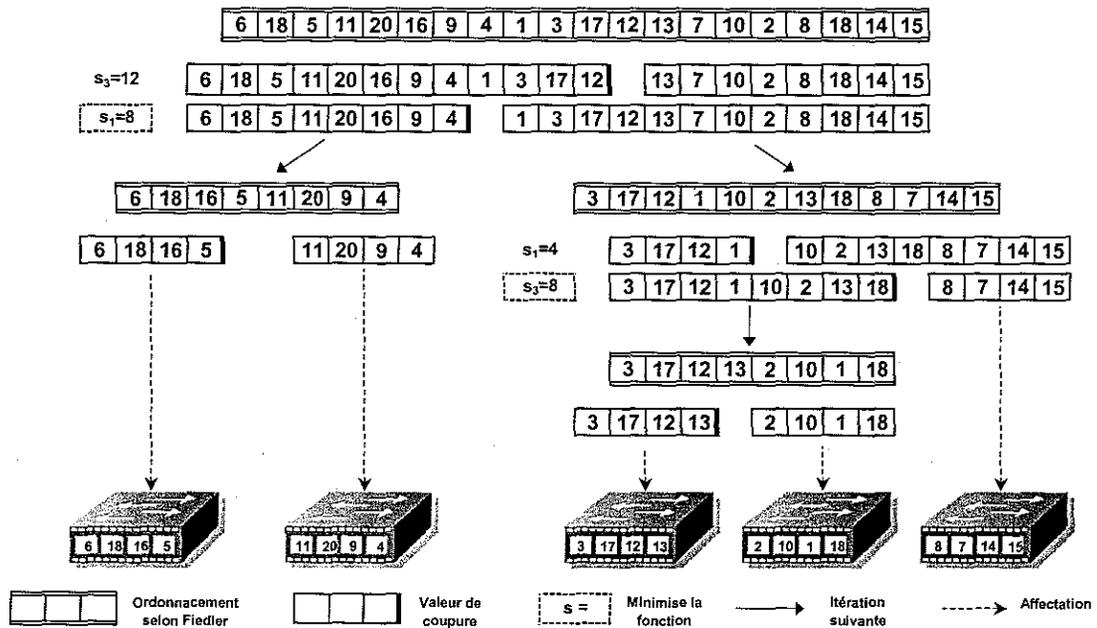


FIG. 3.12 – Décomposition suivant RSB modifié

La figure 3.12 illustre le comportement de l'algorithme *RSB modifié*. La première itération de l'algorithme donne : $Ns = \left\lceil \frac{|V|}{n} \right\rceil = \frac{|V|}{n} = 5$ (Ns impair). Nous obtenons alors deux valeurs de coupure $s_1 = 8$ et $s_3 = 12$. En considérant que s_1 est la valeur de coupure minimisant le nombre d'arêtes, nous obtenons alors deux nouvelles partitions P_1 et P_2 . Le nombre de noeuds dans chaque partition étant supérieur au nombre de ports du commuta-

teur, l'algorithme est réitéré. Dans le cas de la partition P_1 , nous avons un seul séparateur $s_1 = 4 = n$. Par conséquent, l'algorithme n'exécutera pas d'itération supplémentaire sur les deux sous-groupes obtenus.

En revanche, plusieurs itérations sont nécessaires pour obtenir une partition de P_2 . Finalement, nous obtenons une solution minimisant le nombre de commutateurs et un taux de raccordement optimal. Dans la suite, nous proposons de fiabiliser l'architecture obtenue.

3.3.2.3 Architecture fiabilisée : Problème de sérialisation

La topologie linéaire doit également être optimisée de manière à coupler deux à deux les équipements de terminaison ayant de fortes corrélations. Ils seront alors séquencés les uns par rapport aux autres, tout en respectant l'architecture hiérarchique prédéfinie. Ce problème s'apparente à un problème de sérialisation. Les techniques spectrales fournissant les relations d'interdépendance entre les composantes d'un graphe, [ATK 99] propose d'utiliser l'ordre du vecteur de Fiedler, pour résoudre le problème :

Etant donné un ensemble d'éléments n à ordonner, et une fonction de corrélation $f(i, j)$ qui reflète un niveau d'attraction entre deux éléments i et j pour qu'ils se retrouvent l'un à côté de l'autre dans la séquence, l'objectif de la sérialisation est de trouver toutes les séquences possibles entre les éléments en toute cohérence avec les fonctions de corrélation. C'est à dire si π représente la permutation d'éléments et $\pi(i) < \pi(j) < \pi(k)$ alors $f(i, j) \geq f(i, k)$ et $f(j, k) \geq f(i, k)$. [ATK 99] propose un algorithme, basé sur les techniques spectrales et notamment l'ordre du vecteur de Fiedler, pour générer cette séquence.

Algorithme

Pour construire la topologie redondante, nous appliquons l'algorithme de sérialisation sur le graphe dont les sommets représentent les commutateurs identifiés lors de l'élaboration de la topologie hiérarchique et les liens entre ces sommets correspondent aux volumes d'échanges entre les commutateurs. Le principe est le suivant :

La phase de partitionnement de la matrice de trafic (ou graphe d'échanges G) par l'algorithme RSB modifié permet d'identifier des groupes (g_1, g_2, \dots, g_k) de sommets faiblement dépendants. Pour étudier les corrélations entre ces groupes et les ranger suivant un ordre

3.3. RÉOLUTION À L'AIDE DES ALGORITHMES SPECTRAUX

de dépendance, la première opération consiste à contracter le graphe d'échanges G . Le graphe contracté G^* est tel que :

- un sommet de G^* correspond à un groupe g_i de G
- le poids d'une arête reliant deux sommets s_i et s_j de G^* est égal à la somme des poids des arêtes entre deux groupes g_i et g_j de G .

Ensuite, le calcul du vecteur de Fiedler associé au graphe G^* permet de déduire l'ordre des sommets, et donc des groupes de G . Cette séquence est utilisée pour cascader les commutateurs entre-eux [KRO 01b].

Application

Reprenons l'exemple précédent pour lequel l'algorithme RSB modifié a permis de répartir les 20 nœuds sur 5 commutateurs de terminaison, eux-mêmes reliés au commutateur fédérateur. La figure 3.13 illustre l'architecture hiérarchique obtenue à l'aide de l'algorithme.

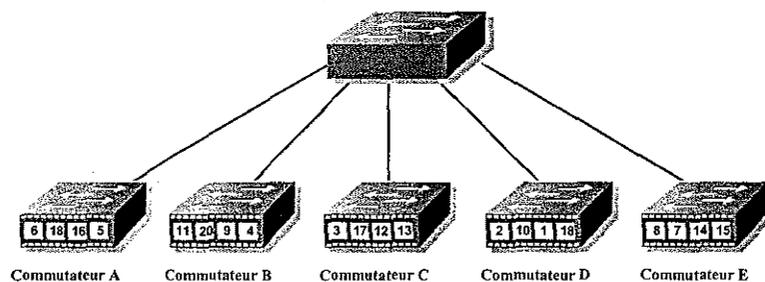


FIG. 3.13 – Architecture hiérarchique déduite de la décomposition suivant RSB modifié

A partir de la topologie hiérarchique, nous pouvons déduire le graphe d'échanges contracté G^* constitué de 5 sommets (A,B,C,D et E) et d'arêtes pondérées représentant le volume des échanges entre deux commutateurs. L'ordonnement des coordonnées du vecteur de Fiedler calculés à partir de cette matrice réduite G^* donne alors le séquençement des commutateurs. La figure 3.14 donne le graphe contracté et l'architecture hiérarchique fiabilisée redondante correspondante.

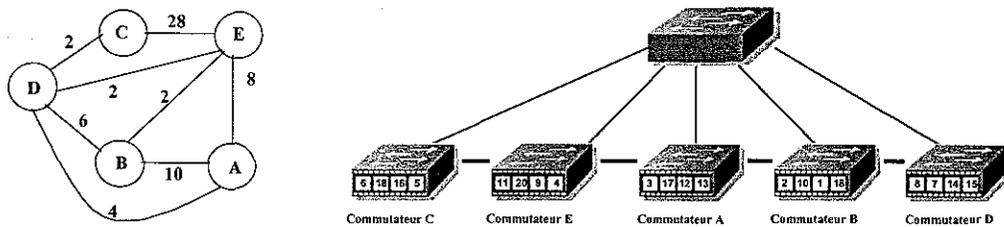


FIG. 3.14 – Sériation des commutateurs

3.4 Résolution par algorithmes génétiques

Les algorithmes génétiques sont très peu standardisés et dépendent fortement des applications pour lesquelles ils sont utilisés. Ainsi, chaque application particulière nécessite le développement d'un algorithme génétique qui lui est propre. Si ceci est souvent perçu comme un inconvénient, il n'en est pas moins un avantage des algorithmes génétiques en terme d'adaptation au problème. Dans un premier temps, nous exposerons les fondements des algorithmes génétiques, puis nous caractériserons ces algorithmes. Enfin, nous présenterons les variantes de notre algorithme développé dans le cadre de la conception de réseaux industriels.

3.4.1 Fondements des algorithmes génétiques

3.4.1.1 De l'évolution des espèces à la théorie de l'évolution

"J'ai nommé ce principe, par lequel toute variation utile chez une espèce est préservée, par le terme de sélection naturelle", Charles Darwin, 1859.

C'est à partir de l'observation de l'évolution des espèces que le naturaliste anglais Charles Darwin initie la théorie de l'évolution. En effet, chaque espèce (animale ou végétale) doit s'adapter aux différentes variations ou modifications de son environnement (climat, nouveaux prédateurs, maladies, surpopulation, ...) afin d'assurer la survie de l'espèce. Soumis à des situations extrêmes, les individus les plus doués (c'est à dire, les plus adaptés) d'une population ont plus de chance de survivre que les individus les plus faibles. Ainsi, par transmission du matériel génétique de génération en génération, les "jeunes" populations devraient hériter majoritairement du bagage génétique des individus les plus forts des populations âgées. C'est le principe que *C. Darwin* appelle la sélection naturelle. Il est basé essentiellement sur les mécanismes de reproduction et le codage génétique qui stocke les informations décrivant l'individu sous forme de gènes.

Malgré tout, certains individus, plus faibles que les autres, réussiront toujours à déjouer le système et à se reproduire (probablement en raison de leur chance). Néanmoins, selon les darwinistes, l'effet à long terme sur les générations futures est que les individus seront de mieux en mieux adaptés à leur milieu.

3.4.1.2 Les algorithmes génétiques

Les algorithmes génétiques (AG) sont l'oeuvre de John Holland (1975) et de ses étudiants de l'Université du Michigan. Ils se sont directement inspirés du modèle des lois de la nature de Darwin, basé sur la survie des espèces les plus adaptées. Tout comme dans la nature où les spécimens se reproduisent, dans le modèle des algorithmes génétiques, les spécimens se reproduiront aussi. Des opérations génétiques sont ainsi effectuées sur les candidats (ou individus) pour obtenir de nouveaux candidats plus performants que leurs propres parents. En biologie, on manipule des gènes et des chromosomes ; il en va de même dans le modèle des AG. Un individu est caractérisé par ses chromosomes. Nous parlons alors d'encodage des solutions du problème. Cet encodage peut prendre différentes formes, une des plus usuelles étant la chaîne de bits. Ces chaînes sont comparables aux chromosomes des systèmes biologiques, alors que les caractères ou les bits qui composent les chaînes sont comparables aux gènes.

Pour trouver une solution à un problème, les AG utilisent un cycle d'évolution de l'ensemble des individus (ou population) formé par trois phases principales :

- La sélection naturelle, qui favorise les "bons" individus en supprimant les "mauvais". Cette sélection utilise une fonction objectif (ou d'adaptation) dont le but est de favoriser les individus les plus adaptés. A chaque individu est alors associé sa qualité généralement représentée par un réel ou un entier,
- Le croisement, qui génère pour chaque couple, des enfants qui héritent des caractéristiques de chacun de leurs parents (transmission du matériel génétique),
- La mutation, qui altère l'individu sur lequel elle s'applique. Elle est représentative des erreurs pouvant survenir lors des phases de reproduction.

Les AG utilisent des règles de transition probabilistes plutôt que déterministes. Le hasard joue alors un rôle important en guidant la recherche vers des régions de l'espace de recherche susceptibles d'améliorer la solution. Ainsi, les candidats à la reproduction sont choisis de façon probabiliste, et les opérateurs de croisement et de mutation s'appliquent

seulement avec une certaine probabilité. De même, dans le processus de croisement (respectivement de mutation), le lieu de croisement (mutation) à l'intérieur des chromosomes est choisi aléatoirement.

En évoluant ainsi de génération en génération, les candidats jugés performants permettront d'obtenir une progéniture plus performante que celle de la génération précédente, s'approchant ainsi d'une solution optimale. La structure générale d'un algorithme génétique est donnée par l'algorithme suivant :

Algorithme 1 *{Algorithme Génétique générique}*

Déterminer une population initiale P_0 constituée de n individus P_0^1, \dots, P_0^n

Faire

{Choisir 2 parents P_0^i et P_0^j dans P_0

$\text{fils} = \text{croisement}(P_0^i, P_0^j)$

$\text{mutation}(\text{fils})$

$\text{replacer}(\text{fils}, P_0)$ }

jusqu'à ce que le critère d'arrêt soit atteint.

3.4.1.3 La théorie des schémas

Du point de vue théorique, les algorithmes génétiques s'appuient sur un résultat qui n'est pas lié aux considérations biologiques mentionnées ci-dessus. Le théorème fondamental des algorithmes génétiques, encore appelé «Schema Theorem», est une des premières analyses du comportement des AG [HOL 75]. Un schéma (ou bloc de construction) est une séquence H décrivant un sous-ensemble de chaînes pour lesquelles on retrouve des gènes de même valeur à des positions particulières, c'est à dire des motifs de similarité. Ces motifs, en plus du codage employé pour la chaîne, utilisent un symbole supplémentaire, en l'occurrence le symbole #, désignant indifféremment une des valeurs du codage. Prenons pour exemple le cas d'un alphabet binaire $A = \{0, 1\}$, un schéma H sera créé à partir de l'alphabet enrichi $A^+ = \{0, 1, \#\}$. En considérant des chaînes (et des schémas) de longueur 6, le schéma $H = 1101\#\#$ permet alors de décrire un sous-ensemble de 4 chaînes $\{110100, 110101, 110110, 110111\}$. En étudiant les effets des opérateurs génétiques sur les schémas, il est alors possible d'observer leur développement ou leur déclin. C'est ainsi qu'est énoncé le théorème des schémas dans [HOL 75] :

"En utilisant une technique de reproduction proportionnelle à l'aptitude de chaque chromosome, nous pouvons prédire la croissance ou la décroissance relative d'un schéma H

dans la prochaine génération. "

[GOL 89] démontre formellement le théorème dans le cadre d'algorithmes génétiques simples utilisant un espace de codage binaire, une méthode de sélection proportionnelle et un opérateur de croisement à un point de coupure. Le nombre de blocs constructifs croît de manière exponentielle tout au long de la recherche, ce qui se traduit par une augmentation théorique de la valeur moyenne de la population courante à chaque génération. Le résultat primordial qui est exprimé de ce théorème, stipule que les algorithmes génétiques construisent la solution sub-optimale à partir des schémas courts et d'aptitude élevée contenus dans la population.

3.4.2 Application à la conception de RLI

Si les algorithmes génétiques sont une famille d'algorithmes basée autour des mêmes idées, il n'en existe pas moins de nombreuses variantes suivant le problème traité. Les AG se caractérisent par un certain nombre d'aspects qui sont : le codage (ou la représentation) du problème, la fonction coût, les opérateurs de sélection et de reproduction, l'arrêt de l'algorithme, etc. Dans cette section, nous allons passer en revue ces différents aspects par rapport au problème de conception de réseaux [KRO 02b, KRO 02a].

3.4.2.1 Codage des solutions

Les AG ne travaillent pas directement avec les solutions du problème mais avec une représentation de celles-ci appelée codage. La plupart des algorithmes génétiques classiques s'appuient fortement sur un codage universel utilisant un alphabet binaire et une longueur fixe des chaînes. Néanmoins, pour de nombreux problèmes complexes, il est très difficile de représenter une solution à l'aide d'une chaîne binaire. Aussi, d'autres formes de codage (réel, matrice, symbole, etc) ont été introduites. Cependant la représentation choisie doit satisfaire les contraintes suivantes :

- Le codage utilisé doit permettre de représenter l'ensemble des solutions possibles. Même s'il peut être intéressant de restreindre l'ensemble des solutions représentables (par exemple en se servant des éventuelles symétries ou des équivalences), il faut cependant veiller à ne pas entraver l'algorithme dans sa recherche en l'empêchant d'explorer certains sous-espaces.
- Les opérateurs de croisement et de mutation utilisent la structure des chromosomes, et les opérateurs les plus courants sont prévus pour être appliqués sur des chromo-

somes en forme de chaînes binaires. Ils doivent donc être adaptés selon le codage utilisé afin que les nouveaux individus engendrés par ces opérateurs soient significatifs, c'est à dire qu'ils puissent coder des solutions valides respectant les contraintes du problème. Ce point sera abordé plus en détail dans le paragraphe suivant.

Dans le contexte de conception topologique de réseau, les approches conventionnelles consistent essentiellement à coder les paramètres entrant dans la composition du réseau, tels que la configuration topologique ou les capacités des liaisons. Dans notre cas, nous utilisons le partitionnement de graphes, afin de définir la répartition des équipements industriels sur le réseau. Chaque individu doit alors représenter, par le biais de son chromosome, une partition du graphe. Nous utilisons la codification suivante : Soit un graphe $G = (V, E)$ composé de n sommets à partitionner en p sous-ensembles. Une partition donnée du graphe est représentée par un vecteur $V = (v_1, v_2, \dots, v_n)$ de taille n . La valeur de v_i indique le numéro de la partition (ou sous-ensemble) contenant le noeud i . La figure 3.15 illustre un exemple de codification. La chaîne $[001221120]$ permet de représenter la solution qui affecte les équipements $\{1, 2, 9\}$ au commutateur 0, $\{3, 6, 7\}$ au commutateur 1 et $\{4, 5, 8\}$ au commutateur 2. Un alphabet binaire $A = \{0, 1\}$ permet de représenter uniquement des bi-partitions d'un graphe. Nous utilisons par conséquent un alphabet constitué de p entiers entre 0 et $p - 1$.

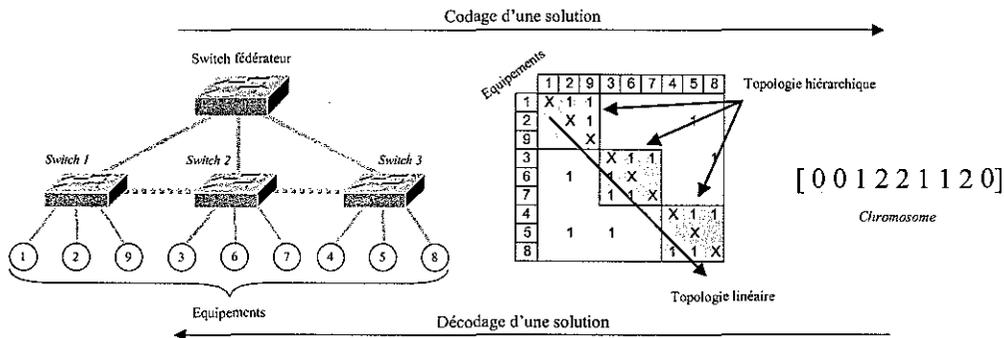


FIG. 3.15 – Codage d'une solution

Généralement, l'ensemble des problèmes de partitionnement restreignent l'espace de recherche par symétrie. Ainsi les chaînes $[001221120]$ et $[110220021]$ représentent le même individu. Dans notre cas, la solution sera identique du point de vue de l'architecture

hiérarchique. En revanche, la topologie linéaire est modifiée. Contrairement aux algorithmes spectraux, nous pouvons donc optimiser les deux architectures simultanément, en distinguant chaque chaîne.

3.4.2.2 Fonction coût (ou d'adaptation)

Soit les éléments de la matrice suivants :

- e , le nombre d'éléments total dans la matrice,
- $e_{ext.}$, le nombre d'éléments à l'extérieur des groupes (ou éléments exceptionnels),
- e_0 , le nombre d'éléments nuls dans les groupes,
- c_g , la charge du groupe g ,
- $c_{moy.}$, la charge moyenne.

Le problème de partitionnement de graphes, présenté dans la section 2.4.1, fait état d'une fonction permettant de :

- minimiser le coût total c des arêtes dont les extrémités appartiennent à deux sous-ensembles différents, c'est à dire la taille de coupure $e_{ext.}$. Afin d'être indépendant du nombre d'éléments total dans la matrice (e), nous exprimerons par la suite c comme le pourcentage de communications inter-groupe. On aura alors $c = \frac{e_{ext.}}{e}$.
- équilibrer la charge des divers sous-ensembles. Ceci revient à minimiser la variance v des poids des noeuds des différents sous-ensembles.

La fonction coût f_1 à optimiser est donc une combinaison linéaire des deux fonctions c et v :

$$f_1 = \min(c + \alpha.v) = \min\left(\frac{e_{ext.}}{e} + \alpha \cdot \sum (c_g - c_{moy.})^2\right) \quad (3.9)$$

avec α , entier positif représentant un compromis entre les deux critères.

Afin de comparer les solutions obtenues avec la méthode spectrale, nous utiliserons une deuxième fonction réduisant f_1 au premier critère, de taille de coupure. Nous avons alors :

$$f_2 = \min(c) = \min \frac{e_{ext.}}{e} \quad (3.10)$$

Une troisième fonction f_3 sera également utilisée [KRO 01a]. Proposée par [KUM 90], elle permet de regrouper les équipements ayant de fortes interactions dans le même groupe. Pour cela, cette fonction prend en compte le nombre d'éléments nuls dans les groupes.

$$f_3 = \frac{e - e_{ext.}}{e + e_0} \quad (3.11)$$

Cette fonction est bornée et permet de traduire les deux cas extrêmes. Si tous les éléments se trouvent à l'intérieur des partitions sans aucun élément vide ($e_{ext.} = e_0 = 0$), alors la fonction vaut 1 (cas idéal). Dans le cas contraire, les partitions sont vides et les éléments sont tous des éléments exceptionnels, alors la fonction vaut 0 ($e_{ext.} = e$). Ceci dit, ces deux situations extrêmes arrivent rarement. La fonction f_3 est cependant à maximiser contrairement aux fonctions f_1 et f_2 . Elle est également indépendante de la taille et de la densité de la matrice, et possède des propriétés de non-négativité.

3.4.2.3 Les opérateurs génétiques

Les opérateurs génétiques sont appliqués lors de chaque génération, de façon à produire, dans le temps, des populations successives de qualité croissante. Nous étudierons dans cette section les trois opérateurs génétiques de base : la sélection, le croisement, et la mutation.

La sélection

L'objectif est de favoriser les meilleures solutions en introduisant régulièrement des copies de ces dernières dans la population. Etant donné que la taille de la population reste constante, cela se traduit par la disparition des solutions de moindre qualité. Il existe plusieurs techniques pour la sélection des chromosomes à reproduire (stratégie élitiste, sélection par tournois, etc). Une des plus utilisées est la technique de la roulette³ dans laquelle la probabilité de reproduction de l'individu est directement proportionnelle à sa valeur d'adaptation (déterminée par la fonction objectif). La roulette est alors divisée en secteurs proportionnels à la valeur des individus. Ainsi, plus la valeur d'une chaîne est élevée, plus cette chaîne a des chances d'être choisie pour la reproduction. La figure 3.16 illustre cette technique et nous pouvons noter qu'en faisant tourner cette roulette, nous aurons beaucoup plus de chance de sélectionner l'individu 1 que l'individu 5.

La sélection consiste donc à lancer la roulette plusieurs fois afin de constituer une population sur laquelle les opérateurs de reproduction seront appliqués. A cette technique est souvent adjointe une stratégie élitiste permettant de choisir systématiquement les deux meilleurs candidats de la population. Elle diffère de la roulette qui, elle, risque d'oublier les meilleurs candidats étant donné son approche probabiliste.

³Plus connu sous le nom de *Roulette Wheel*.

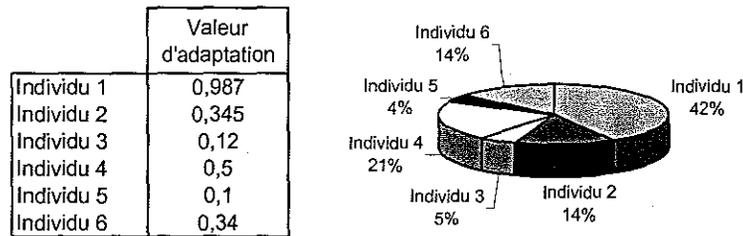


FIG. 3.16 – Exemple de sélection proportionnelle

L'opérateur de croisement (ou combinaison)

Le croisement, à l'instar de son équivalent biologique, est chargé de générer une nouvelle solution (enfant) à partir des solutions précédentes (parents). Il est réalisé en combinant les gènes des chaînes de chaque parent afin de former une nouvelle chaîne (enfant). Par exemple, le croisement peut être effectué selon deux points de coupure (figure 3.17), points à partir desquels les chaînes sont échangées.

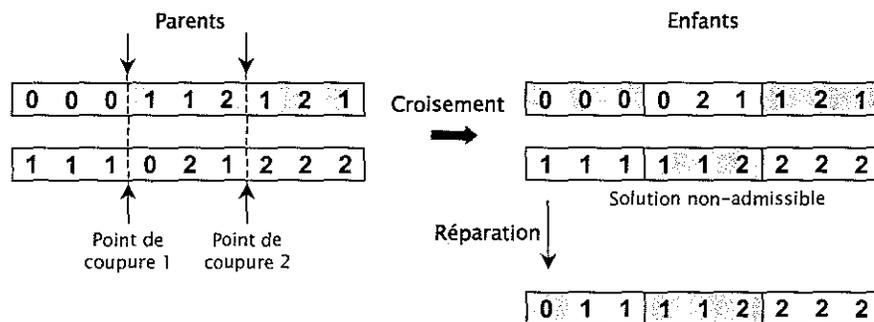


FIG. 3.17 – Opérateur de croisement évolué

Toutefois des solutions "enfants" non-admissibles peuvent être générées. Il s'avère nécessaire d'introduire une procédure auxiliaire dont le rôle est de retrouver les solutions admissibles les «plus proches» des solutions engendrées. L'exemple précédent montre par exemple une solution "enfant" ne respectant pas le nombre de sous-ensembles (partitions). Une phase de réparation est donc utilisée afin de rendre la solution admissible par rapport aux contraintes définies.

La mutation

Dans les algorithmes génétiques, l'opérateur de mutation est souvent considéré comme un opérateur secondaire par rapport au croisement. Toutefois, il ne doit pas être négligé pour autant, puisqu'il diversifie la population afin d'examiner d'autres parties de l'espace de recherche. Ainsi, il permet de prévenir des situations de blocage comme la convergence prématurée, dans des minima locaux de la fonction objectif.

L'opérateur de mutation consiste à changer aléatoirement la valeur d'une position d'une chaîne dans un individu. Avec un codage binaire, l'opérateur de mutation se définit comme le changement d'un gène de l'individu par son complémentaire. Cependant, l'utilisation d'un tel opérateur, avec un alphabet de cardinalité supérieure à 2, peut engendrer des solutions non-admissibles (Fig. 3.18.a). Nous utilisons une autre implantation de l'opérateur qui consiste à permuter 2 gènes (ou groupes) de l'individu parent (Fig. 3.18.b et c).

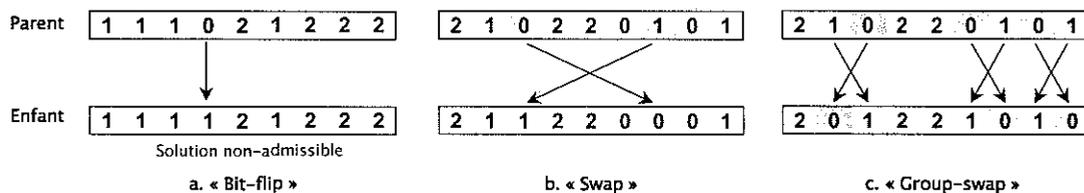


FIG. 3.18 – Opérateurs de mutation

3.4.2.4 Contrôle et mise en oeuvre de l'algorithme

Initialisation

Afin d'initialiser l'algorithme, une population initiale de chaînes doit être créée. Il existe plusieurs méthodes (aléatoire, clonage, résultat précédent, ...). Une population initiale diversifiée, en terme de performance des candidats, permet d'éviter de se retrouver avec des espèces dégénérées et de ne jamais s'approcher de la solution optimale. Nous utilisons alors une population générée aléatoirement. Nous verrons également la possibilité d'utiliser d'autres résultats afin de générer la population initiale, et notamment ceux des algorithmes spectraux.

Critère d'arrêt

Les algorithmes génétiques, tout comme les algorithmes évolutionnaires, sont dans l'absolu sans fin. Un critère d'arrêt doit être utilisé selon l'objectif recherché par l'utilisateur. Nous pouvons distinguer deux grandes familles de critères d'arrêt basés soit sur un critère temporel, soit sur un critère qualitatif de la solution. La première famille de critères d'arrêt représente la majeure partie des critères d'arrêt employés. Ils utilisent le temps d'exécution de l'algorithme ou le nombre d'itérations, et sont très faciles à mettre en oeuvre. La deuxième famille de critères d'arrêt consiste à stopper l'algorithme lorsque la fonction d'adaptation a atteint un seuil acceptable ou stagne depuis plusieurs générations. Cela permet de caractériser un algorithme qui n'arrive pas à trouver de meilleures solutions et qui est dans une configuration de minima locaux. Dans notre étude, ce dernier type de critère d'arrêt n'est pas envisageable puisque que nous avons décorrélé l'évaluation de la solution (du point de vue temporel) de la recherche de la meilleure solution. C'est pourquoi, la politique que nous utiliserons sera d'arrêter l'algorithme au bout d'un temps donné correspondant au souhait de l'utilisateur.

Paramètres

Après avoir choisi entre les divers mécanismes de sélection, de croisement, de mutation et les conditions d'exécution de l'algorithme, nous devons fixer les différents paramètres suivants : probabilité de croisement (p_c), probabilité de mutation (p_m) et taille de la population (t_{pop}). Si diverses études portent sur le choix de ces paramètres et peuvent guider dans le choix de ceux-ci, il n'en demeure pas moins indispensable de réaliser des essais pratiques afin d'optimiser leur valeur en fonction du problème à résoudre. Par exemple, nous veillerons à ne pas faire trop de modifications durant le croisement, de manière à hériter autant que possible des caractéristiques des parents. De même, il faut en réaliser suffisamment de manière à explorer de nouvelles formes de permutation et donc enrichir le domaine de recherche. Nous avons réalisé différents tests en fonction de chaque paramètre, et nous verrons dans la suite leur influence sur les performances de l'algorithme.

3.5 Conclusion

Dans ce chapitre, nous avons proposé une démarche et une modélisation du problème de conception de réseaux *Ethernet Industriel* basés sur une architecture commutée. Le

contexte industriel requiert cependant une fiabilité accrue, c'est pourquoi, nous avons complété l'architecture hiérarchique par une architecture linéaire assurant la connectivité du réseau. Cette redondance augmente également la tolérance aux pannes des noeuds du réseaux.

Dans la partie 3.2, nous avons vu que la minimisation du délai d'acheminement des messages (critère de conception classique) revient à minimiser les échanges entre les équipements réseaux (communications inter-groupe) et à équilibrer la charge des équipements réseaux (communications intra-groupe). Ces deux critères correspondent à ceux utilisés classiquement dans le problème du partitionnement de graphes. C'est pourquoi, nous proposons d'utiliser les techniques de résolution du partitionnement telles que les algorithmes spectraux et les algorithmes génétiques, afin d'optimiser l'architecture. Bien que fondamentalement différents dans leurs concepts, ils présentent des caractéristiques intéressantes.

Les algorithmes spectraux permettent de trouver une répartition des éléments dépendants d'un graphe en étudiant globalement sa connectivité. Ils sont basés sur la minimisation de la taille de coupure du graphe (cf. fonction quadratique 3.2) et ne permettent pas de considérer l'équilibre de charge. De même, ils ne respectent pas la contrainte de coût minimal en terme de nombre de commutateurs. Nous avons alors proposé une modification de l'algorithme afin de réduire le nombre de commutateurs. La fiabilisation de la topologie est obtenue après avoir défini l'architecture hiérarchique. Elle est également réalisée en utilisant les propriétés spectrales du graphe d'échanges, établi à partir de la solution hiérarchique.

Les algorithmes génétiques sont basés sur une approche évolutive qui consiste à faire évoluer un ensemble de solutions du problème, afin de générer de nouvelles solutions plus performantes. Ils se caractérisent par leur représentation du problème. Dès lors, toute la connaissance du problème peut être traduite et exploitée afin d'améliorer la recherche de solutions optimales. Ainsi, l'utilisateur peut définir chacune des caractéristiques de l'algorithme telles que le codage, la fonction coût, ou encore les opérateurs, en fonction du problème à traiter. Nous proposons donc un algorithme génétique adapté au problème de conception de réseaux *Ethernet Industriel*.

Dans le chapitre suivant, nous allons étudier les performances des algorithmes en terme de partitionnement. Ensuite, une évaluation temporelle des solutions est présentée afin de valider les architectures vis-à-vis de la contrainte temporelle fixée par l'application.

Chapitre 4

Résultats et évaluation de performances

Dans ce chapitre, nous présentons les résultats obtenus à l'aide des algorithmes de conception de réseau proposés dans le chapitre précédent. Les résultats présentés ont été obtenus à l'aide d'un outil logiciel de *Conception et d'Évaluation de Réseaux Industriels "Switched Ethernet"* (CERISE) que nous avons développé au laboratoire (Fig. 4.1). Cet outil utilise en entrée, un format de graphe commun, adopté dans le domaine du partitionnement et les divers outils de partitionnement de graphes tels que *Chaco*¹, *Metis*² et *Jostle*³. Cela va nous permettre de tester nos algorithmes sur des graphes servant de référence dans la communauté de ce domaine. À partir de ces graphes, il permet d'optimiser l'architecture selon les deux techniques proposées dans le chapitre précédent. Nous pouvons modifier les paramètres de chaque algorithme comme le nombre de partitions, la probabilité des opérateurs génétiques, le critère d'arrêt, ou encore la fonction coût. Les solutions obtenues peuvent ensuite être évaluées en terme de délai, selon les différentes méthodes présentées dans ce chapitre.

Dans un premier temps, nous réalisons une étude sur l'influence des divers paramètres des algorithmes génétiques, afin d'optimiser leur valeur en fonction du problème de conception de réseaux.

¹Chaco : <http://www.cs.sandia.gov/~bahendr/chaco.html>

²Metis : <http://www-users.cs.umn.edu/~karypis/metis/index.html>

³Jostle : <http://www.gre.ac.uk/~c.walshaw/jostle/>

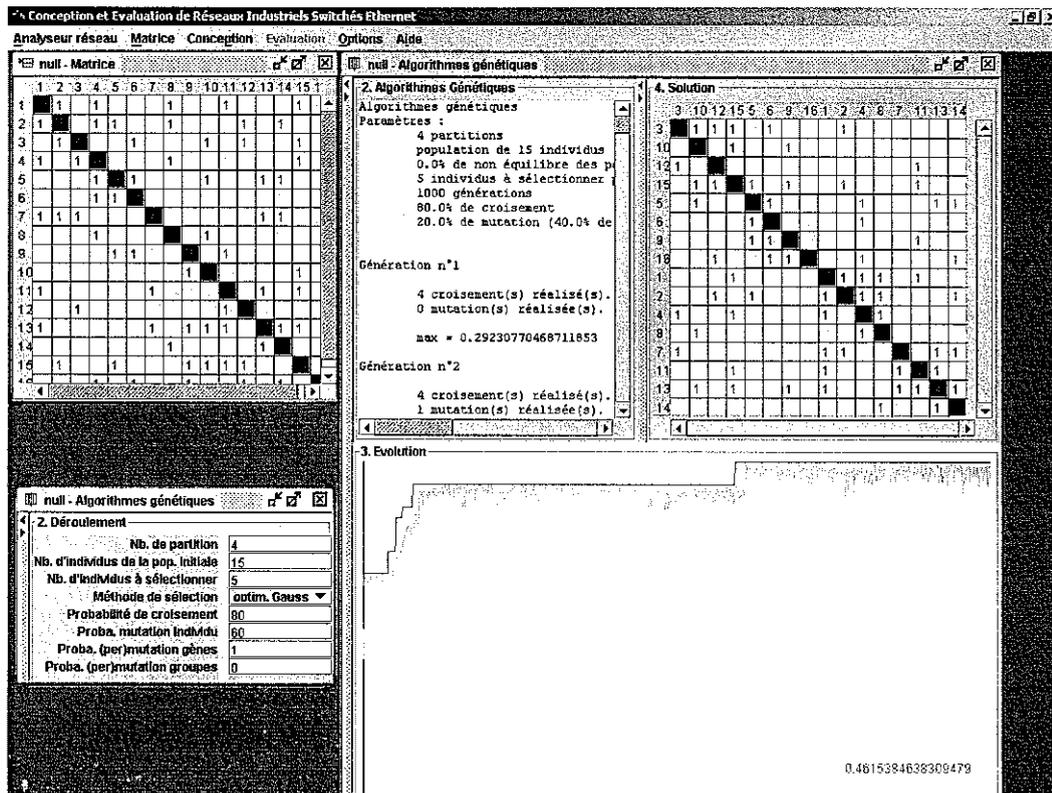


FIG. 4.1 – Interface graphique de l'outil de conception "CERISE"

4.1 Etude des paramètres des AGs

L'ajustement des paramètres d'un algorithme génétique est toujours une opération délicate. Il doit être optimisé pour chaque problème traité, et constitue par conséquent une part importante dans la mise en oeuvre d'un algorithme génétique. Les trois principaux paramètres intervenant sur la performance de l'algorithme, sont la taille de la population, la probabilité de croisement, et la probabilité de mutation. Nous proposons dans cette section d'étudier l'influence de chacun des paramètres afin de déterminer leur valeur donnant les performances optimales.

4.1.1 Influence de la taille de la population t_{pop} .

Quelle doit être la taille de cette population? Les biologistes ont introduit le concept de "*diversité requise*" pour représenter le fait que pour survivre, une espèce doit être

suffisamment hétérogène. Ainsi, une population trop petite évoluera probablement vers un optimum local peu intéressant (convergence prématurée) puisque l'algorithme ne disposera pas assez d'échantillons représentatifs de l'espace de recherche. Par ailleurs, une population trop grande (trop d'informations) augmentera le temps de calcul et ralentira la convergence. La taille de la population doit donc être choisie de façon à trouver un bon compromis entre temps de calcul et qualité du résultat. La valeur de la taille de la population $t_{pop.}$ qui est généralement utilisée, varie entre 30 et 60. Cependant, elle dépend de la puissance de calcul dont on dispose, des méthodes utilisées (codage, sélection, opérateurs génétiques...) et de la fonction d'adaptation. Si la fonction à optimiser comporte peu d'optima locaux et un optimum global, la population nécessaire sera plus petite que dans le cas d'une fonction plus complexe comportant de nombreux optima locaux.

La figure 4.2 illustre l'influence de ce paramètre sur la qualité de la solution et le temps de calcul. Nous avons testé différentes valeurs de $t_{pop.}$ (10, 20, 30 et 200) sur un graphe de 100 noeuds. Pour chaque valeur du paramètre $t_{pop.}$, nous avons deux tracés représentant l'évolution de la valeur optimale de la fonction coût, et de la valeur moyenne pour la population de la génération en cours, en fonction du nombre de générations. Nous avons également fait figurer le temps mis pour atteindre l'optimum commun.

Une valeur de $t_{pop.} = 200$ permet d'obtenir en très peu de génération (≈ 90), la solution optimale. Cependant, nous remarquons que l'algorithme converge difficilement et le temps de calcul devient vite prohibitif ($\approx 7'20$). Une valeur de $t_{pop.} = 30$ permet de diviser par 2 le temps de calcul au détriment du nombre de générations. Par ailleurs, une taille de population $t_{pop.} = 10$ montre que l'algorithme converge vers la solution optimale avec 5 fois plus de générations que dans le premier cas, mais aussi avec un temps de calcul réduit par 3. Ceci s'explique par la faible diversification introduite à l'initialisation.

Néanmoins, la valeur de $t_{pop.} = 20$ permet de trouver l'optimum en réduisant le nombre de génération de $t_{pop.} = 10$ par 2, mais aussi le temps de calcul. Cela confirme la notion de "*diversité requise*". Une taille de population $t_{pop.} = 20$ semble alors être un bon compromis entre qualité du partitionnement et temps de calcul.

4.1.2 Influence de la probabilité de croisement p_c

L'opérateur de croisement a pour but d'enrichir la diversité de la population. Plus le taux de croisement p_c sera élevé, plus il y aura de nouvelles structures dans la population.

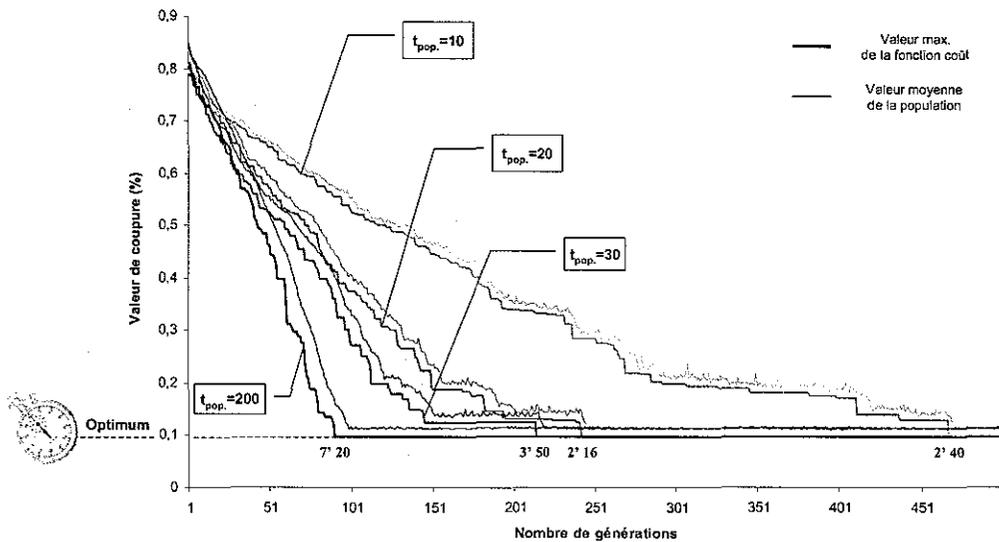
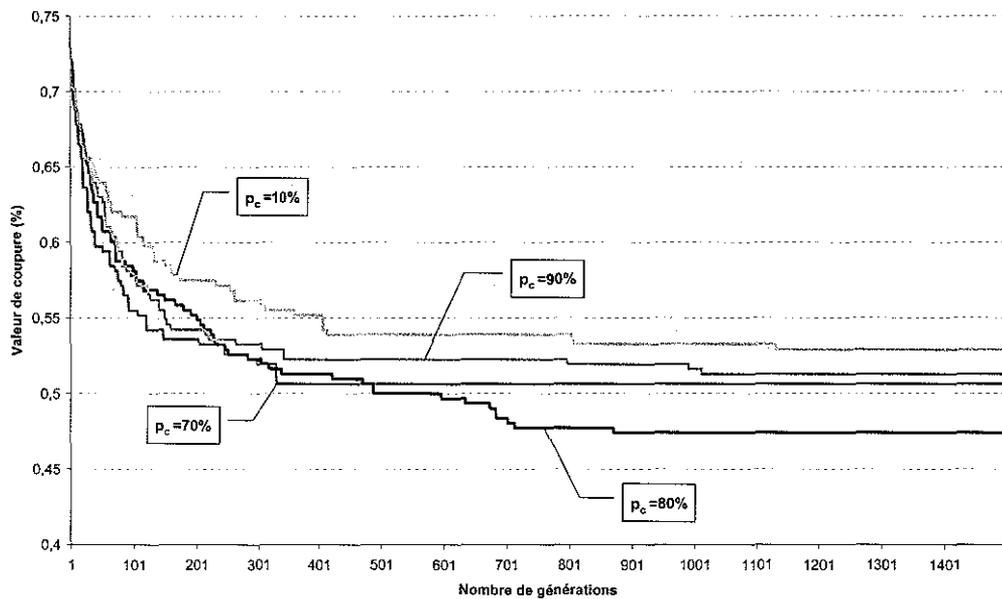


FIG. 4.2 – Influence de la taille de la population ($t_{pop.}$)

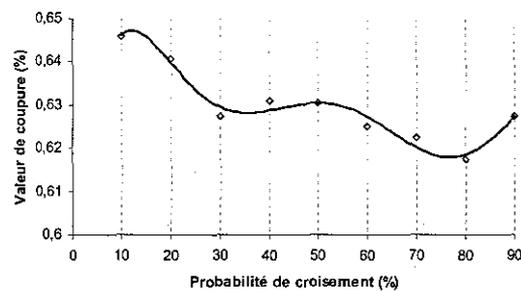
C'est pourquoi, la valeur de p_c est généralement supérieure à 60%. En effet, un faible taux de croisement appauvrit la population et ralentit l'évolution. En revanche, un taux trop élevé risque d'écarter prématurément de bonnes structures sans les avoir exploitées.

Le graphe, présenté figure 4.3, illustre l'influence de cette probabilité sur la qualité de la solution. Il a été réalisé à partir d'un graphe constitué de 80 noeuds. L'algorithme génétique a été exécuté, pour différentes valeurs de p_c , sur 1500 générations avec les autres paramètres fixés à : $t_{pop.} = 20$, $p_m = 40\%$. Le graphe illustre uniquement les tracés obtenus pour une probabilité de 10, 70, 80 et 90. Il montre la nécessité de choisir un taux de croisement important afin de diversifier la population à chaque génération. Ainsi, un taux de 80% permet d'obtenir la meilleure solution. En revanche, un taux trop faible (ici $p_c = 10\%$) ralentit fortement l'évolution de l'algorithme. L'appauvrissement de la population génère des situations de blocage provoquant la stagnation de l'algorithme. Par ailleurs, le tracé de $p_c = 90\%$ montre qu'un taux trop élevé ne permet pas à l'algorithme d'exploiter les individus potentiellement "bon". Il converge alors prématurément vers une solution peu performante.

Afin de fixer la valeur du taux de croisement, nous avons observé la même démarche pour une série de 100 graphes de taille fixe (100). Nous avons ainsi relevé, pour chaque

FIG. 4.3 – Influence de la probabilité de croisement (p_c)

valeur du taux de croisement, la valeur de coupure de la solution. Le tracé, présenté sur la figure 4.4, représente la courbe de tendance obtenue à partir des valeurs moyennes sur les 100 graphes. Par la suite, nous choisirons une valeur de p_c de 80%.

FIG. 4.4 – Choix de la probabilité de croisement (p_c)

4.1.3 Influence de la probabilité de mutation p_m

La diversification est également introduite par l'opérateur de mutation. Cet opérateur a pour but de libérer les situations de blocage (convergence prématurée, etc) dans des

minima locaux de la fonction objectif. Il ne doit pas agir trop souvent sous peine de nuire à la recherche guidée par l'opérateur de croisement. La probabilité de mutation p_m est donc généralement faible et de l'ordre de 0,1% à 5%.

Comme dans le paragraphe précédent, le graphe, présenté figure 4.5, illustre l'influence du taux de mutation sur la qualité de la solution. Il a été réalisé dans les mêmes conditions d'expérience à savoir : un graphe constitué de 80 noeuds, 1500 générations, $t_{pop.} = 20$, $p_c = 80\%$. L'algorithme génétique a été exécuté pour différentes valeurs de p_m et le graphe illustre uniquement les tracés obtenus pour une probabilité de 10, 30, 80 et 50.

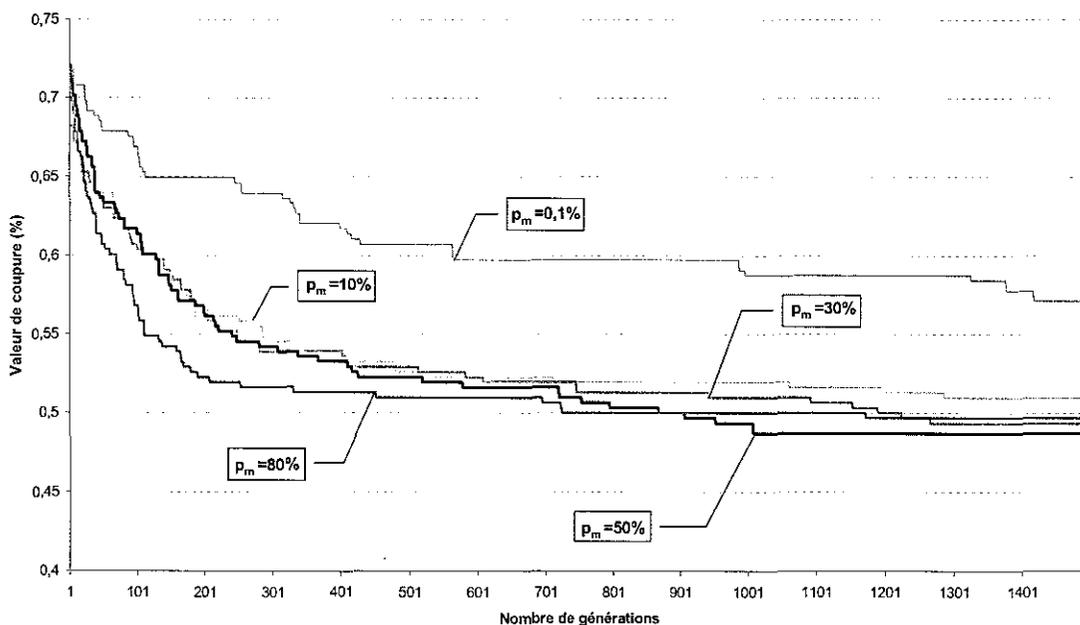


FIG. 4.5 – Influence de la probabilité de croisement (p_m)

Dans notre cas, un taux de mutation de 50% semble être le meilleur compromis. Nous pouvons constater que nous sommes loin des valeurs préconisées ($0,1\% < p_m < 0,5\%$). Cependant, les paramètres dépendent du problème traité et notamment du codage utilisé. Les valeurs proposées sont généralement utilisées dans des algorithmes simples utilisant un codage binaire, de faible longueur, un croisement un point et une mutation simple. Notre opérateur a été adapté pour manipuler des chaînes utilisant un alphabet constitué de p entiers, p étant le nombre de partition. De plus, il réalise des permutations de gènes

afin de ne pas engendrer des solutions non-admissibles. Tout ceci montre l'intérêt d'étudier le comportement des paramètres afin d'optimiser leur valeur en fonction du problème à résoudre.

Afin de fixer la valeur du taux de mutation, nous avons utilisé la série de 100 graphes de taille fixe (100), pour laquelle nous avons relevé, pour chaque valeur du taux de croisement, la valeur de coupure de la solution. Le tracé, présenté sur la figure 4.6, représente la courbe de tendance obtenue à partir des valeurs moyennes sur les 100 graphes. Par la suite, nous choisirons une valeur de p_c de 50%.

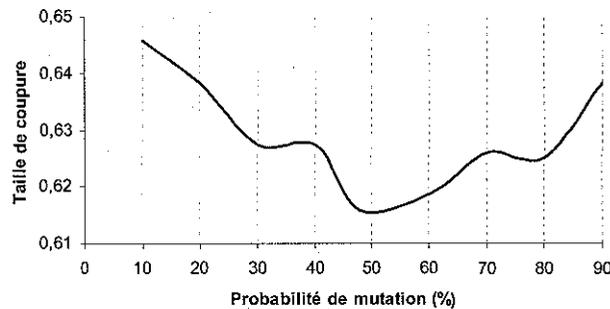


FIG. 4.6 – Choix de la probabilité de croisement (p_m)

En résumé, l'efficacité d'un algorithme génétique dépend fortement de la façon dont est menée l'exploration de l'espace des solutions. Si une diversification trop faible freine l'évolution, une diversification trop forte ne permet pas d'exploiter les solutions. Le choix des valeurs des paramètres de l'algorithme ($t_{pop.}$, p_c , p_m) permet d'optimiser la recherche et d'améliorer la performance. L'ensemble des tests, présenté dans cette section, a permis de déterminer la valeur de ces paramètres pour notre algorithme.

Dans la suite, nous allons évaluer et comparer la performance des algorithmes spectraux et des algorithmes génétiques (avec les paramètres fixés). Cependant, la nature aléatoire des algorithmes génétiques les caractérise comme des algorithmes "*bruités*". C'est pourquoi, contrairement aux algorithmes spectraux, nous exécuterons chaque test 10 fois de suite afin de prendre une moyenne sur l'ensemble des solutions obtenues.

4.2 Performances intrinsèques des algorithmes

Afin de comparer les performances des deux algorithmes, nous avons d'abord réalisé l'évaluation indépendamment du contexte d'application, c'est à dire en considérant que le meilleur algorithme est celui qui fournit la meilleure valeur de coupure $|E_0|$ dont la définition est présentée section 2.4.1.

Pour cela, nous avons utilisé, dans un premier temps, des "graphes tests" servant de référence dans la communauté scientifique du partitionnement des graphes que l'on peut trouver sur le site de l'Université de Greenwich⁴. Ce site répertorie les meilleures partitions trouvées jusqu'ici pour une série de graphes, dans le but de fournir un référentiel et des ressources pour l'expérimentation. Bien que ces "graphes tests" ne soient pas toujours représentatifs de notre domaine applicatif de part leur taille n ($2395 < n < 448695$) et leur densité⁵ d ($d < 1\%$), nous avons testé nos algorithmes sur quelques graphes. Par exemple, la meilleure bisection du graphe "Add_20" (constitué de 2395 noeuds, 7462 arêtes) obtenue jusqu'ici, a été trouvée à l'aide de l'outil de partitionnement de graphes *Chaco*, basé sur une méthode *multilevel-KL* (cf. 2.4.2.3). Le tableau 4.1 présente les résultats obtenus en terme de taille de coupure pour un partitionnement de la matrice en 2, 4 et 8 parties équilibrées. Malgré l'adaptation de nos algorithmes au problème de conception de réseaux *Ethernet Industriel*, nous remarquons qu'ils restent efficaces pour le problème de partitionnement classique.

| Nombre de parties | Meilleure partition connue (outil) | k-Metis | Cerise (AS) | Cerise (AG) |
|-------------------|------------------------------------|---------|-------------|-------------|
| 2 | 630 (Chaco) | 774 | 820 | 750 |
| 4 | 1203 (Jostle) | 1214 | 1325 | 1260 |
| 8 | 1758 (Iterated-Jostle) | 1858 | 2100 | 1850 |

TAB. 4.1 – Résultats obtenus à partir du graphe Add20

Dans un deuxième temps, nous avons réalisé d'autres tests à l'aide d'une série de 1000 graphes générés aléatoirement avec les caractéristiques suivantes : densité $d = 10\%$ et taille du graphe $n = 80$. La figure 4.7 illustre les résultats obtenus par chaque algorithme. Nous

⁴<http://www.gre.ac.uk/~c.walshaw/partition/>

⁵La densité du graphe correspond au rapport du nombre d'arêtes présentes dans le graphe sur le nombre d'arêtes possibles.

obtenons dans 76% des cas, une meilleure solution avec les algorithmes génétiques.

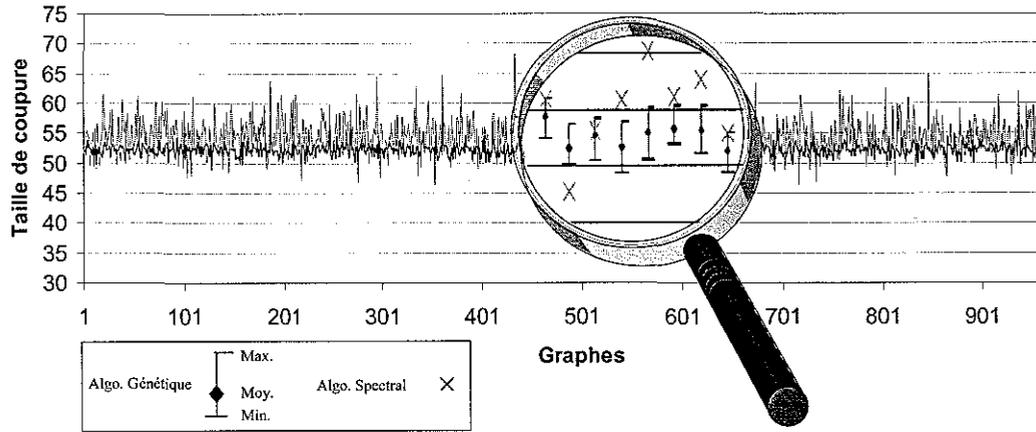


FIG. 4.7 – Evaluation du partitionnement

Il est alors intéressant d'observer le comportement de chaque algorithme en fonction des caractéristiques des graphes. Dans les sections suivantes, nous étudions l'influence de chacune des caractéristiques du graphe, et comparons les performances de chaque algorithme. Les algorithmes génétiques seront paramétrés suivant les valeurs obtenues dans la section 4.1 ($t_{pop.} = 20$, $p_c = 80$ et $p_m = 50$).

4.2.1 Influence de la taille des graphes

Afin d'observer l'influence de la taille des graphes, nous avons généré une série de 200 graphes de densité d fixe ($d = 10\%$) et de taille variable ($20 < n < 220$). La figure 4.8 illustre les résultats obtenus. Dans 88% des cas, soit 176 graphes, nous obtenons une meilleure solution avec les algorithmes génétiques. De plus, nous pouvons constater des écarts plus importants entre les 2 méthodes pour des graphes de petite taille (inférieure à 40). Par exemple, l'amélioration la plus importante apportée par les algorithmes génétiques par rapport aux algorithmes spectraux, concerne le graphe de taille $n = 26$, avec une réduction de la taille de coupure de 50%.

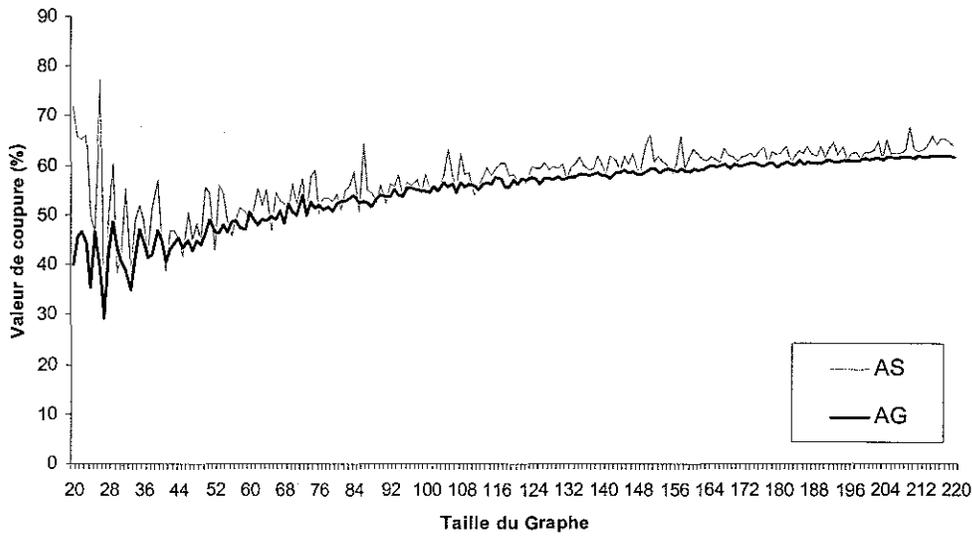


FIG. 4.8 – Influence de la taille des graphes

4.2.2 Influence de la densité des graphes

L'influence de la densité des graphes a été étudiée sur un ensemble de 280 graphes de taille fixe ($n = 80$), et de densité d variable ($1\% < d < 90\%$).

La figure 4.9 illustre les résultats obtenus. Dans un premier temps, nous pouvons voir que plus le graphe est dense, plus il est difficile de satisfaire le critère de minimisation des communications inter-groupe. Dès lors, les valeurs de coupure des différentes solutions sont proches. Par ailleurs, les algorithmes génétiques montrent un meilleur comportement que les algorithmes spectraux puisqu'ils fournissent dans 94% des cas, une meilleure solution.

Comme dans l'étude précédente, nous pouvons encore constater des écarts plus importants entre les 2 méthodes pour des graphes de faible densité (inférieure à 20). Par exemple, l'amélioration la plus importante apportée par les algorithmes génétiques par rapport aux algorithmes spectraux, concerne le graphe de taille $d = 9\%$, avec une réduction de la taille de coupure de 53%.

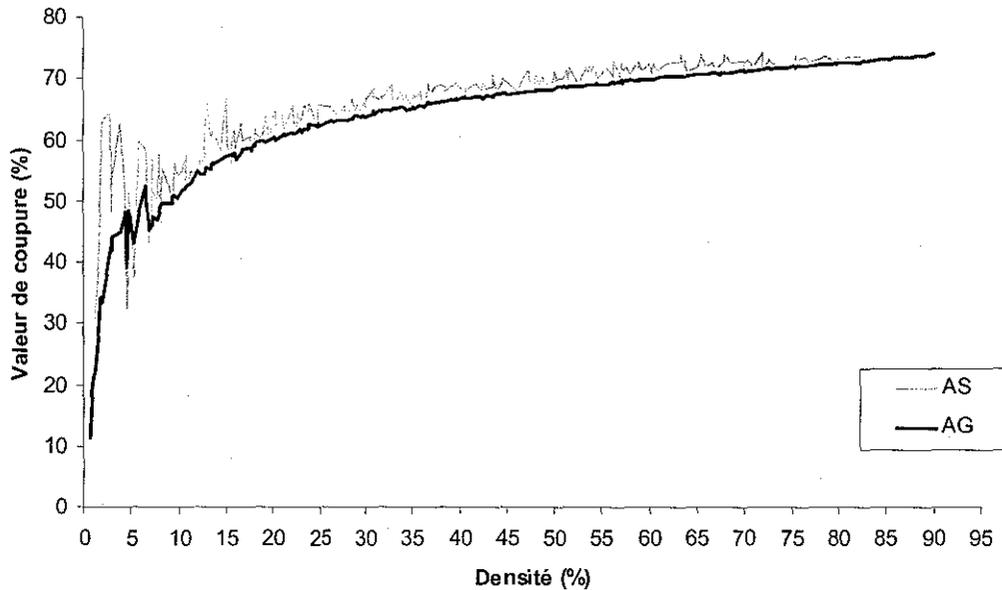


FIG. 4.9 – Influence de la densité des graphes

4.2.3 Synthèse

La comparaison des deux algorithmes de conception a été réalisée indépendamment de l'application à l'aide de la fonction objectif qui consiste à minimiser la taille de coupure du graphe. Nous avons présenté les résultats permettant de comparer les méthodes développées en fonction du type de graphe à traiter. Les algorithmes génétiques semblent présenter une plus grande fiabilité et robustesse que les algorithmes spectraux. Cette performance peut notamment s'expliquer par leur plus grande exploration de l'espace de recherche, mais aussi par l'approche non-récurrente développée dans les algorithmes génétiques.

Dans la section 3.2, nous avons observé la corrélation entre la minimisation du délai d'acheminement des messages dans le réseau, et la minimisation des critères liés au partitionnement de graphes. Après avoir comparé la performance des algorithmes, nous proposons d'évaluer la solution par rapport au contexte de l'application, c'est à dire le délai d'acheminement des messages. Nous verrons dans un premier temps les travaux permettant d'évaluer la performance des réseaux Ethernet commutés. Ils reposent sur l'évaluation des délais moyens d'acheminement des messages. Puis, nous utiliserons une méthode de calcul

de majorant du délai, qui a été développée au laboratoire, afin de valider l'architecture réseau optimisée par rapport aux contraintes temporelles fixées par l'application.

4.3 Evaluation du délai moyen

Outre les outils de simulation réseau (*ComNet*, *OpNet*, *NS*, ...), il existe à ce jour divers travaux sur l'évaluation de performance des réseaux Ethernet commutés dans le milieu industriel [KAM 99, RÛP 99, SON 01, TOR 00]. Tous reposent sur une analyse de file d'attente et ne diffèrent que par leurs hypothèses de modélisation des équipements et du trafic. Le paramètre de performance étudié est le délai d'acheminement de bout en bout des messages. Ce critère est fortement corrélé avec l'activité des équipements d'interconnexion. En effet, plus le commutateur est chargé, moins il peut servir de messages et par conséquent ceux-ci seront mis en attente dans les mémoires de ces équipements avant d'être traités. Nous devons donc déterminer la charge des commutateurs pour en déduire le délai d'acheminement des messages.

Dans la suite, nous présentons une méthode de calcul de la charge des commutateurs [TOR 00], puis nous déterminerons les délais d'acheminement des messages selon la modélisation proposée par [KAM 99].

4.3.1 Etude de la charge des commutateurs

4.3.1.1 Méthode analytique

Pour étudier la charge des commutateurs dans des architectures hiérarchiques ou linéaires, [TOR 00] propose une solution systématique utilisant une représentation sous forme de graphe. A partir de cette structure, le concepteur réseau doit dimensionner les équipements réseau en faisant une étude prédictive de leur charge relativement à une modélisation du trafic. Ainsi, pour assurer qu'aucun paquet ne soit rejeté, la charge maximale de tous les nœuds peut être calculée en se référant au trafic réseau maximum.

Soit T_{ij} le flux d'informations exprimé en paquets par unité de temps entre deux nœuds de terminaison i et j . Ces différents flux peuvent être rassemblés dans une matrice $n \times n$ appelée matrice de trafic $T = (T_{ij})$ où n représente le nombre de nœuds de terminaison. La matrice T peut représenter le flux maximum, le flux moyen, ... entre deux nœuds de terminaison. Cette matrice ne contient naturellement que des zéros sur la diagonale.

Parallèlement, la topologie arborescente du réseau est modélisée par un graphe orienté $G = [X, U]$, où X est l'ensemble des sommets du graphe représentant l'ensemble des nœuds du réseau, et U l'ensemble des arcs entre deux sommets (Fig. 4.10). Chaque arc est une paire ordonnée de sommets (i, j) où i est appelé le sommet source, et j le sommet puits. Le sommet j est un successeur de i , et i un prédécesseur de j , s'il existe un lien (i, j) dans le graphe. L'ensemble des successeurs de i est noté Γ_i . Comme le graphe G décrit une arborescence, il existe un sommet racine $r \in X$ tel que pour tous les sommets $j \in G$, il existe un chemin allant de r jusqu'à j . La profondeur d'un sommet $x \in X$ noté $d(x)$ correspond à la longueur du plus grand chemin débutant à partir de x . La profondeur du graphe est égale à la profondeur maximale de ses sommets.

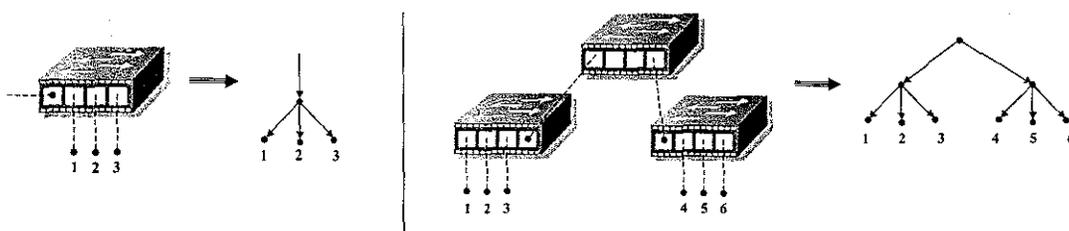


FIG. 4.10 – Modélisation d'un commutateur et d'un réseau commuté

Dans la suite, on considère que le réseau contient n nœuds de terminaison numérotés de 1 à n , et l'ensemble de ces nœuds est noté X_t . Pour tout ensemble de $X = \{x_1, \dots, x_m\} \subseteq X_t$, $e_{n,x}$ est un vecteur de dimension n avec des 1 pour tous les éléments de X et des 0 ailleurs. S'il n'y a pas d'ambiguïté sur la taille du vecteur, alors on utilise la notation simplifiée e_X .

Le théorème suivant est la première étape pour analyser la charge de réseaux en arbre. Il montre que le flux total à travers tous les nœuds de profondeur 1 peut être obtenu par un simple calcul matriciel sur la matrice de trafic.

Théorème 1 : Le flux total traversant un nœud i ayant une profondeur de 1 est égal à :

$$\lambda_i = e_{\Gamma_i}^T T^m e_{X_t - \Gamma_i} + e_{X_t - \Gamma_i}^T T^m e_{\Gamma_i} + e_{\Gamma_i}^T T^m e_{\Gamma_i}$$

avec :

$e_{\Gamma_i}^T T^m e_{X_t - \Gamma_i}$ représente le trafic du nœud i généré par ses nœuds successeurs à destination de son nœud prédécesseur.

$e_{X_t - \Gamma_i}^T T^m e_{\Gamma_i}$ représente le trafic du nœud i généré par son nœud prédécesseur à destination de ses nœuds successeurs.

$e_{\Gamma_i}^T T^m e_{\Gamma_i}$ représente le trafic du nœud i entre ses nœuds successeurs.

T^m est la matrice de trafic apparent définie ainsi :

$$T^m = \begin{cases} T_{ij} & ; j \neq k \\ T_{ij} & ; j = k \text{ et } d(\Gamma_j^{-1}) > 1 \\ 0 & ; j = k \text{ et } d(\Gamma_j^{-1}) = 1 \end{cases}$$

avec $d(\Gamma_j^{-1})$, la profondeur des prédécesseurs de j .

Le deuxième théorème permet de compléter l'analyse de charge du réseau en montrant une procédure de réduction de graphe qui permet d'appliquer de manière récursive le théorème 1.

Théorème 2 : Considérons un réseau représenté par un arbre T , et une matrice de trafic T . T a n nœuds et une profondeur de d . Soient $V = \{v_1, v_2, \dots, v_q\}$ l'ensemble des nœuds de T ayant une profondeur de 1, $X_t^* = \Gamma_v$ l'ensemble des nœuds de terminaison de T avec un prédécesseur dans V et supposons que $|X_t^*| = p \leq n$. Considérons maintenant T' , une version réorganisée de T , obtenue par déplacement de tous les nœuds de terminaison dans X_t^* et des liens associés. Les propriétés de T' sont les suivantes :

1. T' a une profondeur de $d - 1$
2. T' a m nœuds de terminaison avec $m = n - p + q$
3. En renumérotant les nœuds de terminaison de T^m de 1 à m , et en définissant une matrice E de succession $n \times m$ comme suit : $E_{ij} = 1$ si le nœud de terminaison i dans T est un successeur ou le même que le nœud de terminaison de j dans T_j , sinon $E_{ij} = 0$, alors le flux total traversant le nœud j de T^m est donné par la somme des éléments de la ligne j et la colonne j du produit de la matrice : $T_j = E^T T_j E$ avec T^m , la matrice de trafic apparente définie dans le théorème précédent.

De ce deuxième théorème, découle l'algorithme d'analyse de charge des commutateurs dans une structure arborescente :

Considérons un réseau modélisé par le graphe T et la matrice de trafic T . T définit une arborescence de profondeur d . Soit $T_d = T, T_d = T$ et $i = d$.

1. Soit m_i l'ordre de T_i (numéros des nœuds de terminaison de T_i)
2. Faire les étapes suivantes pour $j = 1, \dots, m_i$

Nous commençons tout d'abord par étudier la topologie hiérarchique. Elle est constituée de quatre commutateurs de terminaison (A, B, C, D) et d'un commutateur fédérateur E jouant le rôle de backbone. Elle a donc une profondeur de 2, cela signifie que deux itérations (Fig. 4.11) sont suffisantes pour déterminer les charges des commutateurs et que $L = T_2$.

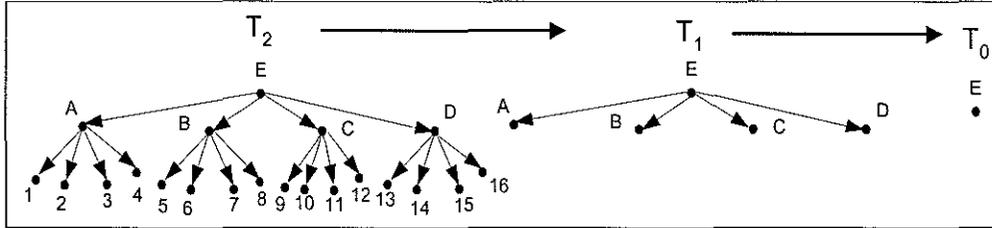


FIG. 4.11 – Graphes d'arborescence représentant la topologie hiérarchique

Pour calculer la matrice T_1 , il faut construire la matrice E_1 que l'on trouve à partir du théorème 2 avec $T_2'' = T_2$.

$$E_2^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad T_1 = E_2^T T_2'' E_2 = \begin{bmatrix} 17,4 & 17,6 & 11,2 & 24,2 \\ 28,8 & 17,4 & 17,6 & 29 \\ 35,2 & 28,8 & 17,4 & 33,8 \\ 50,6 & 45,8 & 41 & 38,7 \end{bmatrix}$$

En sommant les lignes et les colonnes de même indice, nous obtenons directement les charges des commutateurs A, B, C et D puisque les quatre nœuds de T_1 ont une profondeur de 1. D'où :

$$\lambda_A = \lambda_B = \lambda_C = 185 \text{ paquets/s et } \lambda_D = 263,1 \text{ paquets/s}$$

Pour déterminer la charge du commutateur E , il suffit de placer des 0 sur la diagonale de T_1 et on a T_1'' . Ensuite on applique la formule suivante :

$$\lambda_E = T_0 = E_1^T T_1'' E_1 = 363,6 \text{ paquets/s., avec } E_1^T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Le même principe est repris pour la deuxième topologie (linéaire), mais avec une profondeur de 4. Les différentes arborescences T sont présentées (Fig 4.12).

La figure 4.13 rassemble l'ensemble des charges des commutateurs en fonction des deux topologies.

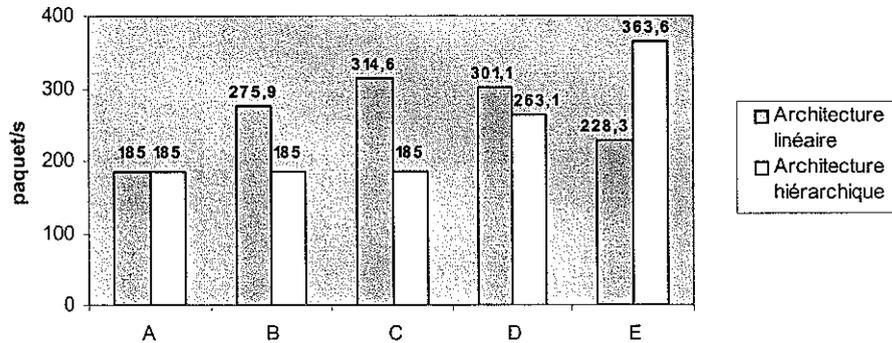


FIG. 4.13 – Charges des commutateurs

Après avoir déterminé les charges des commutateurs de l'architecture, nous pouvons calculer les délais d'acheminement des messages en utilisant la modélisation du réseau proposée dans [KAM 99].

4.3.2 Calcul du délai de bout en bout

4.3.2.1 Méthode analytique

Contrairement à l'approche définie par [RÚP 99], où les échanges sont de type maître / esclave et envoyés suivant des périodes fixes (cf. section 3.1.2.1), on se place ici dans un contexte où tous les nœuds de terminaison peuvent dialoguer entre-eux. Pour cela, [KAM 99] propose d'utiliser le système à file d'attente $M/D/1$ pour calculer les temps d'acheminement des messages de bout en bout. Cela signifie que le trafic généré par tous les nœuds de terminaison a un comportement Poissonien et que les messages ont des tailles fixes. Nous rappelons ici que le délai d'attente moyen dans la file d'un commutateur est alors égal à $W = \frac{\lambda T^2}{2(1-\lambda T)}$, où λ est le taux d'arrivée moyen et T le temps de service. Nous avons également : $\lambda < \lambda_{\max} = \frac{1}{T}$.

En considérant que tous les commutateurs travaillent en *store and forward*, le temps d'acheminement des communications entre deux nœuds de terminaison i et j est obtenu en sommant les temps d'attente dans les files de chaque commutateur traversé et les temps

de transmission (*forwarding*). D'où l'expression :

$$D_{ij} = \sum_{k \text{ commutateurs traversés}} W_k + \sum_{k \text{ forwardings}} T_k$$

Pour faciliter la comparaison entre plusieurs architectures de communication, deux indicateurs de performance sont étudiés. Le premier est le temps moyen d'acheminement d'un message de bout en bout défini ainsi :

$$D_{moyen} = \frac{\left(\sum_i \sum_j l_{ij} D_{ij} \right)}{\left(\sum_i \sum_j l_{ij} \right)}$$

avec :

l_{ij} est le trafic moyen du nœud de terminaison i vers j

D_{ij} est le temps d'attente moyen pour la communication entre i et j .

Le second indicateur est le rendement de l'architecture :

$$\eta = \frac{\left(\sum_i \sum_j l_{ij} \right)}{\left(\sum_i \text{charge du commutateur } i \right)}$$

Il est à noter que η est maximum (100%) pour une simple connexion en étoile, c'est à dire lorsque tous les nœuds de terminaison sont connectés à un seul commutateur. Le fait de traverser plusieurs commutateurs pour aller d'un nœud de terminaison à un autre diminue le rendement du système de communication.

4.3.2.2 Exemple numérique

Reprenons l'exemple traité précédemment en précisant que les commutateurs sont tous à 100Mb/s et la valeur approximative de T est égale à :

$$T = \frac{1518 \cdot 8}{100} = 121,44 \mu s \text{ (en prenant la taille maximale d'un paquet Ethernet).}$$

Calcul sur la topologie hiérarchique :

Les temps d'attente dans chaque commutateur sont donc :

$$W_A = \frac{\lambda_A T^2}{2(1-\lambda_A T)} = W_B = W_C = 1,4 \mu s$$

$$W_D = 2 \mu s$$

$$W_E = 2,8 \mu s$$

Et les temps de forwarding sont égaux :

- 121,44 μs pour les communications confinées à l'intérieur d'un commutateur
- 364,32 μs pour les communications entre deux commutateurs

Nous obtenons alors la matrice D suivante (exprimée en μs) :

$$D = \begin{bmatrix} 0 & 122,84 & 122,84 & 122,84 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 122,84 & 0 & 122,84 & 122,84 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 122,84 & 122,84 & 0 & 122,84 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 122,84 & 122,84 & 122,84 & 0 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 369,92 & 369,92 & 369,92 & 369,92 & 0 & 122,84 & 122,84 & 122,84 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 369,92 & 369,92 & 369,92 & 369,92 & 122,84 & 0 & 122,84 & 122,84 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 369,92 & 369,92 & 369,92 & 369,92 & 122,84 & 122,84 & 0 & 122,84 & 369,92 & 369,92 & 369,92 & 369,92 & 370,52 & 370,52 & 370,52 & 370,52 \\ 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 0 & 122,84 & 122,84 & 122,84 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 \\ 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 122,84 & 0 & 122,84 & 122,84 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 \\ 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 369,92 & 122,84 & 122,84 & 0 & 122,84 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 \\ 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 0 & 123,44 & 123,44 & 123,44 & 123,44 \\ 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 123,44 & 0 & 123,44 & 123,44 \\ 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 123,44 & 123,44 & 0 & 123,44 \\ 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 370,52 & 123,44 & 123,44 & 123,44 & 0 \end{bmatrix}$$

Le calcul des deux indicateurs de performance donne :

$$D_{moyen} = 350,85 \text{ ms} \quad \eta = 38,46 \%$$

Calcul sur la topologie linéaire :

Appliquons la même méthode, nous obtenons :

$$D_{moyen} = 354,57 \text{ ms} \quad \eta = 34,83 \%$$

Toutefois l'évaluation du délai moyen n'est pas suffisante, notamment dans un contexte industriel où les délais doivent être bornés. Nous devons donc aussi évaluer cette borne maximale pour vérifier la validité de nos solutions. Dans la section suivante, nous présentons une méthode de calcul de majorant développée au laboratoire [GEO 02, KRO 02c]. Elle permet de déterminer un majorant des délais de traversée des réseaux Ethernet commutés, et répond mieux au besoin d'évaluation dans un contexte temps réel.

4.4 Détermination de la borne maximale du délai

Pour calculer des majorants, nous proposerons de modéliser les communications en utilisant une loi d'arrivée stricte, et en particulier en utilisant le "calcul réseau". Le calcul réseau introduit dans [CRU 91] suppose que le nombre de données émises sur les liens du réseau ne dépasse pas une courbe d'arrivée (traditionnellement, un seau percé) pouvant à la fois représenter le trafic périodique et apériodique. Il permet de modéliser des commutateurs Ethernet et leurs interconnexions via l'assemblage de composants élémentaires tels que des multiplexeurs, démultiplexeurs, buffers. Les propriétés temporelles de ces composants sont données dans [CRU 91]. Le calcul réseau est alors utilisé pour déterminer le délai maximum de bout en bout de paquets appartenant à un même scénario de communication.

Après avoir présenté les concepts du calcul réseau et la modélisation par composants élémentaires, nous utiliserons alors l'algorithme de détermination de la borne maximale du délai afin d'évaluer les solutions obtenues à partir des algorithmes de conception de réseaux industriels commutés [KRO 02c].

4.4.1 Concepts du calcul réseau

Le calcul réseau, initié par Cruz [CRU 91] et Le Boudec [LEB 01], est une théorie déterministe des systèmes de bufferisation rencontrée dans les réseaux informatiques. Elle fait appel à l'algèbre Min-Plus⁶, où les opérations sont transformées comme suit : l'addition devient le calcul du minimum et la multiplication devient l'addition. Cette théorie présente la particularité d'aborder tout élément en s'intéressant à son niveau de congestion (c'est à dire la quantité d'opérations en attente de traitement au sein de cet élément) dans ce système. Ce niveau de congestion est formulé en fonction des débits et des arrivées. L'originalité du calcul réseau réside dans la recherche des situations les plus défavorables, ce qui permet d'obtenir des majorants.

Pour cela, différentes représentations et concepts sont introduits, notamment dans [CRU 91] :

- Soit $R(t)$, la fonction non négative représentant le taux instantané d'arrivée du trafic sur un lien à un instant t telle que pour tout $y \geq x$, $\int_x^y R(t)dt$ est la quantité de données transmise sur le lien durant l'intervalle de temps $[x, y]$.

⁶issue de la théorie mathématique des dioides, et en particulier, le dioïde Min-Plus.

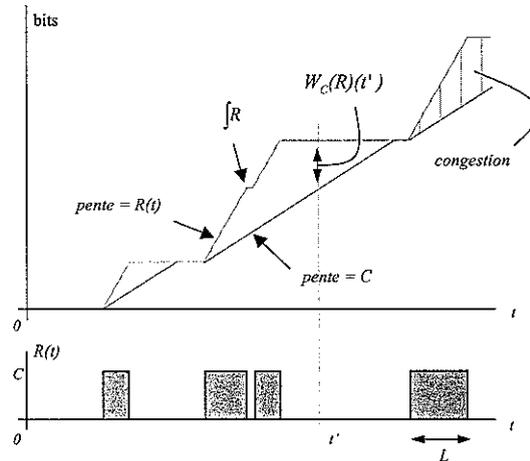


FIG. 4.14 – Arrière de traitement d'un système

- Soit C , la capacité de transmission en bits par seconde du système et L la longueur maximale des paquets en transit.

L'arrière de traitement⁷ (quantité de travail non encore traité à l'instant t) dans un équipement, est alors défini par $W_C(R)(t)$ tel que :

$$W_C(R)(t) = \max_{s \leq t} \left[\int_s^t R(t) dt - C(t - s) \right] \quad (4.1)$$

Ce phénomène est graphiquement mis en évidence à la figure 4.14. Sur cette figure, on retrouve dans un premier temps le chronogramme d'arrivée des données. On voit bien que ces données arrivent à la vitesse du lien et que la durée d'arrivée des paquets est proportionnelle à leur taille. Il est alors à noter qu'à l'instant t' , la courbe de pente C n'a pas encore rejoint la courbe d'arrivée suivant $R(t)$. Cela signifie qu'une partie des données reçues n'a pas encore été traitée. Nous sommes à cet instant en présence de congestion. Nous pouvons donc dire que lorsque l'arrière de traitement est nul, i.e. $W_C(R)(t) = 0$, le système ne présente pas de congestion (inversement pour tout cas où $W_C(R)(t) > 0$).

Le délai d'acheminement des messages sur le réseau dépend de l'arrière de traitement des équipements du réseau. Dès lors, la détermination de l'arrière nécessite la modélisation de l'arrivée des données ainsi que la description du comportement du système.

⁷ou encore *Backlog*.

4.4.2 Modélisation du trafic

A partir de l'étude des communications industrielles présentées dans la section 3.1.1, une hypothèse de respect d'arrivée des données est convenue. Elle repose sur le concept du seau percé [LEB 01, définition 1.3.2], permettant de modéliser les rafales de trafic (Fig. 4.15). La génération du trafic est alors bornée par une fonction affine notée $b(t)$, et la contrainte déterministe retenue indique que le nombre de bits émis par un acteur à un instant t n'est pas supérieur à la valeur $b(t)$.

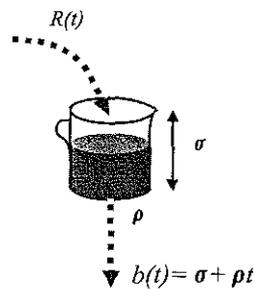


FIG. 4.15 – Le concept du seau percé

L'écriture mathématique du seau percé revient à associer la hauteur maximale du liquide dans le seau, notée σ , à la quantité de données en rafale et le taux de fuite, noté ρ , au taux instantané d'arrivée des données. La formulation mathématique de cette limitation est donnée par l'équation (4.3).

$$b(t) = \sigma + \rho t \quad (4.2)$$

$$R(t) \sim b(t) \iff \int_{t=x}^y R(t) dt < \sigma + \rho(y - x) \quad (4.3)$$

où $R(t)$ représente le taux instantané, σ le volume maximal des rafales et ρ un majorant du taux d'arrivée.

4.4.3 Modélisation du réseau Ethernet commuté

Les commutateurs sont des systèmes complexes qui introduisent différents mécanismes tels que la bufferisation des paquets, la commutation. La figure 4.16 présente les différents composants permettant de modéliser un commutateur utilisé dans le contexte *Ethernet Industriel*.

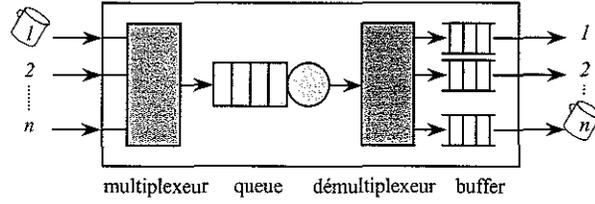


FIG. 4.16 – Modèle de commutation

En considérant chaque composante du commutateur comme un système indépendant, nous pouvons alors exprimer la valeur maximale de son arriéré de traitement selon l'équation 4.1, et ainsi le majorant du délai de traversée, noté \overline{D} . Pour de tels composants, [CRU 91] montre que la traversée d'un composant par un flux de trafic $b_{in}(t) = \sigma_{in} + \rho_{in}t$ est caractérisé par une augmentation du volume des rafales liée au délai. Ce qui peut être traduit dans l'équation 4.2 par :

$$\sigma_{out} = \sigma_{in} + \rho_{in}\overline{D} \quad (4.4)$$

Dès lors, pour un flux I traversant le commutateur, le retard total introduit par le commutateur est donné par la somme des retards de chaque composant élémentaire :

$$\overline{D_{commutateur,I}} = \overline{D_{mux,I}} + \overline{D_{queue,I}} + \overline{D_{mux,I}} + \overline{D_{buffer,I}} \quad (4.5)$$

A partir des équations 4.1, 4.4 et 4.5, [GEO 02] propose de formuler le majorant du délai de la manière suivante :

Le délai appliqué à un paquet entrant par un port I dans un commutateur à M entrées, à capacité interne C , à capacité d'entrée C_{in} , et à capacité de sortie C_{out} , est majoré par $\overline{D_{s,I}}$, tel que :

$$\overline{D_{s,I}} = \alpha [\sigma_1 + \sigma_2 + \dots + \beta\sigma_I + \dots + l\sigma_K + \dots + \sigma_M - \delta] \quad (4.6)$$

En résumé, le majorant du délai est déterminé par la valeur maximale de l'arriéré du commutateur, dont l'expression correspond à la somme des données en rafale moins la capacité de transmission, notée δ . Elle dépend des coefficients β (permettant d'identifier le lien dont on cherche à caractériser le délai de traversée), et de l (étant donné que la valeur du délai d'un flux en entrée du multiplexeur dépend du lien le plus chargé).

4.4.4 Calcul du délai de bout en bout

4.4.4.1 Méthode analytique

L'équation 4.6 fait intervenir le volume des rafales de chaque flux en entrée du commutateur. Or celui ci varie au fur et à mesure de la traversée des commutateurs suivant les délais de traversée (équation 4.4). C'est pourquoi, [GEO 02] reformule l'expression du délai de traversée par la valeur du volume des rafales d'un flux en sortie d'un commutateur. L'ensemble de ces équations est ensuite regroupé dans un système d'équations dont les inconnues sont les volumes des rafales de chaque flux aux différents points du réseau. La résolution de ce système permet alors d'obtenir les valeurs numériques des volumes des rafales de chaque flux tout au long de leur route et finalement, en utilisant la propriété de l'équation 4.4, un majorant du délai de bout en bout peut être calculé.

En résumé, l'algorithme de détermination de la borne maximale du délai de bout en bout est le suivant :

Algorithme

1. Pour chaque entité communicante, identifier les communications et déterminer les paramètres de leur seau percé initial.
2. Sur chaque commutateur, formuler les équations des volumes des rafales de sortie des flux.
3. Définir le système d'équation sous la forme mathématique

$$a_n\sigma_1 + b_n\sigma_2 + \dots + z_n\sigma_m = \delta_n.$$
4. Calculer les volumes des rafales.
5. Déterminer les délais de bout en bout à l'aide de la formule

$$\overline{D}_i = \frac{\sigma_i^h - \sigma_i^0}{\rho_i}$$

où h représente le nombre de commutateurs traversés.

4.4.4.2 Exemple numérique

Pour expliciter d'avantage l'algorithme précédent, étudions le réseau de la figure 4.17. Dans cet exemple, deux processus industriels identiques cohabitent ensemble. Chacun de

ces processus est constitué d'un automate, de deux cartes d'entrées ainsi que d'une station de supervision. Le scénario est alors le suivant : durant le cycle des automates (20 ms), automates, entrées et stations émettent chacun des trames de 64 octets (soit au plus 46 octets de données). A1 émet 4 trames vers PC1 qui lui répond par une trame et E11, E12 émettent chacun deux trames vers A1. De même pour le second processus. Les liens présentent une vitesse de transmission de 10 Mb/s et les commutateurs ont une capacité interne de 1 Gb/s.

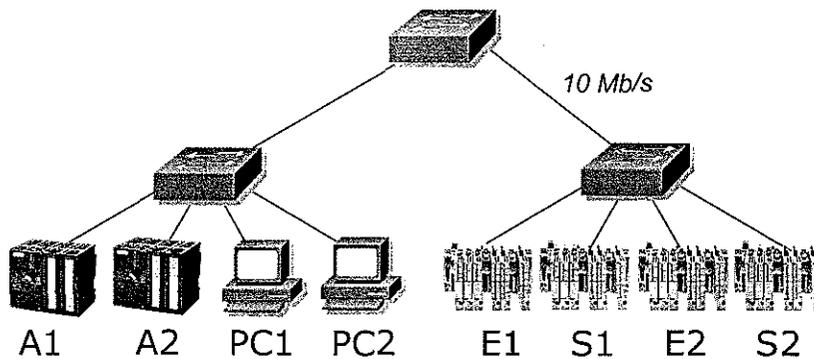


FIG. 4.17 – Exemple de réseau *Ethernet Industriel*

Le déroulement de l'algorithme est alors le suivant.

1. Identification des communications

Nous prenons pour σ la taille maximale des paquets circulant sur le réseau, soit 64 octets. Dans la mesure où les courbes d'arrivées des automates, des stations et des entrées sont identiques entre elles, seulement trois communications sont à identifier.

- communications issues des automates vers les stations, $b_A(t) = 64 + 12800t$;
- communications issues des stations, $b_{PC}(t) = 64 + 3200t$;
- communications issues des entrées, $b_E(t) = 64 + 6400t$;

2. Formulation des nouveaux volumes des rafales

Dans la mesure où les communications issues des entrées présentent la même courbe d'arrivée, la traversée du commutateur de droite peut être caractérisé simplement par :

$$\sigma_E^1 = a_{E_1} [3 * \sigma_E^0 + B_{E_1} \sigma_E^0 - g_{E_1}]$$

Pour le commutateur de tête, l'équation est similaire, si ce n'est que cette fois, il faut prendre en compte le fait que toutes les données arrivent par le même lien.

$$\sigma_E^2 = a_{E_2} [3 * b_{E_2} \sigma_{E_2}^1 + B_{E_2} \sigma_{E_2}^1 - g_{E_2}]$$

Et pour le commutateur de gauche, les volumes de rafales en sortie seront déterminés par :

$$\begin{aligned} \sigma_A^1 &= a_A [2 * \sigma_{PC}^0 + 4 * l * \sigma_E^2 + \sigma_A^0 + B_A \sigma_A^0 - g_A] \\ \sigma_{PC}^1 &= a_{PC} [2 * \sigma_A^0 + 4 * l * \sigma_E^2 + \sigma_{PC}^0 + B_{PC} \sigma_{PC}^0 - g_{PC}] \\ \sigma_E^3 &= a_{E_3} [2 * \sigma_A^0 + 2 * \sigma_{PC}^0 + 3 * b_{E_3} \sigma_E^2 + B_{E_3} \sigma_E^2 - g_{E_3}] \end{aligned}$$

3. Ecriture du système d'équations

En rajoutant l'hypothèse de modélisation du trafic du point 1, on obtient :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1+B_A & -1/a_A & 2 & 0 & 0 & 0 & 4 * l & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1+B_{PC} & -1/a_{PC} & 0 & 0 & 4 * l & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3+B_E & -1/a_{E_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3b_{E_2} + B_{E_2} & -1/a_{E_2} & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 3b_{E_3} + B_{E_3} & -1/a_{E_3} \end{bmatrix} * \begin{bmatrix} \sigma_A^0 \\ \sigma_A^1 \\ \sigma_{PC}^0 \\ \sigma_{PC}^1 \\ \sigma_E^0 \\ \sigma_E^1 \\ \sigma_E^2 \\ \sigma_E^3 \end{bmatrix} = \begin{bmatrix} 64 \\ g_A \\ 64 \\ g_{PC} \\ 64 \\ g_{E_1} \\ g_{E_2} \\ g_{E_3} \end{bmatrix}$$

4. Résolution du système

La résolution des volumes de rafales est dans ce cas relativement simple.

$$\begin{bmatrix} \sigma_A^0 \\ \sigma_A^1 \\ \sigma_{PC}^0 \\ \sigma_{PC}^1 \\ \sigma_E^0 \\ \sigma_E^1 \\ \sigma_E^2 \\ \sigma_E^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1+B_A & -1/a_A & 2 & 0 & 0 & 0 & 4 * l & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1+B_{PC} & -1/a_{PC} & 0 & 0 & 4 * l & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3+B_E & -1/a_{E_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3b_{E_2} + B_{E_2} & -1/a_{E_2} & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 3b_{E_3} + B_{E_3} & -1/a_{E_3} \end{bmatrix}^{-1} * \begin{bmatrix} 64 \\ g_A \\ 64 \\ g_{PC} \\ 64 \\ g_{E_1} \\ g_{E_2} \\ g_{E_3} \end{bmatrix}$$

5. Détermination des délais de bout en bout

Par calcul, on obtient :

$$\overline{D_A} = \frac{\sigma_A^1 - \sigma_A^0}{\rho_A} = 246 \mu s, \quad \overline{D_{PC}} = \frac{\sigma_{PC}^1 - \sigma_{PC}^0}{\rho_{PC}} = 246 \mu s, \quad \overline{D_E} = \frac{\sigma_E^3 - \sigma_E^0}{\rho_E} = 400 \mu s$$

A titre de comparaison, cet exemple a été repris pour la même architecture réseau mais un partitionnement différent, regroupant chaque équipement d'un même processus sur le même commutateur. Dans ce cas, le délai maximum d'acheminement des messages de bout en bout est unique et vaut $246 \mu s$.

Dans le contexte industriel, la méthode de détermination de la borne maximale du délai de bout en bout permet d'évaluer le bon respect de la contrainte temporelle. Ainsi, dans l'exemple précédent, si l'on estime que le délai ne doit pas dépasser $300 \mu s$, la méthode de calcul nous permet de dire que l'architecture de la figure 4.17 n'est pas satisfaisante et qu'il convient de regrouper les équipements d'un même processus.

4.5 Evaluation de performance des solutions

Dans cette section, nous proposons de traiter le problème de conception et d'évaluation de réseaux industriels sur un exemple significatif, représentant l'interconnexion de 90 équipements sur une architecture hiérarchique *Ethernet Industriel*. Nous disposons de commutateurs 16 ports et de liaisons à 10 Mb/s. La contrainte temporelle est fixée par des informations périodiques produites selon un temps de cycle de 1,5 ms. L'ensemble des communications est représenté dans la matrice d'échanges donnée en figure 4.18. Le volume des informations étant inférieur à 46 octets, la matrice est unitaire et chaque '1' de cette matrice représente l'émission d'un paquet de 64 octets toutes les 1,5 ms. La solution représentée par la figure 4.18 est une solution "*non optimisée*" qui consiste à affecter les noeuds aux commutateurs comme suit : les 15 premiers automates de la matrice sont reliés au premier commutateur, les 15 suivants au second, et ainsi de suite.

4.5.1 Evaluation selon le critère de partitionnement

Nous avons utilisé les trois algorithmes de conception : *RSB*, *RSB modifié* et *AG*. Cependant, l'algorithme *RSB* ne permet pas de réaliser une architecture à coût minimal puisqu'il utilise 8 commutateurs au lieu des 6 utiles. Afin de déterminer l'influence d'un mauvais taux de raccordement sur la qualité du partitionnement, nous avons réalisé une série de test sur 160 graphes de 90 noeuds à partitionner en 6 (Fig. 4.19). Dans 100% des cas, l'algorithme génétique fournit de meilleur résultat que l'algorithme *RSB*. En revanche, ce taux est réduit à 60% avec l'utilisation de l'algorithme *RSB modifié*. La modification

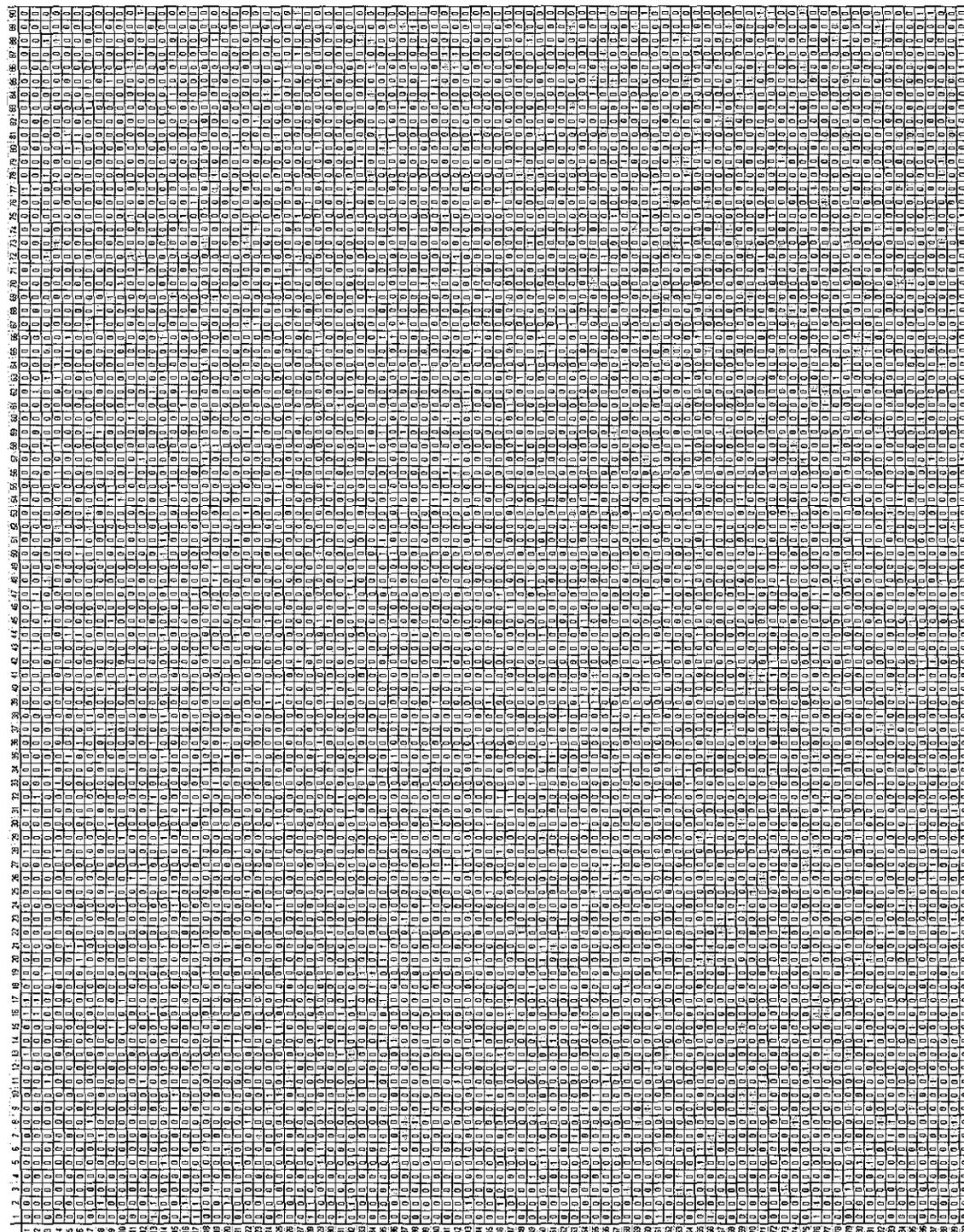


FIG. 4.18 – Scénario d'échanges avec 90 noeuds : solution non optimisée (zones grisées : échanges dans les commutateurs ; cellules grisées : échanges sur le réseau)

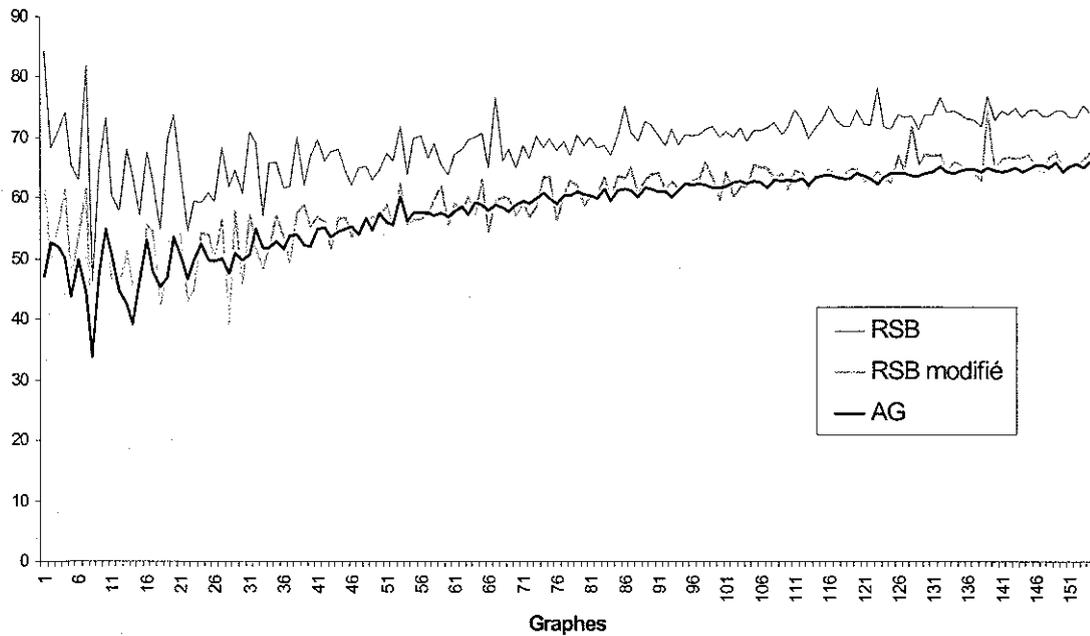


FIG. 4.19 – Apport de la modification de l'algorithme RSB

apporté sur le *RSB* permet donc d'obtenir des solutions plus proches de celles des algorithmes génétiques.

La qualité du partitionnement des solutions obtenues a été évaluée suivant les 3 fonctions coût f_1 , f_2 , et f_3 (cf. 3.4.2.2). Le tableau 4.2 regroupe l'ensemble des valeurs.

| méthode de conception | "non optimisée" | RSB | RSB modifié | A.G. |
|-------------------------------|-----------------|------|-------------|------|
| f_1 (taille coupure+charge) | 0,9 | 0,62 | 0,15 | 0,15 |
| f_2 (taille coupure) | 619 | 327 | 70 | 70 |
| f_3 (fortes interactions) | 0,04 | 0,29 | 0,47 | 0,47 |

TAB. 4.2 – Comparatif des méthodes

Les algorithmes génétiques (indépendamment de la fonction coût) et l'algorithme *RSB modifié* ont convergé vers la même solution. Celle-ci est illustrée sur la figure 4.20 et montre les 6 partitions obtenues ainsi que la minimisation des communications inter-groupe.

Au niveau du partitionnement, l'amélioration apportée par les méthodes de conception est significative. Elle a permis de réduire de 90% la taille de coupure. Bien que dévalorisé

4.5. EVALUATION DE PERFORMANCE DES SOLUTIONS

The figure displays a dense grid of numerical data, organized into 90 columns corresponding to nodes 1 through 90. Each row represents a specific node's performance or exchange scenario. The values are small integers, ranging from 0 to 10, distributed across the grid in a complex, non-uniform pattern. The overall structure suggests a highly optimized solution for a network of 90 nodes.

FIG. 4.20 – Scénario d'échanges avec 90 noeuds : solution optimisée

par son plus grand nombre de partition, l'algorithme *RSB* améliore la solution empirique de 50%.

4.5.2 Evaluation temporelle

Les méthodes développées dans ce chapitre ont été utilisées afin de vérifier le respect de la contrainte temporelle. Un outil de simulation réseau (*ComNet*) a également été utilisé afin de simuler les architectures et comparer les résultats (Fig. 4.21). Ce type d'outil repose sur une modélisation par file d'attente, des équipements, et permet de diversifier les lois d'arrivées des messages (loi de poisson, bernoulli, ...).

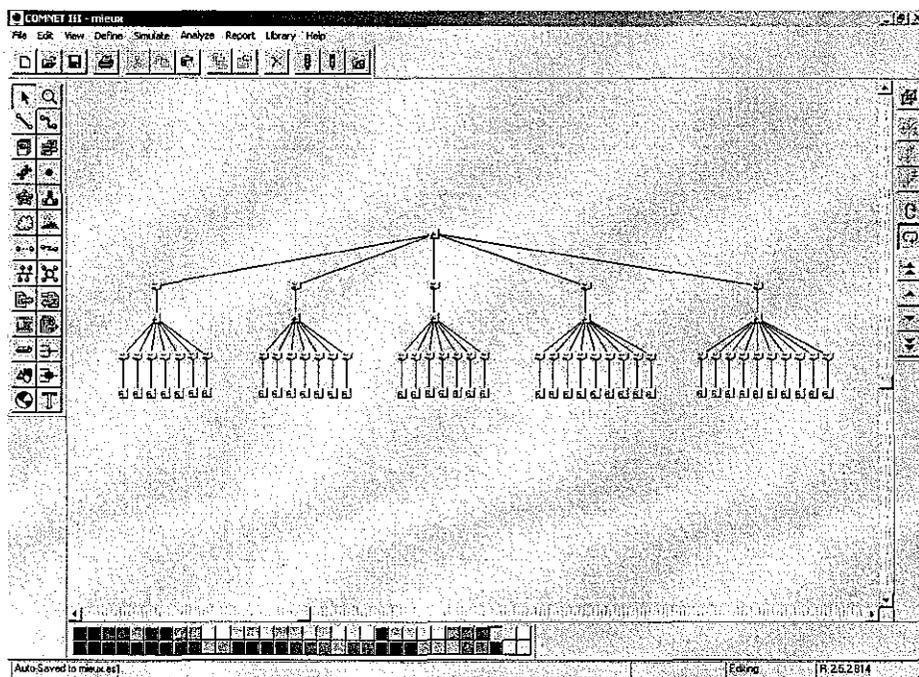


FIG. 4.21 – Outil de simulation de réseaux

Le tableau 4.3 rassemble tous les résultats numériques obtenus. La solution "*non optimisée*" présente, quel que soit l'outil d'évaluation, un délai trop important vis-à-vis de la contrainte temporelle. De même, la solution *RSB* qui affiche un délai inférieur à 1,5 ms (simulation et délai moyen), présente une borne maximale du délai au delà de la contrainte temporelle fixée par l'application. Ces deux solutions ne sont donc pas valables car le délai ne peut être garanti.

| méthode de conception | "empirique" | RSB | RSB modifié | A.G. (f_1) | A.G. (f_2, f_3) |
|--------------------------|-------------|------|-------------|----------------|---------------------|
| délai moyen | 1,70 | 1,07 | 0,67 | 0,67 | 0,67 |
| délai max | 1,96 | 1,83 | 1,32 | 1,31 | 1,32 |
| simulation (délai moyen) | 1,81 | 1,4 | 0,88 | 0,86 | 0,88 |

TAB. 4.3 – Evaluation temporelle des solutions

L'algorithme génétique a été exécuté avec les différentes fonctions coût afin de les comparer. Nous avons alors obtenus la même solution pour $AG(f_2, f_3)$ et RSB modifié (Fig. 4.20). Cette solution permet de garantir la contrainte temporelle. Par ailleurs, l' $AG(f_1)$ cherche une solution optimale suivant le critère de taille de coupure et d'équilibre de charge. Le graphe 4.22 représente le délai maximal en fonction des deux critères.

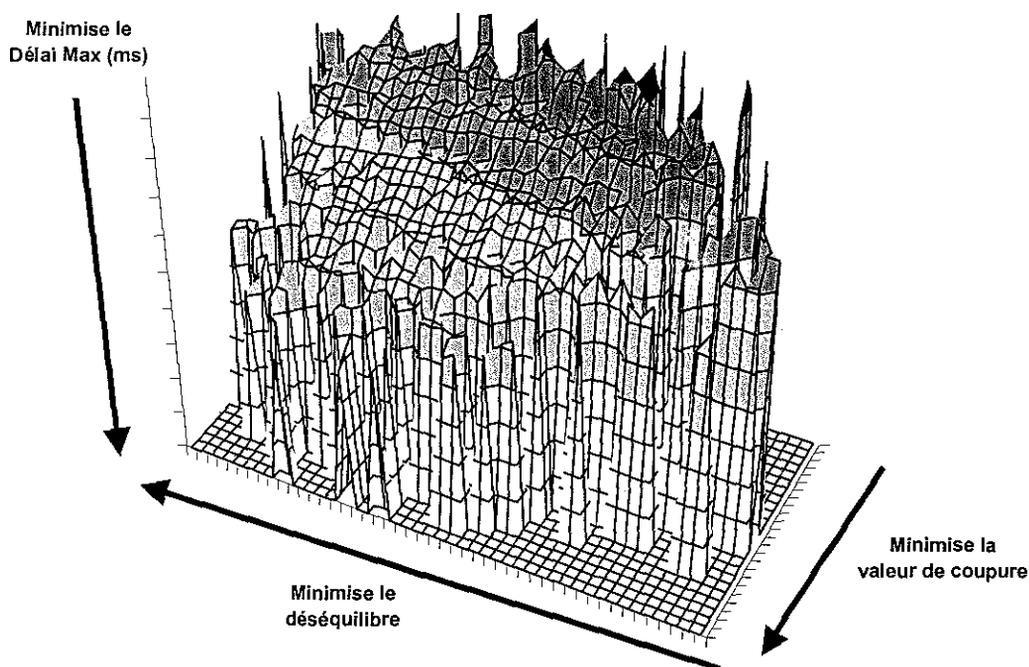


FIG. 4.22 – Influence de la taille de coupure et de l'équilibre sur le délai max

Nous pouvons remarquer que la minimisation de ces deux paramètres, revient à minimiser le délai. Cependant leur impact sur le délai est différent. Pour cela, nous avons tracé les variations du délai en fonction de chaque critère : taille de coupure (Fig. 4.23) et

équilibre (Fig. 4.24). Nous constatons que le gain apporté par la minimisation de l'équilibre est 3 fois moins important que celui obtenu par la minimisation de la taille de coupe. La solution obtenue par l'algorithme $AG(f_1)$ est donc proche de celle de $AG(f_2, f_3)$ et ne présente pas un gain significatif au niveau temporel.

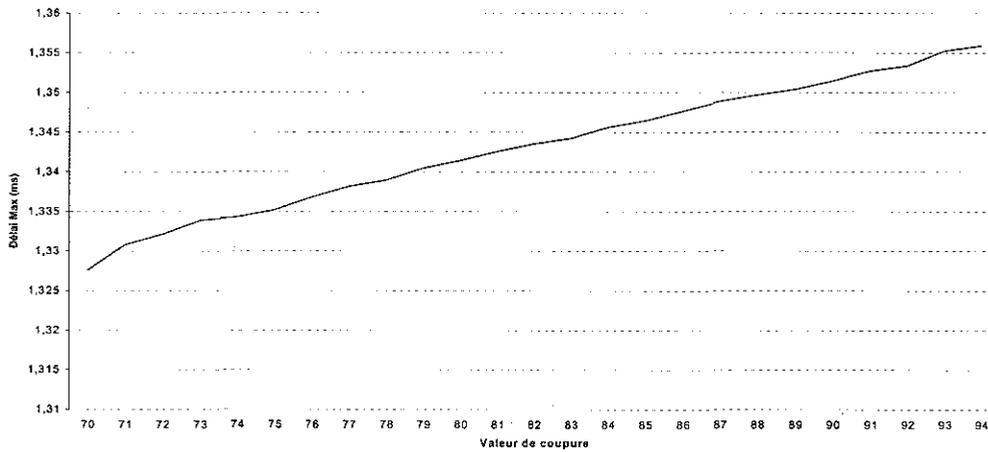


FIG. 4.23 – Influence de la valeur de coupe

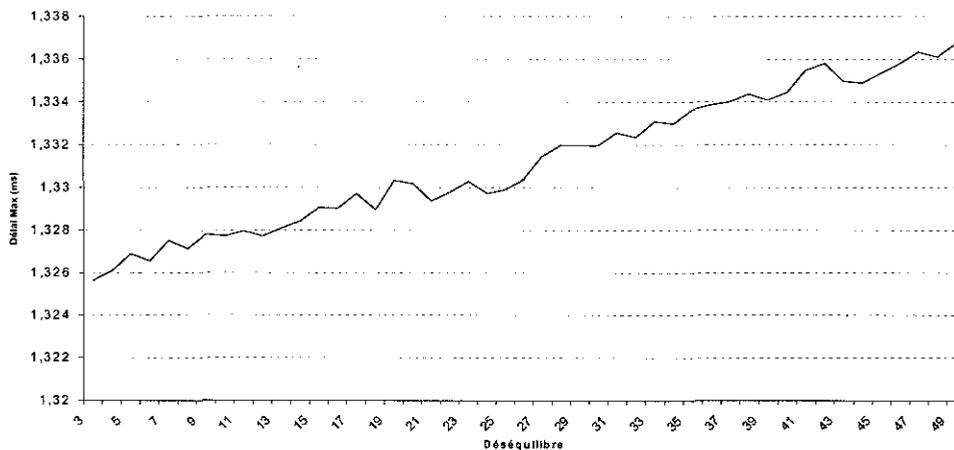


FIG. 4.24 – Influence de l'équilibre

4.6 Conclusion

Dans ce chapitre, nous avons présenté l'ensemble des résultats obtenus par les différents algorithmes de conception ainsi que leur évaluation. Ils ont été obtenus à l'aide d'un outil logiciel de Conception et d'Evaluation de Réseaux Industriels "*Switched Ethernet*" (CERISE), développé au laboratoire.

Dans un premier temps, une comparaison des algorithmes de conception a été réalisée indépendamment de l'application, à l'aide de la fonction objectif (f_2), qui consiste à minimiser la taille de coupure du graphe. Nous avons montré que les solutions obtenues donnent un avantage aux algorithmes génétiques, et ce, quel que soit les caractéristiques du graphe. Cela peut s'expliquer par leur plus grande exploration de l'espace de recherche, mais aussi par leur approche non-réursive. Néanmoins, cette performance nécessite un bon paramétrage des algorithmes génétiques, qui peut s'avérer parfois difficile.

Ensuite, nous avons proposé d'étudier les performances temporelles des solutions à partir de deux méthodes d'évaluation : le calcul du délai moyen d'acheminement et la détermination de la borne maximale de ce délai. Les résultats ont également été validés par un outil de simulation réseau. Ces évaluations ont permis de montrer que :

- Le problème de conception de réseau peut être résolu à l'aide du partitionnement de graphes, même si le délai n'est pas directement pris en compte dans la fonction coût.
- La détermination de la borne maximale du délai de bout en bout permet de vérifier, pour l'architecture optimisée, le respect de la contrainte temporelle.

Conclusion et perspectives

Au même titre que les réseaux informatiques il y a quelques années, les réseaux industriels sont à leur tour en pleine évolution pour répondre aux nouveaux besoins des utilisateurs. Ceux-ci souhaitent s'affranchir de l'hétérogénéité des architectures afin de réduire les coûts d'interconnexion, en termes de câblage, mais aussi d'interfonctionnement. D'autre part, les industriels désirent également bénéficier des nouveaux services liés à l'émergence des technologies Internet. Cette "*mutation*" des réseaux industriels amène alors de nombreux débats dont celui d'*Ethernet Industriel*. En effet, si Ethernet est sans conteste le réseau local informatique par excellence, ses propriétés telles que le non déterminisme, ne lui permettent pas encore de s'imposer dans le monde industriel. Dès lors, différentes études ont été menées pour remédier à ces inconvénients. Cependant, les solutions proposées sont généralement difficiles à mettre en oeuvre par les contraintes matérielles ou logicielles qu'elles imposent. C'est dans ce contexte que nous inscrivons nos travaux.

L'approche que nous avons proposée, est basée sur la conception de réseaux. Elle permet d'optimiser la topologie en tenant compte à la fois des besoins des utilisateurs et de ceux des applications, en particulier la garantie d'un délai d'acheminement des messages. L'originalité de cette approche tient dans le fait qu'elle ne remet en cause ni les concepts de base d'Ethernet, ni le trafic. Nous avons orienté nos travaux vers la théorie du partitionnement de graphe. En effet, la conception d'un réseau peut être vue comme le partitionnement d'un graphe représentant les échanges d'informations sur ce réseau. D'une part, nous montrons que ce partitionnement conduit à une solution similaire à celles qui seraient fournies par une approche classique de type modélisation par file d'attente. D'autre part, notre approche permet non seulement d'intégrer des outils issus de la théorie du partitionnement (méthode spectrale), mais aussi d'effectuer une évaluation temporelle uniquement à partir de la solution optimisée. Notons que, parmi les méthodes de résolution, nous avons retenu les algorithmes spectraux (méthode globale), ainsi que

les algorithmes génétiques (méthode locale) pour leur efficacité reconnue dans le domaine du partitionnement.

Par ailleurs, l'évaluation temporelle des solutions a été effectuée à l'aide de méthodes d'estimation du délai moyen (type files d'attente), mais également de méthodes de détermination de la borne maximale du délai. Nous avons pu alors tester, de façon objective, les solutions par rapport à la contrainte temporelle imposée par l'application industrielle. Ainsi, l'obtention de la borne maximale du délai a permis d'observer que certaines solutions, basées uniquement sur le délai moyen, ne pouvaient être retenues.

L'ensemble des résultats expérimentaux a mis en évidence l'intérêt d'utiliser une approche de type conception de réseau. Ainsi, il apparaît que le fait de rationaliser les échanges, contribue à l'intégration d'Ethernet dans les milieux industriels.

En perspective de ces travaux, Il serait intéressant :

- d'étudier une approche *hybride*, qui combine les algorithmes spectraux et les algorithmes génétiques. Il s'agirait alors d'identifier dans un premier temps une "bonne solution" à l'aide des algorithmes spectraux. Cette solution serait intégrée à la population initiale afin de converger plus rapidement vers une meilleure organisation de réseaux,
- de rechercher des solutions en prenant en compte d'autres objectifs tels que la disponibilité, le coût. Cependant, lorsque les objectifs sont non commensurables, il est difficile de trouver une solution optimisant tous les objectifs. Une approche d'optimisation multi-objectifs peut alors être utilisée. Par exemple, le concept de Pareto-optimalité permet de distinguer l'ensemble des solutions sous-optimales (qui ne sont pas d'une efficacité maximale), des solutions dites optimales au sens de Pareto⁸. Dès lors, le concepteur de réseaux intervient dans le processus de décision en choisissant une solution adaptée à son cahier des charges[COE 99],
- de changer d'objectifs en cours d'exécution de l'algorithme génétique. Ainsi pour obtenir rapidement une bonne solution, l'objectif pourra être basé sur la taille de coupure du graphe d'échanges, comme proposé dans ce mémoire. Puis, la fonction d'évaluation pourra ensuite être établie relativement à des critères temporels. Pour

⁸Une situation est dite optimale au sens de Pareto s'il n'en existe pas d'autre qui permette d'accroître le niveau de satisfaction d'un ou plusieurs objectifs sans faire diminuer celui d'au moins un autre objectif.

cela, nous envisageons d'utiliser les formules de détermination du délai maximal définies dans les travaux de [GEO 02] reposant sur la théorie du calcul réseau,

- de pouvoir prendre en compte simultanément des trafics exigeant des niveaux de qualité de service différents (en termes de délai, de gigue, de fiabilité, ...) afin de proposer une organisation de réseau adaptée répondant au mieux à chacun de ces indicateurs.

Bibliographie

- [ALP 95] ALPERT C. J., KAHNG A. B., "Recent Developments in Netlist Partitioning : A Survey", *Integration : The VLSI Journal*, vol. 19, pp. 1-81, 1995.
- [ALV 00] ALVES M., TOVAR E., FOHLER G., BUTTAZO G., "CIDER - Envisaging a Cots Communication Infrastructure for Evolutionary Dependable Real-Time", *Proceedings of the 12th Euromicro Conference on Real-Time Systems*, Stockolm, pp. 19-22, 2000.
- [ATK 99] ATKINS J. E., BOMAN E. G., HENDRICKSON B., "A Spectral Algorithm for Seriation and the Consecutive Ones Problem", *SIAM Journal on Computing*, vol. 28, n° 1, pp. 297-310, 1999.
- [AUT 01] AUTOMATION C., "Réseaux de Terrain et Ethernet", Recueil des textes de la journée d'information et de débats, 2001.
- [BAR 93] BARNARD S. T., SIMON H. D., "A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems", *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, pp. 711-718, 1993.
- [BAR 94] BARNARD S. T., SIMON H. D., "A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems", *Concurrency : Practice and Experience*, vol. 6, pp. 101-107, 1994.
- [BAS 01] BASSO C., MACQ O., TIGREAT P., "Traçabilité D'environnement de Fabrication sur L'ensemble Des Lignes de Production Dans Le Secteur Pharmaceutique", Euroforum, Paris, juin 2001.
- [BEL 02] BELKACEM O., "Les Bus et Les Réseaux de Terrain En Automatisme Industriel", *InterSections*, Schneider Electric, 2002.
- [BER 87] BERGER M., BOKHARI S. H., "A Partitioning Strategy for Nonuniform Problems on Multiprocessors", *IEEE Transactions on Computers*, vol. 36, n° 5, pp. 570-580, 1987.

- [BOU 86] BOUDENANT J., FEYDEL B., ROLIN P., "Lynx : An Advanced Deterministic CSMA-CD Local Area Network Prototype", *Proceedings Advanced Seminar on Real-Time Local Area Networks*, Bandol, pp. 177-192, April 1986.
- [BRO 01] BROOKS P., "EtherNet/IP : Industrial Protocol White Paper", *ETFA'2001*, Rockwell Automation, Octobre 2001.
- [BUI 92] BUI T. N., JONES C., "Finding Good Approximate Vertex and Edge Partitions is NP-Hard", *Information Processing Letters*, vol. 42(3), pp. 153-159, 1992.
- [BUI 96] BUI T. N., MOON B. R., "Genetic Algorithm and Graph Partitioning", *IEEE Transactions on Computers*, vol. 45, n° 7, pp. 841-855, 1996.
- [CAP 02] CAPONETTO R., LOBELLO L., MIRABELLA O., "Fuzzy Traffic Smoothing : Another Step Towards Statistical Real-Time Communication over Ethernet Networks", *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA-ECRT'02)*, Vienne, pp. 33-36, 2002.
- [CER 01] CERN, "Workshop on Ethernet as Field Network", 2001.
- [CHA 96] CHARON I., GERMA A., HUDRY O., *Méthodes d'Optimisation Combinatoire*, Masson, 1996.
- [CHA 97] CHAN T. F., CIARLET, JR. P., SZETO W. K., "On the Optimality of the Median Cut Spectral Bisection Graph Partitioning Method", *SIAM Journal on Scientific Computing*, vol. 18, n° 3, pp. 943-948, 1997.
- [CHI 94] CHIUEH T., VENKATRAMANI C., "Supporting Real-time Traffic on Ethernet", *Proceedings of IEEE Real-time Systems Symposium*, pp. 282-286, 1994.
- [CIA 94] CIARLET P., ET F. LAMOUR J., "Spectral Partitioning Methods and Greedy Partitioning Methods : A Comparison on Finite Element Graphs", rapport technique n° 94-9, Department of Mathematics, University of California at Los Angeles, April 1994.
- [COE 99] COELLO C., "An Updated Survey of Evolutionary Multiobjective Optimization Techniques : State of the Art and Future Trends", ANGELINE P. J., MICHALEWICZ Z., SCHOENAUER M., YAO X., ZALZALA A., Eds., *Proceedings of the Congress on Evolutionary Computation*, Mayflower hotel, Washington D.C., USA, IEEE Press, pp. 3-13, 1999.

- [COS 95] COSTA D., "Méthodes de Résolution Constructives, Séquentielles et Évolutives Pour Des Problèmes D'affectation Sous Contraintes", PhD thesis, Thèse de l'École Polytechnique Fédérale de Lausanne, No 2132, Lausanne, 1995.
- [CRU 91] CRUZ R., "A Calculus for Network Delay, Part I : Network Elements in Isolation", *IEEE Transactions on Information Theory*, vol. 37, pp. 114-131, Janvier 1991.
- [DEC 01] DECOTIGNIE J.-D., "A Perspective on Ethernet-TCP/IP as a Fieldbus", *International Conference on Fieldbus Systems and their Application (4th FET)*, IFAC, pp. 138-143, Novembre 2001.
- [DUR 01] DURRIEU G., BEL G., BONIOL F., "Evaluation Des Performances D'une Plate-Forme Avionique Modulaire Intégrée", journée "Architecture Logicielle", octobre 2001.
- [ELB 96] ELBAUM R., SIDI M., "Topological Design of Local-Area Networks Using Genetic Algorithms", *IEEE/ACM Transactions on Networking*, vol. 4, n° 5, pp. 766-778, 1996.
- [ERS 93] ERSOY C., PANWAR S., "Topological Design of Interconnected LAN/MAN Networks", *IEEE Journal Selected Areas Communications*, vol. 11, n° 8, pp. 1172-1182, 1993.
- [EUR 01] EUROFORUM, "Ethernet TCP/IP : Vers Une Utilisation Comme Bus de Terrain", 2001.
- [FAR 94a] FARRELL C., KIERONSKA D., SCHULZE M., "Genetic Algorithms for Network Division Problem", *Proc. On IEEE Conference on Evolutionary Computing*, Orlando, USA, pp. 422-427, 1994.
- [FAR 94b] FARRELL C., SIMPSON J., SCHULZE M., KIERONSKA D., "Analyser : A Genetic Algorithm Based Network Divider", *Proc. Of the 1994 Perth AUUG Summer Technical Conference*, Perth, Australia, pp. 43-49, 1994.
- [FET 91] FETTEROLF P., ANANDALINGAM G., "Optimizing Interconnexions of Local Area Networks : An Approach Using Simulated Annealing", *ORSA Journal on Computing*, vol. 3, n° 4, pp. 275-287, 1991.
- [FID 82] FIDUCCIA C. M., MATTHEYSES R. M., "A Linear-Time Heuristic for Improving Network Partitions", *Proceedings of the 19th IEEE Design Automation Conference, IEEE*, pp. 175-181, 1982.

- [FIE 75] FIEDLER M., "A Property of Eigenvectors of Nonnegative Symmetric Matrices and its Application to Graph Theory", *Czech. Math.*, vol. 25(100), pp. 619-633, 1975.
- [FJA 98] FJALLSTROM P., "Algorithms for Graph Partitioning : A Survey", *Linkoping Electronic Atricles in Computer and Information Science*, 3, 1998.
- [GAR 76] GAREY M., JOHNSON D., STOCKMEYER L., "Some Simplified NP-Complete Graph Problems", *Theoretical Computer Science*, vol. 1(3), pp. 237-267, 1976.
- [GAR 79] GAREY M., JOHNSON S., *Computers and Intractability : A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
- [GEO 73] GEORGE A., "Nested Dissection of a Regular Finite Element Mesh", *SIAM Journal of Numerical Analysis*, vol. 10(2), pp. 345-363, 1973.
- [GEO 02] GEORGES J.-P., "Calcul Des Délais de Bout En Bout Dans Les Réseaux Industriels Ethernet Commutés", Master's thesis, Ecole Doctorale IAEM Lorraine DEA Contrôle, Signaux et Communication, 2002.
- [GIL 98] GILBERT J. R., MILLER G. L., TENG S.-H., "Geometric Mesh Partitioning : Implementation and Experiments", *SIAM Journal on Scientific Computing*, vol. 19, n° 6, pp. 2091-2110, 1998.
- [GLO 86] GLOVER F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research*, vol. 13(5), pp. 533-549, 1986.
- [GLO 89] GLOVER F., "Tabu Search - Part I", *ORSA Journal on Computing*, vol. 1, pp. 190-206, 1989.
- [GLO 90] GLOVER F., "Tabu Search - Part II", *ORSA Journal on Computing*, vol. 2, pp. 4-32, 1990.
- [GMB 01] GMBH. A., "AFDX Network Test", *AIM View Newsletter*, vol. 4, 2001.
- [GOL 89] GOLDBERG D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison- Wesley, 1989.
- [GUA 98] GUATTERY S., MILLER G. L., "On the Quality of Spectral Separators", *SIAM Journal on Matrix Analysis and Applications*, vol. 19, n° 3, pp. 701-719, 1998.
- [GUP 96] GUPTA A., "Fast and Effective Algorithms for Graph Partitioning and Sparse Matrix Ordering", *IBM Journal of Research and Development*, vol. 41, n° 1/2, pp. 171-184, 1996.

- [HAG 92] HAGEN L., KAHNG A. B., "New Spectral Methods for Ratio Cut Partitioning and Clustering", *IEEE Transactions on Computer-Aided Design*, vol. 11, pp. 1074-1085, 1992.
- [HEA 95] HEATH M. T., RAGHAVAN P., "A Cartesian Parallel Nested Dissection Algorithm", *SIAM Journal on Matrix Analysis and Applications*, vol. 16, n° 1, pp. 235-253, 1995.
- [HEN 93a] HENDRICKSON B., LELAND R., "Multidimensional Spectral Load Balancing", rapport technique n° SAND93-0074, Sandia National Laboratories, 1993.
- [HEN 93b] HENDRICKSON B., LELAND R., "A Multilevel Algorithm for Partitioning Graphs", rapport technique n° SAND93-1301, Sandia National Laboratories, 1993.
- [HEN 95] HENDRICKSON B., LELAND R., "A Multilevel Algorithm for Partitioning Graphs", KARIN S., Ed., *Proc. Supercomputing '95*, ACM Press, 1995.
- [HIR 99] HIRSCHMANN, "Highlights : Automation and Networking Solutions", rapport technique, Hirschmann GmbH, 1999.
- [HIR 00] HIRSCHMANN NETWORK SYSTEMS, "Distributed Communication Architecture : Industrial Networking Solutions with a Future", rapport technique, Hirschmann, 2000.
- [HOL 75] HOLLAND J., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
- [HOL 98] HOLZRICHTER M., OLIVEIRA S., "New Graph Partitioning Algorithms", The University of Iowa TR-120, 1998.
- [HOP 01] HOPTROFF C., WILLIAMS S., "L'utilisation Des Technologies Web Dans un Atelier de Tôlerie", Euroforum, Paris, juin 2001.
- [IEE 98] IEEE, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", Standard 802.3, 1998.
- [JOH 89] JOHNSON D., ARAGON C. R., MCGEOCH L. A., SCHEVON C., "Optimization by Simulated Annealing : An Experimental Evaluation ; Part I, Graph Partitioning", *Operations Research*, vol. 37, n° 6, pp. 865-892, 1989.
- [JUN 01] JUNIER N., DESVAUX E., VENET F., "Gestion Technique Centralisée sur Ethernet TCP/IP Dans L'agro-Alimentaire", Euroforum, Paris, juin 2001.

- [KAM 99] KAMEN E. W., TORAB P., COOPER K., CUSTODI G., "Design and Analysis of Packet-Switched Networks for Control Systems", *Proceedings of the 38th Conference on Decision and Control*, pp. 4460-4465, 1999.
- [KAR 95] KARYPIS G., KUMAR V., "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs", rapport technique n° TR 95-035, Department of Computer Science, University of Minnesota, Minneapolis, 1995.
- [KAR 97] KARISCH S. E., RENDL F., CLAUSEN J., "Solving Graph Bisection Problems with Semidefinite Programming", rapport technique n° DIKU-TR-97/9, Dept. of Computer Science, University of Copenhagen, Copenhagen, Denmark, July 1997.
- [KAR 98] KARYPIS G., KUMAR V., "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs", *SIAM J. Sci. Comput.*, vol. 20, n° 1, pp. 359-392, 1998.
- [KER 70] KERNIGHAN B. W., LIN S., "An Efficient Heuristic Procedure for Partitioning Graphs", *The Bell System Technical Journal*, vol. 49(2), pp. 291-307, 1970.
- [KHA 91] KHALIL K. M., SPENCER P. A., "A Systematic Approach For Planning, Tuning And Upgrading Local Area Networks", *GLOBECOM '91*, pp. 658-663, 1991.
- [KIR 83] KIRKPATRICK S., GELLAT C., VECCHI M., "Optimization by Simulated Annealing", Science, 1983.
- [KRO 01a] KROMMENACKER N., DIVOUX T., RONDEAU E., "Configuration of Network Architectures for Co-Operative Systems by Genetic Algorithm", *Workshop Representation and Operators for NETWORK Problem in the Genetic and Evolutionary Computation Conference (GECCO'2001)*, San Francisco, pp. 272-275, july 2001.
- [KRO 01b] KROMMENACKER N., RONDEAU E., DIVOUX T., "Study of Algorithms to Define the Cabling Plan of Switched Ethernet for Real-Time Applications", *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2001)*, Antibes, pp. 223-230, Oct. 2001.
- [KRO 02a] KROMMENACKER N., DIVOUX T., RONDEAU E., "Genetic Algorithms for Industrial Ethernet Network Design", *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*, pp. 149-155, Août 2002.
- [KRO 02b] KROMMENACKER N., DIVOUX T., RONDEAU E., "Using Genetic Algorithms to Design Switched Ethernet Industrial Networks", *International Symposium on Industrial Electronics (IEEE-ISIE'2002)*, L'Aquila (Italie), pp. 152-157, july 2002.

- [KRO 02c] KROMMENACKER N., GEORGES J., RONDEAU E., DIVOUX T., "Designing, Modelling and Evaluating Switched Ethernet Networks in Factory Communications Systems", *1st International Workshop on Real-Time in LANs in the Internet Age (RTLIA-ECTRS'02)*, Vienne, pp. 55-58, Juin 2002.
- [KUM 90] KUMAR C., CHANDRASEKHARAN M., "Grouping Efficacy : A Quantitative Criterion for Goodness of Block Diagonal Forms of Binary Matrices in Group Technology", *Journal of Production Research*, vol. 28(2), pp. 233-243, 1990.
- [KWE 99] KWEON S.-K., SHIN K., ZHENG Q., "Statistical Real-Time Communication over Ethernet for Manufacturing Automation Systems", *Proceedings of Fifth IEEE Real-Time Technology and Applications Symposium*, 1999.
- [LAN 93] LANN G. L., RIVIÈRE N., "Real-Time Communications over Broadcast Networks : The CSMA-DCR and the DOD-CSMA-CD Protocols", rapport technique n° Technical Report 1863, INRIA, 1993.
- [LAS 93] LASZEWSKI G. V., "A Collection of Graph Partitioning Algorithms", unpublished Technical Report, 1993.
- [LEB 01] LÉBOUDEC J. Y., THIRIAN P., *Network Calculus*, vol. 250, Springer Verlag, Juillet 2001.
- [MAI 94] MAINI H., MEHROTRA K., MOHAN C. K., RANKA S., "Genetic Algorithms for Graph Partitioning and Incremental Graph Partitioning", *Proc. Supercomputing'94*, pp. 449-457, 1994.
- [MES 02] MESSERCHMIDT R., "Real-Time with Ethernet", *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA-ECRT'02)*, Vienne, pp. 21-24, 2002.
- [MET 53] METROPOLIS N., ROSENBLUTH A., ROSENBLUTH M., TELLER A., TELLER E., "Equation of State Calculations by Fast Computing Machines", *Journal of Chemical Physics*, vol. 21, pp. 1087-1092, 1953.
- [MET 76] METCALFE R. M., BOGGS D. R., "Ethernet : Distributed Packet Switching for Local Computer Networks", *Communications of the ACM*, vol. 19, pp. 395-404, July 1976.
- [MOL 02] MOLDOVANSKY A., "Utilization of Modern Switching Technology in Ethernet/IP Networks", *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA-ECRT'02)*, Vienne, pp. 25-28, 2002, Rockwell Automation.

- [NUS 87] NUSSBAUMER H., *Conception Des Réseaux - Réseau - Transport*, Téléinformatique Vol. II, Presses Polytechniques et Universitaires Romandes, ISBN : 2-88074-145-9, 1987.
- [NUS 97] NUSEKABEL M., "Switched Network Partitioning Using Tabu Search", Master's thesis, University of South Florida, 1997.
- [PAP 82] PAPADIMITRIOU C. H., STEIGLITZ K., *Combinatorial Optimization : Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [PAT 98] PATRA A., KIM D., "Efficient Mesh Partitioning for Adaptive Hp Finite Element Methods", *International Conference on Domain Decomposition Methods*, 1998.
- [POT 90] POTHEN A., SIMON H. D., LIOU K., "Partitioning Sparse Matrices with Eigenvectors of Graphs", *SIAM Journal on Matrix Analysis and Applications*, vol. 11(3), pp. 430-452, 1990.
- [POT 94] POTHEN A., "An Analysis of Spectral Graph Partitioning Via Quadratic Assignment Problems", American Mathematical Society, *Domain Decomposition Methods in Science and Engineering*, vol. 180, pp. 105-110, 1994.
- [POT 97] POTHEN A., "Graph Partitioning Algorithms with Applications to Scientific Computing", *Parallel Numerical Algorithms*, Kluwer Academic Press, pp. 323-368, 1997.
- [PRI 95] PRITTY D. W., MALONE J. R., SMEED D. N., BANERJEE S. K., LAWRIE N. L., "A Real-Time Upgrade for EthernetBased Factory Networking", *Proc. of IECON*, pp. 1631-1637, 1995.
- [ROC 98] ROCKWELL, "Ethernet for Industrial Control : An Ethernet White Paper", rapport technique, Rockwell International Corporation, 1998.
- [ROL 96] ROLLAND E., PIRKUL H., GLOVER F., "A Tabu Search for Graph Partitioning", *Annals of Operations Research, Metaheuristics in Combinatorial Optimization*, vol. 63, pp. 209-232, 1996.
- [RÜP 99] RÜPING S., VONNAHME E., JASPERNEITE J., "Analysis of Switched Ethernet Networks with different Topologies used in Automation Systems", *Fieldbus Conference (Fet'99)*, Magdebourg, Allemagne, Springer, pp. 351-358, 1999.
- [SCH 77] SCHWARTZ M., *Computer-Communication Network Design and Analysis*, Prentice-Hall, INC., Englewood Cliffs, NY 07632, 1977.

- [SCH 02a] SCHNEIDER, "Modbus on TCP/IP", rapport technique, Schneider Electric Industries S.A., 2002.
- [SCH 02b] SCHNEIDER, "Transparent Factory : Internet-Based Technologies for the Factory Floor", rapport technique, Schneider Electric Industries S.A., April 2002.
- [SIE 99] SIEMENS, "SIMATIC NET Whipe Paper Industrial Ethernet", rapport technique, SIEMENS AG, 1999.
- [SIM 91] SIMON H. D., "Partitioning of Unstructured Problems for Parallel Processing", *Computing Systems in Engineering*, vol. 2, pp. 135-148, 1991.
- [SON 94] SONGERWALA M., "Efficient Solutions to the Network Division Problem", Honours Thesis, Department of Computer Science, Curtin University of Technology, 1994.
- [SON 01] SONG Y., "Time Constrained Communication over Switched Ethernet", *International Conference on Fieldbus Systems and their Application (4th FET)*, IFAC, pp. 152-169, Novembre 2001.
- [THO 00] THOMAS G., "Introduction to Switch Technology - Improving the Performance of Ethernet Networks", *Control Network*, vol. 1, n° 9, 2000.
- [TOR 00] TORAB P., "Performance Analysis of Packet Switched Networks with Tree Topology", PhD thesis, School of Electrical and Computer Engineering, 2000.
- [TUR 96] TURIEL J., MARINERO J., GONZÁLEZ J., "CSMA/PDCR : A Random Access Protocol Without Priority Inversion", *Annual Conference of the IEEE Industrial Electronics Society (IECON'96)*, pp. 910-915, 1996.
- [YOU 00] YOUSSEF H., SAIT S., KHAN S., "Fuzzy Simulated Evolution Algorithm for Topology Design of Campus Networks", *Proc. Of the 2000 Congress on Evolutionary Computation*, pp. 180-187, 2000.
- [ZAM 90] ZAMBRE R., "Design Considerations for Extended Local Area Networks", *International Conference on Computer Communication, ICC '90*, pp. 432-442, 1990.

Publications

Conférences Internationales avec Comité de Lecture

N. Krommenacker, E. Rondeau and T. Divoux. "Genetic Algorithms for Industrial Ethernet Network Design." *4th IEEE International Workshop on Factory Communication Systems (WFCS'2002)*, Mälardalen University, Västerås, Sweden, p. 149-155, august 2002.

N. Krommenacker, T. Divoux and E. Rondeau. "Using Genetic Algorithms to Design Switched Ethernet Industrial Networks." *International Symposium on Industrial Electronics (IEEE-ISIE'2002)*, L'Aquila (Italie), p. 152-157, july 2002.

N. Krommenacker, J.P. Georges, E. Rondeau and T. Divoux. "Designing, Modelling and Evaluating of Switched Ethernet Network in Factory Communication Systems." *14th Euromicro Conference on Real-Time Systems (Workshop on Real-Time LANs in the Internet Age - RTLIA'2002)*, University of Vienna, Austria, p. 55-58, june 2002.

N. Krommenacker, E. Rondeau and T. Divoux. "Study of Algorithms to Define the Cabling Plan of Switched Ethernet for Real-Time Applications." *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2001)*, Antibes, p. 223-230, oct. 2001.

N. Krommenacker, T. Divoux and E. Rondeau. "Configuration of Network Architectures for Co-operative Systems by Genetic Algorithm." *Workshop Representation and Operators for NETWORK problem in the Genetic and Evolutionary Computation Conference (GECCO'2001)*, San Francisco, p. 272-275, july 2001.

N. Krommenacker, F. Lepage. "Remote Operations in Virtual Company : Video Transmission Optimization." *IFAC/IFIP/IEEE Second Conference on Management and Control of Production and Logistics (MCPL'2000)*, Grenoble, july 2000.

Conférences Nationales avec Comité de Lecture

N. Krommenacker, E. Rondeau et T. Divoux. "Comparaison d'heuristiques d'Aide à la Conception d'Architecture Réseau." *Troisièmes Rencontres Francophones sur les aspects Algorithmiques des Télécommunications (Algotel'2001)*, p. 89-94, mai 2001.

N. Krommenacker, E. Rondeau. "Organisation des Architectures de Communications dans les Environnements de Travail Coopératif." *Journées Doctorales d'Automatique (JDA '99)*, p. 357-360, Sept. 1999.

Monsieur KROMMENACKER Nicolas

DOCTORAT de l'UNIVERSITE HENRI POINCARÉ, NANCY 1
en INFORMATIQUE INDUSTRIELLE

VU, APPROUVÉ ET PERMIS D'IMPRIMER

n° 773

Nancy, le 9 Janvier 2003

Le Président de l'Université



Cl. BURLET



Heuristiques de conception de topologies réseaux : application aux réseaux locaux industriels

RÉSUMÉ. *L'avènement d'Ethernet dans les environnements industriels bouleverse le schéma classique des réseaux de terrain basés sur des méthodes déterministes d'accès au médium. De nombreux travaux de recherche portent sur l'amélioration des capacités d'Ethernet pour qu'il offre les performances déterministes requises par les applications critiques. Les solutions proposées sont généralement technologiques et nécessitent de remettre en cause soit les concepts de base d'Ethernet, soit le trafic applicatif. Les propositions avancées dans ces travaux, exploitent les techniques de partitionnement afin de concevoir des architectures topologiques optimisées en terme de délais d'acheminement des messages et de charges dans les nœuds du réseau. La qualité de service recherchée intègre également la disponibilité du système de communication et les architectures proposées sont tolérantes aux fautes par la mise en place de redondances pour pallier la défaillance de liens ou de nœuds du réseau. Deux heuristiques de partitionnement ont été développées et adaptées à ce problème de conception d'architectures de communication. Les résultats obtenus montrent l'intérêt de la démarche proposée.*

MOTS-CLÉS : *Ethernet Industriel, Partitionnement de Graphes, Algorithmes Spectraux, Algorithmes Génétiques.*

ABSTRACT. *Industrial Ethernet is a controversial discussion topic mainly due to its non-determinism. Many researches focus on the improvement of Ethernet's capabilities to deliver the deterministic performance required by critical applications. Generally, that is obtained by modifying the Ethernet protocol or the traffic generated by the applications. Furthermore, the industrial context imposes a links redundancy in order to provide reliable architectures. That is why we propose to improve the switched Ethernet topology performances by optimizing the network organization at the level of the physical layer. Some graph partitioning techniques have been used in order to design optimized topologies which guarantee bounded maximum end to end delays. Two heuristics have been developed and adapted to this network design problem. The obtained results show the interest of the proposed method.*

KEYWORDS : *Industrial Ethernet, Graph Partitioning, Spectral Algorithms, Genetic Algorithms.*
