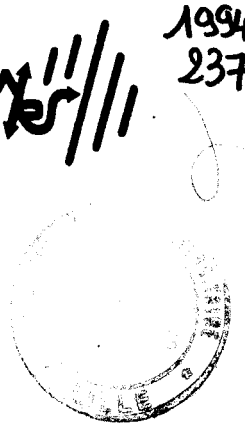
NUMERO D'ORDRE : 1290 ANNEE : 1994



THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le titre de

DOCTEUR en INFORMATIQUE

par

Aklesso BEKLEY

Contribution à l'étude des réseaux d'interconnexion des machines parallèles - Utilisation des Hyperfréquences -

Thèse soutenue le 28 Octobre 1994, devant la commission d'examen :

Président :	M. MERIAUX	LIFL, EUDIL
Rapporteurs :	D. ETIEMBLE	LRI, Université Paris Sud
	D. LITAIZE	IRIT, Université Paul Sabatier
Directeur de Thèse	B. TOURSEL	LIFL, EUDIL
Examinateurs :	G. GONCALVES	LIFL, Université d'Artois
	T. HSU	LIFL, Université d'Artois
	P. A. ROLLAND	IEMN, EUDIL

UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE
U.F.R. d'I.E.E.A. Bât M3. 59655 Villeneuve d'Ascq CEDEX
Tél. 20.43.47.24 Fax. 20.43.65.66

UNIVERSITE DES SCIENCES
ET TECHNOLOGIES DE LILLE

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M. H. LEFEBVRE, M. PARREAU

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PARREAU, J. LOMBARD, M. MIGEON, J. CORTOIS, A.DUBRULLE

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

M. P. LOUIS

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CHAMLEY Hervé	Géotechnique
M. CONSTANT Eugène	Electronique
M. ESCAIG Bertrand	Physique du solide
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. LABLACHE COMBIER Alain	Chimie
M. LOMBARD Jacques	Sociologie
M. MACKE Bruno	Physique moléculaire et rayonnements atmosphériques

M. MIGEON Michel	EUDIL
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. TRIDOT Gabriel	Chimie appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre	Astronomie
M. BIAYS Pierre	Géographie
M. BILLARD Jean	Physique du Solide
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean Pierre	Chimie-Physique
M. BOSCQ Denis	Probabilités
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie Végétale
M. BRASSELET Jean Paul	Géométrie et topologie
M. BREZINSKI Claude	Analyse numérique
M. BRIDOUX Michel	Chimie Physique
M. BRUYELLE Pierre	Géographie
M. CARREZ Christian	Informatique
M. CELET Paul	Géologie générale
M. COEURE Gérard	Analyse
M. CORDONNIER Vincent	Informatique
M. CROSNIER Yves	Electronique
Mme DACHARRY Monique	Géographie
M. DAUCHET Max	Informatique
M. DEBOURSE Jean Pierre	Gestion des entreprises
M. DEBRABANT Pierre	Géologie appliquée
M. DECLERCQ Roger	Sciences de gestion
M. DEGAUQUE Pierre	Electronique
M. DESCHEPPER Joseph	Sciences de gestion
Mme DESSAUX Odile	Spectroscopie de la réactivité chimique
M. DHAINAUT André	Biologie animale
Mme DHAINAUT Nicole	Biologie animale
M. DJAFARI Rouhani	Physique
M. DORMARD Serge	Sciences Economiques
M. DOUKHAN Jean Claude	Physique du solide
M. DUBRULLE Alain	Spectroscopie hertzienne
M. DUPOUY Jean Paul	Biologie
M. DYMENT Arthur	Mécanique
M. FOCT Jacques Jacques	Métallurgie
M. FOUQUART Yves	Optique atmosphérique
M. FOURNET Bernard	Biochimie structurale
M. FRONTIER Serge	Ecologie numérique
M. GLORIEUX Pierre	Physique moléculaire et rayonnements atmosphériques
M. GOSSELIN Gabriel	Sociologie
M. GOUDMAND Pierre	Chimie-Physique
M. GRANELLE Jean Jacques	Sciences Economiques
M. GRUSON Laurent	Algèbre
M. GUILBAULT Pierre	Physiologie animale
M. GUILLAUME Jean	Microbiologie
M. HECTOR Joseph	Géométrie
M. HENRY Jean Pierre	Génie mécanique
M. HERMAN Maurice	Physique spatiale
M. LACOSTE Louis	Biologie Végétale
M. LANGRAND Claude	Probabilités et statistiques

M. LATTEUX Michel	Informatique
M. LAVEINE Jean Pierre	Paléontologie
Mme LECLERCQ Ginette	Catalyse
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique atomique et moléculaire
M. LEROY Jean Marie	Spectrochimie
M. LHENAFF René	Géographie
M. LHOMME Jean	Chimie organique biologique
M. LOUAGE Françis	Electronique
M. LOUCHEUX Claude	Chimie-Physique
M. LUCQUIN Michel	Chimie physique
M. MAILLET Pierre	Sciences Economiques
M. MAROUF Nadir	Sociologie
M. MICHEAU Pierre	Mécanique des fluides
M. PAQUET Jacques	Géologie générale
M. PASZKOWSKI Stéfan	Mathématiques
M. PETIT Francis	Chimie organique
M. PORCHET Maurice	Biologie animale
M. POUZET Pierre	Modélisation - calcul scientifique
M. POVY Lucien	Automatique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. RAMAN Jean Pierre	Sciences de gestion
M. SALMER Georges	Electronique
M. SCHAMPS Joël	Spectroscopie moléculaire
Mme SCHWARZBACH Yvette	Géométrie
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
M. SLIWA Henri	Chimie organique
M. SOMME Jean	Géographie
Melle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. THIEBAULT François	Sciences de la Terre
M. THOMAS Jean Claude	Géométrie - Topologie
M. THUMERELLE Pierre	Démographie - Géographie humaine
M. TILLIEU Jacques	Physique théorique
M. TOULOTTE Jean Marc	Automatique
M. TREANTON Jean René	Sociologie du travail
M. TURRELL Georges	Spectrochimie infrarouge et raman
M. VANEECLOO Nicolas	Sciences Economiques
M. VAST Pierre	Chimie inorganique
M. VERBERT André	Biochimie
M. VERNET Philippe	Génétique
M. VIDAL Pierre	Automatique
M. WALLART Françis	Spectrochimie infrarouge et raman
M. WEINSTEIN Olivier	Analyse économique de la recherche et développement
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ABRAHAM Francis	
M. ALLAMANDO Etienne	Composants électroniques
M. ANDRIES Jean Claude	Biologie des organismes
M. ANTOINE Philippe	Analyse
M. BALL Steven	Génétique
M. BART André	Biologie animale
M. BASSERY Louis	Génie des procédés et réactions chimiques
Mme BATTIAU Yvonne	Géographie
M. BAUSIERE Robert	Systèmes électroniques
M. BEGUIN Paul	Mécanique
M. BELLET Jean	Physique atomique et moléculaire
M. BERNAGE Pascal	Physique atomique, moléculaire et du rayonnement
M. BERTHOUD Arnaud	Sciences Economiques
M. BERTRAND Hugues	Sciences Economiques
M. BERZIN Robert	Analyse
M. BISKUPSKI Gérard	Physique de l'état condensé et cristallographie
M. BKOUCHE Rudolphe	Algèbre
M. BODARD Marcel	Biologie végétale
M. BOHIN Jean Pierre	Biochimie métabolique et cellulaire
M. BOIS Pierre	Mécanique
M. BOISSIER Daniel	Génie civil
M. BOIVIN Jean Claude	Spectrochimie
M. BOUCHER Daniel	Physique
M. BOUQUELET Stéphane	Biologie appliquée aux enzymes
M. BOUQUIN Henri	Gestion
M. BROCARD Jacques	Chimie
Mme BROUSMICHE Claudine	Paléontologie
M. BUISINE Daniel	Mécanique
M. CAPURON Alfred	Biologie animale
M. CARRE François	Géographie humaine
M. CATTEAU Jean Pierre	Chimie organique
M. CAYATTE Jean Louis	Sciences Economiques
M. CHAPOTON Alain	Electronique
M. CHARET Pierre	Biochimie structurale
M. CHIVE Maurice	Composants électroniques optiques
M. COMYN Gérard	Informatique théorique
Mme CONSTANT Monique	Composants électroniques et optiques
M. COQUERY Jean Marie	Psychophysiologie
M. CORIAT Benjamin	Sciences Economiques
Mme CORSIN Paule	Paléontologie
M. CORTOIS Jean	Physique nucléaire et corpusculaire
M. COUTURIER Daniel	Chimie organique
M. CRAMPON Norbert	Tectolique géodynamique
M. CURGY Jean Jacques	Biologie
M. DANGOISSE Didier	Physique théorique
M. DE PARIS Jean Claude	Analyse
M. DECOSTER Didier	Composants électroniques et optiques
M. DEJAEGER Roger	Electrochimie et Cinétique
M. DELAHAYE Jean Paul	Informatique
M. DELORME Pierre	Physiologie animale
M. DELORME Robert	Sciences Economiques
M. DEMUNTER Paul	Sociologie
Mme DEMUYNCK Claire	Physique atomique, moléculaire et du rayonnement
M. DENEL Jacques	Informatique
M. DEPREZ Gilbert	Physique du solide - christallographie

M. DERIEUX Jean Claude	Microbiologie
M. DERYCKE Alain	Informatique
M. DESCAMPS Marc	Physique de l'état condensé et cristallographie
M. DEVRAINNE Pierre	Chimie minérale
M. DEWAILLY Jean Michel	Géographie humaine
M. DHAMELINCOURT Paul	Chimie physique
M. DI PERSIO Jean	Physique de l'état condensé et cristallographie
M. DUBAR Claude	Sociologie démographique
M. DUBOIS Henri	Spectroscopie hertzienne
M. DUBOIS Jean Jacques	Géographie
M. DUBUS Jean Paul	Spectrométrie des solides
M. DUPONT Christophe	Vie de la firme
M. DUTHOIT Bruno	Génie civil
Mme DUVAL Anne	Algèbre
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FARVACQUE Jean Louis	Physique de l'état condensé et cristallographie
M. FAUQUEMBERGUE Renaud	Composants électroniques
M. FELIX Yves	Mathématiques
M. FERRIERE Jacky	Tectonique - Géodynamique
M. FISCHER Jean Claude	Chimie organique, minérale et analytique
M. FONTAINE Hubert	Dynamique des cristaux
M. FORSE Michel	Sociologie
M. GADREY Jean	Sciences économiques
M. GAMBLIN André	Géographie urbaine, industrielle et démographie
M. GOBLOT Rémi	Algèbre
M. GOURIEROUX Christian	Probabilités et statistiques
M. GREGORY Pierre	I.A.E.
M. GREMY Jean Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLOT Jean	Chimie organique
M. GUELTON Michel	Chimie physique
M. GUICHAOUA André	Sociologie
M. HAIMAN Georges	Modélisation,calcul scientifique, statistiques
M. HOUDART René	Physique atomique
M. HUEBSCHMANN Johannes	Mathématiques
M. HUTTNER Marc	Algèbre
M. ISAERT Noël	Physique de l'état condensé et cristallographie
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et statistiques
M. JEAN Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme
M. JOURNEL Gérard	Spectroscopie hertzienne
M. KOENIG Gérard	Sciences de gestion
M. KOSTRUBIEC Benjamin	Géographie
M. KREMBEL Jean	Biochimie
Mme KRIFA Hadjila	Sciences Economiques
M. LANGEVIN Michel	Algèbre
M. LASSALLE Bernard	Embryologie et biologie de la différenciation
M. LE MEHAUTE Alain	Modélisation,calcul scientifique,statistiques
M. LEBFEVRE Yannic	Physique atomique,moléculaire et du rayonnement
M. LECLERCQ Lucien	Chimie physique
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Marc	Composants électroniques et optiques
M. LEFEVRE Christian	Pétrologie
Melle LEGRAND Denise	Algèbre
M. LEGRAND Michel	Astronomie - Météorologie
M. LEGRAND Pierre	Chimie
Mme LEGRAND Solange	Algèbre
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne

M. LE MAROIS Henri	Vie de la firme
M. LEMOINE Yves	Biologie et physiologie végétales
M. LESCURE François	Algèbre
M. LESENNE Jacques	Systèmes électroniques
M. LOCQUENEUX Robert	Physique théorique
Mme LOPES Maria	Mathématiques
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU François	Sciences économiques
M. MAHIEU Jean Marie	Optique - Physique atomique
M. MAIZIERES Christian	Automatique
M. MANSY Jean Louis	Géologie
M. MAURISSON Patrick	Sciences Economiques
M. MERIAUX Michel	EUDIL
M. MERLIN Jean Claude	Chimie
M. MESMACQUE Gérard	Génie mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MOCHE Raymond	Modélisation,calcul scientifique,statistiques
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie organique
M. MORE Marcel	Physique de l'état condensé et cristallographie
M. MORTREUX André	Chimie organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
M. NIAY Pierre	Physique atomique,moléculaire et du rayonnement
M. NICOLE Jacques	Spectrochimie
M. NOTELET Francis	Systèmes électroniques
M. PALAVIT Gérard	Génie chimique
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. PERTUZON Emile	Physiologie animale
M. PETIT Daniel	Biologie des populations et écosystèmes
M. PLIHON Dominique	Sciences Economiques
M. PONSOLLE Louis	Chimie physique
M. POSTAIRE Jack	Informatique industrielle
M. RAMBOUR Serge	Biologie
M. RENARD Jean Pierre	Géographie humaine
M. RENARD Philippe	Sciences de gestion
M. RICHARD Alain	Biologie animale
M. RIETSCH François	Physique des polymères
M. ROBINET Jean Claude	EUDIL
M. ROGALSKI Marc	Analyse
M. ROLLAND Paul	Composants électroniques et optiques
M. ROLLET Philippe	Sciences Economiques
Mme ROUSSEL Isabelle	Géographie physique
M. ROUSSIGNOL Michel	Modélisation,calcul scientifique,statistiques
M. ROY Jean Claude	Psychophysiologie
M. SALERNO Françis	Sciences de gestion
M. SANCHOLLE Michel	Biologie et physiologie végétales
Mme SANDIG Anna Margarette	
M. SAWERYSYN Jean Pierre	Chimie physique
M. STAROSWIECKI Marcel	Informatique
M. STEEN Jean Pierre	Informatique
Mme STELLMACHER Irène	Astronomie - Météorologie
M. STERBOUL François	Informatique
M. TAILLIEZ Roger	Génie alimentaire
M. TANRE Daniel	Géométrie - Topologie
M. THERY Pierre	Systèmes électroniques
Mme TJOTTA Jacqueline	Mathématiques
M. TOURSEL Bernard	Informatique
M. TREANTON Jean René	Sociologie du travail

M. TURREL Georges	Spectrochimie infrarouge et raman
M. VANDIJK Hendrik	
Mme VAN ISEGHEM Jeanine	Modélisation,calcul scientifique,statistiques
M. VANDORPE Bernard	Chimie minérale
M. VASSEUR Christian	Automatique
M. VASSEUR Jacques	Biologie
Mme VIANO Marie Claude	
M. WACRENIER Jean Marie	Electronique
M. WARTEL Michel	Chimie inorganique
M. WATERLOT Michel	géologie générale
M. WEICHERT Dieter	Génie mécanique
M. WERNER Georges	Informatique théorique
M. WIGNACOURT Jean Pierre	
M. WOZNIAK Michel	Spectrochimie
Mme ZINN JUSTIN Nicole	Algèbre

Remerciements

Je remercie Monsieur Michel Mériaux, professeur à l'EUDIL de m'avoir fait l'honneur de présider le jury de cette thèse.

Je remercie également Messieurs Daniel Etiemble et Daniel Litaize d'avoir accepté ce difficile travail de rapporteurs. Je les en remercie pour leurs commentaires, critiques et observations qui m'ont permis de corriger ce document et pour leurs encouragements qui ont renforcé ma motivation.

L'équipe PALOMA (PArallélisme LOgiciel et MAtériel) est dirigée par Bernard Toursel, Professeur et directeur adjoint de l'EUDIL. Je lui suis reconnaissant de m'avoir accueilli dans son équipe et pour la confiance qu'il m'a accordée. Ses précieux conseils et son aide ont été déterminants pour la rédaction de cette thèse.

Monsieur Paul Alain Rolland s'est intéressé à nos travaux, en particulier en participant à ce jury. Je lui en suis très reconnaissant.

Je tiens aussi à remercier Messieurs Gilles Goncalves et Tienté Hsu de m'avoir soutenu tout au long de ces trois années d'études. Leurs multiples conseils, encouragements et leur gentiesse m'ont souvent aidé à traverser des périodes difficiles.

Mes derniers remerciements s'adressent à toutes les personnes qui ont contribué de près ou de loin au bon déroulement de ce travail, en particulier Pierre Vangeluwe de l'IEMN et l'ensemble des membres de l'équipe PALOMA.

À ma famille,

À mes amis.

Table des matières

Introduction

Pour un grand nombre d'applications essentiellement dans le domaine scientifique (prévisions météorologiques, analyse des flux aérodynamiques etc.), la puissance de calcul requise nécessite le développement de calculateurs parallèls très puissants. Cependant la puissance crête des machines parallèles n'est presque jamais atteinte, et ceci pour des raisons se situant à tous les niveaux de l'exploitation : applicatif, logiciel de base et matériel.

- **le niveau applicatif** : pour une taille de problème donnée, une loi fondamentale, la *loi d'Amdahl* dit que l'accélération sur toute machine parallèle est limitée par la fraction séquentielle du code. En d'autres termes, la puissance crête ne peut être atteinte que sur des programmes complètement parallélisés.

- **le niveau logiciel de base** : à la difficulté de conception du logiciel exploitant pleinement les ressources matérielles des machines (les paralléliseurs automatiques, etc), s'ajoutent des problèmes tels que le placement de tâches beaucoup plus spécifiques à certains types d'architectures parallèles.

- **le niveau matériel** : malgré les innovations technologiques et architecturales pour améliorer les débits des machines parallèles afin de suivre les performances des processeurs dans le but de construire des architectures équlibrées (puissance des processeurs, taille mémoire, réseau d'interconnection), il reste néanmoins que les communications constituent les aspects les plus limitatifs des machines parallèles.

Sur le plan technologique, si l'architecture des ordinateurs a bénéficié de la très rapide évolution de la microélectronique, évolution encore prévisible pour quelques années encore, il est inévitable que les limites pratiques ou fondamentales finiront par ralentir cette croissance. La question de la contribution possible de technologies autres qu'électroniques dans la conception des ordinateurs n'est donc pas nouvelle [11] [57]. Cette question concerne les experts des domaines aussi bien de la chimie, de la physique, de la biologie, des hyperfréquences que de l'optique. Nous avons ainsi étudié au LIFL[1] en collaboration avec le DHS[2], les contributions possibles de la technologie hyperfréquences dans la conception des réseaux d'interconnexion des machines parallèles à passage de messages. Après avoir montré la réalisation possible des liaisons hyperfréquences au travers d'un guide d'ondes entre un ensemble de processeurs, nous avons étudié une utilisation de ces liaisons en hyperfréquences pour la conception de grands réseaux. Ce document est structuré en cinq chapitres :

[1] Laboratoire d'Informatique Fondamentale de Lille
[2] Département Hyperfréquences et Semiconducteurs de l'I.E.M.N.

Le premier chapitre présente l'évolution des systèmes informatiques, des monoprocesseurs jusqu'aux machines massivement parallèles, en accordant une plus grande attention aux performances de leur réseau d'interconnexion en termes de débit et de latence. La dernière partie de ce chapitre exposera les techniques mises en oeuvre pour augmenter la bande passante des réseaux d'interconnexion.

Dans le second chapitre, après une ébauche des études réalisées à partir d'autres technologies pour l'ordinateur, essentiellement l'optique, nous présenterons la technologie hyperfréquences et son impact actuel dans des domaines comme les réseaux locaux et mondiaux.

Le troisième chapitre met en exergue les contraintes nous ayant amenés à l'utilisation de guides d'ondes et présente le protocole mis en oeuvre pour la communication entre des noeuds connectés à un guide d'ondes. Des estimations sur les performances physiques attendues seront données. Nous concluons ce chapitre par un bref exposé des réalisations électroniques effectuées au DHS pour valider notre modèle.

Le quatrième chapitre définit la topologie d'un réseau d'interconnexion pour machines massivement parallèles MIMD, construit à base de guides d'ondes pouvant intégrer un nombre quelconque N d'éléments de calcul. Des réseaux étudiés durant la dernière décennie trouvent ici une application grâce à l'avance technologique de ces dernières années dans le domaine de l'intégration des composants hyperfréquences. La topologie du réseau retenu sera une réactualisation du dual-bus-hypercube[83].

Nous effectuons dans le dernier chapitre des mesures de performances du réseau décrit plus haut. A la méthode d'évaluation choisie qui est celle des files d'attentes, nous apporterons d'abord une résolution analytique pour un modèle plus simple du réseau. Le second modèle du réseau plus compliqué pour une résolution analytique sera étudié par simulation. Le simulateur ayant été validé d'abord sur le premier modèle par comparaison avec les résultats analytiques.

Nous terminons le document par une présentation de quelques solutions purement théoriques pour la construction de grands réseaux à partir de guides d'ondes : la théorie des *hypergraphes* [6] plus élaborée que celle des graphes simples fournit en effet des outils bien adaptés à la communication par groupes, dans la mesure où nous supposerons que tous les noeuds connectés au même guide d'ondes constituent un groupe.

Chapitre 1

Communications dans les architectures parallèles

Quels que soient les développements en terme d'architecture, les performances globales d'un supercalculateur reposent sur l'équilibre entre la puissance de calcul, la bande passante de communication, la taille mémoire et les entrées/sorties. Plus les processeurs élémentaires sont puissants et rapides, plus la mémoire doit l'être pour alimenter les processeurs en données dans un flux optimisant leurs traitements. L'efficacité des systèmes parallèles à base de processeurs RISC à architecture load-store dépend par exemple principalement de l'optimisation des accès mémoires, celui des systèmes massivement parallèles dépend essentiellement des performances du réseau d'interconnexion. Nos travaux sont orientés vers la diminution du temps requis pour les communications dans les architectures massivement parallèles. Ce chapitre présente après un bref exposé de la problématique qui est celle de la puissance de calcul des machines, une étude de quelques architectures parallèles avec comme critère de comparaison leur réseau d'interconnexion. Pour une approche plus complète du domaine, citons les ouvrages de références tels que HWANG et BRIGGS [44], STONE [75], HOCKNEY et JESSHOPE [40] [41], HENNESSY et PATTERSON [37] [38], HWANG [43] .

1.1 L'évolution et les limitations des systèmes monoprocesseurs

Actuellement, nous observons une croissance exponentielle des performances des technologies CMOS. Comme conséquence de la diminution des structures formant les composants élémentaires (souvent exprimée en termes de *technologie* 1,5 μm, technologie 0,7 μm etc.), la densité d'intégration est multipliée tous les ans par un facteur **1,5** pour les mémoires et un facteur **1,35** pour les processeurs (cf figure 1.1). Quant à la fréquence d'horloge, elle est multipliée par un facteur **1,24** tous les ans (cf figure 1.2). Les longueurs des canaux des transistors n'ont donc cessé de diminuer depuis plusieurs années et il est certain que l'on peut encore progresser dans la miniaturisation. Les longueurs de 0,8 μm sont maintenant courantes, le passage en 0,5

μm est déjà planifié, et les constructeurs envisagent des structures de 0,3 μm. Cette évolution des technologies CMOS et de la lithographie associée est prévisible, au moins jusque vers des technologies 0,2 ou 0,1 μm, même si l'évolution ultérieure est beaucoup moins certaine, par exemple à l'occasion de l'apparition des effets quantiques dans des structures plus petites que 100 nm.

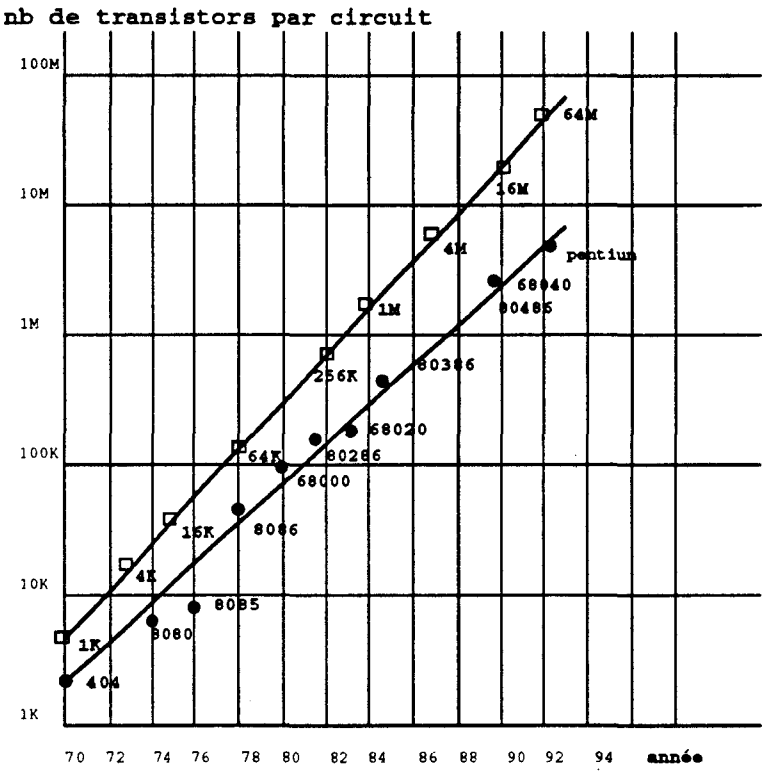


Figure 1.1 : Evolution exponentielle de la densité d'intégration des mémoires : 1.5 par an et des processeurs : 1.35 par an

Parallèlement à cette évolution technologique des processeurs, des progrès sur le plan architectural ont été faits et ont conduit par exemple à la spécification de processeurs de types **RISC**[1] dont la commercialisation se généralise (exemple des processeurs SPARC, T9000, Mips R4000, Dec 21064, T.I. Supersparc, PowerPC [71] et [72]). On parle également d'architecture **pipeline**, **Superpipeline**, **Superscalaire** et de **VLIW** dont une étude peut être trouvée dans [24].

Cependant cette évolution des performances des processeurs ne va pas sans induire des problèmes qui sont essentiellement ceux liés à la vitesse, au packaging et aux débits mémoires (une étude plus approfondie du domaine est effectuée dans [25] et [38]). Nous décrivons ci-dessous succintement ces trois principales difficultés avec quelques techniques mises en oeuvre pour les pallier.

[1] Reduced Instruction Set Control

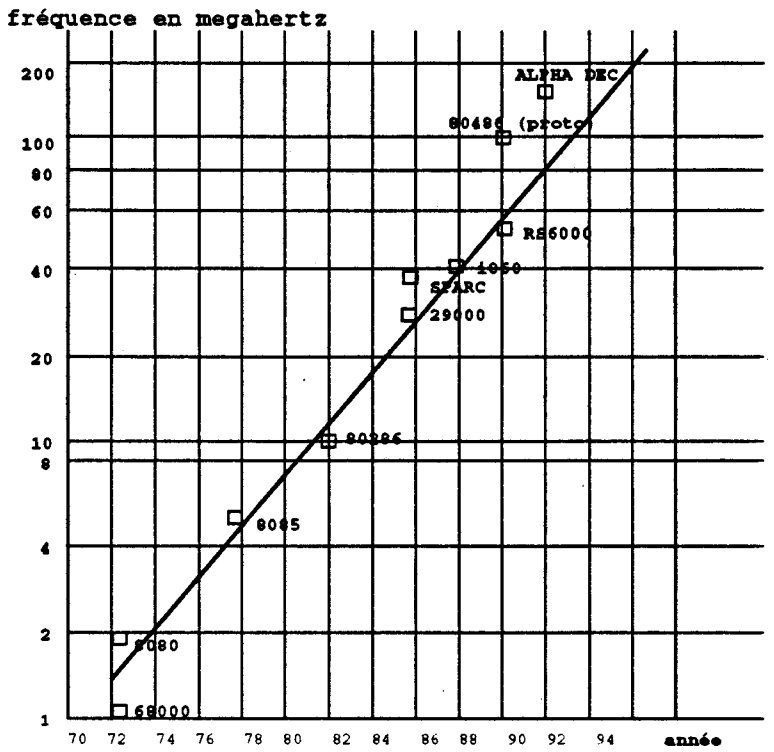


Figure 1.2 : Croissance exponentielle de la fréquence d'horloge des processeurs CMOS

1.1.1 Problèmes liés à la vitesse

Il est bien connu que plus la fréquence de transmission augmente, plus les phénomènes de diaphonie (influence d'une ligne sur une autre) et de réflexion sont deviennent critiques au niveau des connexions métalliques (les changements d'impédances créent une surtension ou une sous-tension suivant le type de réflexion), limitant de ce fait les interconnexions. De plus les problèmes de dissipation de chaleur dans les circuits intégrés sont aussi liés à la vitesse de transmission.

1.1.2 Problèmes liés au packaging

La puissance dissipée est une grandeur directement liée au taux d'intégration des transistors. Ainsi, pour limiter la puissance dissipée, les constructeurs sont obligés de diminuer la tension de commutation (V_{dd}) des logiques. D'autres problèmes liés au packaging concernent la limitation du brochage puisque plus de la moitié de la chaleur dissipée par un composant est le fait des plots d'entrée-sorties [81] et cette puissance, proportionnelle à la fréquence F de l'horloge est donnée par $P = \sum C \times V_{dd}^2 \times F$ [43] (les processeurs *Alpha AXP* [56] [23] dissipent plus de 23W pour un temps de cycle de 6,6 ns (150 MHz) et 431 broches (V_{dd}=3,3 Volts)), C désignant la capacité du composant.

1.1.3 Problèmes liés aux débits mémoire

L'évolution des performances des processeurs s'accompagne aussi d'un écart croissant entre l'unité centrale et la mémoire principale, les temps d'accès à la mémoire n'évoluant plus lentement que les performances des processeurs (cf figure 1.3 issue de [38]). Diverses améliorations sur le plan architectural ont été apportées pour pallier cet écart : l'utilisation systématique de caches, des mots mémoires plus larges (1 mot MP = k mots UC), l'entrelacement des adresses mémoire (m bancs mémoires), etc.

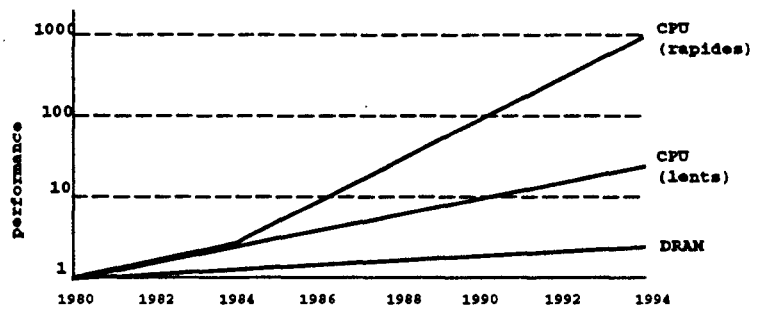


Figure 1.3 : Evolution des DRAMs et des processeurs

1.1.4 Conclusion

Malgré l'accroissement de la puissance de calcul individuelle des processeurs, les performances des systèmes monoprocesseurs restent toujours en deçà de la puissance de calcul requise actuellement dans de nombreux domaines. Cette limitation n'est pas dûe uniquement aux limitations technologiques d'intégration, mais aussi aux limitations des interconnexions processeur-mémoire. L'exploitation du parallélisme reste de ce fait vraisemblablement l'approche idéale pour un véritable saut de puissance, et augmenter les performances des calculateurs. Par le terme parallélisme, nous entendons la coopération de plusieurs éléments de calcul pour l'exécution d'une même tâche.

1.2 Les machines parallèles

De manière standard, les machines parallèles sont composées de plusieurs noeuds de calcul reliés entre eux par un réseau d'interconnexion. Un noeud est généralement constitué par (cf par exemple figure 1.4) :

- le processeur de calcul scalaire auquel peut s'ajouter une ou plusieurs unités flottantes ou des processeurs vectoriels.
- une mémoire locale ou un accès à une mémoire globale. (L'accès à cette mémoire partagée se fait par le réseau).
- une horloge interne
- une unité de gestion de communications, qui gère le routage des messages entre les noeuds : calcul de la fonction de routage, multiplexage des liens.

A cette description, peuvent s'ajouter :

- des mémoires caches de données et d'instructions
- un noyau système du type UNIX, permettant une meilleur gestion des ressources du noeud (accès aux fichiers sur disque, affichage de données, mémoire virtuelle)
- un analyseur de performances qui enregistre certains événements qui se déroulent sur le le noeud. Ceci permet de visualiser sur un écran le déroulement du programme en chaque point du réseau.

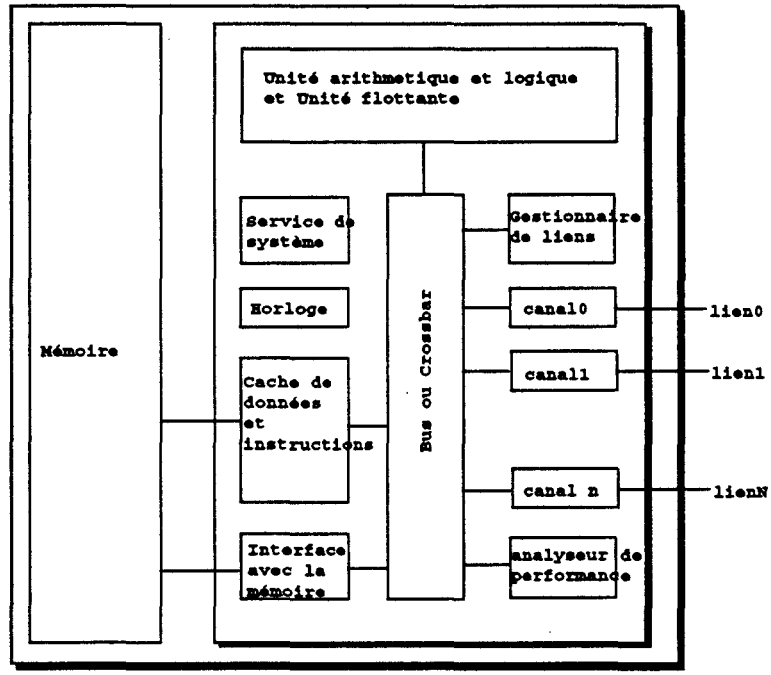


Figure 1.4 : Exemple de noeud dans les machines parallèles à mémoire distribuée, à passage de messages (la mémoire est privée ici et le gestionnaire de communication est chargé du routage des messages)

De plus, lors de la description algorithmique sur une machine parallèle, un noeud peut comporter plusieurs processeurs, ou encore en simuler un grand nombre, on parle alors de processeurs virtuels. Nous parlerons donc dans la suite du document indifféremment de processeur, d'éléments de calcul ou de noeuds, ces trois termes se rapportant à la description du noeud donné ci-dessus.

La grande diversité des machines parallèles a donné lieu, à plusieurs taxonomies parmi lesquelles on peut citer celle de Flynn [27]. Cependant, parmi les critères de différenciation liés au architectures parallèles, on peut noter principalement :

- comment les multiples noeuds sont connectés (topologie)
- si chaque unité de traitement a son propre flux d'instructions : le modèle d'exécution (SIMD, MIMD, etc.)

- si les unités de traitement possèdent leur propre mémoire privée ou partagent un même espace mémoire (mémoire partagée, distribuée ou virtuellement partagée).

Néanmoins, de toutes les machines commerciales nous pouvons retenir trois principaux groupes :

1. SIMD (Single Instruction Multiple Data Stream)
2. MIMD (Multiple Instruction Multiple Data Stream) avec mémoire partagée
3. MIMD avec mémoire distribuée

Cette classification n'est bien sûr pas exhaustive. Il existe en effet d'autres modes de parallélisme, comme par exemple les architectures vectorielles pipelines (MISD pour Multiple Instruction Simple Data) ou systoliques [49] ou les réseaux de neurones [2]. Cependant, les processus de circulation de données s'effectuent dans ces derniers modèles très différemment et ne peuvent relever de la même étude.

Nous décrivons très succintement les trois types de machines énumérés plus haut, en insistant plus sur les architectures MIMD à mémoire distribuée qui constituent le thème central de cette thèse. Nous consacrerons enfin une section entière à l'étude des réseaux d'interconnexion des machines MIMD à mémoire distribuée.

1.2.1 Architectures SIMD

Dans les architectures parallèles SIMD (cf figure 1.5), chaque unité de traitement exécute la même instruction. En d'autres termes, il y a seulement un seul compteur ordinal contrôlant l'exécution du programme. Chaque unité de traitement est réduite à une Unité Arithmétique et Logique (UAL) et une mémoire locale (ML). Ainsi, une seule unité de traitement (appelée frontal) génère le flux de contrôle et envoie les instructions à toutes les unités de calcul ; les couples (UAL-ML) étant généralement appelés processeurs élémentaires (PE). Les processeurs élémentaires sont reliés entre eux par un réseau d'interconnexion. Les réseaux les plus usités sont les hypercubes et les réseaux maillés 2D. Le frontal joue le rôle du "host" qui génère les instructions. Il a accès au réseau d'interconnexion, aux canaux d'entrée/sortie et à sa propre mémoire. Parmi les machines de cette catégorie, citons la Connection Machine CM-2 [39] et les machines MasPar MP-1 [62] et MP-2.

1.2.2 Architectures MIMD à mémoire partagée

Dans les architectures à mémoire partagée (cf figure 1.6), tous les processeurs partagent un même espace de données global. Dans ce cas, tous les processeurs peuvent accéder à n'importe quel banc mémoire via un réseau d'interconnexion, chaque processeur ayant le même temps d'accès en l'absence de conflits. Ce modèle de multiprocesseur est appelé *modèle UMA* pour Uniform Memory Access. Il arrive aussi assez souvent que chaque processeur ait sa propre antémémoire (une mémoire cache) et/ou une mémoire locale qui peut être privée ou non (suivant que les autres processeurs peuvent y accéder ou pas). Dans ces cas, des protocoles doivent être mis en oeuvre pour maintenir la cohérence des mémoires caches locales. Diverses topologies de réseaux ont été mises en oeuvre pour connecter les processeurs aux différents bancs mémoires : l'Alliant FX/8 et FX/2800 utilisent un simple bus ; la machine M3S[2] [67] [53] utilise une autre

[2]Multiprocesseur à Mémoire Multiport Série

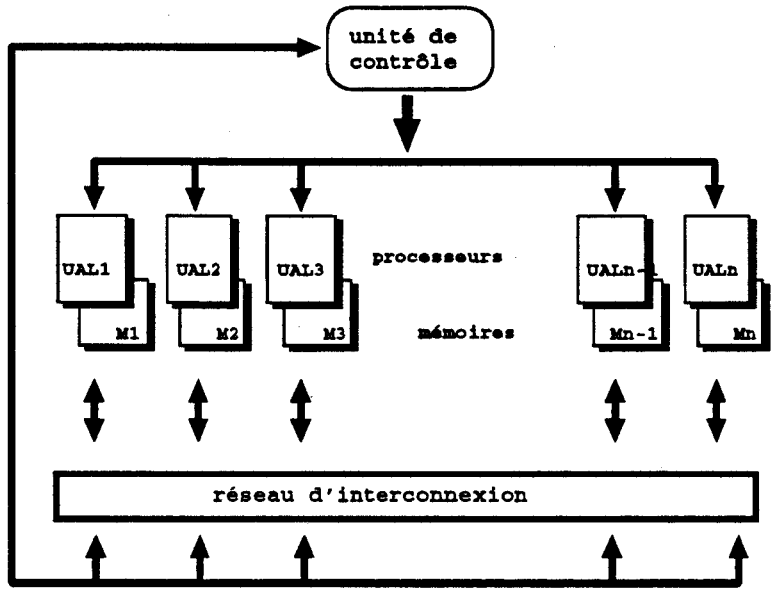


Figure 1.5 : Architecture parallèle SIMD

approche consistant à relier les processeurs aux mémoires par des liens séries à haut débit (cf figure 1.7). Pour des systèmes utilisant un petit nombre de processeurs (moins de 16), il est possible d'utiliser le crossbar qui est le réseau reconfigurable idéal (parce que permettant non seulement la diffusion, mais aussi toutes les permutations des entrées vers les sorties). Les machines Cray (à l'exception de la T3D) et l'IBM 3090 font partie de de la famille de machines utilisant un crossbar pour les accès mémoires.

1.2.3 Architectures MIMD à mémoire distribuée

1.2.3.1 Le fonctionnement standard

Dans les architectures parallèles MIMD à mémoire distribuée, chaque Unité de Traitement (UT) a sa propre mémoire locale et exécute les instructions indépendamment des autres Unités de traitements (cf figure 1.8). Les UT n'ont pas accès à la mémoire des autres unités de traitements. Pour avoir accès aux données d'une quelconque unité de traitement dans le système, une requête explicite doit être émise à travers le réseau d'interconnexion ; on parle alors de communication par messages par opposion à la communication par variables partagées dans les architectures à mémoire commune. Plusieurs réseaux d'interconnexion peuvent être utilisés pour ce type d'architecture : anneaux, étoiles, arbres, hypercubes, cube-connected cycles, réseaux omega et butterfly, etc. Cependant, si l'hypercube a été adopté dans les premières machines MIMD à mémoire distribuée, actuellement, nous observons un regain d'intérêt pour les réseaux meshes 2D (W. DALLY montre dans [19] que pour une même largeur de bissection[3], les réseaux de faible dimensions comme les tores possèdent des temps de latence[4] plus faibles et des débits plus élevés que les réseaux de dimensions élevées, en l'occurence les hypercubes). Les systèmes à base de Transputers, le Cosmic Cube du Caltech [70] et ses successeurs industriels, les hypercubes

[3] Le nombre minimum de liens qui, une fois retirés, séparerait le réseau en deux sous réseaux
[4] Temps requis pour l'envoi d'un message de la source vers la destination

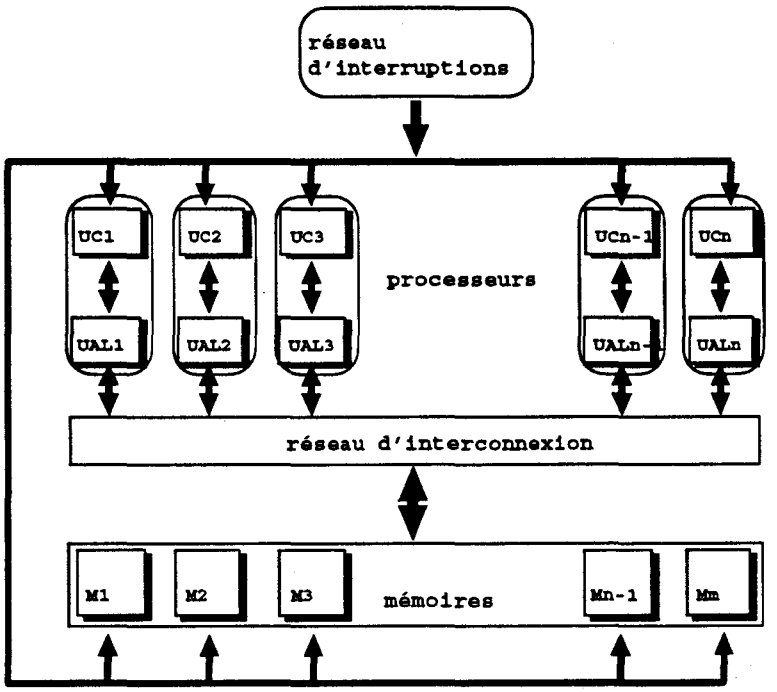


Figure 1.6 : Architecture parallèle MIMD à mémoire partagée

commerciaux dont les plus connus sont les iPSC/1 et /2 et la machine Paragon [47] sont des exemples de ce type d'architecture.

1.2.3.2 La tendance actuelle

Actuellement, de nouveaux modes de fonctionnement sur ce type d'architecture commencent à être généralisés ; le premier se distingue par le modèle de programmation se rapprochant des architectures SIMD et le second est caractérisé essentiellement par une gestion de la mémoire similaire à celle des architectures à mémoire partagée.

le mode SPMD (Single Program Multiple Data) : Un même programme s'exécute sur tous les noeuds, mais les instructions qui s'exécutent à un instant donné sur des processeurs différents peuvent être différents (par exemple en cas de branchement conditionnel). Les machines ayant ce mode de fonctionnement possèdent généralement un ou plusieurs dispositifs de synchronisation qui peuvent être implémentés de manières différentes. Ainsi, si la notion de *barrière de synchronisation* a été introduite sur la CM5 pour synchroniser les processeurs par l'utilisation d'un réseau spécial, le réseau de contrôle (différent du réseau de données), le Cray T3D fournit des primitives matérielles plus élaborés [63] dont le *Barrier Synchronization* et le *Eureka Synchronization* pour les recherches parallèles. Il faut aussi noter que contrairement au SIMD qui impose un mode de fonctionnement synchrone et un réseau capable de fournir un flux d'instructions tous les cycle de calcul, la synchronisation dans le mode SPMD est lié au programme à exécuter et non pas à l'architecture. La machine Paragon [47] par exemple contrairement aux deux machines citées ci-dessus utilise uniquement des mécanismes logiciels fonctionnant à partir du réseau de données de type mesh 2D. On parle aussi de plus en plus de programmes SPMD sur des réseaux

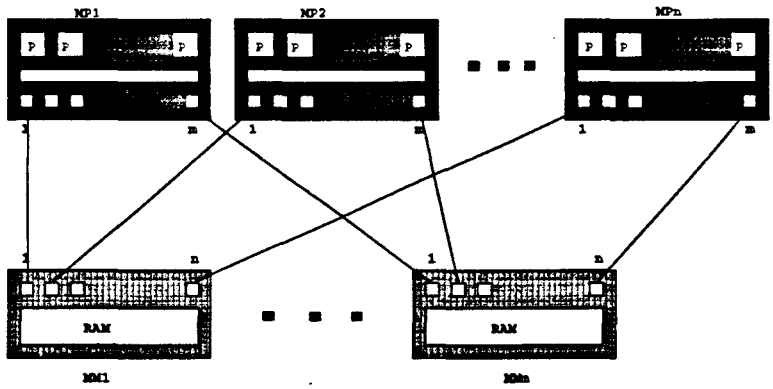


Figure 1.7 : Architecture Multiprocesseur M3S

de stations de travail.

la mémoire virtuellement partagée : La mémoire locale à chaque UT n'est plus privée. On distingue généralement deux modèles de multiprocesseurs à mémoire distribuée et virtuellement partagée : le modèle NUMA pour Non Uniform Memory Access et le modèle COMA pour Cache-Only Memory Architecture. Ces deux modèles étant étudiés pour améliorer les performances des modèles UMA en réduisant évidement les accès au réseau d'interconnexion (processeur-bancs mémoires).

- le modèle NUMA : Les données étant réparties sur l'ensemble des mémoires locales, les temps d'accès sont différents suivant que la donnée est présente dans la mémoire locale ou dans une autre mémoire. On appelle ainsi *mémoire home* d'une donnée, le module mémoire qui la contient. Il est évident pour ce modèle que les performances de ce type de machine sont étroitement liées à la répartition des données dans les différentes mémoires locales. Le Cray T3D [63] et la machine DASH [51] sont des exemples de ce modèle de multiprocesseurs.
- le modèle COMA : Tous les UTs ont le même espace d'adressage composé de plusieurs grosses mémoires caches distribués sur les UTs, d'où la terminologie *ALLCACHE memory*, l'endroit où se trouve une donnée dans la machine étant totalement indépendant de son adresse physique. Quand un défaut de bloc survient, la requête est envoyée vers une autre mémoire cache à travers le réseau d'interconnexion. Le bloc ainsi demandé devient alors, suivant le type de requête, soit partagé, le système devant fournir les moyens de maintenir la cohérence des copies multiples, soit non partagé, le noeud demandeur gardant alors l'exclusivité des accès au bloc, les autres copies du bloc devant alors être invalidées. La machine KSR1 [33] [48] de Kendall Square Research a le même mode de fonctionnement avec un réseau hiérarchique à base d'anneaux.

1.2.4 Conclusion

Pour exécuter un programme N fois plus vite, il ne suffit pas de mettre N processeurs ensembles. Les N processeurs doivent échanger des informations, or les communications entre processeurs prennent du temps (et constituent même dans certains cas l'essentiel du temps total), celui-ci étant fortement lié au réseau d'interconnexion. De la description rapide des machines parallèles faite ci-dessus, il ressort qu'une des caractéristiques essentielles des architectures parallèles est

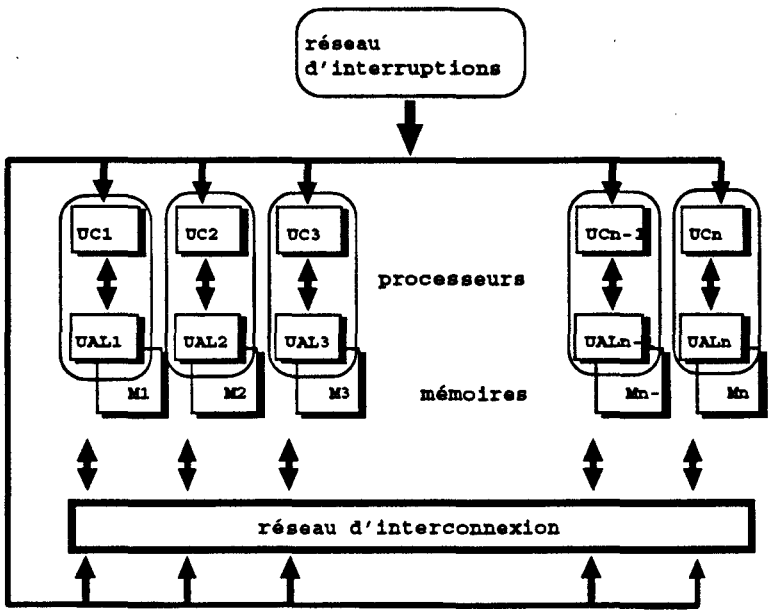


Figure 1.8 : Architecture parallèle MIMD à mémoire distribuée

celle des ***communications entre processeurs***, nous présentons donc dans la section suivante les différents types de réseaux et les techniques mises en oeuvre pour augmenter leur efficacité en terme de débit et de latence.

1.3 Les réseaux d'interconnexion

Les ressources (liaisons) utilisées lors d'une communication entre deux noeuds dans une architecture parallèle peuvent être allouées dynamiquement c'est-à-dire à la demande et au cours de l'exécution de l'application, on parle alors de topologies dynamiques, ou sont conçues et fixées à la construction de la machine, il s'agit dans ce cas de topologies statiques [66].

1.3.1 Les topologies dynamiques

Dans les topologies dynamiques, le support physique est fixé, mais des commutateurs ou connecteurs permettent de modifier le schéma de connexion du réseau. Utilisés surtout pour le partage des ressources telles que la mémoire (les bancs mémoires), une bonne classe des topologies dynamiques concerne les *réseaux multi-étages* (les réseaux de Clos [12], de Benès [5] etc.). Le plus simple de ces réseaux est le bus utilisé en temps partagé, mais ne permettant de connecter qu'une trentaine au plus de noeuds pour des raisons évidentes de saturation [52]. Le plus complexe des réseaux dynamiques est le crossbar que l'on peut représenter par une grille N*M, les processeurs étant en tête des lignes, les mémoires en tête des colonnes. Le crossbar est dit *réarrangeable* parce qu'il permet toute permutation des entrées vers les sorties, il est aussi *non bloquant* parce que les connexions entre les entrées et les sorties peuvent être établies indépendamment sans changer les autres connexions. **Le bus et le crossbar bornent le spectre du prix et de la performance pour pratiquement tous les réseaux d'interconnexion à topologie dynamique.**

Les réseaux multi-étages ont fait l'objet de nombreuses études ces quinze dernières années, et l'on en dispose actuellement de plusieurs modèles. L'intérêt pour la plupart de ces réseaux est qu'ils conservent quelques qualités du réseau idéal (le crossbar) avec une complexité moindre. Les réseaux de Benès, Omega, Baseline etc sont quelques exemples de ces réseaux largement commentés dans l'ouvrage de F. THOMSON LEIGHTON [50]. Parmi les projets de machines utilisant ce type de réseau, on peut citer le projet PTAH [10] qui utilise un réseau non bloquant pour réduire au minimum les temps de latence réseau qui doivent de plus être constants (5 cycles processeur pour 1K PE) pour répondre aux exigences de l'architecture, le projet Supernode [59] qui utilise un réseau de Clos formé de crossbars C004 *d'Inmos*, la machine CHiP [74] de l'université de Purdue, formée d'un réseau de processeurs interconnectés par une matrice de circuits d'aiguillage (switch) programmables de façon statique, et la machine RPP [60] (Reconfigurable Parallel Processor) de l'Université de Kyushu. Le SP1 [35] d'IBM et de CS2 [46] de PCI sont des exemples de machines commerciales utilisant un réseau multi-étage pour la communication.

L'avantage des topologies dynamiques réside essentiellement dans leurs reconfigurabilité, simplifiant de ce fait le placement de tâches sur la machine parallèle, puisque tout noeud peut être considéré comme voisin immédiat d'un autre physiquement éloigné. Le mérite des réseaux multi-étages est qu'ils essaient de relier tous les noeuds du réseau, simulant un graphe complet. Parmi les projets actuels dans ce domaine, nous pouvons citer le projet ARP [42] (Architecture à Réseau Partagé) mené actuellement à l'Université de Lille. Le réseau ARP est un réseau d'interconnexion à reconfiguration dynamique et asynchrone permettant l'allocation de ressources de communication (à la demande) pendant l'exécution d'une application parallèle.

1.3.2 Les topologies statiques

Le réseau d'interconnexion dans les topologies statiques est fixé à la construction de la machine. La définition de ces réseaux inclut d'une part la géométrie (topologie) modélisée par un graphe et d'autre part l'algorithme de routage défini sur la topologie. D'une manière beaucoup plus fine, on peut aussi définir les propriétés matérielles des liens essentiellement par la largeur du canal, c'est-à dire la quantité d'informations élémentaires d'un transfert physique entre processeurs voisins. Bien qu'un graphe complètement connecté évite de passer par des noeuds intermédiaires pour toute communication entre noeuds, l'utilisation de tels graphes est limitée à un petit nombre de noeuds, limitations dues essentiellement aux contraintes physiques, ne serait-ce que celles liées au nombre de broches ou liées à la surface VLSI des circuits. Le graphe modélisant la topologie est en général un graphe non complètement connecté dont les sommets sont les noeuds de la machine et les arêtes les liens de communication. A cette formalisation sont attachées plusieurs notions classiques largement exploités dans la littérature [21] :

- *Le degré* (Δ) d'un sommet (d'un noeud) est le nombre de sommets auxquels il est relié. Le graphe est dit regulier si tous les sommets ont le même degré, qui est alors appelé degré du graphe (du réseau).
- *La distance* entre deux sommets est la longueur du plus court chemin qui les relie.
- *Le diamètre* (**D**) du graphe est le maximum des distances dans le graphe.

Ces propriétés purement géométriques des graphes nous fournissent un premier ensemble de critères de comparaison des réseaux. D'autres propriétés beaucoup plus fines nous paraissent importantes et utilisées essentiellement lors de la spécification de notre réseau au chapitre 4 :

- *Le degré physique* d'un sommet est le nombre de connexions physiques (ou ports) que possède un noeud.
- *Le degré logique* d'un sommet est le nombre de sommets auxquels il peut directement accéder.

Les deux notions définies ci-dessus *a priori* identiques trouvent leurs sens quand on suppose qu'une arête peut relier un groupe de noeuds et non plus un seul noeud comme dans les cas classiques. On parle alors dans ce cas d' *hypergraphes* [6] dont une brêve présentation est fournie en annexe. Evidemment le groupe de noeuds peut se réduire à un seul noeud. Dans ce cas les degrés physique et logiques sont identiques et nous rejoignons la définition initiale du degré relative aux graphes simples.

La géométrie du réseau d'interconnexion d'une machine massivement parallèle devrait présenter les caractéristiques suivantes :

1. un grand nombre de sommets (**N**) : le nombre de processeurs doit être élevé pour autoriser un taux de parallélisme élevé.
2. un faible degré physique (Δ) : chaque processeur ne peut avoir qu'un petit nombre de connexions (des ports), essentiellement à cause des limitations technologiques d'encapsulation.
3. un faible diamètre (D) : caractéristique essentielle si l'architecture doit supporter de nombreuses classes d'applications avec des topologies logiques (graphes d'exécution de ces applications) très différentes. La distance entre les processeurs doit donc être petite.

La théorie des graphes a fournit de nombreux résultats [8] relatifs aux réseaux d'interconnexion et a notamment montré qu'il existe une relation entre les trois paramètres **N**, **D** et Δ puisqu' évidemment les concepteurs auraient souhaité un réseau satisfaisant les trois exigences contradictoires ci-dessus. On montre ainsi que si le degré (Δ) et le diamètre (D) sont fixés, il existe une limite théorique au nombre de sommets du graphe, connue comme la borne de Moore et égale à :

$$1 + \Delta + \Delta(\Delta - 1) + \Delta(\Delta - 1)^2 + \ldots + \Delta(\Delta - 1)^{(D-1)}$$

Un autre critère de comparaison des réseaux d'interconnexion est la densité d'interconnexion qui réflète essentiellement les difficultés technologiques lors de leur réalisation physique puisqu'une des limitations essentielles des systèmes à base de VLSI réside essentiellement dans les connexions [69]. En particulier pour faciliter la fabrication des composants de VLSI, il faut que les modules d'interconnexion soient homogènes ; ce qui suppose que les sommets aient le même degré (ce qui n'exclut pas que certaines petites irrégularités de degré soient exploitées à d'autres fins, essentiellement pour connecter des périphériques : les entrée-sorties etc.).

Plusieurs réseaux ont été proposés pour les machines parallèles soit parce qu'ils offraient de très bonnes propriétés géométriques comme les graphes de De Bruijn [6], soit parce qu'ils présentaient moins de difficultés lors de leur réalisation VLSI comme les grilles, les tores ou les hypercubes [26] dont on dispose de bons algorithmes de layout[5]. Cependant, des machines commercialisées, nous pouvons retenir :

l'hypercube de dimension n possède un degré et un diamètre égaux à n. Assez modulaire, il peut être construit récursivement à partir d'hypercubes de dimensions inférieures (cf

[5]Cette caractéristique est toutefois de moins en moins considérée car la plupart des machines parallèles actuelles (cf table1.1 en fin de chapitre) tendent plutôt à utiliser des processeurs existants qu'à spécifier des noeuds entiers dans un seul chip

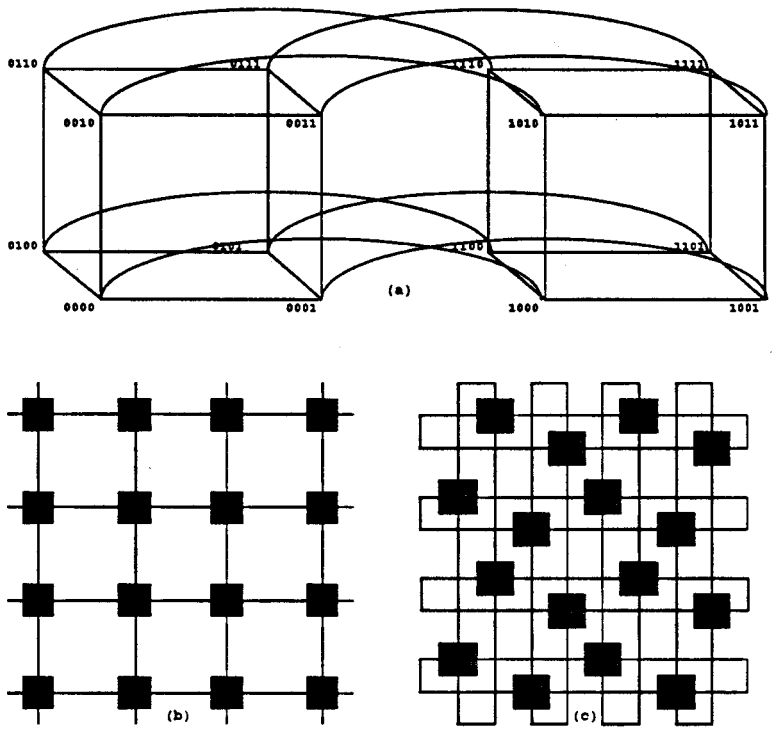


Figure 1.9 : Les réseaux (a) hypercube de dimension4 (b) grille 2D 4x4 et (3) tore2D 4x4

figure 1.9(a)). le Cosmic Cube [70], le Ncube, l'iPSC [76], la CM-2 sont des machines dont la topologie est un hypercube.

la grille ou mesh de dimension n et de base k a k^n sommets que l'on peut voir comme les points de coordonnées entières comprises entre 0 et k-1 dans un espace euclidien de dimension n (cf figure 1.9(b)). La grille n'est pas un graphe régulier puisque les sommets internes ont un degré 2n et ceux situés à la périphérie un degré inférieur. Un des intérêts majeurs de la grille est son extensibilité : on peut ajouter indéfiniment des noeuds au réseau sans modifier la structure des noeuds existants, en l'occurence le degré (ou le nombre de ses ports physiques). La grille et sa variante la grille torique sont utilisées par des machines comme l'Ametek, la machine Paragon [33] (cf figure 1.11) et la machine Cray T3D. La grille torique ou tout simplement le tore est une amélioration de la grille, permettant la

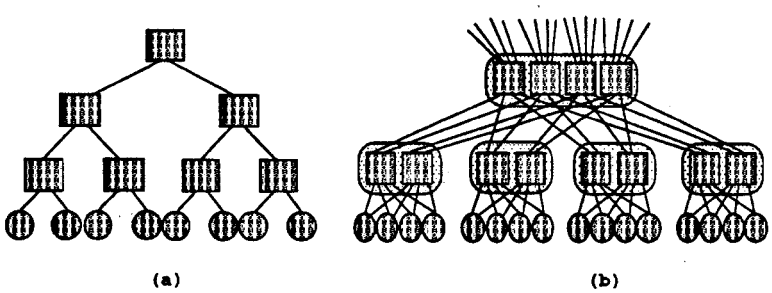


Figure 1.10 : L'arbre binaire complet (a) et le Fat-tree implanté sur la CM5 (b)

connexion physique des deux sommets les plus éloignés sur chaque direction (cf figure 1.9(c)). l'Illiac mesh assez particulière de la machine Illiac IV, est aussi une variante de la grille torique.

l'arbre avec ses améliorations telles que l'hyper-arbre et le fat-tree (cf figure 1.10) implantée sur la CM-5, offre l'avantage des topologies hiérarchiques permettant essentiellement des communications locales plus efficaces.

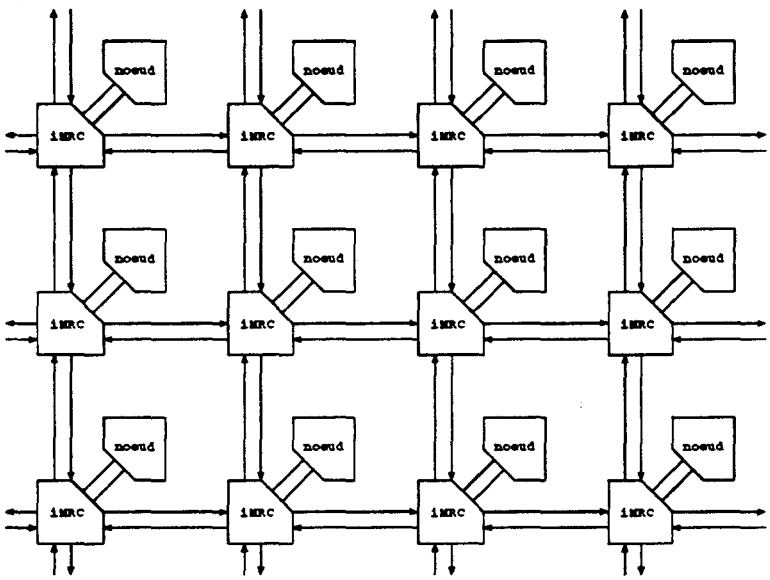


Figure 1.11 : Le réseau Mesh de la machine Paragon

Pour construire de grands réseaux de milliers de noeuds, les topologies statiques sont plus adaptées, ceci grâce aux liens directs entre les modules, simplifiant les protocoles de communication. D'autre part, ce type de topologie supporte plus aisément les communications locales puisque dans le cas des topologies dynamiques, toute communication requiert la même complexité, dans la mesure où la notion de localité n'a plus de sens dans les topologies dynamiques.

1.3.3 Le routage

Pour une machine parallèle de quelques dizaines de noeuds, on peut concevoir un réseau d'interconnexion permettant de relier toute paire de noeuds, et ces connexions sont possibles grâce aux topologies dynamiques. Mais au delà, seul un réseau non complètement connecté peut être utilisé. Et pour que ce type de réseau puisse permettre la communication de tout couple de noeuds, les messages doivent franchir une suite de noeuds du réseau entre l'émission et la réception. Tous les noeuds doivent donc pouvoir fournir des moyens pour l'expédition, la réception d'un message et surtout permettre la réexpédition (routage) efficace de messages qui ne leur sont pas destinés. Ceci est aussi valable dans le cas des réseaux multi-étages pour lesquels tous les switches doivent être convenablement configurés pour acheminer le message d'un noeud à un autre. Le routage adopté lors de la conception du réseau résulte de plusieurs choix parmi lesquels on peut distinguer d'une part l'algorithme de routage qui intuitivement s'occupe de la sélection du chemin à emprunter par le message, et d'autre part la technique

de commutation qui définit le mode de réalisation physique du chemin logique. Une autre précaution à prendre lors de la spécification des routeurs concerne la gestion des conflits pour les cas où plusieurs messages sont en compétition pour l'utilisation d'un même lien (cette situation entraîne généralement la mémorisation ou le déroutement de l'un au moins des messages).

1.3.3.1 L'algorithme de routage

Les messages arrivant à chaque noeud intermédiaire doivent être réexpédiés vers un noeud qui les rapproche de la destination finale. L'efficacité d'un réseau se mesurant essentiellement en terme de *temps de latence* (temps mis par un message de la source à la destination finale) et de *débit* (nombre de messages délivrés par unité de temps), l'algorithme de routage doit pouvoir sélectionner le meilleur chemin pour le message arrivant et ceci en un temps court. La complexité de l'algorithme utilisé est en général très lié à la topologie sous-jacente. On trouve donc des algorithmes de routage pour les hypercubes, les tores etc. Cependant certaines propriétés peuvent résulter d'un choix du concepteur dépendant essentiellement du comportement du réseau dans des situations critiques (réseau fonctionnant à pleine charge...). Le routage pouvant donc être :

déterministe : il existe un seul chemin entre la source et la destination sans tenir compte de l'état du réseau. Même si ce chemin est indisponible et qu'il en existe d'autres libres de la même longueur, le message reste bloqué, ce qui suppose des capacités de mémorisation assez importantes à chaque noeud.

adaptatif : le choix du chemin dépend de l'état du réseau, évalué avant chaque transmission ou chaque retransmission.

1.3.3.2 La technique de commutation

Pour communiquer, les machines parallèles ont longtemps connu deux techniques : la commutation de circuits où un chemin physique et complet est établi entre l'envoyeur et le destinataire et la commutation de messages (store and forward) dans laquelle les messages peuvent être complètement tamponnés à chaque noeud (store), ce dernier se chargeant de réexpédier le message (forward) jusqu'à ce qu'il arrive au destinataire. Cette technique a été utilisée dans le prototype Cosmic Cube [70] et beaucoup de machines parallèles de première génération comme l'iPSC-1 et le Ncube1.

D'autres modes de commutation beaucoup plus efficaces ont été étudiés, des techniques pipelinées essentiellement comme le virtual cut-through d'où découle le routage wormhole qui peut être considéré comme un compromis entre les deux modes standards (la commutation de circuits et la commutation de messages). Le wormhole [61] devient de plus en plus systématiquement adopté pour le routage (exemple du réseau de la machine Paragon et des réseaux à base de T9000). En négligeant les temps de commutation des switches dans le réseau, le temps de latence passe de $t = T_c \times D \times \frac{A+L}{W}$ pour le store and forward à $t = T_c(D \times \frac{A}{W} + \frac{L}{W})$ pour un routage wormhole[6] (cf figure 1.12 issue de [16]). Cette technique permet donc asymptotiquement de masquer la distance entre les deux noeuds communiquants si $L \gg D$. La différence fondamentale entre le virtual cut-through et le wormhole vient de la manière dont les messages sont traités en cas de blocage de l'en-tête : dans le cas du wormhole, le message est bufferisé sur tout le chemin alors que dans le cas du virtual cut-through, tout le message s'accumule sur

[6] T_c : le temps de cycle sur les liens, D : le diamètre du réseau, L : la taille du message en bits, A : la taille de l'en-tête et W : la largeur des liens (liens parallèles)

le dernier lien ; ce dernier type de routage impose donc de ce fait des tampons de liens aussi importants que dans le cas du store and forward.

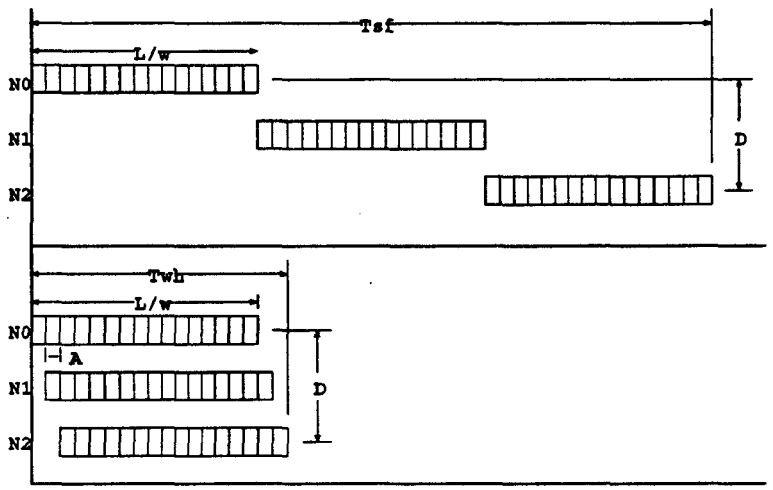


Figure 1.12 : Les temps de latence du store and forward (haut) et du wormhole (bas)

1.3.3.3 La gestion des conflits

La gestion des conflits ne se situe pas uniquement au niveau de l'arbitrage des requêtes pour l'utilisation d'un lien puisque dans ces cas il suffit par exemple de fixer des priorités (éventuellement tournantes) pour résoudre la contention. Les problèmes typiques concernent surtout d'une part les cas d'*interblocages* où un ensemble de messages se bloquent mutuellement, aucun d'entre eux ne pouvant plus progresser, et d'autre part des situations dans lesquelles un message peut circuler indéfiniment dans le réseau (suite à des déroutements), provocant alors ce qu'on appelle un éblouissement du réseau. Le réseau est alors sous utilisé. Il existe deux méthodes bien connues de résolution de conflits. Soit on s'arrange pour que l'algorithme de routage soit exempt d'interblocage (c'est le cas des algorithmes non déterministes où il n'existe pas qu'un seul chemin pour un message donné, en supportant en plus un bonne tolérance aux pannes), soit l'interblocage est admis et il faut alors fournir des moyens pour le lever. Mais les algorithmes de routages étant généralement distribués, aucune connaissance de l'état global du système n'est possible, et très souvent les moyens mis en oeuvre pour déceler ces situations (tout message devant être suivi d'une attente d'acquittement) sont eux aussi consommateurs de la bande passante globale du réseau qui rappelons le, est la ressource la plus chère de la machine parallèle.

De nombreuses études ont été menées pour l'utilisation d'algorithmes de routages adaptatifs c'est-à-dire tenant compte de la charge du réseau mais sans blocage. Ainsi W.J.Dally et C.L.Seitz [17] ont proposé une méthode évitant les interblocages par l'utilisation de la notion de canaux virtuels (le canal physique est divisé en plusieurs canaux virtuels, l'ensemble des canaux virtuels utilisant le canal physique par multiplexage temporel). La condition suffisante pour un routage sans interblocages est l'absence de cycles dans le graphe de dépendance des canaux [28]. Pour un réseau quelconque et une fonction de routage donnée, les cycles de canaux peuvent être supprimés en divisant les canaux physiques en canaux virtuels.

1.3.4 Les primitives de Communication globales et de Synchronisation

Tout environnement de programmation distribuée à passage de messages devrait posséder une bibliothèque de communication générale. Ces primitives sont de plus en plus implémentées par un matériel indépendant pour augmenter les performances en communication des machines parallèles. Parmi ces primitives de communication, les plus importantes sont :

- le transfert : c'est la primitive de communication la plus élémentaire, consistant à envoyer un message d'un noeud vers un autre non nécessairement physiquement reliés (*One-to One*).
- la diffusion : correspond à l'envoi par un processeur d'une même donnée à tous les autres (*Broadcasting, One-to-All, etc.*). Il existe d'autres variantes dont la multidiffusion pour lequel le message n'est plus envoyé à tous les autres noeuds du réseau mais à un ensemble bien précis, et la diffusion personnalisée (*One-to-All personnalisé ou distribution*) où le processeur émetteur doit faire parvenir un message différent à chacun des processeurs. Ce type de communication est évidemment à la base de nombreux algorithmes comme l'élimination de Gauss (diffusion de la ligne pivot), le gradient conjugué ou le produit matrice-matrice (cf chapitre 5). La CM5 par exemple utilise un réseau indépendant *le réseau de contrôle* pour effectuer toutes les opérations globales dont la diffusion. Le réseau multi-étage du CS2 de PCI par exemple intègre dans ses switches (des crossbars 8x8) des fonctions permettant des diffusions efficaces au même titre que des transferts simples de données.
- le regroupement : à l'inverse de la distribution, on trouve le regroupement : on part de messages distribués sur le réseau et on veut les amener sur un même processeur. La réduction est une variante du regroupement, avec combinaisons par des opérations intermédiaires, de tout petits messages.
- l'échange total : correspond à plusieurs diffusions simultannées (*All-to-All*) de tous les processeurs. On parle aussi de *All-to-All personnalisé* où des parties différentes d'un même message sont envoyées de chaque processeur vers tous les autres. Ce dernier type de communication est utile par exemple lors de la multiplication matrice-vecteur quand la matrice est répartie en colonnes sur les processeurs (cf [15] page 419).

Les primitives de communication citées ci-dessus permettent non seulement d'implanter plus efficacement plusieurs classes d'algorithmes mais aussi des mécanismes de synchronisation essentiels en particulier liés au modèle de programmation SPMD. A titre d'exemple, citons le CRAY T3D qui utilise un réseau séparé permettant d'implémenter efficacement une *Barrière de Synchronisation* (un ET logique (combinaison) permettant à tous les processeurs de se synchroniser à une étape donnée de l'application) et le *Eureka Synchronization* (un OU logique sur l'état des processeurs).

1.3.5 Les réseaux locaux : support de communication pour le traitement parallèle

La recherche dans le domaine de réseaux d'interconnexion pour machines parallèles vise naturellement à spécifier des réseaux ayant un débit le plus élevé possible pour supporter les besoins en mouvement de données des applications, et un temps de latence (matériel et logiciel) le plus faible pour satisfaire les exigences des applications à grain très fin. Nous n'avons

Tableau 1.1 : Performances des réseaux d'interconnexion de quelques machines à grand nombre de PE : la colonne **débit** représente le débit du lien reliant chaque PE au réseau, la **latence matérielle** représente en général le temps minimal mis par le premier octet d'un message pour être transporté à travers le réseau.

machine	année	processeur	**réseau**	**débit** (par PE)	latence (1K PE)
Tera	199?	(1200 Mips 1200 Mflops)	tore 3D	3200 Mo/s	
Parsytec GC5	199?	T9000 (200 Mips 25 Mflops)	*	80 Mo/s?	6 μs
CRAY T3D	1992	Alpha (150 Mips 150 Mflops)	tore 3D	300 Mo/s	?
Telmat CS2	1992	SPARC/μVP (150 Mips 150 Mflops)	multi-étage	100 Mo/s	1 μs
IBM SP1	1992	RS 6000	multi-étage	40 Mo/s ?	0,5 μs
CM5	1992	SPARC/C.Vect (128 Mips 128 Mflops)	Fat tree	20 Mo/s	3-7 μs
Intel Paragon	1992	i860XP (50 Mips 75 Mflops)	grille 2D	200 Mo/s	1,5 μs
KSR1	1992	Sharp (40 Mips 40 Mflops)	anneau	32 Mo/s	Mémoire Partagée
iPSC/860	1990	i860XR (40 Mips 60 Mflops)	Hypercube?	22 Mo/s	
iPSC/2	1988	80386/80387 (3 Mips 0,3 Mflops)	Hypercube	22 Mo/s	
iPSC/1	1985	80286/80287 (1 Mips 0,05 Mflops)	Hypercube	10 Mo/s	

de ce fait parlé que des réseaux d'interconnexion pour machines "parallèles" (le tableau 1.1 montre une récapitulation des performances des machines citées dans ce chapitre). Néanmoins, pour faire du calcul parallèle (ou du traitement distribué) à faibe coût, une solution consiste à utiliser un réseau de stations UNIX. Les réseaux utilisés dans ce cas ont naturellement des performances de plusieurs ordres de grandeurs en deça des réseaux d'interconnexion : temps de latences élevés et débits plus faibles. L'utilisation de ce type de réseau pour faire du calcul parallèle n'est envisageable que dans le cas d'applications à très gros grain, c'est-à-dire nécessitant des communications moins fréquentes, chaque communication pouvant porter sur des tailles de données assez importantes. Le réseau n'étant pas très performant, le mode SIMD n'est évidemment pas envisageable parce qu'il faudrait pouvoir distribuer un flux d'instruction à chaque cycle processeur, ces dernier utilisant le même réseau pour leurs communications. Les modèles de programmation possibles sont donc le MIMD et le SPMD, la synchronisation logicielle (sous PVM, Express, etc.) étant effectuée toujours en utilisant le réseau de données.

Il existe dans ce domaine, plusieurs type de réseaux locaux pour le traitement parallèle suivant les exigeances des applications, les débits allant de quelques Mbps[7] à quelques Gbps[8]. Citons

[7] Méga bits par seconde : 10^6 bps
[8] Gigabits par seconde : 10^9 bps

comme exemple le parc de stations *Alpha* de DEC (*Alpha AXP Farms*). Pour l'interconnexion de l'ensemble de stations on peut utiliser :

- un réseau *Ethernet* avec un débit de l'ordre de 10 Mbps, pas très adapté au calcul parallèle, surtout quand plusieurs utilisateurs partagent le même réseau,
- un réseau *FDDI*[9] possédant un débit de 100 Mbps, adapté aux applications parallèles ne nécessitant pas beaucoup de mouvements de données entre les différents processus, comme la simulation par la méthode de Monte Carlo,
- un réseau à commutation très rapide *GIGAswitch à base de FDDI*, adapté à la plupart des applications parallèles comme le 3D FFT pour le traitement du signal. En effet, selon le constructeur (*Digital*), le *GIGAswitch* est un crossbar pouvant offrir un débit crête de 3,6 Gbps (latence ?).

1.4 Conclusion

Nous avons essayé dans ce chapitre de dégager l'importance des machines parallèles, en insistant plus sur leur réseau d'interconnexion comme point critique, d'une part parce que l'efficacité d'une machine parallèle repose essentiellement sur son réseau de communication, et d'autre part parce que la puissance des processeurs élémentaires augmente sans cesse, creusant de ce fait l'écart entre temps de calcul et débit en communication lors du calcul parallèle. Nous avons ensuite mis en exergue les techniques mises en oeuvre actuellement pour arriver à des débits souhaitables (des techniques de routage de plus en plus performantes et des techniques architecturales de plus en plus complexes). Si le transfert entre deux noeuds quelconques constitue la primitive de communication de base dans les systèmes distribués, des primitives de communications globales sont aussi importantes et font d'ailleurs l'objet de beaucoup de recherches actuellement. Une nouvelle approche tendant à se généraliser (parce que moins chère) consiste aussi à utiliser des réseaux locaux comme support de communication pour faire du calcul distribué. Pour augmenter les performances globales des machines parallèles, les inovations ne sont pas qu'architecturales, elles sont aussi technologiques et consistent de ce fait en l'exploitation de technologies moins classiques comme l'optique. Le chapitre suivant expose donc brièvement quelques études dans le domaine de l'optique avant de présenter la technologie Hyperfréquence faisant l'objet de cette thèse.

[9] Fiber Distributed Data Interface

Chapitre 2

De nouvelles technologies pour l'ordinateur

Nous avons jusqu'à présent observé une croissance des performances de la micro-électronique. Bien que la saturation de cette croissance pratiquement exponentielle ait été mainte fois prédite et n'ait jamais été confirmée, on peut penser que des limitations fondamentales se feront sentir dans la décennie 1990, par exemple à l'occasion de l'apparition des effets quantiques dans les structures plus petites que 100 nm. D'autres technologies dont l'optique pour une plus grande part commencent à suppléer les technologies actuelles non pas entièrement (machine totalement optique) mais en prenant en charge certaines fonctions comme les interconnexions, les périphériques de stockage (disques optiques), etc.

2.1 L'utilisation de l'optique

La technologie optique apparaît sans doute aujourd'hui comme la technologie des prochaines années en considérant par exemple son impact actuel dans le domaine des télécommunications. Cependant les recherches dans le domaine de l'optique ne se limitent pas uniquement aux communications et concernent aussi bien les systèmes de stockage de données, les processeurs spécialisés (par exemple pour le traitement d'images ou le traitement numérique), que les réseaux d'interconnexion des machines massivement parallèles. Pour une étude plus approfondie du domaine, A. LOURI [54] montre la faisabilité d'une machine SIMD tout optique, avec une description du modèle de calcul.

2.1.1 Les processeurs optiques

Considérées assez longtemps comme étant des produits "laboratoires", les architectures à base d'optique ont connu un regain d'intérêt avec les exigences en puissance de calcul de plus en plus grandes des applications temps-réel (cf JPDC[1] vol 17, 1993). Les processeurs optiques peuvent donc bénéficier des bonnes propriétés de l'optique : le parallélisme intrinsèque, la vitesse (temps de commutation), l'évolution des connecteurs optiques, et une faible consommation d'énergie.

Un processeur (digital) optique utilise quatre types de sous-systèmes (cf figure 2.1) :

- une source lumineuse (une diode électroluminescente ou LED[2] ou une diode laser)

[1] Journal of Parallel and Distributed Computing
[2] Light Emitting Diode

- un ou plusieurs modulateurs (spatial light modulators SLM[3])
- des lentilles
- un photodétecteur

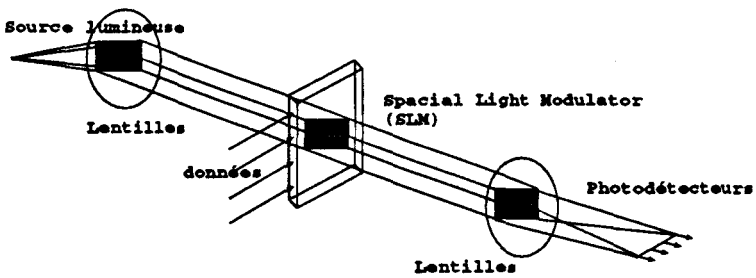


Figure 2.1 : Architecture Optique simplifiée

Le faisceau laser est généré et transmis à travers plusieurs lentilles dans un modulateur de lumière à commande optique et/ou électronique. Le modulateur transforme les données électroniques en entrée, les envoie sous forme de photons vers des lentilles, et enfin un photodétecteur transforme le signal optique résultant sous forme électronique.

L'architecture classique d'un calculateur optique est composée de simples liaisons point-à-point de sources lumineuses entre les composants. Ces calculateurs ont deux panneaux en entrée **A** et **B** appelés modulateurs (SLM) et un panneau de détection en sortie. Les lentilles étant capables de représenter toute source issue d'un panneau vers un autre panneau, elles peuvent projecter le produit (binaire) des sources issues des deux panneaux en entrée vers le panneau de sortie. Le calculateur optique peut réaliser ce type d'opération sur plusieurs faisceaux lumineux (formant ce qu'on appelle une image) simultanément. Ainsi, le processeur optique peut exécuter simultanément plusieurs fonctions mathématiques de base.

Une amélioration de cette technique, appelée *interconnexions optiques 3D* permet de ne plus se restreindre aux connexions "parallèles". Tout point (source lumineuse) issu d'un panneau peut être connecté à tout point du panneau suivant (cf figure 2.2), ceci étant réalisé grâce à l'une des propiétés inhérentes à l'optique : les faisceaux lasers de longueurs d'ondes différentes n'interfèrent pas entre eux.

2.1.1.1 Digital Optical Computer

La machine *digital optical computer* (**DOC II**) de la société Opticomp [36] est en cours de fabrication. Cette machine entièrement programmable devrait atteindre des performances de l'ordre de $0,8 \times 10^{12}$ opérations binaires par seconde. Une station sous UNIX servira de frontal. L'unité arithmétique et logique optique sur 32 bits est constituée de trois panneaux d'entrée/sorties. Le premier panneau module le laser, générant des sources digitalisées (les variables en entrée). Le second panneau est commandé par le microcode devant générer les opérations de base. Le dernier calcule les minterms de l'expression (AND-OR). Cette machine devrait fonctionner à une fréquence de 100Mhz, peut traiter des données à raison de 12,8 Gbits par secondes et peut

[3]composant actif optique temps réel capable de modifier dans le temps ou dans l'espace quelques caractéristiques (la polarisation, la phase, l'amplitude ou l'intensité) d'un faisceau lumineux, commandé par un signal électrique et/ou par l'intensité d'un autre faisceau lumineux

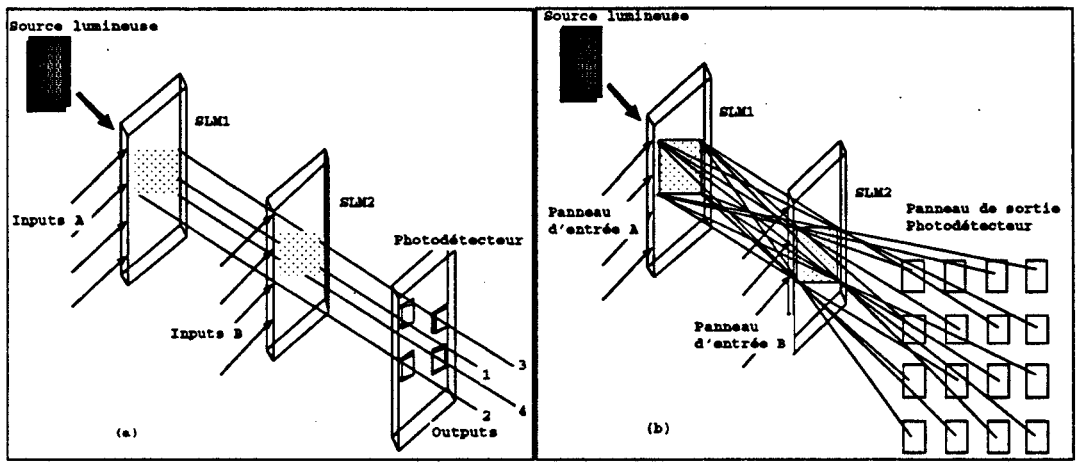


Figure 2.2 : Les processeurs Optiques : (a) implémentation classique, (b) interconnexions 3D

atteindre des performances crêtes de l'ordre du tera (10^{12}) opérations binaires par seconde pour une consommation de l'ordre de 200 à 300 watts.

La machine DOC III [36] est actuellement en cours de spécification toujours par la même société, et devrait bénéficier des interconnexions optiques 3D dont nous avons parlé plus haut. DOC III fonctionnant sur 64 bits devrait permettre d'atteindre des performances de l'ordre de 10^{14} opérations binaires par seconde, avec des primitives plus complexes comme le ou-exclusif, l'addition et la multiplication binaire.

2.1.1.2 Exemples de calcul optique

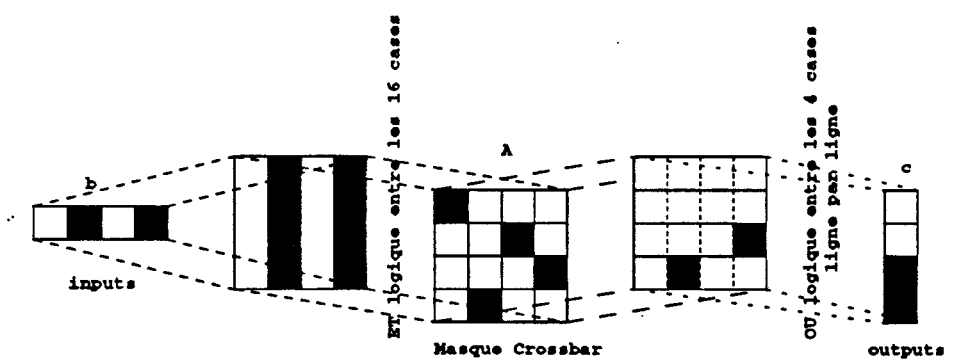


Figure 2.3 : Calcul *parallèle* du produit matrice-vecteurs

- **Produit matriciel** : la figure 2.3 montre un système issu de [68] capable d'exécuter un produit de la forme $c = A \times b$. Les N éléments du vecteur **b** en entrée génère un tableau de N diodes lasers (LEDs) désignées par *inputs* sur la figure, avec un signal binaire, le 1 binaire est représenté par une lumière d'une intensité fixe, et le 0 binaire par une intensité nulle. Le système optique suivant les entrées étend le signal de chaque entrée en colonnes verticales illuminant le masque crossbar. Le crossbar consiste en une matrice carré NxN de cellules représentant les entrées de la matrice de permutation **A**. Une entrée 0 dans A

correspond à une transmission de lumière d'intensité nulle ou une cellule opaque, et le 1 binaire de A correspond à une transmission de lumière d'une intensité fixe ou une cellule non opaque donc laissant passer de la lumière (venant évidemment des entrées *inputs*). Ici le masque crossbar peut être un SLM classique à commande externe. Enfin le dernier système optique de la chaîne collecte les faisceaux transmis par chaque ligne du masque et transfert le résultat à un tableau de photodétecteurs correspondant aux N éléments du vecteur résultat **b**. Le procédé décrit ci-dessus réalise ce type d'opération en un seul cycle d'horloge. Le produit matriciel pour des données de largeur M bits est exécuté en Mcycles d'horloge. Dès que le masque crossbar est configuré, les données digitales ou analogiques peuvent y circuler, d'une manière synchrone ou asynchrone, la bande passante n'étant limitée que par les sources et les détecteurs.

- **Architectures systoliques** : les architectures systoliques ayant pour propriété d'être rapidement reconfigurables, trouvent une bonne implémentation en optique [68] [32]. La figure 2.4 montre le type d'architecture utilisé toujours pour l'exécution de l'équation $c = A \times b$. Comme dans toute architecture systolique, nous avons besoin de dispositifs capables de mémoriser des informations et de les envoyer aux connecteurs dans la bonne séquence et au bon moment. Les éléments (a_{ij}) de la matrice intercalés par des '0' conceptuellement entrent dans le système par l'utilisation d'un SLM (qui peut être réalisé physiquement par un dispositif acousto-optique multicanaux ou un tableau de LED). Les éléments b_j du vecteur également intercalés par des 0 entrent par un autre SLM agissant comme un registre à décalage. Les signaux sont synchronisés de telle manière que l'élément a_{ij} est émis du centre de son SLM quand l'élément b_j aussi arrive au centre de son SLM. La lumière issue de la cellule a_{ij} passe ainsi à travers la cellule b_j et le résultat du produit $a_{ij}b_j$ est enregistré par le photodétecteur. Le photodétecteur agit comme un registre de décalage suivi d'une addition (shift-and-add) générant des résultats séquentiellement. Par exemple, au cours du premier cycle, $a_{11}b_1$ apparaît au milieu du photodétecteur, au second cycle, $a_{11}b_1$ sera décalé (par une commande électronique) vers le bas, l'élément a_{12} sera généré et l'élément b_2 sera aussi décalé vers le haut, au même niveau que a_{12}, le photodétecteur aura comme résultat à ce niveau la valeur $a_{11}b_1 + a_{12}b_2$. Ainsi, cycle après cycle, tous les termes de c_1 seront calculés et additionnés dans le temps, ce procédé étant identique à celui utilisé dans le cas des architectures systoliques électroniques classiques effectuant le calcul matrice-vecteur.

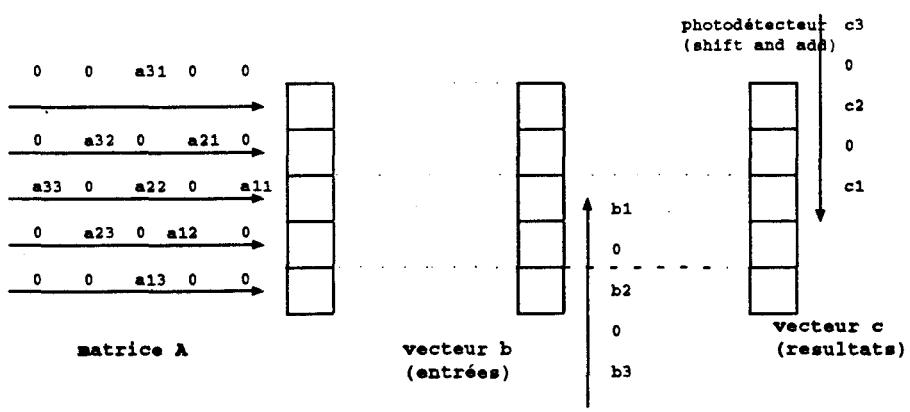


Figure 2.4 : Calcul systolique optique du produit matrice-vecteur

Comparées à l'architecture *parallèle* décrite plus haut (du produit matriciel), les architectures systoliques offrent des débits plus lents mais avec aucune perte en intensité lu-

mineuse, puisque dans l'autre cas, l'efficacité est de $1/N$ au mieux parce que $(N-1)/N$ du faisceau de chaque entrée ne traverse pas le masque crossbar (à cause de la distribution des éléments des entrées sur les colonnes du crossbar).

2.1.2 Les réseaux d'interconnexion

Il demeure que les propriétés les plus attractives de l'optique pour les systèmes massivement parallèles concernent les communications [34] si l'on se réfère aux écarts de plus en plus grands entre la puissance des PEs et les débits en communication des machines parallèles. Dans le domaine des télécommunications où les progrès de la technologie optique est la plus rapide, on dispose actuellement de fibres optiques de 0,1 db/km d'atténuation et 3,4 Gbit/s avec 50 km de distance entre les répéteurs et des tests prouvant des débits possibles à 15 Gbit/s ont été effectués [64] (*... and it seems safe to say that the laser switching rate will likely be limited by the drive electronics before it is limited by laser itself*). Bien que les besoins dans les réseaux locaux(plusieurs liens sur de petites distances) ne soient pas les même que ceux des télécommunications (peu de liens sur de très longues distances), l'impact de la technologie optique dans les réseaux locaux commence à se généraliser : Les réseaux FDDI[4] à 100 Mbit/s (avec un taux d'erreurs proche de 10^{-10}) sont de plus en plus utilisés, souvent comme réseaux fédérateurs de réseaux locaux grâce à leur débit plus important. De la même manière que dans le calcul optique, les transmissions optiques utilisent trois dispositifs :

- une source optique : un LED (composant plus simples offrant des débits modérés et utilisés par exemple dans les réseaux FDDI) ou une diode laser (composant plus difficile à mettre en oeuvre mais nécessaires si l'on veut atteindre des débits plus élevés),
- un *receiver* optique (des photodétecteurs) et
- un support de transmission (une fibre optique qui peut être *multimode* si plusieurs faisceaux de longueur d'onde différentes peuvent y circuler simultanément ou *monomode* si le diamètre de la fibre est réduite dans des proportions telles qu'un seul rayon puisse se propager, dans ce dernier cas, une diode laser est nécessaire pour l'émission)

L'évolution des réseaux locaux, métropolitains et mondiaux jusqu'aux débits de l'ordre du Gigabits par seconde devrait fournir des bandes passantes assez élevées pour nombre d'applications distribuées. D'autre part, en divisant la bande passante de la fibre optique en plusieurs petits sous-canaux par l'utilisation du multiplexage en longueur d'onde (WDM[5]), une seule fibre optique peut simuler plusieurs liens de communication. La fibre optique peut donc véhiculer plusieurs communications en parallèles pouvant aller jusqu'à 128 [36]. Cette propriété de la fibre optique a initié plusieurs études sur les réseaux d'interconnexion à base de fibres optiques.

2.1.2.1 Réseau à topologie statique à base de fibres optiques

L'utilisation de nouvelles technologies pour les communications a fait apparaître la nécessité d'utiliser de nouveaux modes de commutation autres que ceux fondés sur les techniques classiques comme la commutation de circuits ou le store and forward. VETTER et DU [82] ont proposé un modèle original de commutation à base de fibres optiques structuré en couches : la couche physique désignant les liens physiques à base de fibres optiques interconnectant des

[4] Fiber Distributed Data Interface
[5] wavelength-division multiplexing

noeuds physiques (cette terminaison n'a aucun rapport avec la couche physique du modèle OSI) et pouvant simuler diverses topologies suivant les exigences de l'application, la couche système et la couche application.

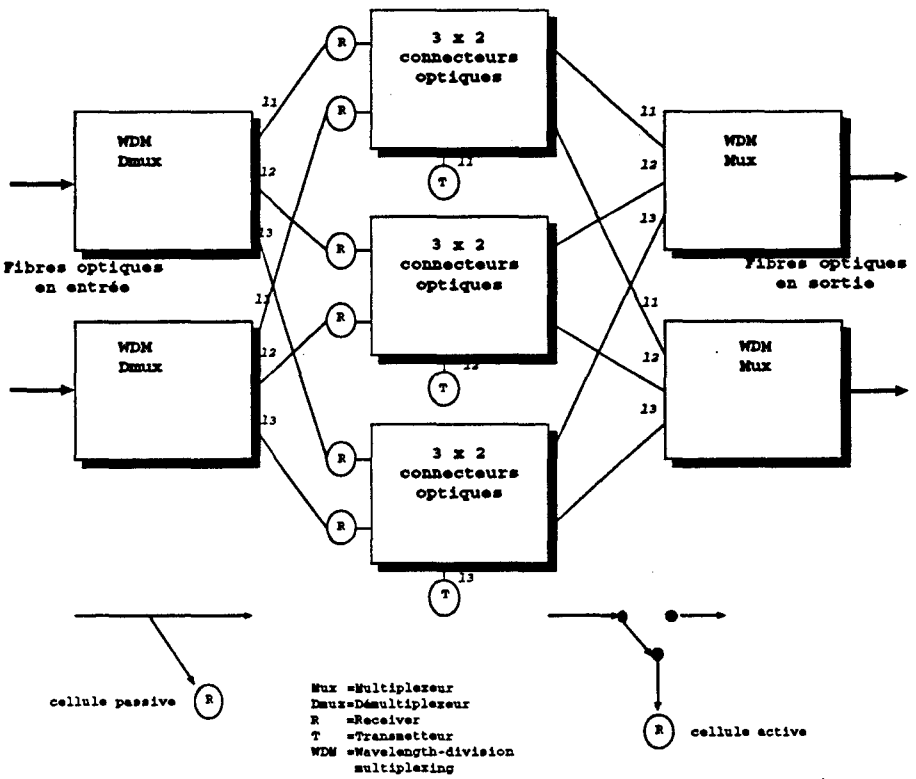


Figure 2.5 : Un noeud dans le réseau optique WDM, chaque fibre optique possède 3 canaux *l1*, *l2* et *l3*

2.1.2.2 La couche physique : le réseau virtuel

Constituée de liaisons point-à-point à base de fibres optiques entre les noeuds, la couche physique permet la connexion directe de tout noeud à un autre qui n'est pas physiquement adjacent sans conversions optique/électronique. Ce type de liaisons est rendu possible grâce à l'utilisation de la technique WDM et l'utilisations de connecteurs à l'intérieur de chaque noeud. Un signal en entrée caractérisé par sa longueur d'onde (λ) arrivant au noeud peut être aiguillé directement vers un autre canal en utilisant des connecteurs. La figure 2.5 montre l'exemple d'un noeud dans ce système, relié à deux fibres en entrée et deux fibres en sortie avec trois longueurs d'ondes différentes. Chaque fibre en entrée est reliée à un démultiplexeur en longueurs d'onde qui sépare les différentes longueurs d'ondes en canaux discrets, chacun étant identifié par une longueur d'onde donnée $\lambda_1, \lambda_2, \lambda_3$. Ces différents canaux sont reliés à plusieurs connecteurs 3x2 en passant d'abord par des cellules notées **R** sur la figure. Ces cellules peuvent être passives délivrant une fraction du signal au récepteur local et laissant passer l'autre fraction, ou être actives et jouer dans ce cas le rôle d'un aiguilleur c'est-à-dire réceptionner tout le signal ou laisser passer le signal vers le noeud suivant. Pour transmettre une information, le noeud local utilise un canal donné ($\lambda_1, \lambda_2, \lambda_3$) et utilise une des entrées des connecteurs (T). Des *connexions*

virtuelles seront ainsi créées entre les noeuds, permettant ainsi à deux noeuds physiquement non adjacents d'être virtuellement adjacents dans le réseau WDM.

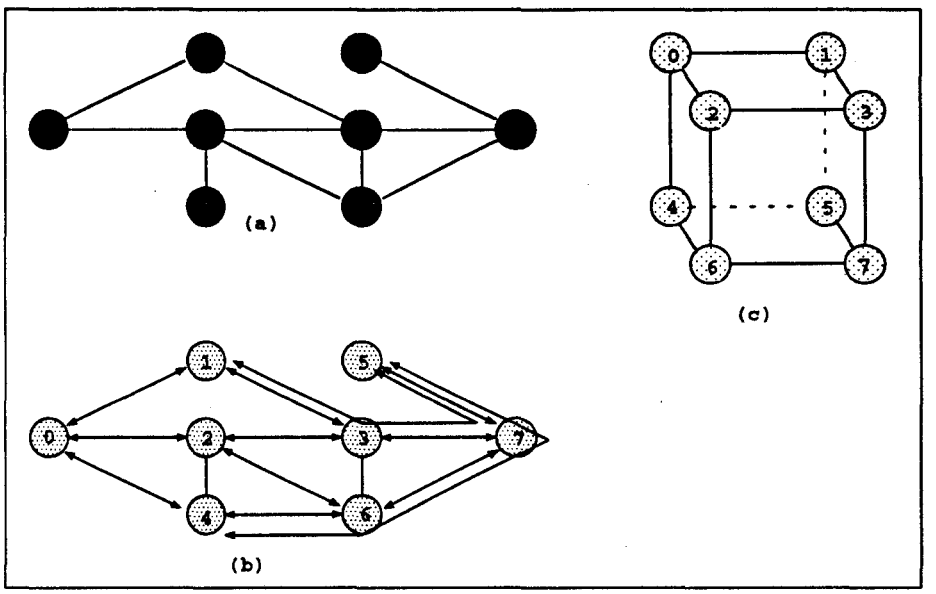


Figure 2.6 : Un graphe avec des liens physiques(a) et la projection (b) d'un hypercube (c) sur le réseau physique

2.1.2.3 La couche système et la couche application

C'est au niveau de la couche système que le réseau virtuel est généré en commandant chaque noeud. Diverses topologies peuvent ainsi être simulées par commande du réseau physique suivant les exigences de l'application.

La figure 2.6(c) montre une topologie virtuelle hypercube créée à partir d'une topologie d'un graphe quelconque de huit noeuds, grâce au multiplexage en longueur d'ondes (WDM) sur les fibres optiques. Quelques liaisons virtuelles créées occupent plusieurs liens physiques comme la connexion entre les noeuds 4 et 5, alors que certains liens physiques sont ignorés. Les noeuds 4 et 5 sont donc virtuellement adjacents.

2.1.2.4 Crossbar optique

La réalisation d'un crossbar avec l'utilisation de l'optique est conceptuellement assez simple [68]. Puisqu'il s'agit de réaliser un dispositif permettant d'effectuer toute permutation de N éléments en entrée vers une sortie, les deux exemples de calculs matriciels décrits ci-haut peuvent être utilisés. La méthode1 par exemple permet toute permutation à de très hauts débits seulement en commandant le panneau réalisant le masque crossbar.

L'inconvenient majeur des crossbars optique est leur temps de reconfiguration plus élevé que celui des composants électroniques. En effet l'électronique permet des temps de reconfiguration rapides de l'ordre de 50 à 100 nanosecondes pour des débits de 10 à 100 mégabits (par connecteur) par seconde alors que les temps de reconfiguration pour l'optique sont actuellement de l'ordre de la microseconde mais offre des débits dépassant le gigabits par seconde. Il en ressort

évidemment que les crossbars optiques trouvent un intérêt dans des domaines où chaque reconfiguration est suivie d'un transfert d'importants paquets de données comme dans le traitement d'images.

2.1.3 Conclusion

De nombreuses études sont actuellement en cours dans le domaine de l'optique pour essentiellement réduire les temps de commande (les reconfigurations sont trop lentes actuellement). Cela implique surtout la réduction des temps de conversions des signaux électroniques vers l'optique et vice-versa, puis la recherche de moyens plus efficaces pour le contrôle des composants.

2.2 Les Hyperfréquences

2.2.1 Introduction

La figure 2.7 issue de [29] montre les subdivisions du spectre électromagnétique. Le terme *hyperfréquences* désigne une bande de fréquences située entre environ 300 Mhz et 300 Ghz. Ce terme permet surtout de localiser le domaine des hyperfréquences entre celui des fréquences plus basses employées pour la diffusion de la radio et de la télévision et celui des fréquences plus élevées caractérisant le domaine des infrarouges.

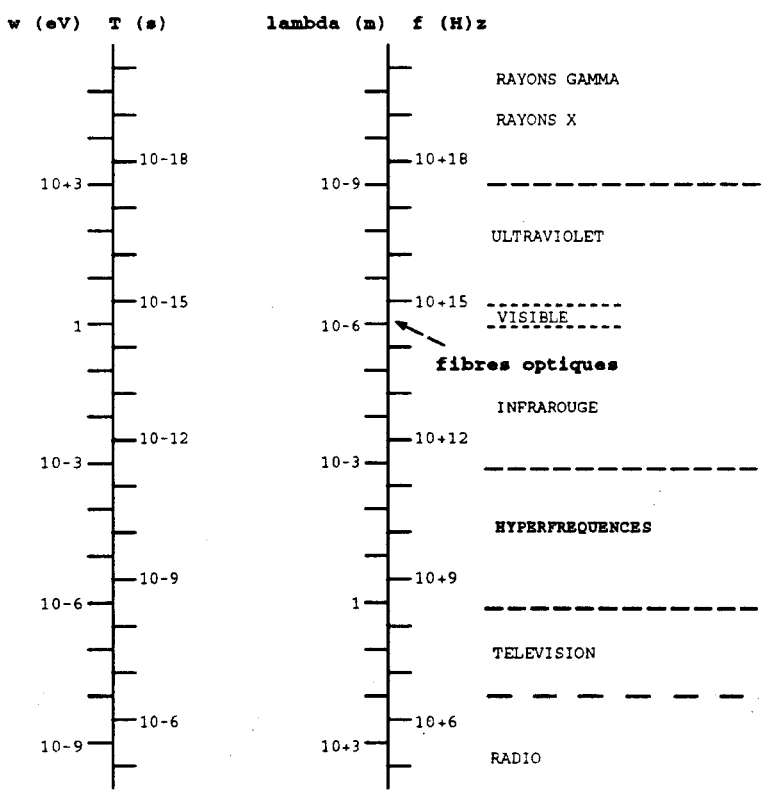


Figure 2.7 : Subdivision du spectre électromagnétique

De plus, pour une onde électromagnétique de fréquence f se déplaçant dans l'espace à la vitesse de la lumière $c_0 = 2,997925...10^8 m/s$, la longueur d'onde, définie par $\lambda = c_0/f$ se situe entre 1m et 1mm pour un signal hyperfréquence. On peut ainsi parler d'ondes *décimétriques,centimétriques* et *millimétriques* toujours pour désigner les hyperfréquences. Un autre synonyme est le terme *micro-ondes* qui n'est qu'une traduction de l'anglais "microwaves". Ce terme dénote tout simplement la petitesse des longueurs d'ondes par rapport à celles dont on fait usage en radiodiffusion, et ne veut en aucun cas désigner des longueurs d'ondes de l'ordre du micron.

Parmi les propriétés des hyperfréquences on peut noter :

- Une grande bande passante. Le débit d'information qui peut être transmis sur un canal étant directement proportionnel à la bande passante de celui-ci, un calcul simple[6] montre que la bande s'étendant de 300 Mhz à 300 Ghz permet d'acheminer 10^3 fois plus d'information en un temps donné que toutes les bandes situées en dessous de 300Mhz. Cette propriété étant directement liée à la fréquence du signal, des quantités encore plus considérables d'information pourraient selon cet argument, être transmises dans les bandes des infrarouges et du visible, dans les systèmes à laser et à fibres optiques.
- La possibilité d'intégration des composants et l'utilisation de guides d'ondes. En effet, la longueur d'onde d'un signal hyperfréquences est du même ordre de grandeur que les dimensions des éléments employés pour le produire. Un choix judicieux des longueurs d'ondes à utiliser permet de satisfaire l'encombrement physique visé pour un domaine d'application donnée (réseaux locaux, réseaux d'interconnexions etc.).

2.2.2 Les hyperfréquences et l'optique

Les avantages (d'ordre technologiques) liés à l'utilisation de l'optique dans les transmissions sont :

- **un haut débit** : le débit d'informations qui peut être transmis sur un canal étant directement proportionnel à la bande passante de celui-ci, il est évident que l'utilisation de l'optique permet d'obtenir de plus grands débits par rapport aux hyperfréquences.
- **faible encombrement** : avec l'utilisation de fibres optiques monomodes, et des guides plus souples.
- de très faibles pertes lors des liaisons importantes (quelques kilomètres) et une haute stabilité fréquencielle des oscillateurs.
- **un faible coût** lié aux progrès rapides de la technologie optique (avec le développement des réseaux câblés).

La technologie hyperfréquence pallie sensiblement les limitations de l'optique, et présente donc les avantages suivants :

- une connectique plus aisée à réaliser, ce qui n'est pas le cas dans le domaine de l'optique,
- les sources hyperfréquences sont plus facilement accordables en fréquence, ce qui est pratiquement impossible en optique (nous utiliserons cet atout lors de la spécification du protocole de communication en hyperfréquence), donc la possibilité de réaliser plus aisément des liaisons *multicanaux*en ce sens où la bande passante sera divisée en plusieurs canaux indépendants.

[6] $(3^{11} - 3^8)/3^8 = 999$

- un bon rapport S/B[7] comparé à la technologie optique. En effet l'utilisation de l'optique est sujette au fort bruit généré par les composants, typiquement 20 db pour le récepteur alors qu'il est de 3 à 4 db pour un récepteur identique en hyperfréquence.

2.2.3 Applications dans les réseaux locaux et large bande

Tout comme dans le cas de l'optique, l'utilisation des communications en hyperfréquences a été d'abord développée dans les réseaux mondiaux (WAN[8]) et les réseaux locaux (LAN[9]).

Grâce aux propriétés leur permettant traverser l'ionosphère sans subir de distorsion (l'ionosphère ayant une fréquence de coupure de l'ordre de 10-40Mhz), les hyperfréquences sont très largement utilisées dans les WAN, essentiellement pour les communications par satellites, ainsi que pour maintenir le contact avec les sondes et expéditions spatiales. Citons comme exemple des applications, le premier satellite de télécommunications *telstar* et le premier satellite géostationnaire *Early Bird*.

Dans le domaine de liaisons sans fils, deux types de technologies sont envisagés [22] [18] : les réseaux infrarouge (à partir de 10^{12} Hz) qui sont inopérants lorsqu'une cloison ou un obstacle quelconque sépare deux stations, et les liaisons sur ondes radio et micro-ondes (de 10Khz à 300Ghz). Les liaisons hyperfréquences dans les réseaux locaux commencent à se développer, puisque répondant aux exigeances des entreprises changeant souvent de site. Le problème essentiel pour cette application reste celui des normes internationales (l'exemple de la norme *Dect*[10] de Philips, Alcatel, Ericsson, Siemens AG et Nokia permettant des communications sans fils à partir de téléphones sans fils connecté à un central téléphonique privé[11] dans la bande de fréquence de 1880 à 1900 Mhz), puisque cette bande est utilisée par les *radars de détection* des organismes nationaux (l'armée, la police...). Parmi les réseaux locaux commercialisés, nous pouvons citer *radiolink* et *Wave Lan* de la société NCR, utilisant des liaisons point-à-point pour les liaisons entre machines.

2.2.4 Les communications en hyperfréquences

Pour illustrer les techniques utilisées pour les communications à courte portée en hyperfréquences, nous allons décrire succintement dans ce qui suit (comme dans le cas de la technologie optique) l'idée de base pour la transmission et pour la réception des informations sur un guide d'ondes.

2.2.4.1 La transmission

Le circuit d'émission représenté sur les figures 2.8 (a) et 2.9 (a) est composé d'un ensemble de circuits électroniques dont un oscillateur à commande électronique, un mélangeur, un filtre à ondes de surfaces, un démodulateur et quelques amplificateurs de signaux.

- Un oscillateur à commande en tension (VCO[12]) est un composant qui génère un signal périodique dont la fréquence est une fonction instantannée de la tension de commande. Actuellement, nous disposons de deux types de VCO : VCO à transistors bande X (10 Ghz)

[7] Signal sur Bruit
[8] Wide Area Network
[9] Local Area Network
[10] Digital European Cordless Telecommunications
[11] PABX : Private Branch Exchange
[12] Voltage Control Oscillator

et VCO à cavité bande V (60 Ghz). Il est de plus nécessaire que les VCO possèdent la plus large bande d'accord (bande de fréquences sur laquelle le VCO peut osciller) possible. Il existe des VCO en gamme millimétriques (30 Ghz - 100 Ghz) permettant un accord électrique sur $\pm 5\%$ de la fréquence centrale ; ce qui nous donne 3 Ghz autour de la fréquence centrale (60 Ghz). Nous obtenons de ce fait une bande passante de $B_{VCO} = 6Ghz$ avec des VCO à cavité bande V 60 Ghz.

- L'opération de modulation a pour but de transformer un signal numérique en bande de base en un signal numérique passe-bande. Cette transposition permet de juxtaposer ces signaux dans un même support de transmission, chaque signal possédant sa bande de fréquence propre. La PLL[13] permet de stabiliser la source émettrice.

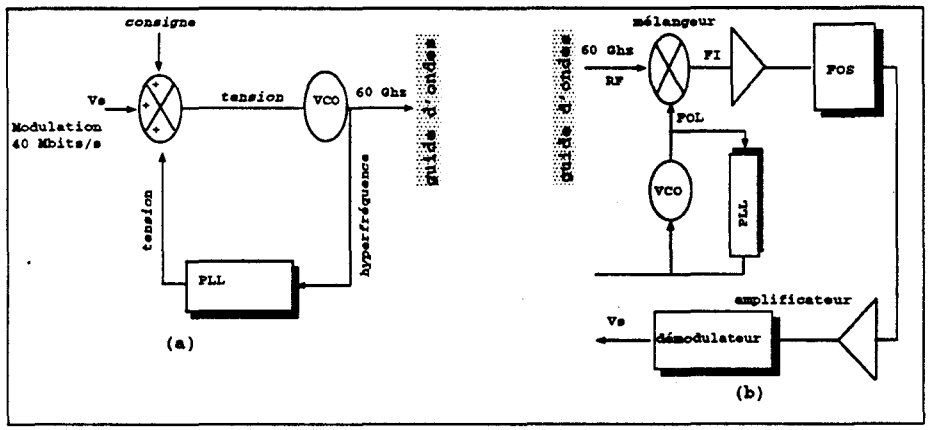


Figure 2.8 : Modulation à basse fréquence : unité d'émission (a) et unité de réception (b)

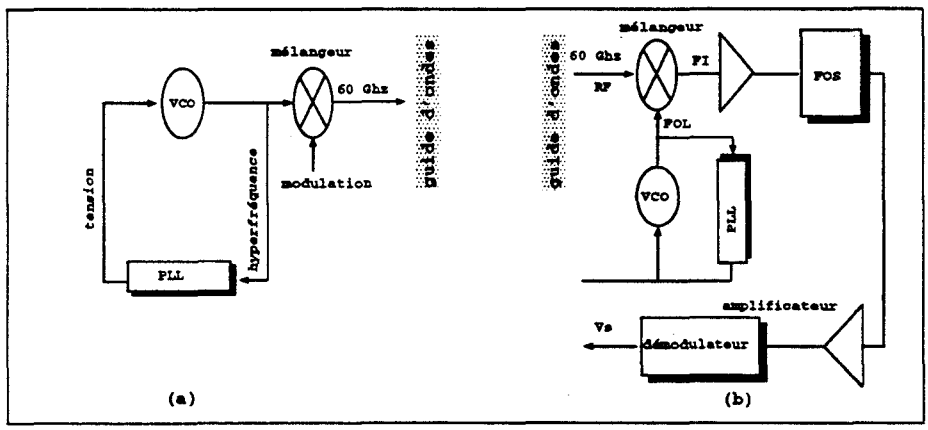


Figure 2.9 : Modulation à haute fréquence : unité d'émission (a) et unité de réception (b)

Pour des transmissions à faibles débits (débits inférieurs à 100 Mbps), le signal de modulation est appliqué à l'entrée d'un sommateur en tension alors qu'à haut débit (débits supérieurs à 100 Mbps), la commande est appliquée directement à un mélangeur.

[13] Phase-locked loop

2.2.4.2 La réception

En plus des éléments cités ci-dessus nécessaires à la transmission, un mélangeur, un filtre à ondes de surface (FOS) et un démodulateur ont été rajoutés (cf figure 2.8 (b) et figure 2.9 (b)) pour la réception des signaux.

- Le mélangeur est nécessaire pour réaliser une conversion de fréquences. Il transpose le signal incident de fréquence RF en une fréquence plus basse FI (fréquence intermédiaire) par mélange avec le signal local de fréquence FOL produit par un oscillateur local. le signal en bande de base FI est destiné au filtre (décrit ci-dessous). L'opération (FI = RF - FOL) est réalisée grâce à un élément non linéaire (un mélangeur pouvant être constitué d'une diode ou d'un transistor à effet de champ).
- Un filtre à ondes de surface permet d'isoler uniquement une bande de fréquences donnée, pour sélectionner un canal donné.

2.2.4.3 Le support de transmission

Le support le plus utilisé dans le domaine des hyperfréquences est sans doute l'espace vu les domaines d'applications actuels cette technologie. Cependant, pour des liaisons à courte portée, divers supports peuvent être utilisés pour véhiculer l'onde hyperfréquence ; parmi ces supports nous pouvons citer:

- les guides d'ondes qui ne sont que des supports métalliques creux à l'intérieur desquels se déplacent les ondes hyperfréquences. Leurs dimensions physiques étant proportionnelles à la longueur d'ondes des signaux, les guides d'ondes offrent un avantage réel puisqu'ils sont facilement adaptables au domaine d'application (il suffit de choisir convenablement la fréquence d'utilisation pour réduire ou augmenter la dimension des guides d'ondes). Les guides d'ondes peuvent aussi être réaliés en plastique métallisé (avec une couche de métal à l'intérieur).
- les câbles coaxiaux dont les avantages restent la souplesse des câbles, cependant ils restent assez coûteux et présentent des pertes plus importantes.
- les lignes de transmission (microstrip, coplanaires, triplaque) dont l'avantage est sans doute une réalisation plus facile, mais offrant cepandant des performances moyennes avec les phénomènes de rayonnement à partir des fréquences de fonctionnement de 60 Ghz.

2.3 Conclusion

Dans le but d'explorer les technologies futures pour l'ordinateur, ce chapitre a montré d'abord l'énorme potentialité de l'optique autant dans le domaine du calcul (processeurs ou opérateurs optiques) que dans les communications. Il reste néanmoins que les composants "optiques" (émetteurs, récepteurs, modulateurs etc.) demeurent très chers, la plupart des projets tendant plutôt à montrer la faisabilité de systèmes optiques que la commercialisation effective. La technologie hyperfréquences a aussi été introduite, insistant surtout sur les domaines d'applications d'aujourd'hui, domaine qui est celui des réseaux large bande, métropolitains et locaux. Le chapitre suivant explore un autre domaine possible d'applications des hyperfréquences, celui des réseaux d'interconnexions pour machines parallèles, puis présente un modèle de base pour une architecture parallèle construite à partir de guides d'ondes.

Chapitre 3

Les communications en hyperfréquences

Les ondes radio-électriques présentées dans le chapitre précédent servent de support de transmission dans de nombreux domaines. Dans ce chapitre, nous proposons un modèle de communication dans les réseaux d'interconnexion des machines parallèles, utilisant les ondes radio-électriques.

Après une brève étude des contraintes liées à l'utilisation des ondes radio-électriques qui justifient l'utilisation de guides d'ondes, nous décrivons le protocole de communication de deux noeuds sur un guide d'ondes. La dernière partie du chapitre présente les tests et réalisations électroniques effectués pour valider notre modèle.

3.1 Les contraintes

Dans le cas de réseaux locaux sans fil, les machines utilisent des antennes d'émission/réception pour communiquer entre elles. De plus, le nombre de machines connectées ne dépasse que très rarement la trentaine. Dans le cas de réseaux d'interconnexion pour machines massivement parallèles avec un nombre d'éléments de calcul pouvant dépasser le millier, ceux-ci devront non seulement se partager la bande passante disponible mais aussi être contenus dans un espace physique 3D plus réduit. Plusieurs problèmes devront donc être résolus, notamment les problèmes d'encombrement physique et les problèmes d'adressage des éléments de calcul.

3.1.1 Les dimensions physiques acceptables

L'utilisation des antennes comme dans le cas des réseaux locaux est inapplicable ici à cause des contraintes physiques à respecter quand le domaine ciblé est celui des machines massivement parallèles. Cependant, comme nous l'avons souligné au chapitre précédent, la longueur d'onde d'un signal hyperfréquence est du même ordre de grandeur que les dimensions des éléments employés pour le produire et le traiter, et cette propriété guidera notre choix quant à la fréquence

à utiliser. La figure 3.1 montre par exemple l'évolution des dimensions d'un guide d'ondes en fonction de la fréquence d'utilisation.

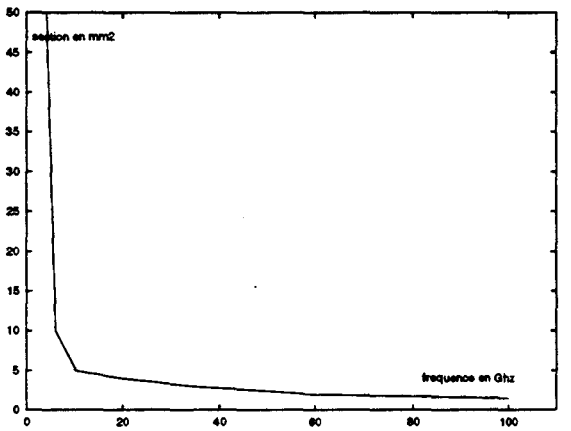


Figure 3.1 : Evolution des dimensions extérieures d'un guide d'ondes en fonction de la fréquence

3.1.2 Limiter les interférences entre les communications

Les interférences dans le cas des réseaux locaux est réduite grâce au faible nombre de machines (les sites) mis en jeux et aux distances assez grandes entre les sites. Pour un nombre assez important de noeuds, la bande passante est divisée par autant de noeuds, un canal plus étroit étant réservé à un noeud. Il est donc plus difficile de limiter avec précision le champs des antennes, augmentant les difficultés d'adressage des noeuds. Pour une meilleure fiabilité des transmissions, il est nécessaire de limiter au maximum les interférences avec l'environnement (problème de bruit), interférences qui croissent avec le nombre d'éléments communicants. Une solution consiste à **mettre en oeuvre des communications par groupes de noeuds**, chaque groupe utilisant évidemment toute la bande passante, en évitant bien sûr les interférences entre groupes disjoints.

3.1.3 Utilisation de guides d'ondes

Des deux contraintes observées ci-dessus, il en ressort que nous devons :

- utiliser de petits composants donc des fréquences élevées
- mettre en oeuvre des communications par groupes de noeuds.

Les guides d'ondes permettent de limiter le champs de propagation des ondes électromagnétiques à l'intérieur d'un support, et de ce fait créer des groupes de noeuds communicants. D'autre part, les dimensions de guides étant proportionnelles à la longueur d'onde des signaux produits, l'observation de la courbe de la figure 3.1 nous conduit au choix des guides pouvant véhiculer des signaux de l'ordre de 60-90Ghz (ondes millimétriques). Les guides à cette fréquence ont des dimensions compatibles avec l'encombrement visé. Un autre avantage inhérent à l'utilisation de guides est l'absence de normes auxquelles sont soumis les concepteurs quand les ondes sont propagées librement dans l'espace.

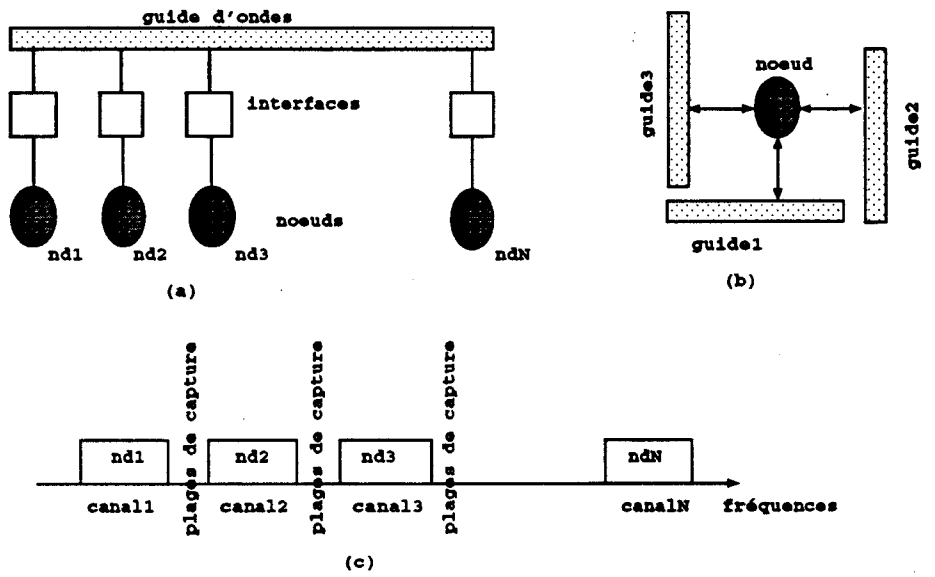


Figure 3.2 : Guide d'ondes interconnectant N noeuds (a), un noeud relié à 3 guides d'ondes (b) et la division de la bande passante des émetteurs/récepteurs (connectés au guide) en canaux (c)

3.1.4 Utilisation de la modulation de fréquences

Des études (historiques) ont été faites sur les différentes manières de coder l'information en utilisant des ondes. Dans le domaine de l'optique, l'étude faite par P.E. GREEN dans [34] a montré que deux techniques peuvent être utilisées pour l'adressage des éléments connectés à un support comme une fibre optique ou un guide d'ondes :

- la modulation de fréquences (ou de longueur d'ondes : WDMA[1]),
- le multiplexage temporel (time slot : TDMA[2]).

Le multiplexage temporel impose plusieurs *slots* par temps-bit, ce qui implique la mise en oeuvre de dispositifs matériels plus rapides que le temps de transmission d'un bit pour synchroniser les signaux à la réception. De ce fait, la technique WDMA semble actuellement être la plus recommandée pour les transmissions par ondes dans un support.

Si nous considérons des circuits d'émission et de réception (E/R) d'une bande passante de BWHz (c'est-à-dire que chaque noeud est connecté à un circuit E/R pouvant émettre et recevoir des informations sur tout le champs de fréquences couvert par BW) et des canaux de BW_i Hz par noeud connecté au guide d'ondes, **un guide d'ondes peut donc interconnecter $N = BW/BW_i$ noeuds, chaque noeud étant identifié par sa fréquence propre** (cf figure 3.2). Chaque unité connectée au guide d'ondes reçoit donc les informations sur sa fréquence propre appelée fréquence de réception. Pour une émission de messages, la transmission se fera évidemment dans la fréquence (propre) de réception du destinataire.

Les guides d'ondes choisis transportent ainsi des signaux dont la porteuse est choisie autour de 60Ghz (il n'est pas nécessaire d'aller au delà de cette valeur puisque l'observation de la courbe

[1] Wavelength-division multiple access
[2] Time-division multiple acces

d'évolution des dimensions des guides d'ondes montre qu'aucun gain significatif n'est constaté après 60Ghz) et l'adressage des unités connectées au guide est réalisé grâce à la modulation de fréquence. A l'intérieur d'un canal BW_i, la transmission est aussi effectuée par modulation de fréquence.

3.2 Communications dans un guide d'ondes

La section précédente nous a conduit à l'utilisation de guides d'ondes comme support de base pour les communications. Une communication mettant en relation un émetteur et un destinataire, et une liaison se décrivant en termes d'actions élémentaires (validation de buffer, activation de signal de chargement de registre...), le *protocole* de la liaison décrit l'enchaînement de ces actions.

Comme les unités connectées à un guide ont un fonctionnement indépendant les unes des autres, plusieurs unités émettrices peuvent tenter de prendre la ressource (canal du destinataire) simultanément. Cette situation conduit à un état de *contention*. D'autre part si deux ou plusieurs unités émettrices utilisent simultanément la ressource, on obtient un cas de *collision* conduisant inévitablement à une perte de l'information. Beaucoup de techniques ont été mises oeuvre pour résoudre contention et collision dans les systèmes distribués, c'est-à-dire des systèmes dépourvus de tout dispositif central de coordination et dont les utilisateurs sont en compétition pour utiliser le canal unique qu'ils doivent se partager.

En supposant qu'il y a N utilisateurs connectés au canal, deux problèmes devront être résolus :

- parmi les N utilisateurs, identifier ceux qui désirent accéder au canal, puis
- allouer le canal à exactement l'un des utilisateurs prêts s'il y en a.

La principale difficulté reste le fait qu'ici, le canal de transmission est aussi le seul moyen de communication pour la coordination des usagers.

3.2.1 Protocoles de communication distribués

3.2.1.1 Les protocoles ALOHA

Le protocole ALOHA [1] considéré comme l'ancêtre des protocoles à contention est basé sur une idée simple : laisser les utilisateurs transmettre en toute liberté. Il y aura bien entendu des collisions et les trames qui en seront victimes seront détruites. Cependant, de par le fonctionnement du réseau de diffusion, un émetteur peut toujours savoir en écoutant le réseau si sa trame a été détruite ou non. L'émetteur ayant constaté que sa trame a été détruite observe un délai d'attente avant de l'émettre à nouveau. Le meilleur taux d'utilisation du réseau par cette technique est de 1/(2e)=18% (avec e=2,71; cf [78] pp.191 pour la démonstration) de la bande passante du réseau. Une amélioration de ce protocole est appelée ALOHA discrétisé. Dans ce dernier protocole, le temps est divisé en intervalles répétitifs de durée constante. Toute station devra attendre le début d'un intervalle de temps avant d'émettre. Les performances sont nettement meilleures que celles du pur ALOHA puiqu'on peut arriver à un taux d'utilisation de 1/e=37% de la bande passante du réseau (cf figure 3.3).

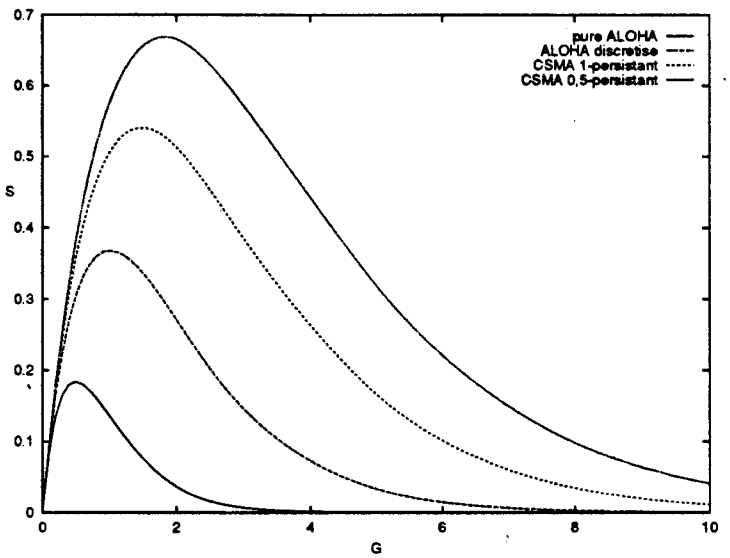


Figure 3.3 : Comparaison de l'efficacité des réseaux : S (nombre de trames transmises sans collision par intervalle de *temps de trame* (temps de transmission d'une trame) en fonction de G (nombre de trames transmises dans le réseau par intervalle de *temps de trame*)

3.2.1.2 Les protocoles CSMA

D'autres améliorations du protocole ALOHA sont issues de l'observation qu'une station peut s'enquérir de ce que font les autres stations et adapter son comportement en conséquence. Les protocoles reposant sur ce principe sont appelés *protocoles à détection de porteuse* (*carrier sense protocole*). Nous présentons ci-dessous quelques exemples de cette famille appelée CSMA pour *Carrier Sense Multiple Access.*

Le CSMA 1-persistant : lorsqu'une station décide d'émettre une trame, elle écoute le canal et maintient cette écoute jusqu'à ce que le canal devienne libre. La trame peut être émise dès que le canal est libre ; si une collision se produit, la station reprend le protocole après un temps aléatoire. Ce protocole est appelé 1-persistant parce que dès que le canal est libre, la station décide d'émettre avec une probabilité égale à 1.

Le CSMA non persistant : avant d'émettre, la station écoute le canal ; s'il est disponible, elle émet sa trame, mais dans le cas contraire, elle ne reste pas en écoute permanente et ne cherche pas à prendre possession du canal dès qu'il est libre, elle reprend tout simplement le protocole après avoir observé un temps aléatoire comme dans le cas d'une collision.

Le CSMA p-persistant : dans ce cas-ci, quand une station décide d'émettre, elle écoute le canal, si celui-ci est libre, elle transmet sa trame avec une probabilité égale à p. Elle attendra donc l'intervalle de temps suivant avec une probabilité égale à q=1-p. Si lors de ce deuxième intervalle de temps le canal est libre, les probabilités sont les mêmes quant au choix de l'émission ou de l'attente. Si la station trouve le canal occupé (même après une attente volontaire de probabilité q), elle reprend tout le protocole après un temps aléatoire.

le CSMA/CA (CA pour Collision Avoidance) : le principe essentiel est d'éviter les collisions par un test de transmission établissant le droit d'envoyer les données.

Le CSMA/CD (CD pour Collision Detection) : de tous les protocoles présentés ci-dessus, même dans le cas d'une collision, toute la trame est quand même transmise. L'amélioration ici vient du fait que toute station constatant une collision cesse immédiatement d'émettre, gagnant ainsi du temps. Dans ce cas aussi, le protocole est repris après un temps aléatoire.

Si le CSMA/CD exhibe de meilleures performances par rapport aux autres CSMA, il ne résoud pas le problème essentiel du déterminisme, ni de l'équité entre plusieurs utilisateurs, puisque ce protocole prend en compte l'aléatoire. Plusieurs schémas de résolution de conflits ont été proposés et évalués. Les mesures considérées pour l'évaluation des performances sont le débit, la latence et la capacité du canal de communication, cette dernière mesure exprimant le rapport entre le temps utilisé pour la résolution de conflits et le temps utilisé pour la communication effective dans le canal. Ainsi, pour résoudre l'indéterminisme dans la sélection d'un utilisateur, CAPETANAKIS [9] a proposé un algorithme basée sur une procédure de recherche arborescente dont l'idée est la suivante. Supposons que chaque utilisateur correspond à une feuille d'un arbre binaire, et que tous les utilisateurs prêts à émettre commencent simultanément à émettre à un instant précis. En accordant le droit d'émettre à tous les noeuds d'un arbre (ou d'un sous-arbre), l'instant d'après, on ne peut observer que trois situations possibles (libre, succès, collision) sur le canal de transmission indiquant le nombre d'utilisateurs prêts dans l'arbre (ou sous-arbre), soit zéro, un, ou plus d'un. S'il n'y avait qu'un seul utilisateur prêt à émettre, il aura transmis ses données avec succès, dans le cas où il y en avait deux ou plus, on aurait constaté un état de collision, l'arbre serait ainsi divisé en deux sous-arbres, et l'algorithme serait repris uniquement sur un seul sous-arbre d'abord puis l'autre ensuite. Dans le plus mauvais des cas, l'algorithme décrit ci-dessus permet de sélectionner un utilisateur parmis k en $(2 \times \log k)$ τ, τ désignant le temps nécessaire à la détection d'une collision sur le canal.

3.2.1.3 Les protocoles sans collision

Une autre famille de protocoles destinés aux systèmes distribués concerne les *protocoles sans collisions* dont une description détaillée peut être trouvée dans [78]. L'idée principale des protocoles de cette famille réside dans la division de la bande passante en périodes de contentions et en périodes de transmissions. Pendant la période de contention, chaque station pendant un intervalle de temps fixe signale si elle a une trame à transmettre ou pas (en positionnant un bit à 1 ou à 0). Pendant la période de transmission, les stations ayant pris connaissance pendant la période de contention des intentions des autres, attendent leur tour pour transmettre. Ceci permet essentiellement d'éviter les pénalités engendrées par la gestion des collisions très fréquentes à forte charge dans les cas précédents.

3.2.1.4 Etude comparative

En considérant les protocoles cités ci-dessus, deux grandes classes peuvent être observées : les protocoles CSMA appelés *protocoles à contention* et les protocoles fondés sur l'absence de collision.

Les protocoles à contention sont préférables lorsque la charge du réseau est faible puisque les temps d'attente sont moindres. Alors que dans le cas des protocoles sans collision, même à faible charge, la gestion des collisions est toujours effectuée, augmentant les délais d'attente pour l'accès au réseau. Mais au fur et à mesure que la charge du canal augmente, l'efficacité des protocoles à contention devient de moins en moins évidente, car les périodes utilisées pour la résolution des situations de contention deviennent de plus en plus fréquentes.

A contrario, dans le cas des protocoles sans collision, l'efficacité du réseau est maximale à forte charge puisque les transmissions sont fréquentes, réduisant ainsi l'overhead dû à la gestion des contentions.

Dans le but de concevoir un protocole ayant les avantages des deux types étudiés ci-dessus, les protocoles dits à *contention limitée* utilisent les deux techniques suivant la charge du réseau. Quand la charge est faible un protocole à contention sera utilisé, et à forte charge l'autre, exhibant de meilleures performances, sera utilisé. Ces protocoles divisent donc l'ensemble des stations en petits groupes (pour réduire la charge) pas nécessairement disjoints, en affectant des périodes de contention à chaque groupe. Ainsi, pendant la période de contention du groupe i, seules les stations appartenant à ce groupe peuvent entrer en compétition pour l'occupation du canal. Ainsi, un protocole à collision est utilisé à l'intérieur des groupes et un protocole sans collision ordonnance les groupes de façon cyclique. Si le réseau est divisé en groupes d'une seule station, nous rejoignons la classe des protocoles sans collision, dans le cas où le réseau est réduit à un seul groupe, nous retrouvons les protocoles à collision. Un choix judicieux lors de l'affectation des stations aux groupes doit donc être fait (dynamiquement si possible) si l'on veut avoir un bon rendement de la bande passante du réseau.

3.2.2 Protocoles de communication entre deux noeuds connectés à un guide d'ondes

Les protocoles décrits ci-dessus ont tous une particularité : ils sont conçus pour des réseaux dits *à diffusion*. Dans ce type de réseaux contrairement aux réseaux multipoints, toutes les unités sont connectées à un seul canal (un bus unique par exemple) sur lequel elles peuvent émettre et recevoir des informations. Dans notre cas où il s'agit d'un guide d'ondes divisé en plusieurs canaux, la ressource à partager demeure un canal dans la mesure où plusieurs unités peuvent décider d'émettre simultanément vers un même destinataire. Pour une distribution uniforme de requêtes vers les destinataires, les protocoles étudiés ci-dessus peuvent être appliqués sur chaque canal avec une probabilité d'arrivée des requêtes de p/N au lieu de p dans le cas où il n'existerait qu'un seul canal auquel seraient connectées N unités.

La probabilité d'occupation d'un canal passant de p à p/N, nous pouvons utiliser les protocoles à collision pour résoudre la contention lors de l'accès à un canal sur le guide d'onde. Cependant plusieurs interrogations demeurent, notamment si nous supposons que toutes les communications sont effectuées dans la fréquence propre du destinataire (par opposition à l'expéditeur qui a initié la communication), comment signaler aux autres unités connectées au guide d'ondes que l'expéditeur est occupé, puisque son canal demeure libre (l'unité d'émission/réception (E/R) ne pouvant émettre et recevoir à la fois). Nous décrivons ci-dessous quelques protocoles pouvant être appliqués pour le transfert d'un paquet entre noeuds connectés à un guide d'ondes, en considérant toutefois quelques points essentiels :

- *tout noeud connecté à un guide d'ondes est identifié par une seule fréquence de réception fi matérialisée physiquement par une unité d'émission/réception pouvant écouter ou (exclusif) émettre sur toutes les N fréquences du guide d'ondes.*
- *tout noeud connecté au guide d'ondes est en scrutation permanente d'un message quelconque sur sa fréquence propre de réception. La réception commence dès détection d'un signal sur cette fréquence.*

3.2.2.1 Protocole 1

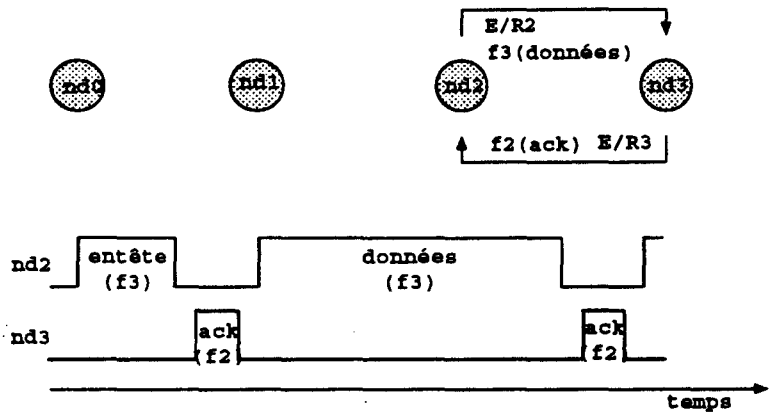


Figure 3.4 : Protocole de communication avec une seule unité E/R par noeud

Nous nous situons dans un contexte où tout noeud ne possède qu'une seule unité de communication (d'émission réception). Pour une communication entre le noeud nd2 et nd3 (cf figure 3.4), le protocole s'établit comme suit : nd2 voulant communiquer avec nd3 devra :

étape1 : attendre l'inactivité de sa cellule E/R2 (la cellule pouvant éventuellement être en réception).

étape2 : écouter la fréquence f3 du noeud3 pendant τp (temps de transmission d'un paquet)[3]. S'il constate une transmission sur cette fréquence, alors il reprend le protocole après un temps aléatoire.

étape3 : envoyer une trame sur f3, trame contenant l'adresse de nd2, puis attendre l'acquittement (ack) sur f2. Si nd2 ne reçoit pas d'acquittement, il reprend à l'étape1 après un temps aléatoire (ou l'application de l'algorithme de sélection arborescente de CAPETANAKIS, mais nous parlerons ici d'attente aléatoire pour ne pas compliquer le protocole).

étape4 : commencer la transmission du paquet sur f3 et recevoir l'acquittement sur f2.

L'utilisation des deux fréquences des noeuds communiquants f2 et f3 permet de signaler aux autres noeuds tels nd0 et nd1 que les canaux des noeuds 2 et 3 sont occupés. Si toute la communication s'était déroulée en utilisant une seule fréquence (du récepteur), il n'y aurait eu aucun moyen de signaler l'occupation de l'autre.

Néanmoins, deux complications peuvent se présenter :

- nd0 cherche au même moment que nd2 à émettre vers nd3. Les étapes1 et 2 seront franchies simultanément, mais nd0 et nd2 ayant deux adresses différentes, pour la suite du protocole, deux cas peuvent être observées :
  - ni nd0, ni nd2 n'auront d'ack de la part de nd3 (puiqu'il y a eu collison entre les deux requêtes) et dans ce cas l'attente aléatoire résoudra la contention.
  - un seul noeud aura un ack sur sa fréquence de réception, et la contention est résolue. L'autre noeud concurrent ne recevant pas d'ack reprendra le protocole plus tard.

[3]Nous verrons plus loin la nécessité de ce temps d'attente

- ndl cherche pendant le protocole de nd2 à émettre vers nd2. Il n'y a pas de risque de collision car l'étapel étant franchie par le noeud ndl, à l'étape2 il n'aura pas de réponse de nd2. On évite ainsi la collision entre les données de ndl vers nd2 et les ack de nd3 vers nd2 puisque ces deux communications se déroulent dans la fréquence f2. De même, le risque de collision entre l'entête ndl vers nd2 et l'ack nd3 vers nd2 peut être nul si le temps d'écoute à la phase2 de ndl est suffisamment grand.

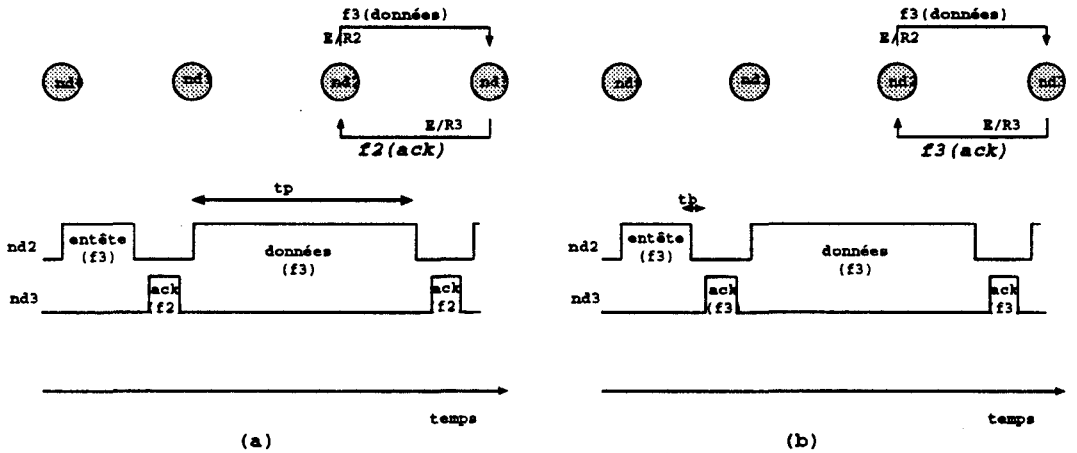


Figure 3.5 : Comparaison entre l'utilisation de deux fréquences f2 et f3 pour la communication (a) et l'utilisation d'une fréquence unique f3 pour la communication : protocole 1 "amélioré" (b)

Une amélioration de ce protocole imposerait que toutes les communications se fassent dans la fréquence de réception du destinataire pour éviter d'une part le changement de fréquence (fréquence-expéditeur vers fréquence-destinataire et vice versa) et d'autre part les éventuelles collisions entre les communications ndl $\leftrightarrow$ nd2 et nd3 $\leftrightarrow$ nd2 citées ci-dessus. En prenant comme exemple le cas présenté ci-dessus, toute la communication se déroulerait dans la fréquence f3. ndl voulant communiquer nd2, ne recevra pas d'ack à l'étape3 et reprendra le protocole après un temps aléatoire. Un des avantages de cette technique est aussi le temps d'écoute à l'étape2 nettement plus court. Il suffit de scruter le canal du destinataire pendant τb le temps de transmission d'un bit pour passer à l'étape2 du protocole, au lieu de τp le temps de transmission d'un paquet (parce qu'un canal occupé par une communication peut être inutilisé pendant tout ce temps : lors de la transmission des données, le canal-ack est libre et vice versa). La figure 3.5 montre une comparaison des deux techniques.

3.2.2.2 Protocole 2

Un noeud est identifié par une fréquence de réception propre mais comporte **deux unités E/R** ; l'une **E/Ra** servant à la réception de données et l'autre **E/Rb** à l'émission d'un *message d'occupation* sur la fréquence de réception de l'expéditeur pour signaler aux autres que son canal est occupé. La figure 3.6 montre une communication entre nd2 et nd3, l'exclusion mutuelle sur la sélection de l'expéditeur est réalisée de la même manière que dans le premier exemple.

L'avantage de cette technique est le temps d'écoute nettement réduit de ndl qui voudrait émettre vers nd2 puisque les deux fréquences des noeuds communicants sont actives. L'inconvenient majeur reste le doublement des unités E/R.

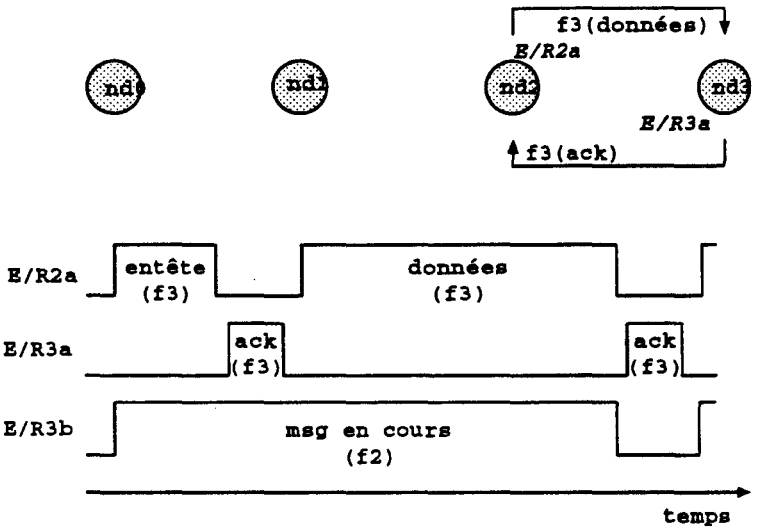


Figure 3.6 : Protocole de communication avec deux unités E/R par noeud

3.2.2.3 Protocole 3

La figure 3.7 montre deux communications simultanées du noeud nd2 à la fois expéditeur dans la communication nd2 ↔ nd3 se déroulant entièrement dans la fréquence nd3 et destinataire dans la communication nd1 ↔ nd2. Cette technique peut être considérée comme identique au protocole1 (cf figure 3.4) avec un doublement de la bande passante du réseau. Cependant elle offre un avantage par rapport au protocole1 "amélioré" (cf figure 3.5) parce qu'un noeud voulant communiquer avec un autre sait déjà à l'étape2 si celui-ci est occupé en réception ou pas, simplement en écoutant son canal de réception (qui ici ne sert qu'à la réception), alors que dans le protocole1, ce n'est qu'à l'étape3 qu'on sait si un noeud n'est pas en communication en tant qu'expéditeur. Considérant toujours l'exemple de la figure 3.7, le noeud nd1 voulant communiquer avec le noeud nd2 sait déjà à l'étape2 que nd2 peut recevoir des données.

3.2.3 Point de vue

Une comparaison des protocoles présentés ci-dessus montre à première vue que le protocole1 "amélioré" et le protocole3 constituent les deux solutions les plus envisageables pour une implémentation.

- Le dernier exhibe de très bonnes performances ; le protocole demeure assez simple et donne une meilleure utilisation de la bande passante du réseau mais avec un coût plus élevé (doublement des unités E/R, des buffers en émission et en réception, gestion des files d'attente...) même s'il est proportionnel (dans le meilleur des cas) à la bande passante offerte,
- Le premier offre un protocole assez simple pour un coût minimal en matériel (lors de l'implantation VLSI).

La principale limitation du nombre N de processeurs connectés à un guide d'ondes venant essentiellement du nombre de canaux, le choix du protocole1 ou du protocole3 (pour doubler

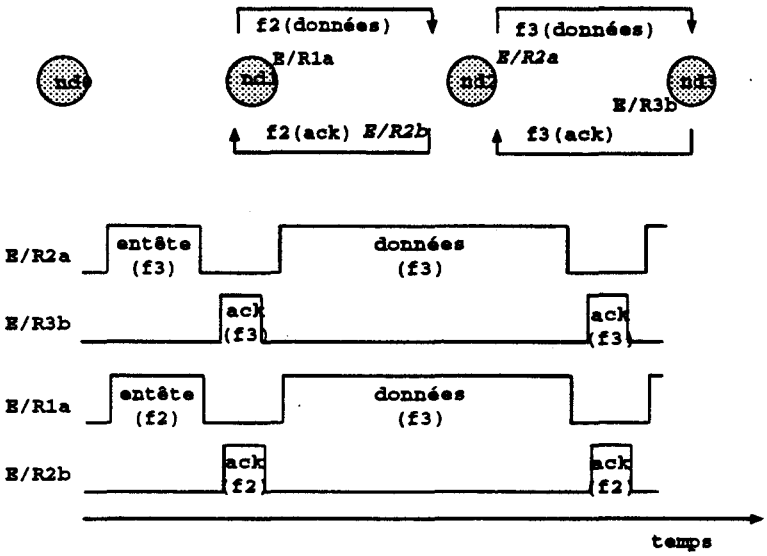


Figure 3.7 : Protocole de communication avec deux unités E/R par noeud permettant deux communications en parallèle

la bande passante du réseau) pour les communications sur un guide d'ondes sera dicté par les exigences en communication de la machine. Nous poursuivrons notre étude en considérant essentiellement le protocole1 "amélioré" (cf figure 3.5) pour la validation du réseau. De plus, ce dernier a l'avantage de présenter un câblage plus aisé vu la simplicité de l'automate associé à l'unité E/R, chargé d'exécuter le protocole (cf figure 3.8). L'automate chargé de contrôler le protocole 3 est pratiquement équivalent à un doublement des états de l'automate du protocole 1. Nous reviendrons dans le chapitre 5 sur la nécessité ou non du doublement des unités de communication.

3.2.4 Le protocole de communication

Nous décrivons ci-dessous les étapes nécessaires pour le transfert d'un paquet entre deux noeuds quelconques **nds** (noeud source) et **ndd** (noeud destination), connectés à un guide d'ondes en utilisant le protocole 1 "amélioré" (représenté par l'automate de la figure 3.8). A l'**état initial**, **nds** et **ndd** écoutent leur fréquences de réception respectives.

- le noeud **nds** voulant émettre vers le noeud **ndd** devra :

  étape1 écouter la fréquence de réception **fd** du noeud destinataire **nd** (pour s'assurer que le canal est libre). Si le canal est occupé, alors revenir à l'état initial.

  étape2 envoyer un en-tête sur **fd** (fréquence du destinataire), entête contenant l'adresse de **nds** et attendre un acquittement (comportant son numéro) sur **fd**. Si l'acquittement n'est pas reçu, aller à l'étape1.

  étape3 commencer la transmission du paquet sur **fd** et recevoir l'acquittement sur **fd**.

- le noeud récepteur **ndd** devra :

  étape1 dès réception de l'en-tête de **nds** sur **fd**, décoder l'en-tête.

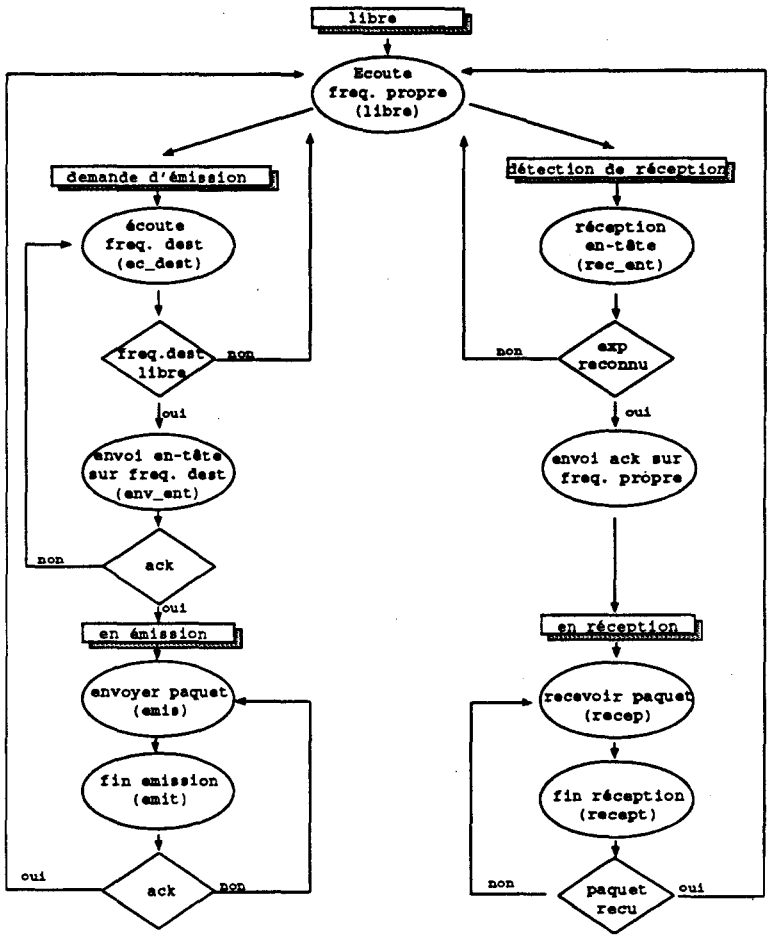


Figure 3.8 : Description du protocole de communication (protocole 1) dans un guide d'ondes

étape2 envoyer un acquittement sur **fd**, une fois **nds** reconnu. Si **nds** n'est pas reconnu, revenir à l'étape initiale.

étape3 commencer la réception du paquet sur **fd** et envoyer l'acquitement sur **fd** si le paquet est bien reçu sinon envoyer un message de non-acquitement sur **fd** puis reprendre à l'étape3.

3.2.5 Le format des messages

A chaque lien de communication est associé un buffer de la taille d'un paquet (parties de message de taille fixe). Comme dans le cas du format de messages mis-en oeuvre pour la communication entre les processeurs T9000, les messages plus longs que la taille maximale sont découpés en paquets de taille fixes (cf figure 3.9).

Conformément au protocole adopté, un message d'acquittement (*ack*) est envoyé dès que le paquet est routé[4]. Dans le cas où le paquet n'est pas bien reçu, un message de non-acquittement

[4] traité ou a quitté le buffer associé au lien de communication

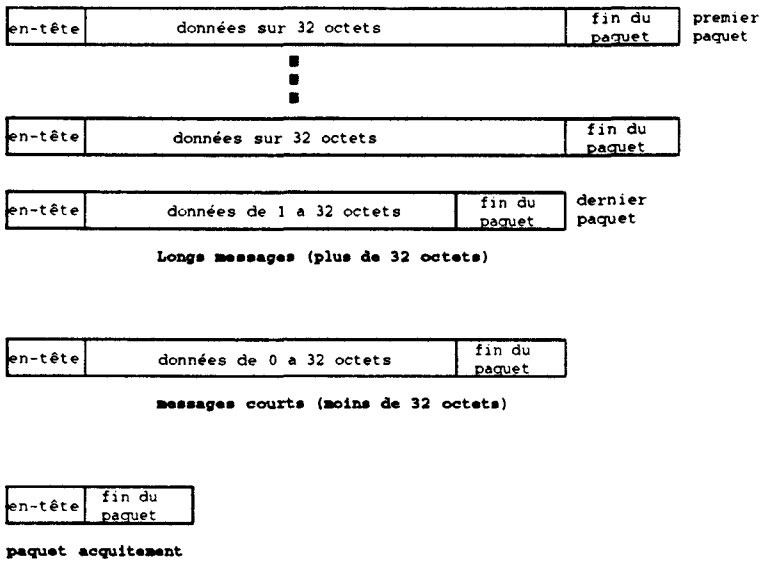


Figure 3.9 : Format de messages et taille des paquets utilisés pour la communication entre les processeurs T9000

nack est envoyé, signalant à l'expéditeur qu'il faut retransmettre le paquet. Trois types de messages d'acquittement suffisent donc pour le protocole de communication :

données : le paquet de données qui a une taille maximale fixée,

ack : le paquet de données est bien reçu sans erreur,

nack : il y a eu erreur de transmission (en vérifiant par exemple la parité), le destinataire se met alors en attente de réception du paquet.

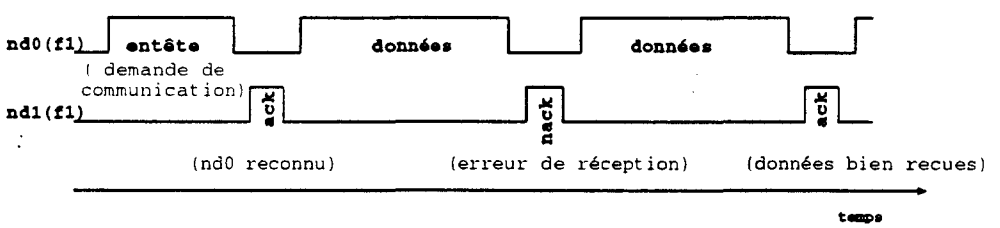


Figure 3.10 : Exemple de communication entre deux noeuds avec les 3 types de messages : d'acquitement, de non-acquitement et de données.

3.3 Le noeud de base

A la description faite d'un noeud au chapitre précédent, nous devons y adjoindre autant d'unités **GCi**[5] (cf figure 3.11) que de liens physiques. **Le GCi peut être vu comme un contrôleur de**

[5] Gestionnaire de communications

canal externe au routeur. Le GCi servant d'interface entre le gestionnaire de communication global du noeud (le **GC** classique qui réalise toute la logique de routage et qu'on retrouve dans beaucoup de routeurs comme celui de la machine Méga [31] [30] de l'université de Paris-Sud) et l'unité d'émission réception hyperfréquence E/R, sa spécification est très importante. Le protocole étudié pour la communication sur le guide d'ondes doit suivre les performances du gestionnaire de communication global du noeud et vice-versa. Le non respect de cette règle risque de dégrader considérablement les performances d'un des éléments couplés (le GC et la cellule hyperfréquence E/R).

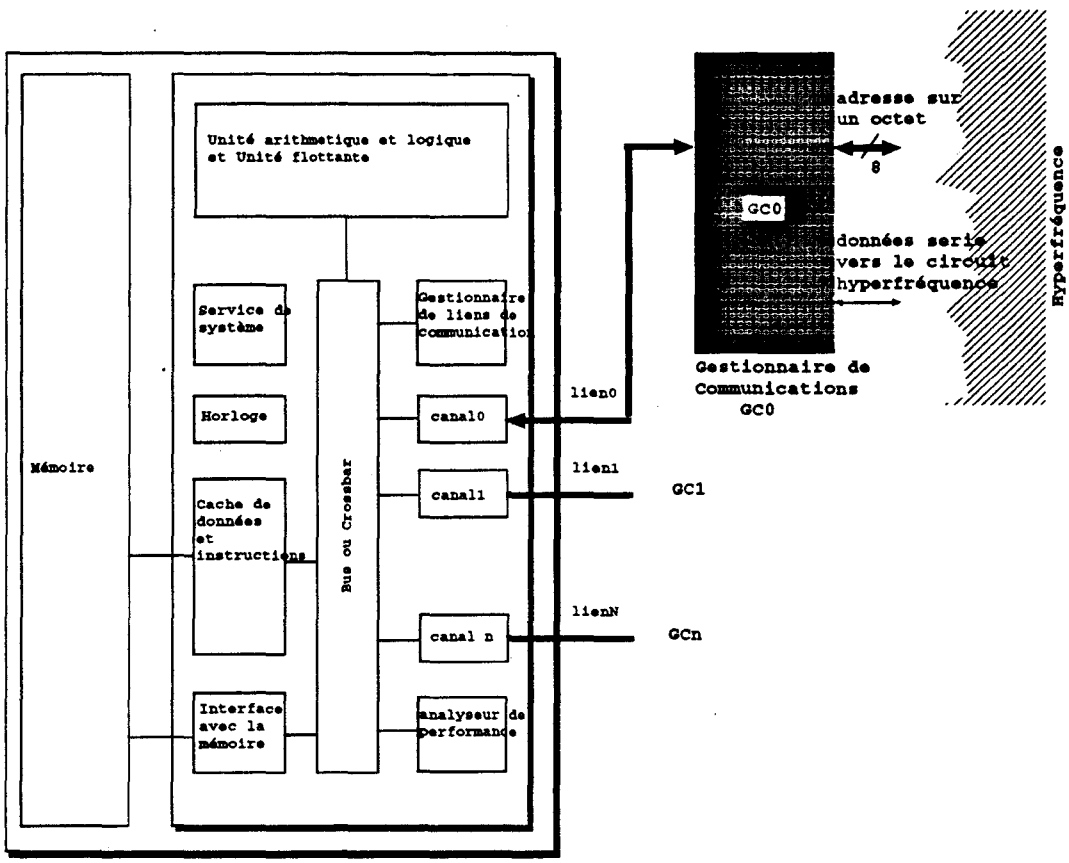


Figure 3.11 : Le noeud de base avec les gestionnaires de communication (GCi) implantant l'automate dédié à la communication sur le guide d'ondes

3.3.1 L'unité de communication GCi

Une unité de communication **GCi** (cf figure 3.11) simplifiée est composée essentiellement de l'unité de contrôle et de quelques éléments dont un comparateur et d'un registre à décalage (cf figure 3.12).

- le comparateur est utilisé pour la vérification de l'expéditeur, en comparant tout simplement son numéro à un nombre représentant le nombre de noeuds maximal connecté au guide (un comparateur 8 bits est donc suffisant). Une vérification plus rigoureuse

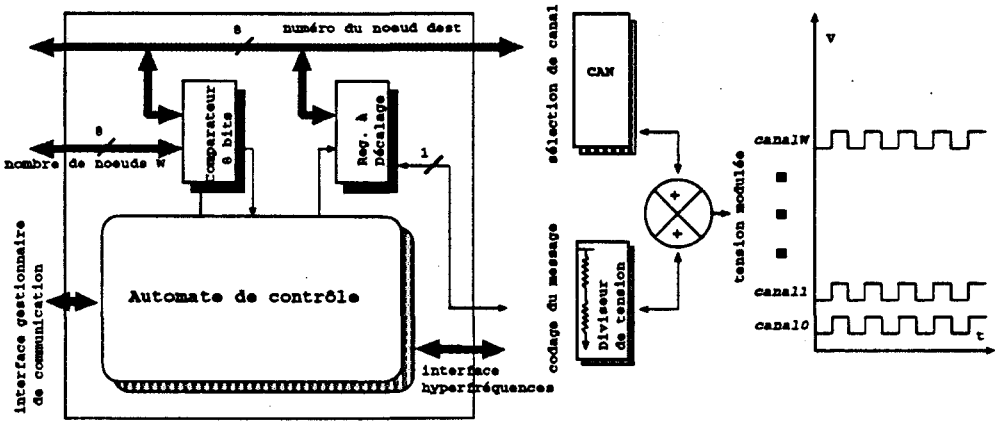


Figure 3.12 : Sélection de canal et transmission de données

peut évidemment être mise en oeuvre comme la consultation d'une table de numéros par exemple.

- le registre à décalage commandé par l'automate sert d'interface entre le lien parallèle (sur un octet par exemple) issu du routeur et le lien série vers l'unité hyperfréquence (unité **XR**).

- l'unité de contrôle est réalisée sous forme d'un automate d'états finis.
  Les entrées de l'automate sont :

  rtransin, rtransXRin requêtes de transmission venant du routeur ou de l'unité XR,
  rfinin, rfinXRin requêtes de fin de transmission venant du routeur ou de l'unité XR,
  ackin, ackXRin signaux d'acquittement venant du routeur ou de l'unité XR,
  nackin, nackXRin signaux de non-acquittement venant du routeur ou de l'unité XR,
  expOK, expNOK signaux issus du comparateur pour la reconnaissance de l'expéditeur

  Les sorties de l'automate sont :

  rtransout, rtransXRout requêtes de transmission vers le routeur ou l'unité XR,
  rfinout, rfinXRout requêtes de fin de transmission vers le routeur ou l'unité XR,
  ackout, ackXRout signaux d'acquittement vers le routeur ou l'unité XR,
  nackout, nackXRout signaux de non-acquittement vers le routeur ou l'unité XR,
  rexpout signaux vers le comparateur pour la reconnaissance de l'expéditeur

  Les règles de transition sont[6] :

[6] notations :
(état-précédent), signal-entrée ⟶ (état-suivant), signal-sortie
{commentaire}

(libre), rtransin {requête venant du routeur}	⟶	(ec-dest), rtransXRout {écoute du destinataire}
(ec-dest), ackXRin	⟶	(env-ent), rtransXRout {transmission de l'entête}
(env-ent), ackXRin	⟶	(emis), ackout {transmission des données}
(env-ent), nackXRin	⟶	(libre), nackout {état libre si l'exp n'est pas reconnu}
(emis), rfinin	⟶	(emit), rfinXRout {fin de transmission}
(emit), nackXRin {erreur de réception}	⟶	(emis), nackout {recommencer la transmission}
(emit), ackXRin {réception OK}	⟶	(libre), ackout {état initial}
(libre), rtransXRin {requête de réception}	⟶	(rec-ent), rexpout {reconnaissance exp}
(rec-ent), expNOK {exp non reconnu}	⟶	(libre), nackXRout {état initial}
(rec-ent), expOK {exp reconnu}	⟶	(recep), ackXRout, rtransout {état réception}
(recep), rfinXRin	⟶	(recept), rfinout {fin de réception}
(recept), nackin {erreur de réception}	⟶	(recep), nackXRout {état réception}
(recept), ackin {réception OK}	⟶	(libre), ackXRout {état initial}

3.3.2 Le circuit d'Emission/Réception

Plusieurs circuits électroniques ont été réalisés dans le cadre de la thèse de P. VANGELUWE [80] dans le département hyperfréquences (DHS) de Lille pour valider notre modèle.

La figure 3.13 montre le schéma résultant de la combinaison du circuit d'émission et du circuit de réception étudiés au chapitre précédent (cf figures 2.8 au paragraphe 2.2.4). La figure3.14 montre le schéma résultant pour une transmission à des débits inférieurs à 100 Mbps (cf figure 3.13) et supérieurs à 100 Mbps (cf figure 3.14). Les mêmes composants servent à la fois pour l'émission et pour la réception, ce qui nous offre des possibilités d'intégration à faible coût du circuit de communication (d'émission/réception). Ce circuit, réalisé au DHS de Lille est

composé essentiellement :

- d'un oscillateur à commande en tension (VCO) à cavité bande V 60 Ghz permettant un accord électrique sur ±5% de la fréquence centrale, offrant une bande passante de $B_{VCO} = 6Ghz$. Le VCO est utilisé à la fois pour l'émission et pour la réception des signaux.

- d'un mélangeur effectuant la conversion du signal modulé (issu du guide d'ondes) pour reconstituer le signal en bande de base qui sera destiné au filtre. Au signal issu du guide d'ondes (RF) est soustrait celui issu du VCO local (FOL), produisant une fréquence plus basse appelée fréquence intermédiaire (FI). Expérimentalement, nous savons qu'un mélangeur génère un nombre important de *raies d'intermodulation*, ceci étant dû aux fréquences F_{min} et F_{max} transposées en fréquences *images* par battements de la FOL issue de l'oscillateur. De ce fait, quand on utilise deux ou plusieurs voies RF, il nous faut générer autant de fréquences FI (correspondant aux canaux alloués à chaque élément connecté). Il est donc nécessaire qu'il n'y ait aucune raie (d'intermodulation) dans les bandes respectives. Heureusement, la puissance des raies d'intermodulation décroît rapidement, et un bon filtrage permet de les éliminer.

- d'un filtre à ondes de surface (FOS) pour la sélection du canal désiré.

- d'une PLL pour stabiliser la source émettrice et d'un démodulateur pour la remise en forme du signal lors de la réception.

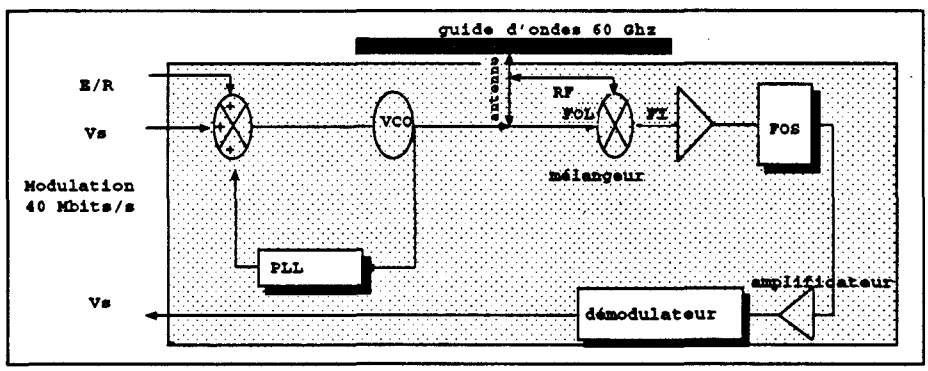


Figure 3.13 : Transmission à faible débit : unité d'émission/réception avec l'utilisation d'un seul oscillateur

3.3.3 Le codage des informations

Le signal en bande de base ayant un débit de d bits par seconde, l'expression du débit optimal par élément connecté est directement liée à la bande passante allouée à chaque canal, aux types de modulations et de codage choisis.

Il existe de nombreux types de codages des signaux dans les transmissions d'informations binaires. Citons comme exemple les codages Manchester, Manchester différentiel, NRZ (non retour à zéro), RZ (retour à zéro), le code biphasé etc. La figure 3.15 montre les particularités de chaque codage :

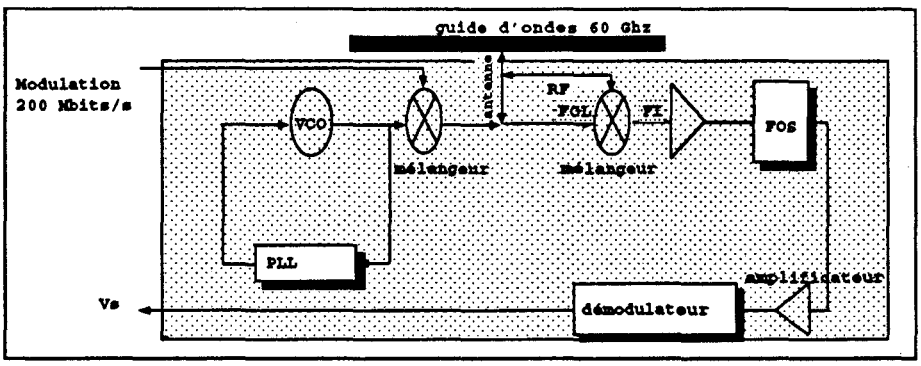


Figure 3.14 : Transmission à haut débit : unité d'émission/réception avec l'utilisation d'un seul oscillateur

- en codage Manchester, chaque intervalle de temps élémentaire pendant lequel un signal binaire est placé sur le câble, nommé souvent *temps bit*, est divisé en deux parties de durées égales avec les conventions suivantes : le signal binaire 1 est représenté par un signal de niveau élevé pendant la première partie du temps bit puis un signal de faible niveau pendant la seconde partie du temps bit, et le signal binaire 0 utilise la convention inverse, c'est-à-dire que le niveau bas est transmis en premier et le niveau haut en second. Ce mode de codage peut aussi être inversé ; on parle alors de codage Manchester Inversé ou de code biphasé.
- la codification Manchester différentielle est une variante de la précédente. Les transitions au milieu de chaque temps bit sont maintenues pour assurer la synchronisation des signaux binaires, mais les signaux binaires sont représentés selon de nouvelles conventions : le signal binaire 1 est représenté par une absence de transition au début du temps bit correspondant, et le signal binaire 0 est représenté par la présence d'une transition au début du temps bit considéré.
- dans le codage RZ, le 1 binaire est codé de la même manière que dans le codage biphasé, et le 0 binaire est caractérisé par une absence de signaux.
- Le codage NRZ n'oblige pas à un retour à zéro lors du codage et est identique au codage binaire classique.

Une comparaison des différents types de codage montre que le codage NRZ permet d'obtenir un débit deux fois plus élevé que les autres types de codages présentés ci-dessus. Les autres types de codages offrent l'avantage d'une transmission plus efficace de la synchronisation temporelle des signaux (l'horloge) simultanément aux informations binaires proprement dites. Nous avons de plus choisi le codage NRZ pour des raisons d'implantation purement électronique suivantes :

- il est simple à réaliser et à décoder.
- il demande moins de bande passante, on peut donc utiliser dans les récepteurs des canaux plus étroits et par conséquent, faire passer plus de canaux simultanés dans le guide d'ondes.
- il permet l'emploi de filtres plus étroits, ce qui a pour effet d'augmenter le rapport S/B.

L'inconvénient majeur d'un codage NRZ étant évidemment une synchronisation moins fiable entre l'émetteur et le récepteur, deux techniques (utilisées d'ailleurs dans les réseaux FDDI) permettent de pallier cette limitation :

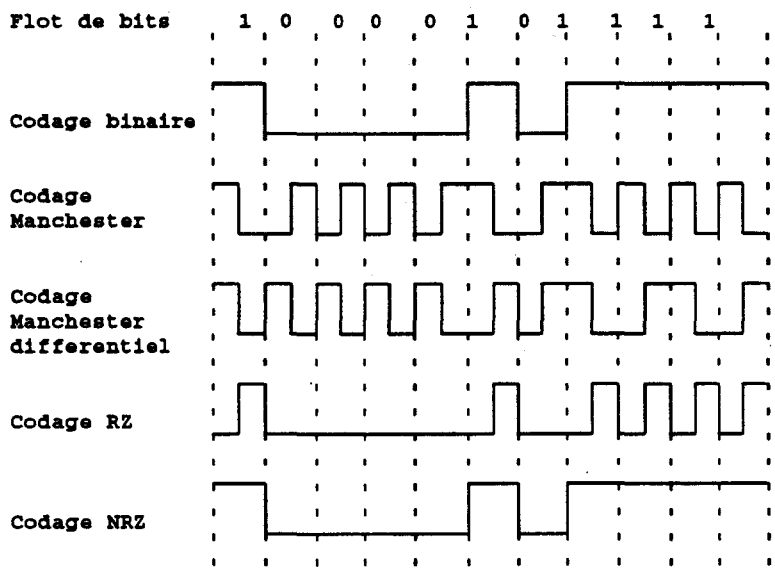


Figure 3.15 : Quelques techniques de codage

- le codage par groupes (4B/5B ou 8B/9B etc.) permettant la suppression par exemple de 3 bits à 0 successivement,
- l'utilisation d'un préambule (une suite de bits précédant tout les messages) pour synchroniser l'émetteur et le récepteur.

3.3.4 Le débit sur un canal

Le débit d sur un canal est directement lié à la bande passante B_s du canal si nous nous référons au théorème de Nyquist[7] [78]. Or la bande passante B_s est dépendante d'autres paramètres et est donnée par la relation :

$$N(B_s + 2 \times B_c) \leq B_{vco} - 2 \times \alpha$$

Ce qui nous donne par exemple une bande passante par canal de B_s = 80 Mhz (avec un taux d'erreurs théorique de 10^{-8}) pour des valeurs de B_{vco}, N, B_c, α données ci-dessous.

- la bande passante des VCO B_{vco}=6Ghz
- le nombre de noeuds connectés N=64
- la plage de capture[8] B_c=6Mhz
- α (plage de capture aux deux extrêmes de la bande passante du VCO) est du même ordre que B_c.

[7] Le débit binaire d maximum est donné par $d_{max} = 2B_s log_2 V$ en bits/s avec V représentant le nombre de niveaux significatifs du signal, ici $V = 2 \Longrightarrow d_{max} = 2B_s$

[8] bande de fréquences de part et d'autre du canal alloué à un noeud pour éviter les interférences

Par ailleurs, le débit d_{max} donné par le théorème de Nyquist considère des filtres *mathématiques* (parfaits) et un canal unique non sujet aux perturbations issues des autres canaux. En considérant des filtres de gabarits classiques, et les perturbations liées à l'usage du *multicanaux*, le signal est modulé dans $B_s/4$, permettant de filtrer convenablement le signal ; un canal de B_s étant toujours alloué à un noeud connecté à un guide d'ondes. Nous obtenons de ce fait un débit de $d = 2\frac{B_s}{4} = B_s/2 = 40$ Mbits/s pour N=64.

Pour N=64, B_s a la valeur 80 Mhz, offrant un débit $d \simeq B_s/2$=40 Mbits/s. Pour N=32, on obtient B_s=160 Mhz et d=80 Mbits/s. Si toute la bande passante du guide d'ondes était allouée à un seul canal, nous obtiendrions un débit de **3 Gbits par secondes**. D'autre part, à cause des problèmes liés aux raies d'intermodulation[9], la connexion d'un nombre N de noeuds dépassant les 64 est assez délicate (les harmoniques générées deviennent très nombreuses et difficile à localiser, elles peuvent donc apparaître dans la plage de fréquences allouée à un autre noeud).

Le débit dans un canal est donc directement lié au choix du nombre de canaux dans le guide d'ondes, en d'autres termes au nombre de noeuds connectés au guide d'ondes : plus le guide d'ondes connecte de noeuds, moins le débit est élevé. Par exemple pour N=64, le débit d=40 Mbps, pour N=16, d=160 Mbps et pour N=8, d=320 Mbps. Il semble donc intéressant de se poser la question de la meilleure configuration de réseau en terme de débit par noeud pour un nombre N de noeuds et un nombre G de guides d'ondes donnés. Une solution consiste à considérer des guides d'ondes autorisant des débits élevés (donc avec un petit nombre de noeuds connectés), puis de construire des réseaux sur plusieurs dimensions. Cette question fera l'objet du chapitre suivant.

3.4 La latence physique et les domaines d'utilisations du réseau

3.4.1 La latence physique

Le protocole adopté pour la validation du projet nécessite avant toute communication, une étape préliminaire pour s'assurer que le canal est libre. Cette étape inclue des procédures d'écoute du canal, d'envoi de l'entête et de réception de l'acquittement. Le temps d'exécution de ces procédures dépend des caractéristiques physiques des composants utilisés, notamment des temps de traversée associés aux composants hyperfréquences (cf table 3.1).

- La surface occupée par la partie hyperfréquence de l'unité de communication est de l'ordre de 50 mm^2 sans compter les interconnexions entre les différents composants, ces derniers nécessitant nettement moins de place, si l'on considère les schémas résultant (disposition régulière des éléments). L'élément le plus encombrant est le FOS qui en notre connaissance n'a pas encore été intégré avec d'autres composants.
- Un signal émis traverse uniquement un VCO et un mélangeur dont les temps de traversée sont de l'ordre d'1ns, soit environ 2ns pour l'émission d'un signal dans un canal (temps qu'il ne faut pas confondre avec le temps d'initialisation du VCO qui est beaucoup plus long et effectué une seule fois).
- La réception est par contre beaucoup plus lente puisqu'elle nécessite la traversée de plusieurs étages dont un mélangeur (1 ns), deux amplificateurs (2×20 ns), un FOS (8

[9] Signaux destinés à un canal et apparaîssant dans une autre bande de fréquences

Tableau 3.1 : Caractéristiques des composants utilisés

composant	surface	temps de traversée	puissance consommée	type de composant
VCO 60Ghz	3 mm^2	1,5 ns	6,25 mW	actif
PLL	5 mm^2	1 - 3 ms (à l'initialisation)	10 mW	actif
mélangeur	2 à 3 mm^2	$\simeq$ 1 ns	0	passif
FOS	25 mm^2	8 ns	0	passif
Ampli	2 mm^2	20 ns (Bipolaire Si)	50 mW	actif
démodulateur	5 mm^2	10 ns	0	passif

ns) et enfin un démodulateur (10 ns), soit environ $\tau_{recept} = 60\ ns$ pour la réception.

Pour déterminer la latence physique il nous faut prendre en compte outre les temps de traversée cités ci-dessus, des temps liés à l'automate de contrôle, ces temps étant fonction de la fréquence de fonctionnement du circuit. En désignant par τ_{recept} le temps nécessaire dû à la traversée des étages des deux circuits repectifs, soit $60ns + 2ns = 62ns$, plus les temps de propagations divers, alors $\tau_{recept} \simeq 70ns$. Nous pouvons faire les estimations suivantes :

- le temps d'écoute du canal doit être strictement supérieur au temps nécessaire à la réception, donc de τ_{recept} au moins,
- le temps d'envoi de l'entête prend $8 \times \tau_{bit} + 2\ ns + \tau_{recept}$ puisqu'il s'agit juste d'envoyer un octet représentant le numéro de l'expéditeur ; τ_{bit} représentant le temps nécessaire à la transmission d'un bit sur le canal (donc dépend de la configuration choisie pour le guide d'ondes : le nombre de noeuds connectés).
- le temps de réception de l'acquittement (supposé être un octet) du récepteur est de tc (vérification de l'entête par le comparateur) + $8 \times \tau_bit$ (temps de transmission) + τ_{recept}.
- l'envoi du paquet de données est de $L \times \tau_{bit} + \tau_{recept}$. Ce temps dépend donc de la longueur L d'un paquet de données.

Conformément au protocole adopté, le temps minimal pour la transmission d'un paquet de longueur quelconque est la somme du temps d'écoute du canal, du temps d'envoi de l'entête, du temps nécessaire à la réception de l'acquittement, du temps de transmission du paquet de longeur L et enfin la réception de l'acquittement. La latence physique minimale du réseau réduit à un guide d'ondes est alors donnée par l'expression :

$$(\tau_{recept}) + (8 \times \tau_{bit} + \tau_{recept}) + (tc + 8 \times \tau_{bit} + \tau_{recept}) + (L \times \tau_{bit} + \tau_{recept}) + (8 \times \tau_{bit} + \tau_{recept})$$

soit, latence(L bits) $= 5\tau_{recept} + (L + 24)\tau_{bit} + tc$.

En considérant des fréquences de fonctionnement de l'unité de communication d'environ 50 Mhz, donc un temps de cycle pour l'horloge de tc=**20** ns, nous obtenons :

$$Latence(Lbits) = 370ns + (L + 24)\tau_{bit} \tag{3.1}$$

le tableau 3.2 donne les temps de latences pour L=1024 bits (transfert de 16 nombres flottants double précision par exemple), pour diverses valeurs de N, le nombre de noeuds connectés au guide.

Tableau 3.2 : Latence du réseau pour un message de L=1 Kbits

N	64	32	16	8
D en Mbps	40	80	160	320
τ_{bit}	25 ns	12 ns	6 ns	3 ns
Latence	26,5 μs	13 μs	6,6 μs	3,5 μs
entete	0,97 μs	0,65 μs	0,51 μs	0,44 μs
overhead (entête/latence) (pourcentage)	3,6	5,0	7,7	12,57

3.4.2 Les domaines d'utilisation du réseau

Un critère important dans la spécification d'une architecture parallèle (choix du processeur de calcul, de la mémoire, du réseau d'interconnexion du mode de fonctionnement, du modèle de programmation, et même de la classe d'applications etc.) est qu'on trouve à tous les niveaux des fonctionalités convergentes. On trouve donc dans les réseaux, des fonctionnalités réflectant le modèle de programmation sous-jacent. Notre démarche ici consiste à considérer les caractéristiques architecturales du réseau à base de guide d'ondes, puis d'essayer de déterminer quel modèle de programmation est le mieux adapté, et pour quels domaines d'utilisation.

Pour déterminer les domaines d'utilisation du réseau à base de guides d'ondes, la latence représente un facteur très important, puisque caractérise en fait l'overhead pour une taille de message donné. Il est donc important d'étudier l'évolution de la latence aussi bien pour de très petits messages que pour des messages de grande taille. Le comportement du réseau pour l'exécution d'opérations globales nous semble aussi être un paramètre important puisqu'il caractérise en fait la capacité du réseau à effectuer efficacement des primitives de communications nécessaires à certains modes de fonctionnement.

- la latence : La table 3.3 donne l'évolution des latences et de l'overhead dû au protocole pour des tailles de messages allant d'un octet au kilo-octet.

  Nous remarquons que le temps de latence pour la transmission d'un octet est assez élevé. Pour N=16 par exemple, **ce temps est de l'ordre de** $0,56\mu s$**, dans le cas où aucun conflit d'accès au canal ne survient**. En considérant $N = 32$, la latence est de $0,75\mu s$.

Tableau 3.3 : Latences pour de différentes tailles de messages

$L(bits)$	8	32	128	512	2K	8K
$Latence(N = 64)$	1,17 μ	1,77 μs	4,17 μs	13,77 μs	52,17 μs	205,77 μs
$Overhead(N = 64)$ (pourcentage)	82,90	54,80	23,26	7,04	1,85	0,47
$Latence(N = 32)$	0,75 μs	1,04 μs	2,19 μs	6,80 μs	25,23 μs	98,96 μs
$Overhead(N = 32)$ (pourcentage)	87,26	63,14	29,99	9,67	2,60	0,66
$Latence(N = 16)$	0,56 μs	0,70 μs	1,28 μs	3,58 μs	12,80 μs	49,66 μs
$Overhead(N = 16)$ (pourcentage)	91,45	72,80	40,09	14,33	4,01	1,03
$Latence(N = 8)$	0,46 μs	0,54 μs	0,82 μs	1,98 μs	6,58 μs	25,02 μs
$Overhead(N = 8)$ (pourcentage)	94,84	82,15	53,51	22,34	6,71	1,76

Par comparaison aux débits offerts par les machines parallèles comme la CM5 avec une latence de l'ordre de 5 μs pour 1024 noeuds, le réseau à base de guides d'ondes, malgré un protocole plus contraignant offre des latences du même ordre puisqu'avec un réseau d'1K noeuds (disposition 32×32 noeuds), nous pouvons obtenir une latence égale à 2 fois celle d'un réseau réduit à un seul guide d'ondes de 32 noeuds, c'est-à-dire $2 \times 0,75\mu s = 1,5\mu s$ pour un message d'un octet (dans le cas où il n'y a pas de conflits d'accès au canal).

Nous pouvons constater de plus que la latence dans un réseau à base de guide d'ondes est indépendante de la localité des noeuds. La communication entre voisins requièrt autant de temps que la communication entre noeuds éloignés, contrairement aux machines utilisant un réseau direct comme la CM5 (un fat tree) ou l'Intel Paragon (un réseau mesh). Le réseau réduit à un guide d'ondes est donc assez proche des réseaux multi-étages qui garantissent en générale un temps de latence assez constant entre tout couple de noeuds comme la Telmat CS2 (1 μs) ou l'IBM SP1 (0,5 μs).

Les latences déterminées ici pour un réseau à base de guide d'ondes sont des **latences minimales** qui peuvent évidemment être atteintes si les communications n'engendrent pas de conflits, en d'autres termes si les applications sont conçues de manière à éviter les conflits de communication. Le chapitre 5 présente une évaluation des performances d'un guide d'ondes en termes de latence moyenne et de bande passante utile.

- les opérations globales concernent essentiellement celles citées dans le premier chapitre : la diffusion, l' échange total et le regroupement.

  Le guide d'ondes étant divisé en plusieurs canaux, un canal peut être réservé uniquement aux diffusions. Tous les noeuds posséderont dans ce cas deux unités de communications, l'une pour les transferts (communications classiques entre deux noeuds) et l'autre toujours connecté au canal de diffusion. Le protocole d'accès à ce canal sera évidement exclusif, assez similaire au protocole étudié pour les transferts sauf que dans ce cas-ci, aucun acquittement n'est attendu après la diffusion. L'échange total peut être réalisé par une

séquentialisation de plusieurs diffusions, le réseau de contrôle de la CM5 pipeline d'ailleurs toutes les diffusions ayant lieu simultanément. Par contre le regroupement est assez délicat et serait difficile à mettre en oeuvre matériellement à partir des guides d'ondes *multi-canaux* (elle peut être implémentée de façon logicielle en log_2(nombre de noeuds) par une collecte arborescente de l'information).

En considérant les latences minimales nous pouvons déjà **préférer les applications nécessitant des échanges moins fréquents de grandes tailles d'informations (transferts de blocs de matrices) aux applications nécessitant des échanges fréquents de données de petites taille (transferts de registres, algorithmes de tris parallèles etc.)**. D'autre part, un mode de fonctionnement SIMD est à exclure puisque ce mode nécessite la diffusion d'une instruction tous les cycles processeur, ce que ne peut supporter le réseau à base de guides d'ondes. Cependant, si l'on dispose d'un canal de diffusion que tous les noeuds écouterait dans un guide d'ondes, un mode de fonctionnement SPMD peut être envisagé, le canal de diffusion étant alors utilisé pour distribuer le code aux noeuds, la synchronisation sera effectuée de façon logicielle.

Enfin, il faut remarquer que la particularité d'un réseau à base de guides d'ondes est de permettre plusieurs communications différentes et simultannées entre noeuds. ***Le réseau à base de guides d'ondes semble donc plus adapté à un mode de fonctionnement MIMD.*** D'autre part, le coût d'une synchronisation évolue de façon logarithmique en fonction du nombre de processeur et un canal de diffusion peut être mis en oeuvre efficacement sur un guide d'ondes.

3.5 Les tests laboratoires

Nous avons effectué en plus des tests à base de générateurs de signaux hyperfréquences, des liaisons entre processeurs (des PC équipés de processeurs i80386) en utilisant leurs ports série. Pour des raisons purement pratiques (limitations du circuit i8250 gérant le port série des PC) nos tests ont été effectués à une vitesse de transmission de 115200 bits par secondes.

Le but de ces tests est de vérifier la possibilité de communications au travers d'un guide d'ondes, puis de déterminer expérimentalement le taux d'erreurs pour de telles liaisons (cf figure 3.16), le taux d'erreur théorique pour un signal S/B=12db étant de 10^{-8}.

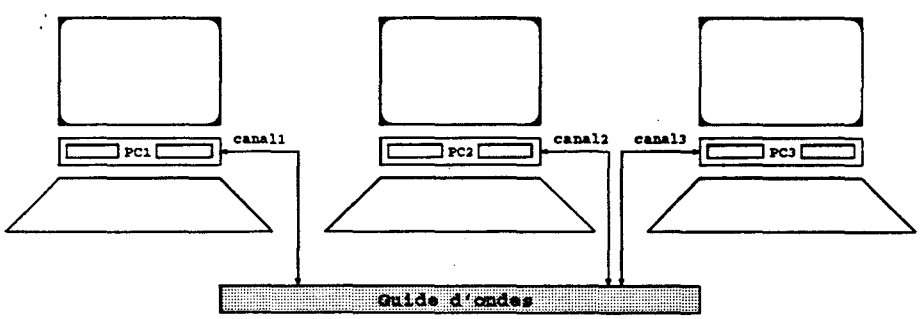


Figure 3.16 : Communications entre PC au travers d'un guide d'ondes

Pour valider la possibilité de communications simultanées sur un guide d'ondes, deux tests ont été effectués (les détails technologiques figurent dans la thèse de P. VANGELUWE [80] actuelle-

ment en cours de rédaction au DHS[10]) :

- Le premier test a mis en oeuvre deux sources hyperfréquences et un récepteur, les trois circuits étant connectés au même guide d'ondes. Les deux sources émettant sur deux fréquences (canaux) différentes et le récepteur écoutant l'une des fréquences pour décoder l'information. Ce test nous a permis de valider la possibilité de sélectionner un canal d'écoute dans un environnement *multicanaux*.
- Le second test a eu pour but la validation de deux communications simultanées au travers d'un même guide d'ondes. Deux générateurs de signaux et deux récepteurs ont été utilisés à cet effet.
- Les tests concernant les mesures des taux d'erreurs pratiques et les mesures des temps de commutation d'une fréquence à une autre puis d'une émission (de données) à une réception (d'acquitement) sont en cours de réalisation.

3.6 Conclusion

Ce chapitre a mis en exergue une application possible des hyperfréquences. Comme dans le cas de la technologie optique, nous avons montré que les techniques (protocoles de communication) jusque là développées pour la communication dans les réseaux locaux peuvent être appliquées aux réseaux d'interconnexion. Des tests effectués en laboratoire ont confirmé la possibilité d'utilisation de guides d'ondes *multicanaux*[11] pour l'interconnexion d'un ensemble de noeuds. Nous pouvons de plus connecter un nombre N de noeuds allant de 2 noeuds à 64 à un guide d'ondes, les débits étant inversement proportionnels au nombre N de noeuds connectés, tout en maintenant un débit théorique de 3 Gbits par seconde sur tout le guide d'onde. Il est évident que des débits plus élevés pourraient être obtenus si nous disposions de VCO dont la bande passante dépassait les 6 Ghz. Nous pouvons donc conclure que les débits annoncés ici suivront très probablement l'évolution de la technologie des hyperfréquences.

Un guide d'ondes constitue donc une brique de base pour la construction de réseaux. Pour ne pas se restreindre à une technologie qui ne propose qu'une machine figée en nombre de noeuds, nous proposons dans le cadre d'une extension du réseau, trois constructions possibles : les deux premières basées sur la théorie des hypergraphes et des Hypernets devraient donner d'assez bons résultats, mais qui restent néanmoins théoriques (cf annexe A) et la dernière plus pragmatique présentée dans le chapitre suivant. Nous étudions donc dans le chapitre suivant les possibilités d'extension d'un réseau construit à partir de guides d'ondes avant de donner une modélisation du débit réel.

[10] Département Hyperfréquences et Semi-conducteurs de Lille

[11] En ce sens où un guide d'onde peut comporter plusieurs canaux de transmission indépendants

Chapitre 4

Topologie d'un réseau *Hypercom*

Dans les pages précédentes, nous avons étudié et montré la faisabilité d'un guide d'ondes pouvant connecter un ensemble de W noeuds (Nous utiliserons W pour représenter le nombre de noeuds connectés à un guide d'ondes, N représentant le nombre total de noeuds dans un réseau construit à partir de plusieurs guides d'ondes), en assurant un débit potentiel global de l'ordre de 3 gigabits par seconde. Mais pour aller au delà de la limite des W noeuds, des architectures plus ou moins complexes sont à envisager. Dans ce chapitre, nous examinons les différentes possibilités de construction de réseaux à base de plusieurs guides d'ondes.

4.1 Extension du réseau

4.1.1 Introduction

Un guide d'ondes peut interconnecter au plus un ensemble de $W = 64$ noeuds, tous pouvant communiquer simultanément deux à deux. Pour un réseau de quelques centaines de noeuds, une structure à bus multiple [58] peut être utilisée mettant en parallèle plusieurs guides d'ondes. Dans ce cas, si tous les noeuds sont connectés à tous les guides d'ondes, cette technique permet juste d'augmenter la bande passante du réseau puisque nous restons toujours dans la limite des W noeuds (cf figure 4.1). Pour interconnecter plus de W noeuds avec une structure à bus multiple (donc un réseau à une seule dimension), il faudra que les guides d'ondes connectent des groupes différents de W noeuds (en ce sens où un guide d'ondes ne peut connecter que W noeuds), en s'arrangeant pour maintenir la connectivité entre l'ensemble des noeuds. Cette façon de procéder pour créer la connectivité entre un ensemble de plus de W noeuds sur une seule dimension sera vue plus loin dans la définition du réseau *Hypercom*.

Dans un cas général, utilisant un guide d'ondes interconnectant W noeuds comme module de base pour un plus grand réseau, deux types de constructions sont possibles.

- Soit utiliser une structure hiérarchique de grappes de noeuds reliées entre elles par un guide d'ondes, les noeuds de la grappe étant eux même reliés entre eux par un guide d'ondes. La machine Cm* de l'Université de Carnegy-Mellon [77] est un exemple de ce type de construction (par utilisation de bus communs). La figure 4.2 montre une construction hiérarchique possible à base de guide d'ondes. Cette architecture est évidemment adaptée aux applications possédant des caractéristiques de forte localité à l'intérieur de chaque grappe.
- Soit utiliser une structure linéaire à 1, 2 ou 3 dimensions dont trois classes de topologies peuvent être envisagées : le Spanning bus hypercube de WITTIE [83], les Hypernets de HWANG [45] et la dernière classe basée sur la théorie des hypergraphes [6]. De ces

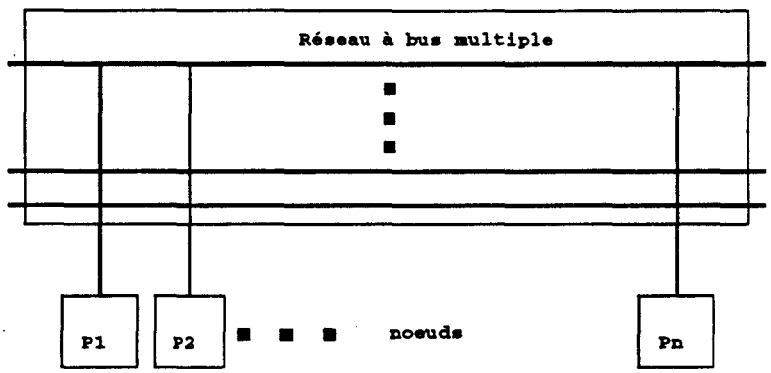


Figure 4.1 : Architecture de machine utilisant un bus multiple

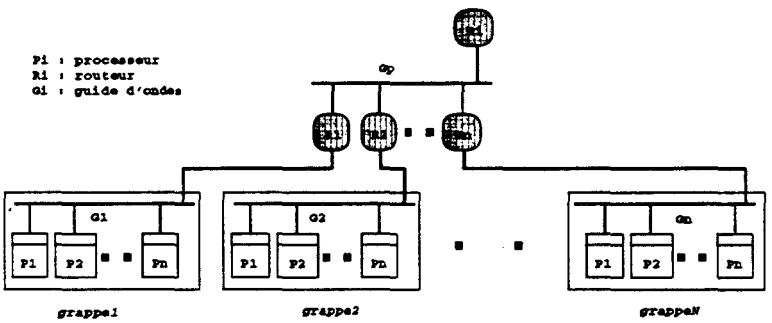


Figure 4.2 : Architecture de machine utilisant une hiérarchie de guide d'ondes

trois classes, seule celle issue de Spanning bus hypercube a un intérêt du point de vue réalisation pratique. Nous décrivons donc dans ce chapitre quelques quelques réseaux issus du Spanning-bus hypercube, l'étude des Hypernets et des Hypergraphes plus théoriques est reportée en annexe.

4.1.2 Le Spanning bus hypercube

Les réseaux *Spanning bus hypercube* et *dual-bus hypercube* ont été proposés par WITTIE dans [83] pour la construction de grands réseaux d'interconnexion. Dans un réseau spanning bus hypercube (cf figure 4.3), les noeuds sont disposés en forme d'hypercube de largeur W et de dimension D (formant un maillage dans l'espace W^D), et chaque bus connecte les W noeuds dans une des D dimensions. Chaque noeud est connecté à exactement D bus et tous les noeuds connectés à un même bus de la *ième* dimension ont les mêmes coordonnées sauf évidemment la coordonnée i.

Par comparaison à un réseau mesh de dimension D, les noeuds dans le cas du spanning bus hypercube requièrent D ports au lieu de 2D liens bidirectionnels pour les réseaux mesh. De plus, en considérant une seule dimension, la distance moyenne entre deux noeuds quelconques dans le cas du spanning bus hypercube est de 1 alors qu'elle est de $\sum_{i=0}^{W-1} min(i, W-i)/W = W/4$ dans le cas du réseau mesh [83] (Un argument en faveur du mesh est que, dans le cas du Spanning bus hypercube, tous les W noeuds partagent l'accès au même bus, et les accès doivent alors être séquentialisés).

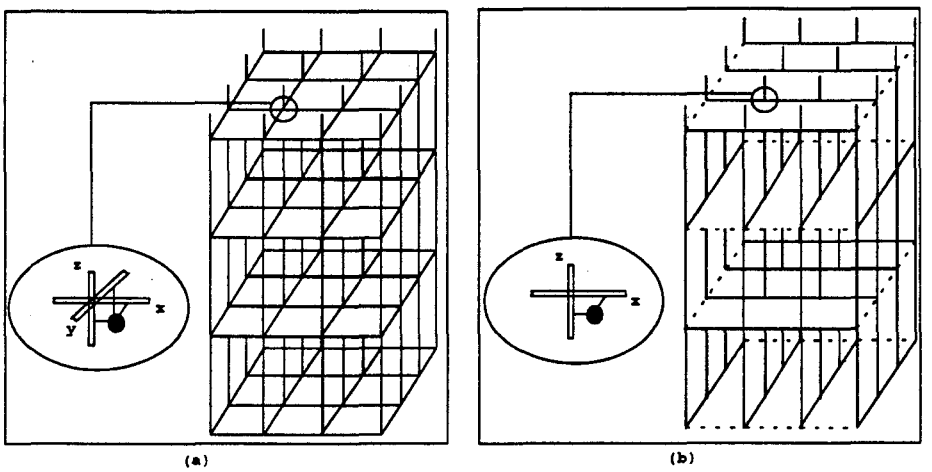


Figure 4.3 : Le Spanning bus hypercube (a) et le Dual bus hypercube (b) de dimension D=3

Le Dual-bus hypercube [83] possède les mêmes caractéristiques quant à la disposition des noeuds, mais avec un nombre de bus beaucoup plus faible, en ce sens où chaque noeud n'est connecté qu'à deux bus (au lieu de D dans le cas du spanning bus hypercube) quelle que soit la dimension D du réseau considéré. Chaque noeud est toujours connecté au bus dans la dimension 0 (choisie arbitrairent et représentée verticalement sur la figure 4.3 (b)), et dans chaque hyperplan de dimension (D-1) perpendiculaire à la dimension 0 tous les noeuds sont connectés à un seul bus dans la même dimension. Si nous supposons que la (dimension "privilégiée") dimension 0 est la dimension verticale (dimension z dans l'espace x, y, z), dans un dual-bus hypercube de dimension 3, les noeuds dont la coordonnée z est paire sont connectés aux bus dans les dimensions z et x, puis les noeuds dont la coordonnée z est impaire sont connectés au bus dans la dimension z et y. Le dual bus hypercube possède donc l'avantage de pouvoir être étendu sans modification du degré physique des noeuds qui reste égal à 2, et le diamètre du réseau n'est environ que deux fois plus grand que celui du spanning bus hypercube (2D-2 exactement). Le Spanning bus hypercube et le dual-bus hypercube possèdent tous les deux un bon algorithme de routage, qui n'est autre qu'une forme simplifiée de ceux relatifs aux hypercubes de largeur W et de dimension D. Un algorithme pour l'envoi d'un message du noeud A au noeud B est le suivant :

- Exprimer les indices de A et de B sous forme de coordonnées dans un maillage W^D. Chaque point étant un nombre de longueur D dans la base W. A chaque étape i, comparer chaque *ième* coordonnée de la source (initialement la source est A) à la *ième* coordonnée de la destination (qui demeure toujours B). Si la source et la destination diffèrent suivant la coordonnée i, router le message suivant la dimension i de sa position actuelle vers le noeud ayant sa *ième* coordonnée identique à celle de la destination (B) et toutes les autres coordonnées identiques à celles de la position actuelle. Le noeud auquel a été envoyé le message devient la nouvelle source et le procédé est réitéré pour le i suivant jusqu'à ce que le message arrive à la destination (B).
- Le routage dans le cas du dual-bus hypercube est sensiblement le même sauf que la réduction de la coordonnée i dans ce cas requiert deux relais au lieu d'un seul. Le premier relais utilise la coordonnée 0 (axe z sur la figure 4.3(b)) pour accéder au noeud connecté au *ième* bus et le second relais utilise la *ième* coordonnée pour arriver à la destination prévue à l'étape i.

4.2 Le réseau d'interconnexion

4.2.1 Introduction

Dans le but de construire des réseaux de tailles plus grandes (nombre supérieur à la limite des W noeuds d'un guide d'ondes) à partir de guides d'ondes connectant W noeuds, une approche consiste à garder l'avantage de la topologie du Spanning bus hypercube de dimension 3 (degré et diamètre du réseau D=3) puis de trouver le moyen d'étendre le nombre de noeuds sur une dimension. Le nombre de noeuds dans une direction pourrait passer de W à $2 \times W$, et permettre la construction de réseaux dépassant un maillage classique $W \times W \times W$. Par exemple, pour garder un débit par canal de 320 Mbits/s avec $W = 8$, un réseau de 1024 noeuds nécessite un maillage $W \times W \times (2 \times W)$, et celui de 2K noeuds nécessite un maillage $W \times (2 \times W) \times (2 \times W)$. Nous proposons donc dans cette partie de la thèse une méthode permettant d'obtenir une telle extensibilité du réseau, en conservant les avantages du Spanning-bus Hypercube.

4.2.2 Méthodologie

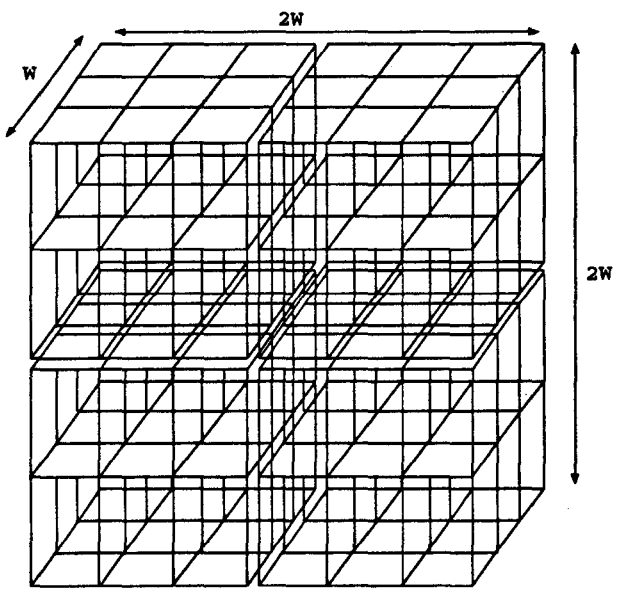


Figure 4.4 : Un réseau de $(2 \times W) \times (2 \times W) \times W$ noeuds formé par quatre spanning bus hypercubes $W \times W \times W$ indépendants

Pour avoir $2 \times W$ noeuds sur une direction, il n'y a pas d'autre moyen que d'utiliser deux guides d'ondes mis bout-à-bout. Les $2 \times W$ noeuds sur une dimension appartiennent ainsi à deux guides différents donnant un réseau formé de quatre spanning bus hypercube indépendants (cf figure 4.4). Il faut alors trouver le moyen de maintenir la connectivité totale (tout noeud pouvant émettre vers tout autre) entre les deux groupes (deux guides d'ondes) ou ensembles disjoints sur une même dimension. Une méthode consiste à adjoindre des guides d'ondes supplémentaires toujours sur la même direction pour créer la connectivité entre les $2 \times W$ noeuds de la direction. Si nous divisons l'ensemble des noeuds connectés à un guide d'ondes en 2 groupes, les $2 \times W$ noeuds de la même direction forment alors 4 groupes et un guide d'ondes permet alors de faire communiquer deux groupes. Pour créer la connectivité totale entre l'ensemble des noeuds,

c'est-à-dire tout noeud pouvant communiquer directement avec un autre sur la même direction, il nous faut autant de guides d'ondes que de deux groupes choisis parmi 4, c'est-à-dire $C_4^2 = 6$ guides d'ondes, chaque noeud étant alors connecté à 3 guides d'ondes sur une seule direction (cf figure 4.5). Pour étendre une direction à $4 \times W$ noeuds (8 groupes), il nous faudra autant de guides d'ondes que de combinaisons de deux groupes choisis parmi 8 c'est-à-dire $C_8^2 = 28$ guides d'ondes, et dans un cas général, pour étendre une direction à $k \times W$ noeuds, il nous faut $C_{2\times k}^2$ guides d'ondes. Le nombre de guides d'ondes à mettre en oeuvre pour étendre un axe tout en conservant l'avantage d'un spanning bus hypercube croît donc de façon exponentielle par rapport à l'incrément en nombre de noeuds.

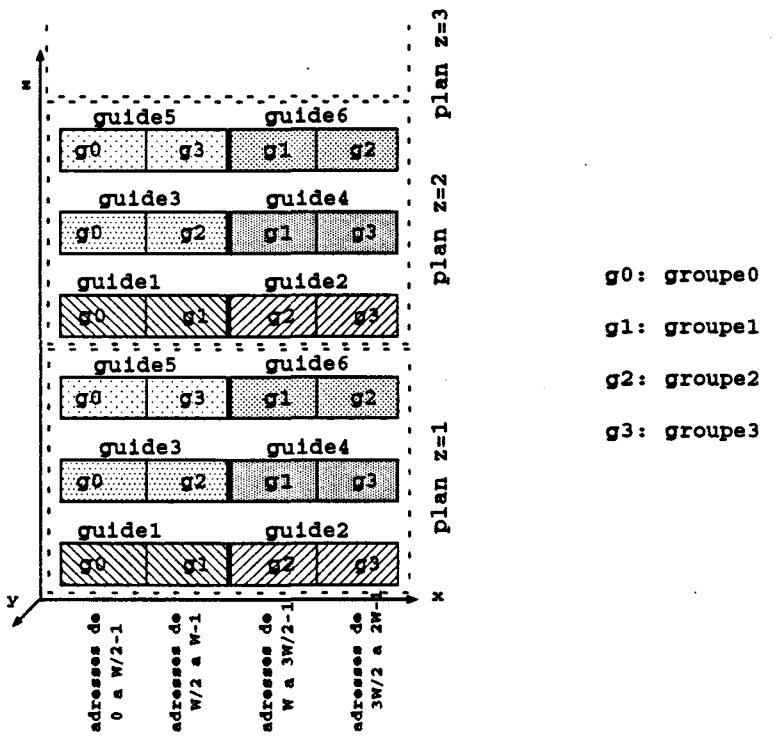


Figure 4.5 : Connectivité totale entre $2 \times W$ noeuds d'un axe utilisant 6 guides d'ondes pouvant connecter W noeuds chacun

Un recouvrement partiel peut être envisagé avec 4 guides d'ondes au lieu de 6 pour la communication entre les $2 \times W$ noeuds d'un même axe, pour diminuer non seulement le nombre de guides d'ondes, mais aussi le degré physique des noeuds qui passe donc de 3 à 2 sur les axes concernés (x et y). La figure 4.6 montre une façon possible de connecter les noeuds. Sur la figure, les noeuds du groupe **ndi** peuvent communiquer directement avec les noeuds du groupe **ndj** puis du groupe **ndl**. Pour communiquer avec un noeud du groupe **ndk**, il faudra un seul intermédiaire qui peut être choisi dans l'un des groupes directement accessibles. Ainsi, les trois quarts des 128 noeuds sont donc toujours directement connectés, un chemin de distance 2 étant requis pour les communications avec le dernier quart.

Il demeure que même le choix d'un recouvrement partiel des noeuds sur les directions comportant $2 \times W$ noeuds présente l'inconvenient majeur d'un doublement des guides d'ondes mis en oeuvre (4 fois plus de guides d'ondes pour connecter 2 fois plus de noeuds : le coût en matériel n'étant pas linéaire en fonction du nombre de noeuds dans le réseau). Un réseau de $W \times W \times W = 2^{3d}$ noeuds (un noeud est donc connecté à $\Delta = 3$ guides et un guide con-

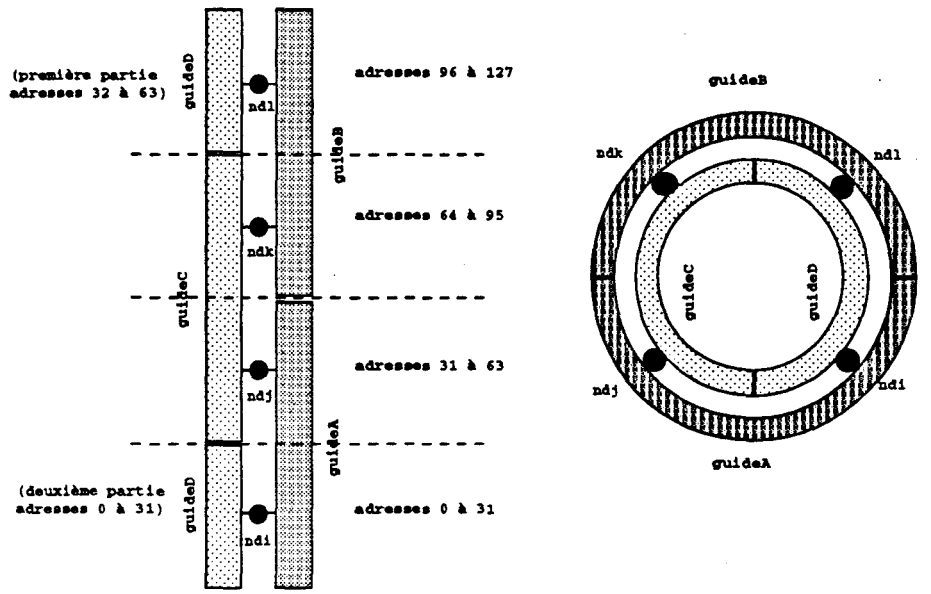


Figure 4.6 : Connectivité presque totale entre les $2 \times W$ noeuds d'une direction x ou y : application pour W=64

necte $W = 2^d$ noeuds) comporte $\frac{2^{\Delta d} noeuds \times \Delta connexions}{2^d connexions/guides} = 2^{(\Delta-1)d} \times \Delta guides = 2^{2d} \times 3$ guides, alors qu'il faut 4 fois plus de guides d'ondes (4 guides d'ondes pour connecter $2 \times W$ noeuds) pour un réseau seulement 2 fois plus grand. Nous avons étudié plus haut le dual-bus hypercube comme solution permettant de réduire le coût du réseau. Nous étudions ci-dessous une approche permettant d'aboutir à un réseau homogène par une simple extension d'une dimension

4.2.2.1 Une classe de réseaux issus du dual-bus hypercube

Pour limiter le coût en nombre de guides d'ondes, nous pouvons utiliser une structure dual-bus hypercube au lieu du spanning bus hypercube. Le diamètre du réseau passe de $D = 3$ pour le spanning bus hypercube à $2 \times D - 2 = 4$ pour le dual-bus hypercube. Sur les directions x et y (comportant quatre guides d'ondes reliant $2 \times W$ noeuds comme représentés sur la figure 4.6), les noeuds sont connectés à 2 guides d'ondes, ce qui diminue le degré physique des noeuds (nombre de ports) qui passe de $\Delta = 5$ à $\Delta = 3$. Nous avons effectué ci-dessus une brève étude du dual-bus hypercube, présenté comme un spanning bus hypercube à faible coût. Mais le facteur important dans notre cas reste la conservation du degré physique des noeuds avec l'inconvénient des quatre guides d'ondes alternés (cf figure 4.3(b)) sur les directions x et y pour chaque plan z, $z \in [0 \ldots W-1]$. Une meilleure redistribution de guides d'ondes dans le réseau nous permettra de pallier cette nécessité de regrouper quatre guides sur une direction.

Le réarrangement de la topologie devant se faire avec un coût en termes de degré et du nombre de guides constant, une étude préalable des contraintes liées aux paramètres du réseau montre que dans tous les cas, l'égalité $N \times \Delta = G \times C$ est toujours vérifiée si N représente le nombre de noeuds total du réseau, Δ le degré physique des noeuds, G le nombre de guides d'ondes dans le réseau et C le nombre de connexions par guides. Cette égalité a une signification physique et montre tout simplement que le nombre total de points de connexions sur les guides d'ondes est nécessairement égal au nombre de ports sur l'ensemble des noeuds.

La formulation du problème de la recherche de la topologie exhibant des performances optimales devient assez aisée, puisque C et Δ sont fixes ($C = W$ et $\Delta = 3$). De plus l'objectif d'intégrer un nombre N d'éléments de calcul nous impose tout simplement de rester dans la limite des G guides au total dans le réseau.

La modification du dual-bus hypercube par éclatement des quatre guides de la direction x sur la direction y (du plan z=0) peut donner une famille de topologies comme celles des figures 4.7. Dans les trois cas, l'égalité $N \times \Delta = G \times C$ est toujours vérifiée et le diamètre du réseau reste toujours le même. Nous allons donc considérer d'autres critères de comparaisons pour le choix de la topologie optimale : La distance moyenne entre deux noeuds quelconques dans le réseau puis la prise en compte de l'algorithme de routage.

4.2.2.2 La distance moyenne

Les comparaisons que nous allons effectuer concerneront les topologies (a) et (b) de la figure 4.7, puisque partant de la topologie (b), une translation dans le plan z=0 suivant la direction x, suivie d'une autre dans le plan z=1 suivant la direction y, permet de retrouver la topologie (c). Il s'agit en fait d'une translation des coordonnées des noeuds modulo W sur les directions x et y. Ces deux topologies (et même plusieurs autres pouvant être générées de la même façon) **sont donc équivalentes et de ce fait appartiennent à la même famille.**

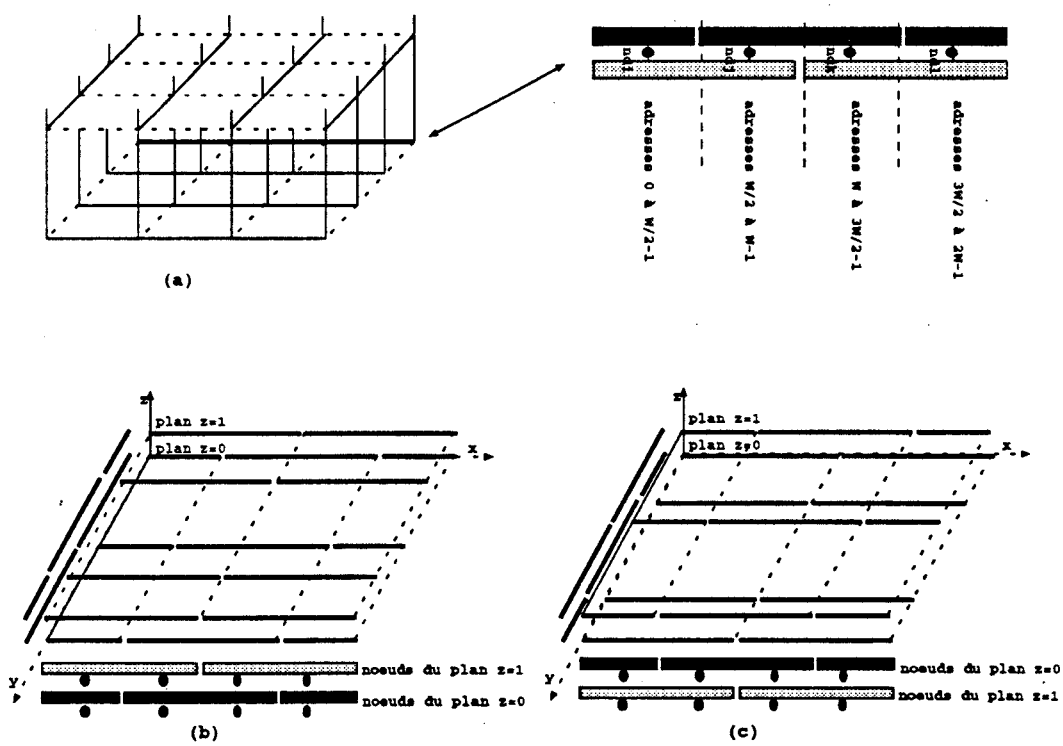


Figure 4.7 : Eclatement des 4 guides d'ondes des directions x et y du dual-bus hypercube (a) pour obtenir une famille de topologies plus régulières (b) et (c)

Dans son expression la plus générale, la distance moyenne Dm d'un réseau quelconque est donnée par la formule $Dm = \sum_{l=1}^{l=lmax} l \times \Phi(l)$, où $\Phi(l)$ est la probabilité pour un message quelconque de traverser l liens, $lmax$ étant le diamètre du réseau. Le routage, que ce soit dans

le cas du spanning bus hypercube ou du dual-bus hypercube est toujours effectué en D étapes (la dimension D=3). Chaque étape consiste en la ***réduction d'une coordonnée x, y ou z*** (cf algorithme de routage proposé dans la section précédente pour le spanning bus hypercube). Dans le cas des deux réseaux présentés ci-dessus, le routage (changement de coordonnées) sur l'axe z se fait toujours en un seul pas, de ce fait, la probabilité d'avoir un seul pas quel que soit le destinataire sur la dimension z est de 1. Seuls les déplacements sur les axes x et y nécessitent suivant la position du destinataire, 1, 2 ou 3 pas. En associant à ces différents déplacements leurs probabilités respectives, on obtient pour les deux topologies, les déplacements moyens sur les axes x et y.

La distance moyenne entre deux noeuds quelconques sur l'axe des x (ou y) pour la topologie (a) (le dual-bus hypercube) est donnée par :

$$\bar{dx}_1 = \frac{1}{2}(\frac{3}{4} \times 1pas + \frac{1}{4} \times 2pas) + \frac{1}{2}(\frac{3}{4} \times 2pas + \frac{1}{4} \times 3pas) = \frac{7}{4}pas$$

Dans le cas ci-dessus correspondant au dual-bus hypercube, dans la moitié des cas, les guides d'ondes existent sur la dimension x (et sont absentes sur la dimension y). Dans ce cas, dans les 3/4 des cas, les deux noeuds en communication sont connectés au même guide d'onde (cf figure 4.4)donc nécessitant un seul pas, et dans 1/4 des cas, un intermédiaire est requis pour envoyer le message à la coordonnée voulue, deux pas étant donc nécessaires. Dans la seconde moitié des cas où les guides d'ondes n'existent pas (les plans z modulo 2 = 1), il faudra au préalable envoyer le message au noeud situé sur le plan au dessus par l'axe z, avant de rerouter le message dans les mêmes conditions que dans le premier cas correspondant au cas où l'axe existe. Dans le calcul de la distance moyenne ci-dessus, un pas supplémentaire est donc pris en compte dans la moitié des cas.

Quand on considère la topologie(b), la distance moyenne est donnée par :

$$\bar{dx}_2 = \frac{1}{2}(1pas) + \frac{1}{2}(\frac{1}{2} \times 2pas + \frac{1}{2} \times 3pas) = \frac{7}{4}pas$$

Le premier pas correspond au cas où les deux noeuds en communication sont connectés au même guide d'onde, la probabilité étant de $W/2W = 1/2$. Dans le cas où le noeud destinataire ne fait pas partie des W premiers correspondant au cas précédent, le message est transmis au noeud juste au dessus sur la direction z, et toujours suivant la position du destinataire, dans la moitié des cas, le message est routé directement (2 pas étant requis alors), soit en passant par un intermédiaire dans la seconde moitié des cas (3 pas).

Nous remarquons que quantitativement, les deux topologies exhibent les mêmes performances : même diamètre et même distance moyenne. L'algorithme de routage discuté dans la section suivante sera donc un facteur important pour le choix d'une topologie.

4.2.2.3 Le routage

Le routage dans un réseau du type dual-bus hypercube est assez aisé et a été largement commenté dans [83]. La principale difficulté dans notre cas concerne le routage sur une direction non totalement connectée, comportant $2 \times W$ noeuds (axes x et y). En considérant le dual-bus hypercube avec un recouvrement partiel des noeuds (figure 4.7(a)), nous remarquons que dans le cas où les supports de connexion (les guides d'ondes) existent, les trois quarts des noeuds sur la même direction sont accessibles en un seul pas, et le dernier quart en deux pas. Un noeud étant connecté à deux guides d'ondes, nous noterons dans la suite **guide x+** et **guide x-**, respectivement le guide sur lequel l'adresse du noeud est strictement inférieure à $W/2$ et

le guide sur lequel l'adresse du noeud est supérieure ou égale à W. En considérant un noeud source quelconque, **ns** voulant émettre vers un noeud destination **nd** sur l'axe x, suivant la position de nd par rapport à ns, c'est-à-dire suivant l'adresse relative de nd **adr(nd)**, tous les cas pouvant se présenter sont :

si adr(nd)∈ [0,W/2-1] alors émettre sur le guide x+ directement

si adr(nd)∈ [-(W/2-1),0] alors émettre sur le guide x- directement

si adr(nd)∈ [W/2,W-1] alors si groupe(ns) est adjacent[1] au groupe(nd), alors émettre directement sur le guide x+, sinon envoyer à un noeud quelconque toujours sur le guide x+

si adr(nd)∈ [-(W-1),-W/2] alors si groupe(ns) est adjacent au groupe(nd), alors émettre directement sur le guide x-, sinon envoyer à un noeud quelconque sur le guide x-

Le noeud quelconque vers lequel il faudra router les données peut être choisi aléatoirement ou fixé, correspondant par exemple à une translation de $W/2$ noeuds sur le même axe.

Le routage pour la seconde topologie envisagée est assez similaire à celui présenté ci-dessus sauf que pour passer d'un guide à un autre toujours sur la même direction, il faudra se servir d'un noeud dans la direction z. De plus il nous faut considérer les adresses physiques des noeuds expéditeur et destinataire, l'adresse relative n'ayant aucune importance parce que si ns et nd ont respectivement pour abscisses 1 et $W-1$, ils peuvent communiquer directement dans la mesure où ils appartiennent au même guide d'ondes, alors que les noeuds d'abcisses $W-1$ et $W+1$ bien que physiquement très proches doivent passer par des intermédiaires parce qu'ils ne sont pas connectés au même guide d'ondes.

4.2.3 Le réseau *hypercom*

Déterminer un algorithme de routage pour la classe de topologies issue du dual-bus hypercube revient à classer les $2 \times W$ éléments d'une direction en groupes, puis à déterminer dans quelles conditions deux groupes peuvent être liés. Cependant, si nous considérons que les W noeuds d'un guide d'ondes constituent un groupe, nous n'avons aucun moyen de créer une connectivité totale entre deux groupes quelconques. Mais si nous supposons qu'un guide d'ondes de W noeuds est constitué de deux groupes, alors nous pouvons lier deux groupes tout simplement parce qu'ils sont connectés au même guide d'ondes.

Sur chaque axe x et y, nous avons ainsi quatre groupes pour lesquels il faudra créer une connectivité, c'est-à-dire tout groupe doit pouvoir être relié à un autre en limitant par exemple la distance entre deux groupes quelconques à 2. Une distance limitée à 1 nous ramènerait à la topologie spanning-bus hypercube initiale dont le coût en nombre de guides d'ondes était exponentiel. La distance moyenne évaluée pour la topologie (b) a mis en exergue le fait qu'il y avait une distance de 3 pas entre certains groupes.

Pour établir la connectivité totale entre les quatre groupes d'une même direction (tout couple de groupes étant connecté), il nous faut autant de liaisons entre groupes que de deux groupes choisis parmi quatre, c'est-à-dire $C_4^2 = 6$. Six liaisons sont donc nécessaires, et puisque chaque plan z permet de connecter 4 groupes en x (ou y) deux à deux, trois plans sont nécessaires sur la dimension z pour disposer d'une connectivité totale entre les groupes d'une même direction. Par exemple sur la figure 4.8, le plan z=0 permet de connecter les groupes 1-2 et 3-4, le plan

[1] les deux groupes possèdent une frontière commune

z=1 permet de connecter les groupes 1-3 et 2-4 et enfin le plan z=2 connecte les groupes 1-4 et 2-3.

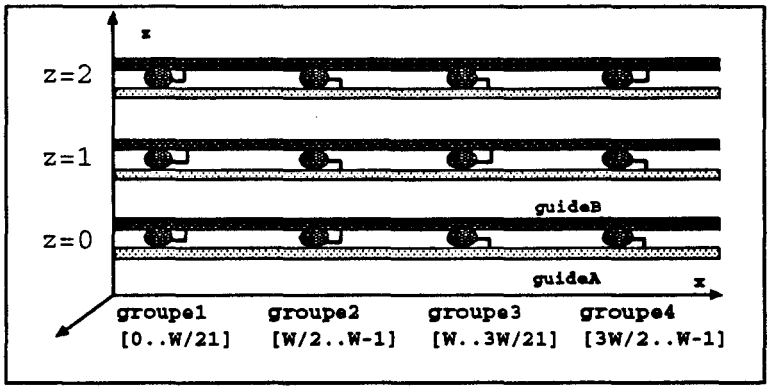


Figure 4.8 : Communication entre groupes sur l'axe des x (ou y)

Le routage dans ce dernier réseau est beaucoup plus aisé que celui présenté plus haut et se rapproche plus de celui du dual bus hypercube. Il suffit dans notre cas de déterminer par quel noeud le message doit passer quand le noeud source et le noeud destination ne sont pas directement connectés. A ce propos, la figure 4.8 montre que :

- quand z mod 3=0, alors les groupes 1 et 2, puis 3 et 4 sont connectés,
- quand z mod 3=1, alors les groupes 1 et 3, puis 2 et 4 sont connectés,
- quand z mod 3=3, alors les groupes 1 et 4, puis 2 et 3 sont connectés.

Alors connaissant le groupe du noeud source **ns** et du noeud destination **nd**, le routage pour la réduction[2] d'une coordonnée x ou y consistera à comparer les deux groupes ; s'ils sont égaux ou si les deux groupes sont connectés dans le plan z, la communication est directe, dans le cas contraire il suffit d'envoyer le message au noeud au dessus ou en dessous sur la direction z à un noeud qui doit être directement connecté à un noeud appartenant au groupe du noeud destinataire.

Le réseau actuel conserve toujours la même valeur du degré ($\Delta = 3$) physique des noeuds par rapport à la classe de réseaux issus du dual-bus hypercube avec recouvrement partiel des noeuds sur les axes, mais la distance moyenne sur les axes x ou y diminue et passe de $\frac{7}{4}$ à :

$$\bar{dx} = \frac{1}{2} \times 1pas + \frac{1}{2} \times 2pas = \frac{3}{2}pas$$

L'expression ci-dessus souligne tout simplement que dans la moitié des cas, la source et la destination sont connectés au même guide d'onde et dans l'autre moitié des cas, la destination est à deux pas, il suffit pour cela de transmettre le message à un seul intermédiaire. Nous avons ainsi supprimé le déplacement nécessitant 3 pas par une meilleure redistribution des adresses des noeuds suivant les plans z=z_0.

Pour transmettre un message du noeud ns de coordonnées (x_0, y_0, z_0) au noeud nd de coordonnées (x_1, y_1, z_1), les trois étapes de l'algorithme de routage dans le réseau global présenté ci-dessus sont les suivantes :

[2] envoie du message à un noeud ayant la même coordonnée que celle du destinataire sur l'axe considéré

routage suivant l'axe x : les déplacements se font dans le plan $y = y_0$:

1- si les groupes des noeuds ns et nd sont connectés dans la direction x, aller au point 2 avec z'=z_0, sinon transmettre dans la direction z à un noeud intermédiaire $n_1(x_0, y_0, z')$ tel que ces deux groupes soient connectés dans la direction x.

2- transmettre dans la direction x au noeud $n_2(x_1, y_0, z')$

routage suivant l'axe y : les déplacements se font dans le plan $x = x_1$:

3- si les groupes des noeuds n_2 et nd sont connectés dans la direction y, aller au point 4 avec z"=z', sinon transmettre dans la direction z à un noeud intermédiaire $n_2(x_1, y_0, z'')$ tel que ces deux groupes soient connectés dans la direction y.

4- transmettre dans la direction y au noeud $n_2(x_1, y_1, z'')$

routage suivant l'axe z : un seul déplacement suivant cet axe :

5- transmettre dans la direction z au destinataire final nd(x_1, y_1, z_1).

4.2.3.1 Généralisation

Dans le cas étudié ci-dessus, nous avions un guide pouvant connecter W noeuds avec un objectif ; celui de pouvoir en connecter $2 \times W$ par direction (x et y), c'est à dire le double. Un guide d'ondes étant divisé en deux groupes, nous avons eu besoin de C_4^2 liaisons entre groupes pour créer la connectivité totale entre les 4 groupes d'une même direction.

Pour étendre le réseau toujours dans un espace physique 3D, nous aurons besoin d'un réseau avec $4 \times W$ noeuds (ou plus) par direction (x, y ou z). Dans le cas de directions de $4 \times W$ noeuds nous aurons besoin de créer la connectivité entre 8 groupes (chacun des 4 guides d'ondes de la direction est toujours composé de 2 groupes). Pour des directions de $8 \times W$ noeuds 16 groupes de noeuds devront être connectés, et dans un cas général, pour $k \times W$ noeuds sur une même direction, il faudra créer la connectivité entre $2 \times k$ groupes. Pour établir la connectivité totale entre k guides d'ondes, il nous faut C_{2k}^2 liaisons. D'autre part, dans chaque plan privilégié (suivant l'axe z par exemple), chaque axe permet d'établir k liaisons deux à deux (chacun des k guides d'ondes connecte 2 groupes différents). Il nous faut donc C_{2k}^2/k plans sur l'axe privilégié (axe z) pour maintenir la connectivité totale sur chaque direction à connecter.

La condition nécessaire et suffisante pour établir la connectivité entre tous les noeuds d'une même direction non totalement connectée est que l'axe privilégié puisse connecter autant de noeuds que de plans nécessaires pour créer la connectivité totale sur les autres axes. Si l'on désigne par W le nombre de noeuds qu'un guide d'ondes peut connecter, la condition nécessaire pour étendre un axe de W à $k \times W$ noeuds s'écrit :

$$W \geq \frac{C_{2k}^2}{k}$$

4.2.3.2 Application

En considérant des guides de $W = 8$ noeuds (chaque noeud pouvant de ce fait émettre à 320 mbps), un réseau de type *dual bus-hypercube* de 2K noeuds peut être construit de deux manières différentes :

- un maillage $W \times (2 \times W) \times (2 \times W)$ dont les directions x et y comportent $2 \times W = 16$ noeuds, donc 2 guides d'ondes. Ici, $k = 2$ et $W = 8$ et nous avons bien l'inégalité $W = 8 \geq \frac{C_{2*k}^2}{k} = \frac{C_4^2}{2} = \frac{6}{2} = 3$.

- un maillage $W \times W \times (4 \times W)$ dont la direction x comporte $4 \times W = 32$ donc 4 guides d'ondes. Dans ce cas-ci, $k = 4$ et $W = 8$ et nous avons bien l'inégalité $W = 8 \geq \frac{C_{2*k}^2}{k} = \frac{C_8^2}{4} = \frac{28}{4} = 7$. La construction du réseau *Hypercom* est donc possible.
  La figure 4.9 montre dans ce cas une distribution possible des connexions sur les 7 plans nécessaires perpendiculaires à l'axe z. Dans le plan z=2 par exemple, le guideA connecte les groupes 0 et 3, le guideB connecte les groupes 1 et 2, le guideC connecte les groupes 4 et 7 et le guideC connecte les groupes 5 et 6.

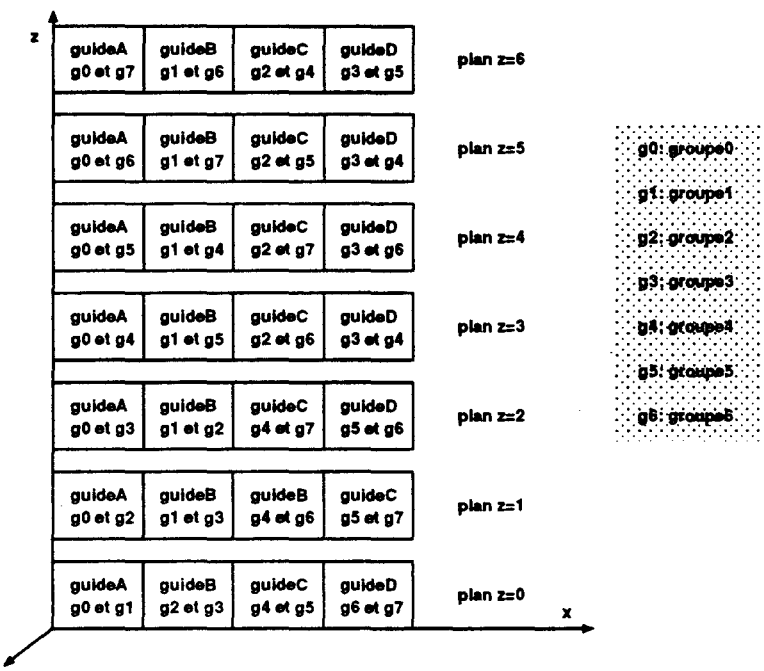


Figure 4.9 : Communication entre 8 groupes sur l'axe des x

4.2.3.3 Le réseau optimal : l'élimination d'un déplacement en z

Pour $k = 2$; Les $\frac{C_{2k}^2}{k} = 3$ plans z ayant servi pour la connexion de l'axe x ont été utilisés pour la connexion de l'axe y. Nous avons ainsi obtenu un réseau de distance moyenne plus faible mais avec un diamètre égal à $2D - 1 = 5$ (D étant le diamètre du réseau Spanning-bus hypercube simulé). Lors de la spécification de la topologie précédente, l'idée sous-jacente était d'effectuer un seul déplacement sur la direction z pour accéder à un plan permettant la connexion directe entre les deux groupes voulant communiquer sur l'axe x, et un autre déplacement si nécessaire pour l'axe y. Cependant, il est possible de combiner en un seul déplacement en z les deux déplacements en z nécessaires pour garantir la connectivité des groupes en x et des groupes en y. Un diamètre de $D + 1$ serait obtenu. Dans ce cas, $(\frac{C_{2k}^2}{k}) \times (\frac{C_{2k}^2}{k}) = 9$ plans sont nécessaires. L'optimisation que nous décrivons ici n'est donc valable que pour

$$W \geq (\frac{C_{2k}^2}{k})^2$$

Pour $W = 16$ et $k = 2$, cette inégalité est vérifiée, permettant par exemple l'extension d'un réseau de 4K noeuds (maillage $16 \times 16 \times 16$) à 16K noeuds (maillage $16 \times (2 \times 16) \times (2 \times 16)$)

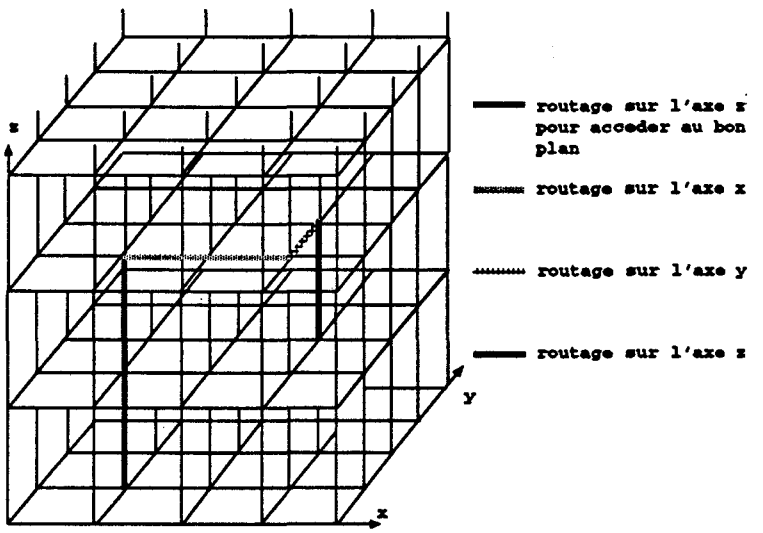


Figure 4.10 : Routage dans le réseau optimal proposé

possédant 4 fois plus de noeuds, avec conservation du degré des noeuds et surtout **presqu'une conservation du diamètre du réseau qui passe de** $D = 3$ **à** $D + 1 = 4$.

Nous donnons ci-dessous la topologie et un exemple de routage pour un réseau de $16 \times (2 \times 16) \times (2 \times 16)$ noeuds ($W = 16$, $k = 2$ et le nombre de plans sur l'axe z nécessaire est $(\frac{C_{2k}^2}{k})^2 = 9$).

La topologie : Un noeud dans le réseau *hypercom* spécifié ci-dessus est identifié par trois coordonnées (ndx, ndy, ndz), et est relié à trois des cinq guides d'ondes passant par sa position. La figure 4.11 montrant une forme possible du noeud de base issue du *méganode* du projet MEGA [31], il faudra trouver suivant sa position (ses coordonnées), les guides auxquels le noeud est relié.

Tout noeud est relié au seul guide d'onde passant par sa position sur la dimension z, mais n'est relié qu'à un seul des deux guides d'ondes sur la direction x (ou y). Nous nommerons dans la suite les deux guides passant par le noeud sur la direction x (respectivement sur la direction y) **guideAx** et **guideBx** (respectivement **guideAy** et **guideBy**). Suivant ses coordonnées sur l'axe des x ou y, le noeud est classé dans le groupe1 si $ndx \in [0 \ldots 7]$, dans le groupe2 si $ndx \in [8 \ldots 15]$, dans le groupe3 si $ndx \in [16 \ldots 23]$, dans le groupe4 si $ndx \in [24 \ldots 31]$(cf figure 4.8). Ainsi, suivant que le plan :

z modulo 9 = 0 , sur la direction x, les noeuds du groupe1 et du groupe2 sont reliés au guideAx, et les noeuds du groupe3 et groupe4 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe2 sont reliés au guideAy, et les noeuds du groupe3 et groupe4 sont reliés au guideBy.

z modulo 9 = 1 , sur la direction x, les noeuds du groupe1 et du groupe3 sont reliés au guideAx, et les noeuds du groupe2 et groupe4 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe2 sont reliés au guideAy, et les noeuds du groupe3 et groupe4 sont reliés au guideBy.

z modulo 9 = 2 , sur la direction x, les noeuds du groupe1 et du groupe4 sont reliés au guideAx, et les noeuds du groupe2 et groupe3 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe2 sont reliés au guideAy, et les noeuds du groupe3 et groupe4 sont reliés au guideBy.

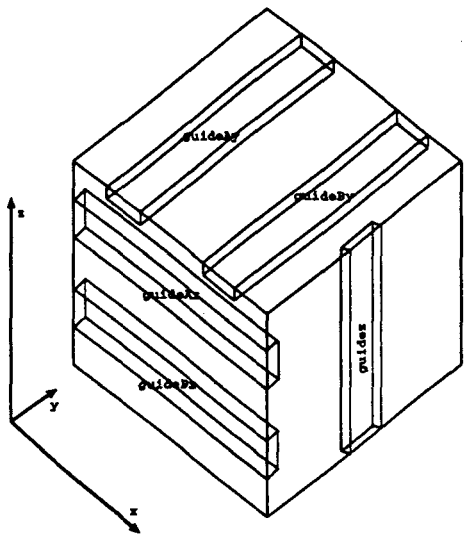


Figure 4.11 : Forme du noeud physique dans le réseau

z modulo 9 = 3 , sur la direction x, les noeuds du groupe1 et du groupe2 sont reliés au guideAx, et les noeuds du groupe3 et groupe4 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe3 sont reliés au guideAy, et les noeuds du groupe2 et groupe4 sont reliés au guideBy.

z modulo 9 = 4 , sur la direction x, les noeuds du groupe1 et du groupe3 sont reliés au guideAx, et les noeuds du groupe2 et groupe4 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe3 sont reliés au guideAy, et les noeuds du groupe2 et groupe4 sont reliés au guideBy.

z modulo 9 = 5 , sur la direction x, les noeuds du groupe1 et du groupe4 sont reliés au guideAx, et les noeuds du groupe2 et groupe3 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe3 sont reliés au guideAy, et les noeuds du groupe2 et groupe4 sont reliés au guideBy.

z modulo 9 = 6 , sur la direction x, les noeuds du groupe1 et du groupe2 sont reliés au guideAx, et les noeuds du groupe3 et groupe4 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe4 sont reliés au guideAy, et les noeuds du groupe2 et groupe3 sont reliés au guideBy.

z modulo 9 = 7 , sur la direction x, les noeuds du groupe1 et du groupe3 sont reliés au guideAx, et les noeuds du groupe2 et groupe4 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe4 sont reliés au guideAy, et les noeuds du groupe2 et groupe3 sont reliés au guideBy.

z modulo 9 = 8 , sur la direction x, les noeuds du groupe1 et du groupe4 sont reliés au guideAx, et les noeuds du groupe2 et groupe3 sont reliés au guideBx. Sur la direction y, les noeuds du groupe1 et du groupe4 sont reliés au guideAy, et les noeuds du groupe2 et groupe3 sont reliés au guideBy.

Un exemple de routage : Le noeud **ns** de coordonnées **(2,29,1)** voulant transmettre un message au noeud **nd** de coordonnées **(10,18,6)**, l'algorithme présenté dans la section précédente procèdera comme suit :

- recherche du plan z=z_0 adéquat pour le routage sur les autres axes x et y
  Le noeud ns(2,29,1) est dans le groupe1 sur la direction x et dans le groupe4 sur la direction y et le noeud nd est dans le groupe2 sur la direction x et dans le groupe3 sur la direction y. De la table des connexions décrite ci-dessus, il en résulte qu'il suffit d'envoyer le message à tout noeud intermédiaire **nil** sur la direction z tel que nil_z **modulo 9 = 0**.
- routage sur l'axe x
  En choisissant nil_z=0, soit **nil** de coordonnées (2,29,0), le noeud nil est relié au noeud nd(10,18,0) à la fois par le guideAx et par le guideBy. le noeud nil(2,29,0) transmet donc le message vers ni2(**10**,29,0) par le guideAx.
- routage sur l'axe y
  puis ni2(**10**,29,0) transmet le message vers ni3(**10**,**18**,0) par le guideBy.
- dernier déplacement sur l'axe z
  Enfin le dernier déplacement sur z : ni3(**10**,**18**,0) transmet le message vers **nd(10,18,6)**.

4.2.4 Utilisation du Store and Forward

Nous avons présenté ci-dessus un algorithme de routage sur la topologie, permettant l'acheminement de messages en $D + 1 = 4$ étapes (recherche du plan adéquat, routage sur les trois axes x,y et z). Si nous reprenons la terminologie (étudiée au chapitre1) associée au routage, plusieurs points doivent être précisés : par exemple la technique de commutation et la gestion des conflits à l'intérieur d'un noeud connecté à trois guides d'ondes.

Pour chacun des choix énumérés ci-dessus constituant la stratégie de routage, se pose outre le problème de leur mode d'implémentation (matériel ou logiciel), la question sur la participation du processeur de calcul dans les mécanismes de communication. Si dans les premières machines parallèles le processeur de calcul participait au moins à l'algorithme de routage (Cosmic Cube [70], IPSC/1 [76]), les machines à passage de messages actuelles (Paragon et les machines à base de T9000 etc.) intègrent dans le noeud de base, en plus du processeur de calcul, un circuit totalement dédié aux communications. Les performances obtenues par l'adjonction d'un circuit de communication sont considérables. W.C. ATHAS et C.L. SEITZ donnent dans [4] un facteur 3 d'ordre de grandeur du temps de latence pour le passage de l'IPS/1 à l'IPS/2, performance due uniquement à l'utilisation de matériel dédié aux communications.

Le temps de latence[3] dans tout le réseau devra tenir compte outre de la longueur L du message, de la distance **d** séparant la source de la destination, c'est-à-dire du nombre de noeuds par lesquels devra transiter le message. Ce temps dépend aussi du mode de commutation choisi comme nous l'avons souligné au chapitre1. Le temps de latence dans un réseau quelconque (utilisant des liens de largeur 1 bit) pour une distance d, un temps de cycle T_c et un message de L bits (dont un en-tête de A bits) est de :

- $t = T_c \times d \times L$ pour un routage *Store and Forward* et
- $t = T_c(d \times A + L)$ pour un routage *wormhole*

On remarque que dans la technique *wormhole*, le temps de latence n'est presque pas influencé par le diamètre du réseau. De ce point de vue, des réseaux à fort diamètre comme les grilles et les réseaux mesh devraient largement en profiter. Dans le cas de notre réseau à très faible

[3]temps total de transmission du message depuis l'envoi de la requête de communication au réseau et la réception du message entièrement par le destinataire

diamètre ($D = 4$), nous sommes très loin du diamètre qu'aurait requis un réseau mesh avec un même nombre de noeuds : pour N=1K noeuds, $D = 32$), le routage *wormhole* n'apporte pas un gain significatif par rapport au *Store and Forward* plus classique.

De plus, le routage *wormhole* est généralement implanté sur des liens ne nécessitant qu'une procedure d'arbitrage très simple : le C104 (cf chapitre1) par exemple utilise une procédure simplifiée pour sélectionner le lien de sortie dès que l'entête du message est disponible sur une entrée. De même dans le cas du réseau mesh de la machine paragon, un lien n'est partagé que par deux noeuds, simplifiant de ce fait l'arbitrage quant à la sélection de l'utilisateur du lien. Dans le cas du réseau *hypercom*, pour implanter efficacement le wormhole, il faudrait que l'unité de communication puisse avoir accès au canal du destinataire avec un temps d'attente presque nul (nous verrons au chapitre 5 que pour chaque noeud, le protocole pour l'accès aux communications sur un guide d'ondes inclut des temps d'attentes liés aux conflits d'occupation de sa propre unité de communication ou de celle du destinataire), ce qui n'est pas le cas pour le réseau hypercom. Un routage Store and Forward peut donc être implanté sur notre réseau sans augmenter significativement le temps de latence, le Store and Forward nécessitant bien-sûr des buffers de taille plus importante.

4.3 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon de différentes topologies pouvant être construites par assemblage de guides d'ondes. Si les *hypernets* et les *hypergraphes* (cf annexe) ont de bonnes propriétés d'extension (pouvant être construits de façon hiérarchique), le *Dual-bus hypercube* possède en plus de très bonnes propriétés algorithmiques (identification des noeuds, routage), une implantation physique beaucoup plus simple. Nous avons ainsi proposé un réseau dérivé du dual-bus hypercube pouvant connecter un nombre quelconque N de noeuds à partir de la double contrainte d'un respect de la dimension physique $D = 3$ et du nombre maximal W de noeuds par guide d'ondes. De plus, les noeuds dans le réseau spécifié ont tous l'avantage d'être homogènes (quelle que soit leur position, ils ont les mêmes caractéristiques quant à la forme (cf figure 4.11)), facilitant de ce fait leur intégration VLSI.

Chapitre 5

Evaluation des performances

Nous avons présenté dans les chapitres précédents un réseau d'interconnexion à base de guides d'ondes pouvant connecter un nombre quelconque de noeuds. Nous proposons dans ce chapitre une modélisation de ce type de réseau en fonction de paramètres architecturaux tels que le débit sur les guides d'ondes et la puissance des noeuds de calcul.

5.1 Choix d'une méthode d'évaluation de performances

Cette section est consacrée à l'étude des performances du réseau présenté dans le chapitre précédent. Différents comportements du réseau devront être analysés en fonction de paramètres caractérisant son état. Ces paramètres étant essentiellement la charge du réseau (fréquence de communication ou fréquence des requêtes émises par les noeuds) et la longueur moyenne des messages dans le réseau. Le **temps d'attente** moyen d'un noeud pour l'accès au réseau à cause des conflits sur les noeuds destinataires et la **bande passante utile** (ou le débit global) du réseau représentent deux mesures de performances importantes dans les systèmes multiprocesseurs [20]. En plus de ces deux mesures importantes, nous essayerons de déterminer d'autres caractéristiques relatives aux réseaux *multicanaux* telles que le taux de requêtes de communication satisfaites sans conflits.

Il existe traditionnellement trois méthodes utilisées pour analyser les performances d'un système : la modélisation (réseaux de Petri, chaînes de Markov, files d'attentes, etc.) à laquelle on peut apporter plusieurs méthodes de résolution, la simulation et les benchmarks. Actuellement les systèmes informatiques sont très complexes à évaluer et un certain nombre d'hypothèses d'approximations sont nécessaires avant une étude des performances. On peut ainsi supposer que les requêtes des processeurs sont uniformément distribuées, alors que dans la réalité, nous observons assez souvent des communications localisées. Ces hypothèses peuvent donc invalider le modèle si elles sont trop sévères. Il est donc souhaitable d'effectuer plus d'une modélisation avec des méthodes différentes avant d'apporter une interprétation sur les résultats obtenus.

5.1.1 La résolution analytique et par simulation

L'intérêt d'une résolution analytique tient d'une part au fait qu'elle est moins complexe à mettre en oeuvre (comparée aux programmes complexes de simulations parfois nécessaires) et d'autre part au fait qu'elle offre un cadre de vérification beaucoup plus rigoureux que les programmes de simulation. Cependant, dès que le système à étudier devient assez complexe nécessitant par exemple des synchronisations entre divers constituants du système ou la concurrence pour l'accès à des ressources partagées (gestion de sémaphores), la résolution analytique devient moins adaptée, d'une part parce qu'elle serait très complexe à effectuer et d'autre part parce qu'elle exigerait des simplifications trop importantes pouvant invalider le modèle.

Une modélisation analytique nous permettra sous certaines contraintes d'effectuer les premières mesures, dans le cas où tout noeud possède deux unités d'émission/réception (permettant alors de ne pas occuper le circuit de réception pendant les émissions de messages). Les méthodes de résolution utilisées ici peuvent être trouvées dans [3] [19]. Des simulations du réseau à l'aide du simulateur *QNAP2* fondé sur les files d'attentes viendront confirmer ces résultats (en ce sens où une description plus fine du comportement du réseau sera faite et la méthode de résolution différente). Dans cette section, nous faisons donc usage conjointement de deux méthodes de résolution, modèle analytique et simulation, pour étudier le comportement du système. Nous validerons par ainsi le simulateur qui sera utilisé pour analyser le second système.

Nous décrivons ensuite le système dans lequel tout noeud ne possède qu'une seule unité d'émission/réception. Dans ce modèle, l'émission et la réception sont liées, puisque partageant la même unité : pendant les réceptions de messages, les émissions sont interrompues et vice-versa. Pour des raisons évidentes de complexité du système, une description assez détaillée du modèle de simulation sera présentée pour l'étude de ce dernier système.

Enfin les valeurs issues des mesures de temps de réponse des composants physiques mis en oeuvres (guides d'ondes, unité de routage) nous permettront de donner un point de vue sur :

- les domaines de bonne utilisation du réseau construit à base de guides d'ondes,
- la nécessité de l'utilisation de deux unités de communication
- les performances par rapport aux réseaux actuels

5.2 Performances du système à deux unités de communication

Dans les systèmes synchrones, les requêtes de communication peuvent être modélisées par la probabilité qu'un processeur génère une requête à chaque début de cycle. Ces requêtes peuvent donc être modélisées par une séquence de suites de Bernouilli indépendantes, p désignant la probabilité qu'un processeur émette une requête et $1 - p$ le cas contraire. Mais dans les systèmes asynchrones, les requêtes peuvent être générées à n'importe quel moment dans le temps ; les composants du système n'ayant pas d'horloge commune. Cependant, une distribution exponentielle des requêtes est généralement considérée [7], ce qui suppose que l'intervalle de temps séparant deux requêtes successives vers un noeud est une variable aléatoire suivant une loi exponentielle.

Disposant de deux unités de communication, un noeud peut recevoir un message même s'il est en émission. Pour modéliser les temps d'attente des messages pour lesquels le noeud est

destinataire, il nous faut décrire en termes de stations et de files d'attentes associées, les différentes unités constituant le système : le processeur comme une source de requêtes, l'unité de décision comme unité chargée de gérer toutes les requêtes issues du processeur et l'unité de communication qui modélise l'occupation du canal.

- Le processeur est vu ici uniquement comme une source de requêtes pour satisfaire les exigences en communication des applications, nécessitant par exemple en moyenne un envoi d'un certain nombre de messages toutes les unités de temps. Si une requête n'est pas satisfaite, elle n'influe pas sur la génération de la suivante. Toute requête générée par le processeur possède un champs supplémentaire : le numéro du noeud destinataire du message. L'algorithme du choix du processeur destinataire lors de la génération des requêtes est implanté de façon indépendante et peut être modifié pour étudier le comportement du réseau pour de différents modes de programmation : communication uniforme, optimale (exemple des applications régulières où les schémas de communications sont connus et sont arrangés de manière à éviter toute attente), diffusion.
- L'unité de décision se charge de prendre en compte toutes les requêtes de communication générées par la source (le processeur associé) puis vérifie si le canal du destinataire est libre. Si celui-ci est libre, la requête est envoyée à l'unité de communication (réception) du destinataire. La file d'attente des messages générés est gérée en FIFO par l'unité de décision. Ceci nous permet de calculer les temps d'attentes moyens par message.
- L'unité de communication se charge de :
  - recevoir les requêtes venant des autres unités de décision,
  - signaler son état d'occupation en positionnant son drapeau *non-comm* à faux
  - procéder à la communication effective pendant un temps constant (le temps de service correspondant à l'envoi du message).
  - réinitialiser son drapeau *non-comm* à vrai pour une prochaine communication.

5.2.1 Distribution uniforme des requêtes

Dans ce modèle, nous supposons que les requêtes générées par les processeurs possèdent une distribution aléatoire et uniforme quant au choix du destinataire, c'est-à-dire que pour un ensemble de W noeuds connectés à un guide d'ondes, les destinataires pour un noeud quelconque sont choisis de façon équiprobable parmi les W-1 noeuds restants.

5.2.1.1 Modèle analytique

Considérons un guide d'ondes de **W** noeuds, chaque noeud émettant un message tous les $E[\tau] = 1/\lambda$ secondes et une durée moyenne de traitement d'un message de $E[s] = 1/\mu$ (cette grandeur quantifie en fait le temps de transmission du message et inclut donc le paramètre longueur du message). λ et μ sont exprimés en nombre de messages par seconde.

Sous les hypothèses d'une distribution uniforme des requêtes, les $W \times \lambda$ requêtes sont destinées aux W noeuds connectés, c'est-à-dire chaque noeud reçoit λ requêtes par secondes (ou reçoit une requête en moyenne toutes les $1/\lambda$ secondes) (cf figure 5.1(a)). L'état du guide d'ondes est ramené à un ensemble de W systèmes de files d'attente Markoviennes M/M/1 indépendants dans lequel chaque noeud destinataire constitue un système (cf figure 5.1(b) et figure 5.2). La

probabilité qu'un noeud reçoive m requêtes pendant un intervalle de temps t est donnée par la *loi de Poisson* :

$$P(m,t) = \frac{e^{-\lambda t}(\lambda t)^m}{m!}$$

De plus, si l'intensité du trafic est définie comme étant le rapport du temps moyen de service (durée moyenne pour l'émission d'un message : $E[s] = 1/\mu$) et du temps moyen séparant deux requêtes ($E[\tau] = 1/\lambda$), le nombre moyen de requêtes en attente (y compris celle en cours de traitement) vers un noeud quelconque à l'état stationnaire s'écrit :

$$N = \rho/(1-\rho)$$

avec $\rho = u/c$ et $u = \lambda/\mu$; u caractérisant l'intensité du trafic, ρ le taux d'occupation du serveur et c le nombre de serveurs, égal à 1 dans notre cas. La quantité ρ doit être inférieure à 1 pour que le nombre de requêtes n'augmente pas sans cesse : le temps de traitement d'une requête serait alors trop élevé par rapport à la fréquence d'arrivée des messages, et les temps d'attente pour l'accès au réseau seraient alors proches de l'infini pour un temps assez long d'utilisation du réseau.

Le temps moyen de traitement d'une requête, y compris le temps d'attente avant le service est donné par la *formule de Little* :

$$T = \frac{N}{\lambda} = \frac{\rho/\lambda}{1-\rho} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu-\lambda}$$

Le temps d'attente moyen T_w d'un message est déduit du temps total de traitement en y soustrayant le temps de service d'un paquet qui est égal à $1/\mu$.

$$T_w = \frac{1}{\mu-\lambda} - \frac{1}{\mu}$$

Si les caractéristiques matérielles telles que le débit sur le guide d'ondes et la puissance des processeurs (le nombre de requêtes de communication par seconde est proportionnel à la vitesse des microprocesseurs) influent largement sur le *temps de latence*, les performances globales du réseau sont étroitement liées aux considérations purement logicielles telles que la longueur des messages (le réseau est occupé plus longtemps) et la fréquence des communications (rapport entre temps moyen passé en communication et temps moyen passé en traitement local). Ces deux notions caractérisent le grain de l'application : Un grain fin nécessitant des communications fréquentes (de petits messages) caractérise les applications du type *i/o-bound* par opposition au gros grain caractérisant les applications du type *compute-bound* [14]. Une brève étude est faite à ce propos en fin de chapitre.

Si nous désignons par **I** le nombre d'instructions exécutées localement par un processeur avant d'émettre un message, et **P** la puissance du processeur donnée en Mips[1], le rapport **I/P** donne le temps moyen entre deux requêtes de communication successives, et de ce fait, $\lambda = P/I$. D'autre part, si **D** est le débit sur un guide d'ondes exprimé en Mbps[2], le temps de traitement d'un message de longueur L est L/D, offrant une fréquence de traitement des messages de $\mu = D/L$.

Nous avons souligné ci-dessus que taux d'occupation des serveurs (unité de communication) ne doit pas être supérieur à 1. Et ce taux $\rho = u/c$ est directement lié au paramètres L et I :

$$\rho = \frac{E[s]}{E[\tau]} = \frac{\lambda}{\mu} = \frac{P/I}{D/L} = \frac{P}{D} \times \frac{L}{I}$$

[1] millions d'instructions par seconde
[2] mégabits par seconde

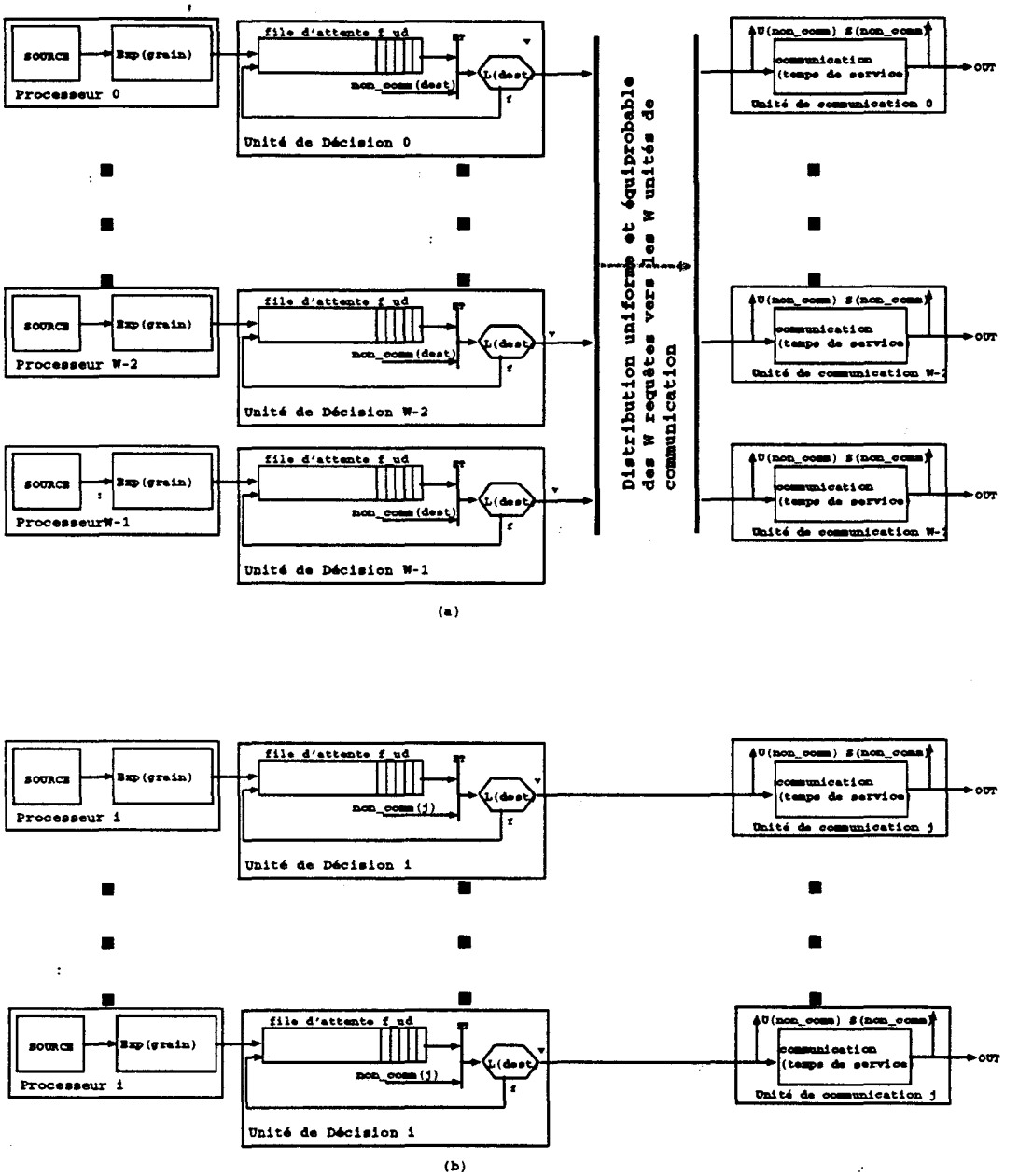


Figure 5.1 : Modélisation des unités fonctionnelles du système à deux unités de communication : (a) W noeuds connectés à un guide, (b) W systèmes indépendants

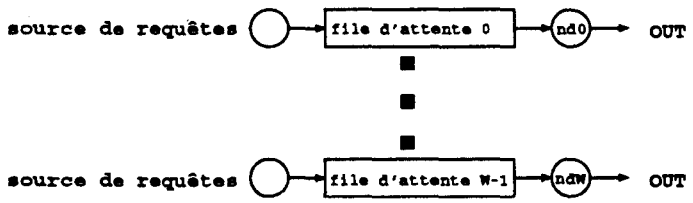


Figure 5.2 : Modélisation analytique des requêtes arrivant à un noeud avec une distribution uniforme des destinataires : bande passante moyenne.

$$\rho < 1 \Longleftrightarrow L/I < D/P$$

Ainsi, pour P et D fixés, le rapport D/P nous fournit une première limite pour des choix judicieux de L et I pour la simulation.

D'autre part, pour étudier les performances du système, les seuls paramètres importants en entrée sont :

- le taux d'arrivée des requêtes de communication : $\lambda = P/I$ et
- le taux de traitement des requêtes : $\mu = D/L$.

En utilisant directement ces rapports, on évite de figer la simulation pour un débit donné sur le guide d'ondes ou pour un processeur d'une puissance donné. De plus, en utilisant ces paramètres qui représentent en fait des rapports, nous obtiendront des résultats non pas pour un ensemble de valeurs données mais pour un ensemble de couples de valeurs. Ainsi, par exemple pour une valeur de $\lambda = 10^5$ messages par secondes, I est égal à 100 si le processeur est un T800 (de 10 mips), I sera égal à 500 pour un processeur $i860XP$ (50 mips) et I sera de l'ordre de 1000 pour un processeur Alpha dont la puissance moyenne sur l'application est de 100 mips. Cependant, pour mieux appréhender les résultats issus des simulations, les paramètres comme I (le nombre d'instructions exécutées localement avant l'émission d'un message) et L (la longueur moyenne d'un message traité) nous semblant être mieux adaptés, nous donnerons donc après chaque étude, un exemple d'applications en considérant des valeurs constantes de de **P, D et L**. D'autre part, si le nombre de requêtes en attente de service N peut être obtenu uniquement par la connaissance du taux d'occupation du serveur (le canal de communication), le temps d'attente issu de la formule de *Little* ne peut être obtenu sans la connaissance du temps de service $1/\mu$ ou du temps séparant deux requêtes de communication $1/\lambda$. Nous ferons donc nos évaluations pour différentes valeurs de I, en fixant des paramètres suivants :

- P=100 Mips, correspondant à la puissance moyenne des processeurs actuels (on peut bien imaginer une grappe de processeurs possédant cette puissance),
- D=300 Mbps, correspondant à un guide d'ondes connectant 8 processeurs de la puissance donnée ci-dessus,
- L=1Kbits, choisi plutôt arbitrairement, puisque nous verrons plus loin l'influence de la longueur des paquets. De l'étude faite au chapitre 3, L=1Kbits nous fournit un temps de latence hors conflit de $3,4\mu s$ et un temps d'envoi de l'entête de $0,44\mu s$.

Les caractéristiques suivantes seront évaluées :

- T_w : le temps d'attente moyen pour l'accès à un canal

- $\tau_{UC} = \rho$: le taux d'occupation de l'unité de communication. Ce taux nous permettra plus tard de déduire le débit moyen du guide d'ondes[3]
- τ_{direct} : le taux moyen de requêtes qui accèdent directement au destinataire sans attente, c'est-à dire sans conflits (occupation de son unité E/R ou de l'unité E/R du destinataire)

Ainsi, pour un noeud ayant une puissance de traitement d'environ 100 Mips connecté à un guide d'ondes, et les valeurs de D=300Mbits par seconde, nous obtenons analytiquement un temps de traitement global du message de $T = 5,38\mu s$ et un temps d'attente pour l'accès au réseau de $T_w = 1,88\mu s$ pour I=1000 instructions exécutées localement avant chaque communication de message de L=1024 bits environ.(cf table5.1).

Tableau 5.1 : Temps d'attente T_w en μs en fonction du taux de communication I sur un guide d'onde (le temps de service $1/\mu = L/D = 3,5\mu s$ et la durée moyenne entre deux requêtes $1/\lambda = I/P = I/10^8$s)

I	350	400	500	700	1000	1500	2000	3000	5000
$1/\lambda$ en μs	3,50	4,00	5,00	7,00	10,00	15,00	20,00	30,00	50,00
$\rho = \frac{\lambda}{\mu}$	1	0,88	0,70	0,50	0,35	0,23	0,17	0,12	0,07
N	∞	7,00	2,33	1,00	0,54	0,30	0,21	0,13	0,08
T	∞	28,00	11,67	7,00	5,38	4,57	4,24	3,96	3,76
T_w	∞	24,50	8,17	3,50	1,88	1,07	0,74	0,46	0,26

5.2.2 Distribution optimale des requêtes de communication

Nous considérons toujours dans ce cas un guide d'ondes de W noeuds, chacun générant λ_i requêtes par seconde ($\lambda = W \times \lambda_i$). Mais cette fois-ci, nous considérons que les requêtes sont distribuées de manière aléatoire et équiprobable vers l'ensemble des W noeuds, en tenant compte de l'état d'occupation des noeuds (cf figure 5.3), c'est à dire que si le message est destiné indifferemment à un noeud parmi un groupe donné, il est envoyé au premier noeud libre dans le groupe. L'algorithme du choix du noeud destinataire a donc été modifié pour déterminer au mieux le destinataire, c'est-à-dire le premier canal libre.

Contrairement au modèle précédent, l'état des files d'attente devant chacun des noeuds est pris en compte lors de l'affectation des requêtes aux différents serveurs (les noeuds destinataires). Ce cas correspond à la communication par groupe : le message est envoyé à un noeud d'un groupe pour le routage et non à un noeud précis, par exemple dans le cas de la réduction de l'axe x, le message peut d'abord être envoyé à un noeud quelconque choisi parmi un groupe donné[4]. Une bonne répartition des communications (ou à l'extême, un algorithme systolique ou pipeline) devrait permettre ce type de communications. Le modèle ainsi ramené à un système unique de files d'attentes à W serveurs (M/M/W) est représenté par le schéma de la figure 5.3.

Le modèle actuel n'est qu'une généralisation du modèle étudié dans la section précédente (nous passons d'un modèle à serveur unique à un modèle à serveur multiple). Nous considérons toujours que l'ensemble des noeud génèrent un message tous les $E[\tau] = 1/\lambda$ secondes et une durée moyenne d'acheminement d'un message de $E[s] = 1/\mu$. La génération des requêtes est toujours considérée comme poissonienne.

[3] $debit = \tau_{UC} \times D \times 8$, avec D=320 Mbps

[4] ceux par exemple dont la coordonnée z mod 3 = 0

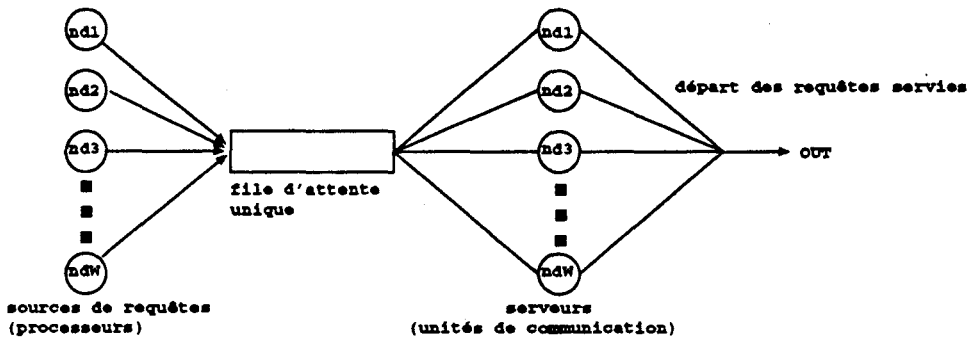


Figure 5.3 : Modélisation des requêtes générées sur un guide d'ondes : bande passante optimale

L'intensité du traffic est donnée par $u = E[s]/E[\tau] = \lambda/\mu$. Cette mesure donne le nombre minimal de serveurs requis pour satisfaire le taux d'arrivée des requêtes. Le taux d'utilisation des serveurs $\rho = u/W$ caractérise, quant à lui, la probabilité qu'un serveur soit occupé et doit être inférieur à 1 pour que le nombre de messages n'augmente pas sans cesse, nous obtiendrions dans ce cas des temps d'attente infinis.

Le temps d'attente moyen pour l'accès à un canal (ou à une communication) est donnée par la formule (issue de [3])

$$T_w = \frac{C(W,u)E[s]}{W(1-\rho)}$$

avec $C(W,u) = \frac{\frac{u^W}{W!}}{\frac{u^W}{W!}+(1-\rho)\sum_{n=0}^{W-1}\frac{u^n}{n!}}$ la formule d'*Erlang*.

Pour W=8, nous obtenons des temps d'attente proches de zéro (cf tableau 5.2) en considérant les mêmes caractéristiques physiques des noeuds de la première modélisation (P=100 mips, D=300 mbps et des messages de 1Kbits), et ceci quel que soit le grain de parallélisme visé (en respectant toutefois les débits sur le guide d'ondes qui sont actuellement de l'ordre de 300 mégabits par seconde pour un réseau de 8 noeuds). Le temps d'attente étant presque nul, une communication sur un guide d'onde requiert uniquement le temps de transfert de tout le message. Le temps de latence dans un réseau global de $8 \times 8 \times (8 \times 2) = 1$K noeuds toujours pour un message de longueur L=1Kbits et un routage *Store and Forward* est donc très proche de quatre fois le temps de transfert sur un guide d'ondes, c'est-à-dire $3,5\mu s \times 4 = 14\mu s$ (cette étude est effectuée plus loin dans la section 5.5).

5.2.3 Une trace des communications entre noeuds

Nous avons fait une étude du comportement du réseau réduit à un guide d'ondes, en déterminant les paramètres comme le temps d'attente ou la charge du réseau à l'état stationnaire, c'est à dire après un temps relativement long d'utilisation. Pour mieux illustrer les performances du réseau (par simulation), nous avons déterminé une trace de la charge en communication du guide d'ondes, évidemment après avoir fixé certains paramètres comme le grain de l'application (I ou λ). La figure 5.4 montre l'évolution en *μsecondes* de la charge pour des valeurs de I=500 et 1000.

Pour I=1000, après un court moment de quelques *μsecondes*, le nombre de noeuds en réception se stabilise autour de 2,69, correspondant exactement au nombre de communications en cours. Les notions de taux d'occupation des unités de communication et du nombre moyen d'unités en

Tableau 5.2 : Temps d'attente T_w en μs en fonction du taux de communication I sur un guide d'onde (le temps de service $1/\mu = L/D = 3,5\mu s$ et la durée moyenne entre deux requêtes $1/\lambda = I/W.10^8$) de W=8 noeuds

I	350	375	400	450	500	600	800	1000	5000
$1/(\lambda/W)$	3,50	3,75	4,00	4,50	5,00	6,00	8,00	10,00	50,00
$u = \frac{\lambda}{\mu}$	8,00	7,44	7,04	6,24	5,60	4,64	3,52	2,80	0,56
$\rho = \frac{u}{W}$	1,00	0,93	0,88	0,78	0,70	0,58	0,44	0,35	0,07
$C(W,u)$	1,00	0,80	0,64	0,41	0,27	0,12	0,03	0,01	0,00
T	ND	8,72	5,72	4,31	3,89	3,63	3,52	3,51	3,50
T_w	ND	5,22	2,22	0,81	0,39	0,13	0,02	0,01	0,00

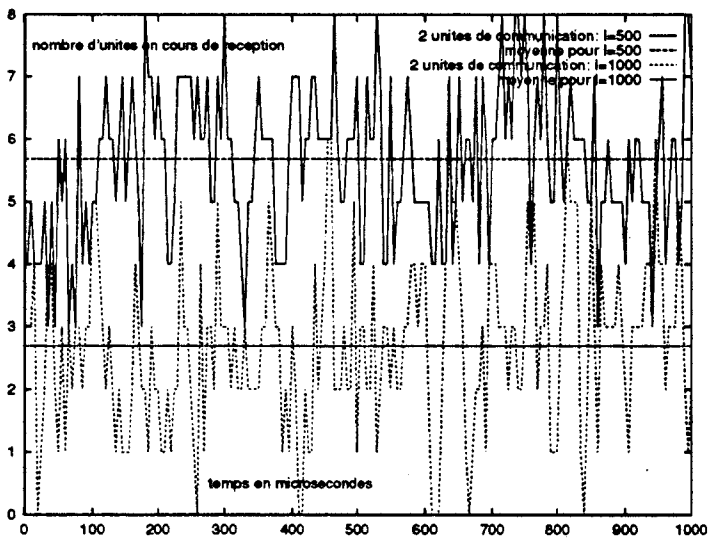


Figure 5.4 : Simulation : système à deux unités de communication : évolution temporelle de la charge du réseau pour I=1000 et I=500

cours de communication sont bien évidemment très liées, l'une permettant de vérifier l'autre : le taux d'occupation de $\rho = \lambda/\mu$=0,35 prévu pour pour I=1000 (cf tableau 5.1) est approché par le taux des unités en communication qui est de $2,69/8 \simeq 0,35$. De même pour I=500, les 5,68 communications en moyenne au cours du temps donnent un taux de 5,68/8 approchant le taux d'occupation des unités de réception qui est de $0,70 = \rho$.

5.3 Performances du système à une seule unité de communication : simulation

Nous avons étudié dans les deux sections précédentes deux modèles de distribution de requêtes ; abstraction faite des cas de collisions liées au partage de l'unité de communication (une unité servait à l'émission et une autre à la réception). L'intérêt de ces deux études précédentes est qu'elles sont simples à modéliser de façon analytique, résultats que nous avons étayé par des simulations logicielles. La prise en compte du protocole réel de la communication en prenant en

compte les cas de collisions et de conflits d'accès au circuit d'émission/réception (unique dans le système actuel) induit une complexité supplémentaire quant à la modélisation d'un noeud. Un noeud quelconque peut être dans quatre états différents dépendants de plus des autres noeuds connectés au guide d'ondes :

- l'état libre,
- l'état occupé en communication : expéditeur,
- l'état occupé en communication : destinataire,
- l'état demande de communication.

De plus, un noeud dans un état *demande de communication* peut être pris en destinataire pour un autre message si son destinataire est occupé pendant ce temps en communication. La requête courante (demande de communication) devra donc être mise en attente et être prise en compte dès la fin de l'occupation de l'unité de communication. Nous nous attacherons donc dans ce qui suit à décrire très précisément tout le modèle en termes de stations et de files d'attentes pour une implantation sous *QNAP2*. Nous commenterons ensuite les résultats issus de la simulation.

5.3.1 Le logiciel QNAP2

QNAP [73] (Queueing Network Analysis Package) est un logiciel de simulation fondé sur la théorie des files d'attente. Son utilisation dans le cadre de la simulation de notre réseau vient d'une part de la facilité de description des modèles dans ce langage, et d'autre part de la multiplicité des méthodes de résolution proposées.

L'écriture d'un service sous *QNAP* est composée de deux parties :

- une partie description d'un ensemble de *stations* entre lesquelles circulent des clients (requêtes...). Une station est composée des éléments suivants :
  - une file d'attente,
  - un ou plusieurs serveurs attachés à cette file,
  - la description algorithmique du service fourni et un algorithme de routage qui permet de définir la circulation des clients dans le réseau.
- une partie résolution dans laquelle un large choix des méthodes de résolution est proposé :
  - un simulateur à événements discrets,
  - des méthodes analytiques exactes (algorithmes de convolution, résolution Markovienne),
  - des méthodes analytiques approchées (méthodes itératives, approximation par diffusion, approches heuristiques).

Le logiciel QNAP fournit plusieurs primitives permettant de réaliser un très grand nombre de routages différents : conditionnel, probabiliste, mécanisme de barrière de synchronisation etc (cf figure 5.5). La principale difficulté d'une modélisation sous QNAP réside essentiellement dans la description du modèle sous la forme d'une structure de réseau de files d'attente, la même approche que dans le cas de la résolution analytique.

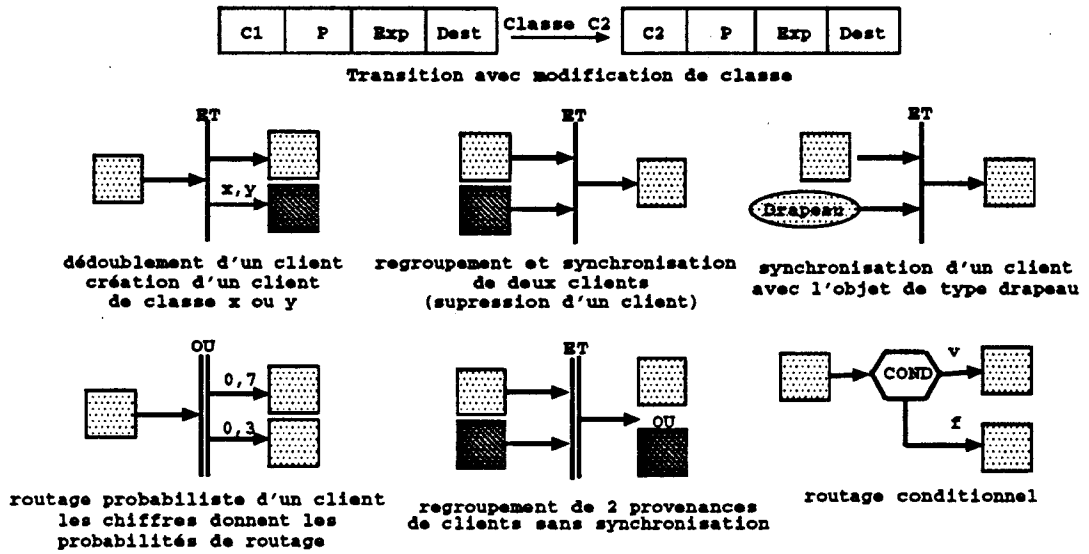


Figure 5.5 : Les routages principaux sous QNAP

Le simulateur à événements discrets est une méthode de résolution basée sur une reproduction du comportement des différents composants du modèle (stations etc.). Elle est dirigée par des séquences de nombres aléatoires (représentant les temps de service, les probabilités de transitions etc.) générés par la machine suivant des distributions spécifiées par l'utilisateur. Ce simulateur à événements discrets est donc proche d'un simulateur réel, facilitant de ce fait le développement de programmes de simulation. Il faut aussi noter que ce simulateur proposé par QNAP2 a été utilisé pour l'étude de comportement de machines parallèles telles que la machine DASH ou M3S [65].

Modéliser les communications sur un guide d'ondes revient à décrire par un ensemble de stations les différentes unités : le processeur, l'unité de gestion de communication (l'unité de décision : UD) et l'unité de communication.

5.3.2 Le processeur

Le processeur est toujours décrit de la même manière que dans les deux modélisations précédentes. Un processeur est donc vu comme une source émettant des requêtes de communication. La requête est l'entité *client* de base circulant dans le réseau, et prend une coloration donnée au cours de son cycle de vie (les colorations sont exprimées par la notion de classe en QNAP). Les différentes classes pour un client dans notre cas sont les suivantes :

- la classe *demande* : la requête vient d'être générée par la source et demande donc de ce fait une communication.
- la classe *non-dest* : le destinataire est occupé par une autre communication, la requête revient dans la file d'attente avec la classe *non-dest.*
- la classe *émis* : la requête a été satisfaite (destinataire libre) et est en cours de communication en tant qu'expéditeur.

- la classe *recept* : le client (initié par un expéditeur quelconque) est en cours de communication en tant que destinataire.

Une requête possède deux champs supplémentaires : le numéro du processeur expéditeur et celui du destinataire.

L'algorithme du choix du processeur destinataire lors de la génération de la requête est implanté de manière indépendante et peut être modifié pour étudier le comportement du réseau pour différents modèles de programmation : communication uniforme, diffusion, etc. Pour notre étude ici, nous considérons une distribution purement aléatoire et uniforme quant au choix du destinataire.

La fréquence de génération des requêtes est toujours déterminée par une loi aléatoire exponentielle de paramètre λ telle que l'intervalle moyen entre 2 requêtes successives corresponde au grain de parallélisme de l'application, c'est-à dire au rapport entre le temps de communication et le temps de traitement. La figure 5.6 montre la modélisation d'un processeur dans le réseau.

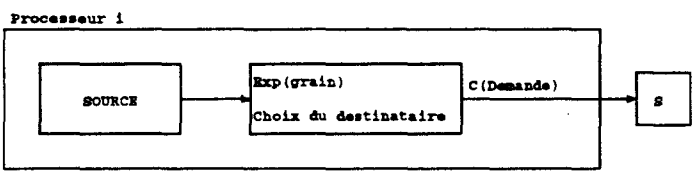


Figure 5.6 : Modélisation d'un processeur

5.3.3 L'unité de décision : l'UD

L'unité de décision se charge de prendre en compte toutes les requêtes de communication générées par la source (le processeur associé) puis effectue les opérations suivantes :

- vérification de la disponibilité de l'unité de communication du noeud associé : le port associé pouvant être en cours de communication. L'automate décrit à la figure 3.8 possède un état représentant cette tâche.

- vérification de la disponibilité de l'unité de communication du noeud destinataire. Cette opération correspond à la phase d'écoute de la fréquence du noeud destinataire.

- si l'une des deux unités de communications est occupée, la requête est mise en attente dans la file d'attente associée à l'unité de décision, et n'influe guère sur la fréquence de génération des requêtes par le processeur associé. Cette propriété est nécessaire pour satisfaire l'exigence sur la constance du grain de l'application.

Dans le cas où les deux unités de communications sont disponibles (représentés par le routage ET QNAP), l'UD envoie la requête courante dans l'unité de communication associé au noeud, puis crée un nouveau *client* de classe *recept* dans l'unité de communication du noeud destinataire. Cette façon de procéder rend compte de l'occupation simultanée des deux unités de communication. La figure 5.7 montre la modélisation d'une unité de gestion de communication.

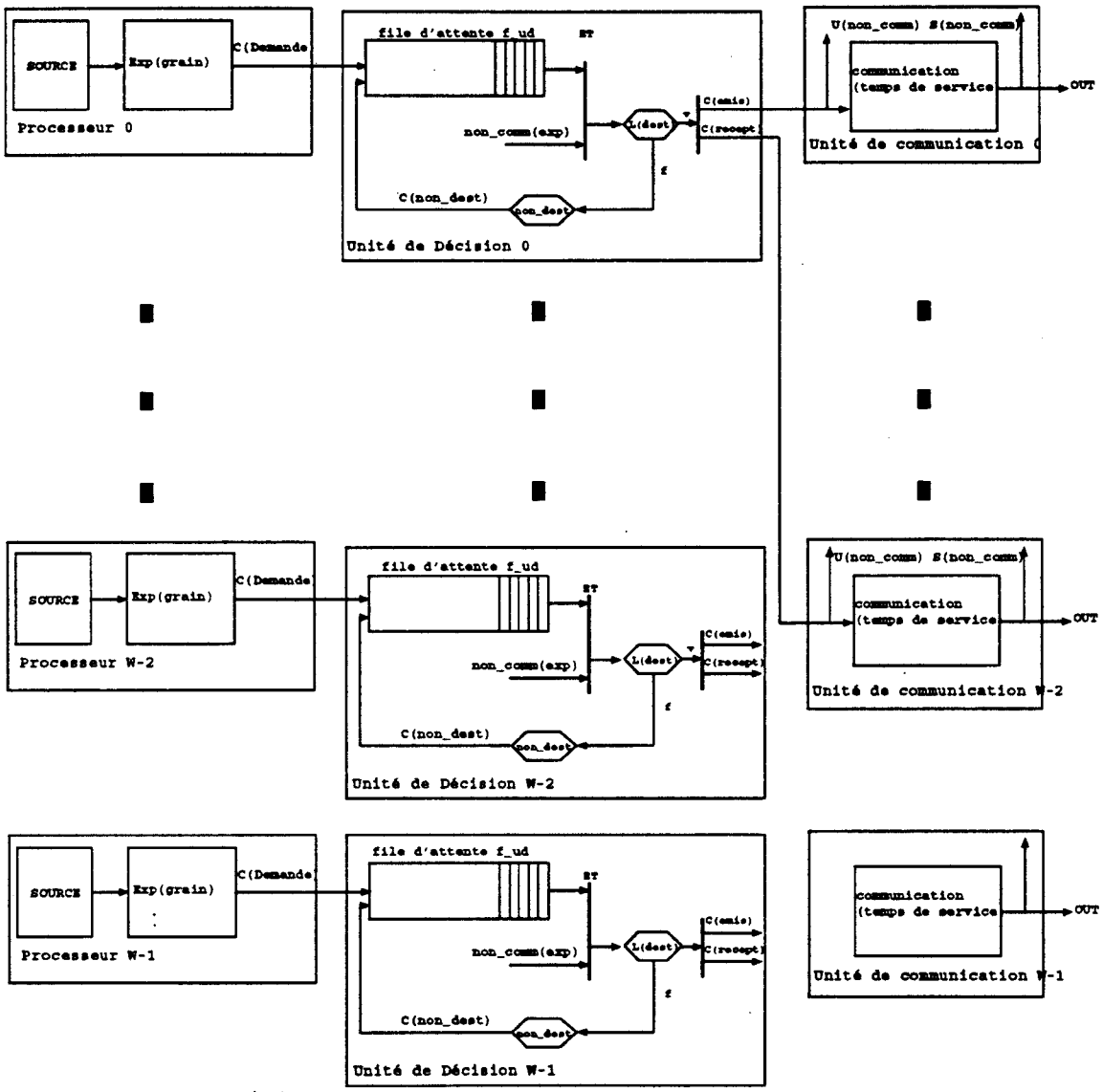


Figure 5.9 : Modélisation des unités fonctionnelles du système à une unité de communication

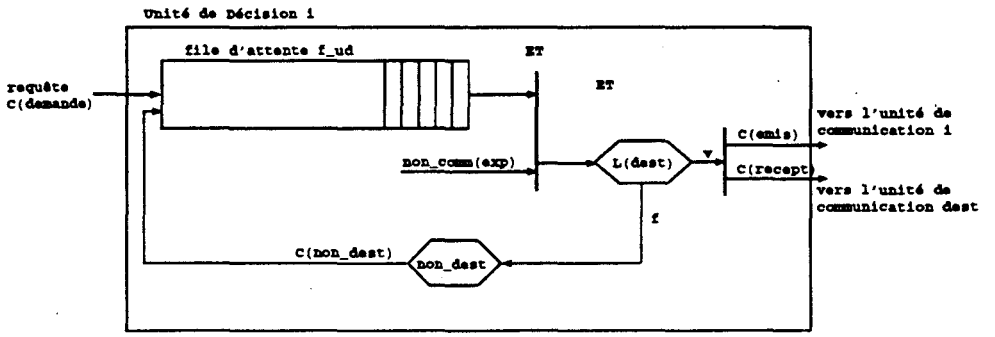


Figure 5.7 : Modélisation d'une unité de décision (UD)

5.3.4 L'unité de communication

L'unité de communication (cf figure 5.8) reçoit les clients de classe *émis* venant de l'unité de décision associé et les clients de classe *recept* venant des autres unités de décisions dans le cas où le noeud associé est en réception.

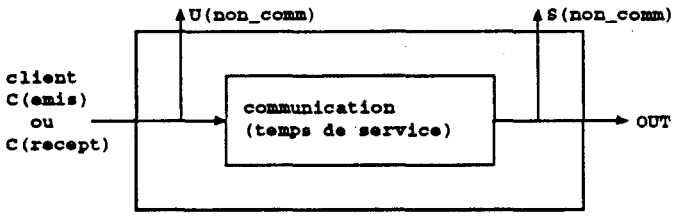


Figure 5.8 : Modélisation d'une unité de communication (UC)

L'unité de communication commence d'abord à signaler qu'elle est occupée en positionnant son drapeau *non-comm* à faux, puis procède à la communication effective pendant un temps constant (le temps de service) dépendant des paramètres logiciels (la longueur du message) et architecturaux (le débit). A la fin de la communication, le drapeau *non-comm* est remis à vrai (à l'initialisation il est naturellement à vrai), puis le client sort du système.

5.3.5 Simulation du guide d'ondes

La figure 5.9 montre un système composé de W noeuds connectés au guide d'ondes. Dans cette description, chaque noeud est constitué d'un processeur, une unité de décision et une unité de communication. Dans cette sous-section, nous allons procéder à l'étude du comportement de tout le système composé de W noeuds en fonction des paramètres physiques comme dans les deux modèles théoriques précédents : P la puissance du processeur associé, D le débit lors des communications, L la longueur des messages échangés et I le nombre d'instructions exécutées localement avant qu'un processeur n'émette une requête de communication. De cette étude nous pourrons :

- déterminer non seulement le temps de latence pour un message d'une longueur donnée, mais aussi les temps de réponse des différentes unités (pour déterminer éventuellement les parties susceptibles d'amélioration),

- déterminer le domaine de validité du réseau ou plus exactement pour des paramètres physiques(P et D) fixes, déterminer le couple des valeurs limites I et L que le système peut supporter.

De l'étude théorique faite dans le cas des deux modèles de simulation précédents, nous avons mis en exergue une condition nécessaire au fonctionnement du réseau : le taux d'occupation des serveurs (unité de communication) ne doit pas être supérieur à 1. Dans le cas où une seule unité de communication est implantée sur les noeuds, cette unité est sollicitée deux fois plus, pour l'émission et pour la réception, donc le taux en réception ρ doit vérifier la condition : $\rho < 1/2$ (dans le système à deux unités de communications, la condition était $\rho < 1$ puisqu'une unité de communication est toujours disponible pour la réception). Et ce taux $\rho = u/c$ est directement lié au paramètres L et I :

$$\rho = \frac{u}{8} = \frac{E[s]}{E[\tau]} = \frac{\lambda}{8 \times \mu} = \frac{8 \times P/I}{8 \times D/L} = \frac{P/I}{D/L} = \frac{P}{D} \times \frac{L}{I}$$

$$\rho < 1/2 \Longleftrightarrow L/I < D/(2 \times P)$$

Ainsi, pour P et D fixés, le rapport $D/(2 \times P)$ nous fournit une première limite pour des choix judicieux de L et I pour la simulation.

Pour comparer ce modèle avec les deux étudiés plus haut (distribution uniforme des requêtes puis distribution optimale des requêtes), nous considèrerons dans un premier temps des messages de L=1 Kbits toujours pour des valeurs de D=320 mégabits par seconde et de P=100mips correspondant à la puissance des processeurs actuels. Les caractéristiques suivantes sont évaluées :

- T_w : le temps d'attente moyen pour l'accès à un canal,
- τ_{UC} : le taux d'occupation de l'unité de communication. Ce taux nous permettra plus tard de déduire le débit moyen du guide d'ondes[5],
- τ_{direct} : le taux moyen de requêtes qui accèdent directement au destinataire sans attente, c'est-à dire sans conflits (occupation de son unité E/R ou de l'unité E/R du destinataire).

La table 5.3 nous donne les valeurs des caractéristiques mentionnées ci-dessus en fonction de I.

Tableau 5.3 : Les valeurs des paramètres T_w (en μs), τ_{UC} et τ_{direct} en fonction de I ; le temps de service=$1/\mu = L/D = 400bits/40.10^7 bits/seconde = 10\mu secondes$

I	700	1000	1500	2000	3000	5000
$1/\lambda$	7,00	10,00	15,00	20,00	30,00	50,00
τ_{UC}	0,90	0,69	0,46	0,34	0,23	0,14
τ_{direct}	0,29	0,21	0,28	0,35	0,47	0,61
T_w	∞	12,42	3,62	2,12	1,14	0,60

La figure 5.10 montre une comparaison de l'évolution des deux taux mentionnés ci-dessus. Pour de grandes valeurs de I, l'unité de communication est plus rarement sollicitée, avec naturellement moins de conflits. Pour de petites valeurs de I, les communications plus fréquentes entraînent évidemment beaucoup de conflits (soit parce que l'unité de communication propre est occupée

[5] $dbit = \tau_{UC} \times D \times 8$, avec D=320 mbps

ou parce que le destinataire est occupé), avec un meilleur taux d'occupation de l'unité de communication.

Nous remarquons de plus que l'unité de communication n'est jamais occupée entièrement : le meilleur taux est de 0,82 ; et cette valeur n'est presque jamais atteinte parce que correspondant à une situation d'attente infinie à l'état stationnaire du réseau (pour une utilisation assez longue). Cette valeur limite de τ_{UC} est dûe au fait que même si l'unité de communication est libre, celle des destinataires peuvent être occupées, et donc une utilisation continue de la ressource *Unité de communication* est impossible.

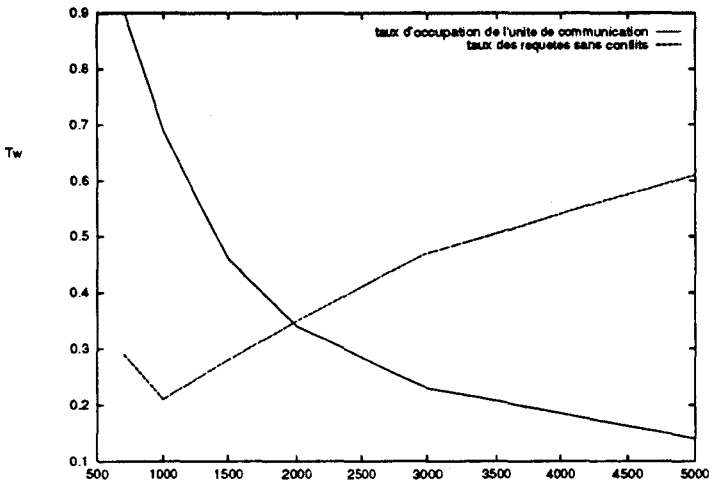


Figure 5.10 : Modèle à une seule unité E/R : évolution des taux d'occupation de l'unité de communication et de requêtes sans conflits en fonction de I

Les temps d'attente pour l'accès au réseau représentés dans le tableau 5.3 confirment un fait assez attendu pour des petites valeurs de I. Pour I assez proche de 700 ($7\mu secondes$ entre deux requêtes de communication), nous obtenons des temps d'attente proches de l'infini, même pour un temps de service de $3,5\mu secondes$. Ceci met en exergue une conséquence du partage de la même ressource (l'unité de communication) à la fois pour l'émission et pour la réception de messages.

Un taux moyen de messages (50% environ) ne subissent pas d'attentes supplémentaires liées aux conflits. Nous pouvons donc déduire compte tenu des temps d'attentes moyens observés que dès qu'un message est en attente pour une transmission (suite à un conflit), celui-ci est de l'ordre de deux fois le temps d'attente moyen : si nous désignons par T_1 le temps d'attente des messages ne subissant pas de conflits ($T_1 = 0$) et T_2 le temps d'attente moyen des messages subissant une attente effective, on a :

$$T_w = \tau_{direct} \times T_1 + (1 - \tau_{direct}) \times T_2$$

pour $T_1 \simeq 0$, on a : $T_2 = \frac{T_w}{1-\tau_{direct}} \simeq 2 \times T_w$ puisque τ_{direct} est proche de 1/2.

5.3.6 Une trace des communications entre noeuds

Nous montrons ci-dessous une trace de la charge du réseau comme dans le système précédent. S'il est plus difficile de déterminer le nombre de noeuds en communication en fonction du nombre de communications dans le système précédent, dans ce cas ci, toute communication

occupe deux noeuds indépendamment des autres communications. Dans le système à deux unités de communications par exemple, deux communications simultanées peuvent occuper uniquement 3 noeuds : le premier émettant vers le second, celui-ci émettant à son tours vers le troisième.

La figure 5.11 montre l'évolution en *µsecondes* de la charge pour des valeurs de I= 300 et 500.

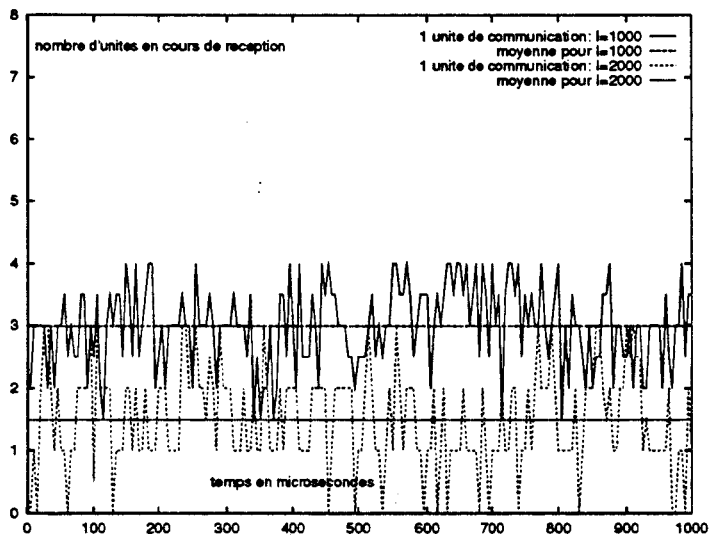


Figure 5.11 : Système à une unité de communication : évolution temporelle de la charge du réseau pour I= 1000 et I=2000

5.4 Comparaison des deux systèmes

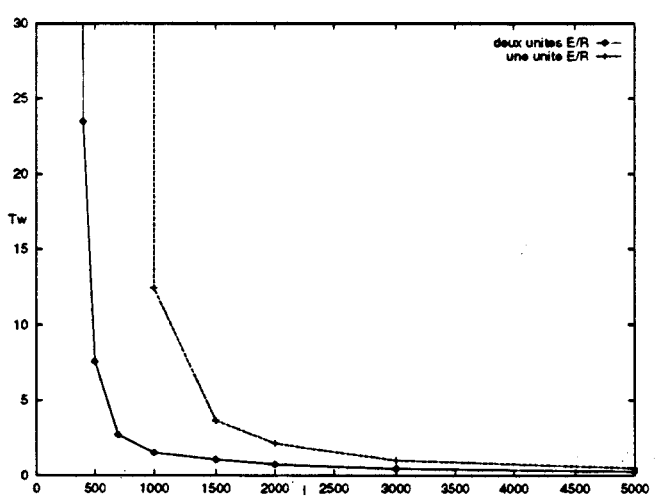


Figure 5.12 : Comparaison des temps d'attente en fonction de I, pour un système à deux unités E/R et un système à une unité E/R

Nous comparons ici le système à deux unités E/R et le système à une seule unité E/R suivant deux critères : les temps d'attentes et le taux d'occupation du canal alloué au processeur (il faut souligner que même avec deux unités E/R, chaque noeud ne possède qu'un seul canal de réception).

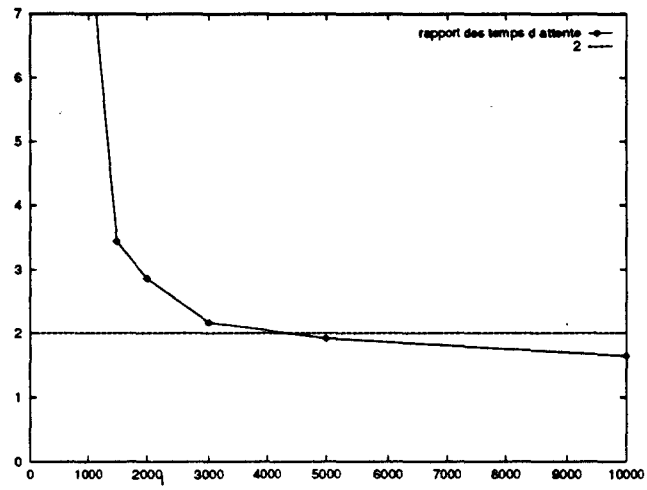


Figure 5.13 : Rapport des temps d'attente en fonction de I, pour les deux systèmes : à une unité de communication et à deux unités de communication

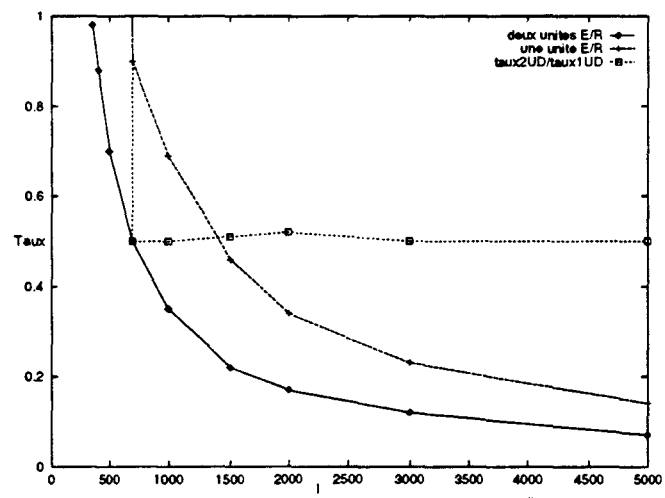


Figure 5.14 : Comparaison des taux d'occupation des unités de communication en fonction de I, pour le modèle à deux unités et pour le modèle à une seule unité

La figure 5.12 montre les écarts entre les deux temps d'attentes suivant le grain de parallélisme I. Pour I de l'ordre de 1000, l'écart est maximal : le réseau à une seule unité E/R est inutilisable dans ce contexte (aucun des deux modèles architecturaux n'est évidemment pas adapté pour des valeurs de I inférieures à 400, en considérant bien-sûr des messages de 1000 bits). Au fur et à mesure que I augmente, l'écart se réduit pour finalement s'annuler, **la probabilité pour que le noeud veuille émettre un message pendant une réception devenant moindre, les temps d'attentes sont dûs uniquement aux conflits pour l'accès à un destinataire quelconque**. Ces taux de conflits restent évidemment identiques quel que soit le modèle (une

ou deux unités de communication).

Il existe alors une valeur de I pour laquelle l'implantation de deux unités d'émission/réception n'est plus nécessaire. Dans le cas où le réseau est destiné à supporter des applications nécessitant des valeurs de I inférieures à 1000 (toujours pour L=1000), l'utilisation de deux unités devient nécessaire. Pour I variant autour de 1500, le choix sera étroitement lié à l'application (on pourrait effectuer une meilleure redistribution des tâches pour réduire les communications comme dans le cas de la distribution *optimale* étudiée dans le modèle précédent). A partir de $I \simeq 5000$, la figure 5.13 montre que le rapport des temps d'attente correspondant au modèle à une seule unité d'émission/réception et le modèle à deux unités prend des valeurs en dessous de 2, alors que ce rapport croît et tend vers l'infini à mesure que le grain de l'application (I) diminue. Nous déduisons de ce fait deux conclusions dans le cas où tous les noeuds du système ne sont connectés qu'à un seul guide d'ondes :

- pour des applications nécessitant une émission de messages environ tous les I=1000 instructions ($\frac{temps de communication}{temps de traitement local} = \frac{1/\mu=3,5\mu secondes}{1/\lambda=10\mu secondes} = \frac{1}{3}$), l'utilisation de deux unités de communication permet de réduire considérablement les temps de latence des communications. Le non respect de cette règle entraînerait par exemple pour I=1000 des temps d'attente 8 fois plus élevés, et pour I=500, l'utilisation de deux unités de réception est obligatoire.

- pour des applications nécessitant une émission de messages environ tous les I=3000 instructions ($\frac{temps de communication}{temps de traitement local} = \frac{1/\mu=3,5\mu secondes}{1/\lambda=300\mu secondes} = \frac{1}{10}$), l'utilisation d'une seule unité d'émission/réception suffit puisque l'utilisation d'une seconde unité ne permet pas d'améliorer les temps d'attente de manière proportionnelle.

La comparaison des taux d'occupation des unités E/R (figure 5.14) montre un bon taux d'utilisation dans le cas d'une seule unité à la fois pour la réception et pour l'émission (évidemment elles supportent les émissions et les réceptions de messages). Dans le cas de deux unités, les unités de communication sont évidemment sous-utilisées, mais elles permettent de supporter un grain de parallélisme plus fin, c'est-à-dire un plus gros débit de communication.

D'autre part, nous constatons presque toujours un rapport de 1/2 entre ces deux taux (cf figure 5.14) (sauf pour des valeurs de I proches de 400 ; valeurs pour lesquelles un état stationnaire du réseau à une seule unité d'émission/réception est difficile à obtenir). Ces résultats confirment un comportement tout à fait attendu : pour une même charge en communication (I constant), l'unité de communication dans le cas du modèle à une seule unité par noeud est sollicitée à la fois pour l'émission et pour la réception qui requièrent les mêmes temps de service. Elle est donc nécessairement deux fois plus chargée que dans le cas du modèle à deux unités de communication.

5.5 Influence du grain de l'application

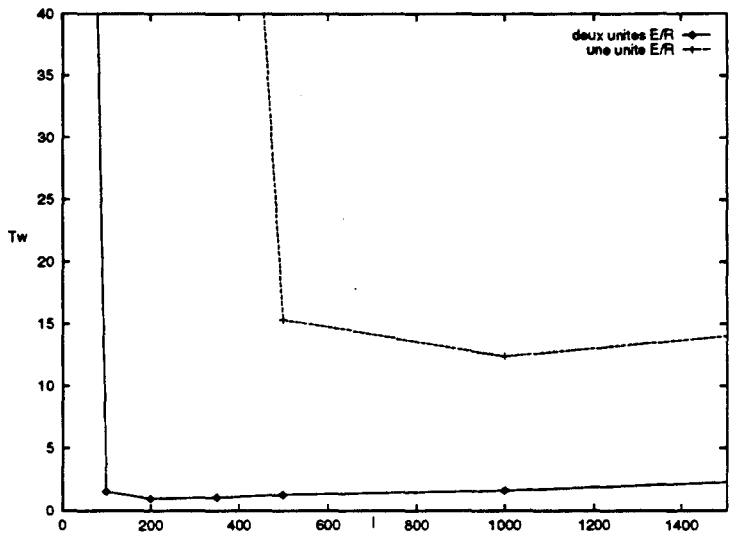


Figure 5.15 : Évolution des temps d'attentes en fonction du grain de l'application (L/I constant=1) : choix entre échanges fréquents de messages courts et échanges moins fréquents de longs messages

Les études ci-dessus ont été faites pour un même type de message : nous avons supposé qu'un message était réduit à un seul paquet de 1Kbits correspondant par exemple à un transfert de 16 nombres flottant double précision.

Il est évident que plus la taille des paquets est grande, plus les canaux de communication sont occupés($1/\mu$ est grand), et pour une même fréquence de communication λ, les temps d'attente issus de la formule de *Little* $T_w = T_{total} - T_{service} = 1/(\mu - \lambda) - 1/\mu$ ne seront que plus élevés. Par contre il semble intéressant d'étudier le comportement du réseau pour un rapport L/I constant. Pour un même besoin en mouvement de données ($\tau_{comm}/\tau_{traitement}$ constant), il s'agit de montrer l'influence entre communications fréquentes de petits messages et communications moins fréquentes de longs messages.

Considérons pour simplifier un rapport $L/I = 1$, ce qui correspond dans nos évaluations précédentes à $I = 1000$, et $\rho = \frac{P}{D} \times \frac{L}{I} = 0,35$.

Tableau 5.4 : Système à deux unités de communication : les valeurs des paramètres T_w (en μs), τ_{UC} et τ_{direct} en fonction de L pour L/I constant=1

L (en bits)	10	50	100	200	350	500	1000	2000	5000
$1/\lambda$ en μs	0,10	0,50	1,00	2,00	3,50	5,00	10,00	20,00	50,00
$1/\mu$ en μs	0,47	0,59	0,74	1,04	1,49	1,94	3,44	6,44	15,44
τ_{UC}	0,97	0,99	0,74	0,52	0,43	0,38	0,34	0,32	0,31
T_w	∞	∞	1,53	0,97	1,06	1,27	1,60	2,92	6,76

Dans le cas de messages très courts de quelques dizaines d'octets, le temps requis pour l'envoi d'un message (temps de service) est proche de l'infini puisque $\rho > 1$ (cf tableau 5.5), conséquence

Tableau 5.5 : Système à une unité de communication : les valeurs des paramètres T_w (en μs), τ_{UC} et τ_{direct} en fonction de L pour L/I constant =1

L (en bits)	10	50	100	200	500	600	750	1000	2000	5000
$1/\lambda$ en μs	0,10	0,50	1,00	2,00	5,00	6,00	7,50	10,00	20,00	50,00
$1/\mu$ en μs	0,47	0,59	0,74	1,04	1,94	2,24	2,69	3,44	6,44	15,44
τ_{UC}	0,97	0,99	0,74	0,52	0,76	0,74	0,71	0,69	0,64	0,62
T_w	∞	∞	∞	∞	15,37	13,27	11,74	12,42	15,63	31,65

de l'overhead requis avant toute communication : Le temps requis pour la trnsmission d'un message est borné à 350 ns. Ensuite, dès que L atteint une valeur correspondant à une charge de travail réaliste ($\rho < 1$), le temps d'attente décroît régulièrement à mesure que L augmente, puis recommence à croître[6]. Les taux d'occupation restent sensiblement conformes à la charge de travail puisque ρ reste inchangé.

Nous avons esquissé dans le chapitre 3 au vu de l'overhead lié au protocole avant toute communication que le réseau est plus dédié aux échanges moins fréquents de longs messages que le contraire. La figure 5.15 confirme que pour définir le grain de l'application, le rapport global entre temps de communication et temps de traitement local pour chaque noeud (L/I) est insuffisant et que le facteur fréquence de communication est aussi à prendre en compte ; aussi, pour un rapport L/I constant = 1 par exemple :

- le réseau ne peut supporter des échanges dont la longueur moyenne est de quelques dizaines d'octets,
- suivant que chaque noeud dispose de deux ou d'une unités de communications, les attentes optimales se situent à $L = 300$ octets dans le premier système et à $L = 700$ dans le second système. A delà de ces valeurs, les attentes ne sont que proportionnelles à la longueur des messages.

5.6 Modélisation du réseau Hypercom

Nous avons effectué ci-dessus une modélisation d'un réseau *multicanaux* réduit à un seul guide d'ondes connectant 8 noeuds. Nous allons présenter dans ce qui suit les valeurs des temps de latence pour le réseau *hypercom* défini au chapitre 4 (maillage $W \times (k \times W) \times (k \times W)$).

Le temps de latence global dans tout le réseau devra tenir compte outre de la longueur L du message, de la distance **d** séparant la source de la destination, c'est-à-dire du nombre de noeuds par lesquels devra transiter le message. Ce temps dépend aussi du mode de routage choisi comme nous l'avons souligné au chapitre 4.

Pour un routage Store and Forward, le temps de latence dans notre réseau pour une distance $\bar{d}$, un temps de cycle $T_c = 3ns$ et un message de L bits (dont un en-tête de A bits) est de : $t = T_c \times \bar{d} \times L$.

[6]La croissance s'explique à partir de la formule de *Litte* : pour I=L,

$$T_w = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = L(\frac{P}{D(D-P)})$$

T_w est une fonction linéaire croissante de L.

Au cours des études précédentes sur la modélisation d'un guide d'ondes, nous avons utilisé comme paramètre I, représentant le nombre d'instructions exécutées localement par un processeur avant une émission de requête de communication. Dans le cas où chaque noeud est connecté à plusieurs guides d'ondes, c'est-à-dire un noeud dont le degré physique Δ est supérieur à 1, pour un grain donné I, chaque guide d'ondes ne recevra en moyenne qu'une requête toutes les $I \times \Delta$ instructions. Les requêtes étant supposées uniformément distribuées sur l'ensemble des guides d'ondes connectés au noeud.

Les temps d'attentes pour l'accès à un canal sur un guide d'ondes dépendant essentiellement du taux d'émission des requêtes (λ ou I) pour une longueur moyenne de messages fixée, il nous faut prendre en compte non seulement les requêtes issues des processeurs associés, mais aussi les messages routés ; le nombre de messages routés dépendant essentiellement de la distance moyenne dans le réseau. Si chaque message doit être routé $\bar{d}$ fois avant d'arriver à destination, ce message correspondra à $\bar{d}$ requêtes vers les différents guides d'ondes qu'il aura emprunté.

Avant de déterminer le temps de latence global dans un réseau quelconque à base de guides d'ondes, il nous faut au préalable déterminer le nombre moyen de requêtes par guides d'ondes qui dépend de :

- $\lambda = P/I$ le grain de parallélisme de l'application (les besoins en communication au niveau de chaque noeud),
- Δ le degré physique de chaque noeud : plus le degré physique est élevé, plus il y a de guides d'ondes dans le système et moins le réseau est chargé (puisque les requêtes sont uniformément distribuées sur les guides d'ondes auxquels le noeud est connecté),
- $\bar{d}$ la distance moyenne dans le réseau : un message émis par un noeud étant routé en moyenne $\bar{d}$ fois, il correspond à $\bar{d}$ requêtes pour le réseau entier.

Considérons un réseau de N noeuds composé de guides d'ondes de $W = 2^3$ noeuds de degré physique Δ, le réseau ayant un diamètre d, la distance moyenne dans le réseau étant notée $\bar{d}$ et les noeuds émettant λ requêtes de communication par seconde.

- le nombre de requêtes par seconde dans le système est de $\lambda \times N$ messages $\times \bar{d}$ requêtes/messages $= \lambda \times N \times \bar{d}$ requêtes de communication.
- le nombre de guides d'ondes dans le système est de : $\frac{Nnoeuds \times \Delta connexions}{Wconnexions/guides} = \frac{N \times \Delta}{W}$.

Le nombre de requêtes par guide d'ondes est donc de : $\frac{\lambda \times N \times \bar{d}}{N \times \Delta / W} = \lambda \times N \times (\bar{d}/\Delta) req/guide$. Pour déterminer le temps de latence dans le réseau global, il nous faut déterminer les temps de service par guides d'ondes pour d'autres valeurs de λ. On pourrait alors poser pour un guide d'ondes, $\lambda' = \lambda \times (\bar{d}/\Delta)$ (ou $I' = I \times (\Delta/\bar{d})$)[7], puis considérer $\bar{d}$ fois le temps de service obtenu pour la transmission sur un guide d'ondes. Nous donnons ci-dessous les trois cas possibles pour les valeurs de $\bar{d}$ et de Δ.

5.6.1 Le cas où $\bar{d} < \Delta$

Si les communications sont très localisées, nous obtenons des temps d'attente très faibles pour l'accès aux canaux de communication puisque $\lambda' = \lambda \times (\bar{d}/\Delta) < \lambda$ et évidement $I' > I$. Les

[7] λ désigne toujours le nombre de requêtes générées par le processeur local par unité de temps et λ' le nombre de requêtes générées par le noeud local par unité de temps en prenant en compte le nombre de guides d'ondes dans le réseau et la distance moyenne dans le réseau

deux tables ci-dessous montrent les temps de latence pour $\bar{d} = 1$ ($I' = I \times \Delta$) ; Δ et d gardant respectivement les valeurs 3 et 4 (la topologie définie au chapitre 4). Nous obtenons des temps de latence inférieurs à ceux obtenus dans le cas où le réseau est réduit à un guide d'ondes ; ceci s'explique par le fait qu'on dispose de beaucoup plus de ressource de communication (les guides d'ondes) pour un faible nombre de requêtes.

Tableau 5.6 : Système à deux unités de communication par noeud : le temps de latence T_N en μs en fonction du taux de communication I dans le réseau Hypercom d'1K noeuds pour un routage *Store and Forward* pour $\bar{d} = 1$ et L 1Kbits

I	150	200	300	500	1000
$I' = I \times \Delta$	450	600	900	1500	3000
$1/\lambda_{guide}$	4,50	6,00	9,00	15,00	30,00
τ_{UC}	0,75	0,57	0,38	0,22	0,12
T_{guide}	11,10	7,45	5,40	4,50	3,97
$T_N = T_{guide} \times d$	11,10	7,45	5,40	4,50	3,97

Tableau 5.7 : Système à une unité de communication par noeud : le temps de latence T_N en μs en fonction du taux de communication I dans le réseau Hypercom d'1K noeuds pour un routage *Store and Forward* pour $\bar{d} = 1$ et L 1Kbits

I	250	300	500	1000	2000
$I' = I \times \Delta$	750	900	1500	3000	6000
$1/\lambda_{guide}$	7,50	9,00	15,00	30,00	60,00
τ_{UC}	0,88	0,75	0,45	0,23	
T_{guide}	340,00	22,74	6,93	4,64	
$T_N = T_{guide} \times d$	340,00	22,74	6,93	4,64	

5.6.2 Le cas où $\bar{d} = \Delta$

Tableau 5.8 : Système à deux unités de communication par noeud : le temps de latence T_N en μs en fonction du taux de communication I dans le réseau Hypercom d'1K noeuds pour un routage *Store and Forward* pour $\bar{d} = \Delta$

I	350	400	500	700	1000	1500	2000	3000	5000
$1/\lambda$ en μs	3,50	4,00	5,00	7,00	10,00	15,00	20,00	30,00	50,00
$\tau_{UC} \simeq \rho = \frac{\lambda}{\mu}$	0,98	0,88	0,70	0,50	0,35	0,22	0,17	0,12	0,07
T_{guide}	∞	27,00	11,07	6,20	5,00	4,55	4,04	3,66	3,60
$T_N = T_{guide} \times d$	∞	81	33,21	18,60	15,00	13,65	12,12	10,98	10,80

Il nous suffit de considérer $\bar{d}$ fois les temps de service pour l'accès au réseau pour obtenir les temps de latence dans le réseau global, puisque $\lambda' = \lambda$. Le diamètre du *spanning bus hypercube* étant égale au degré physique des noeuds, ce cas correspond par exemple aux cas les plus défavorables dans le cas où le réseau sous-jacent est un spanning bus hypercube, mais aussi à une distribution moyenne dans le réseau *hypercom* défini.

Tableau 5.9 : Système à une unité de communication par noeud : le temps de latence T_N en μs en fonction du taux de communication I dans le réseau Hypercom d'1K noeuds pour un routage *Store and Forward* pour $\bar{d} = \Delta$

I	700	1000	1500	2000	3000	5000
$1/\lambda$	7,00	10,00	15,00	20,00	30,00	50,00
τ_{UC}	0,90	0,69	0,46	0,34	0,23	0,14
T_{guide}	∞	15,92	7,12	5,62	4,64	4,10
$T_N = T_{guide} \times d$	∞	47,76	21,36	18,83	13,92	12,30

5.6.3 Le cas où $\bar{d} > \Delta$

Tableau 5.10 : Système à deux unités de communication par noeud : le temps de latence T_N en μs en fonction du taux de communication I dans le réseau Hypercom d'1K noeuds pour un routage *Store and Forward* pour $\bar{d} > \Delta$, $(\bar{d} = d = 4)$

I	500	1000	2000	3000	5000
$I' = 3/4I$	375	750	1500	2250	3750
$1/\lambda_{guide}$	3,75	7,50	15,00	22,50	37,50
τ_{UC}	0,92	0,46	0,46	0,26	0,15
T_{guide}	22,92	6,06	4,55	4,11	3,83
$T_N = T_{guide} \times d$	91,68	24,24	20.70	16,44	15,32

Tableau 5.11 : Système à une unité de communication par noeud : le temps de latence T_N en μs en fonction du taux de communication I dans le réseau Hypercom d'1K noeuds pour un routage *Store and Forward* pour $\bar{d} > \Delta$ $(\bar{d} = d = 4)$

I	1000	1500	2000	3000	5000
$I' = 3/4I$	750	1125	1500	2250	3750
$1/\lambda_{guide}$	7,50	11,25	15,00	22,50	37,50
τ_{UC}	0,92	0,60	0,45	0,30	0,18
T_{guide}	∞	10,90	6,93	5,19	4,32
$T_N = T_{guide} \times d$	∞	43,60	27,72	20,76	17,28

Considérons le cas extrême où tous les messages passent par d guides d'ondes avant d'arriver à destination : ce cas peut se produire si la répartition de tâches sur les noeuds de la machine n'est pas optimisée $(\bar{d} = d)$. Nous obtenons évidemment $\lambda' = \lambda \times (d/\Delta) > \lambda$ et aussi $I' < I$. Il nous suffit alors de considérer d fois les temps de service sur un guide d'ondes pour obtenir le temps de latence global dans le réseau. Dans le cas du réseau *Hypercom* où Δ=3 et d=4, les deux tableaux 5.10 et 5.11 montrent les temps de latence maximaux dans un réseau $W \times W \times (2 \times W)$.

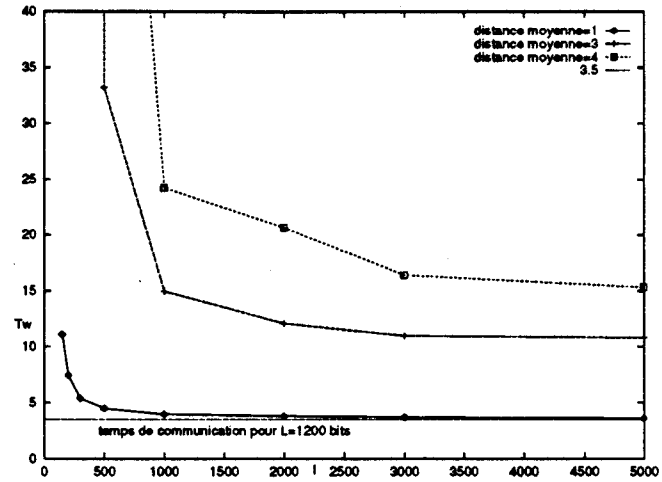


Figure 5.16 : Modèle à deux unités de communications : temps de latence dans le réseau $8 \times 8 \times (2 \times 8)$=1K noeuds pour des communications très localisées ($\bar{d} = 1$), des communications avec $\bar{d} = \Delta$ et $\bar{d} = D$

5.7 Exemple d'application : le produit matriciel par blocs

Le produit matriciel est l'une des opérations les plus courantes dans le domaine du calcul numérique intensif, et une bonne implémentation de ce type d'opération est très déterminant pour les performances globales de toute machine. De nombreux algorithmes ont donc été étudiés tant pour les machines séquentielles que pour différents types de machines parallèles avec comme paramètre important la topologie du réseau sous-jacent. D'autre part, suivant le type de matrice traitée, creuse ou pleine, les algorithmes varient considérablement. Notre but ici n'est pas d'étudier la meilleure implémentation du produit matriciel sur le réseau *Hypercom* mais d'essayer de quantifier les performances sur un telle application puis de montrer la conformité avec les études précédentes.

Nous considérons donc trois matrices carrées A, B et C de dimension n pour lesquelles on veut calculer :

$$C_{i,j} = \sum_{k=1}^{n} A_{i,k} \times B_{k,j}$$

Pour simplifier, on suppose que le réseau de p noeuds est réduit à un maillage 2D $\sqrt{(p)} \times \sqrt{(p)}$ (ou un maillage 3D déplié [8]). Si la dimension n des matrices est supérieure au nombre de processeurs du réseau, on associe plusieurs éléments à un même processeur (par exemple un bloc carré de $n/\sqrt{p}$ par $n/\sqrt{p}$). Nous associerons naturellement le petit bloc C_{ij} de la matrice à calculer au processeur correspondant aux coordonnées cartésiennes i et j dans le maillage $\sqrt{p} \times \sqrt{p}$ (cet algorithme est issu de [15]).

A un instant donné, si les éléments A_{ik} et B_{kj} pour un indice quelconque sont présents dans le processeur [i,j], celui-ci n'a qu'à additionner leur produit à son résultat partiel C_{ij}. L'idée de base est donc de faire circuler les éléments des deux matrices A et B de manière que les blocs A_{ik} et B_{kj}, pour k variant de 1 à $\sqrt{p}$, se rencontrent dans le processeur [i,j] (cet algorithme est évidemment inspiré du systolique).

[8] Réseau des machines Wavetracer par exemple

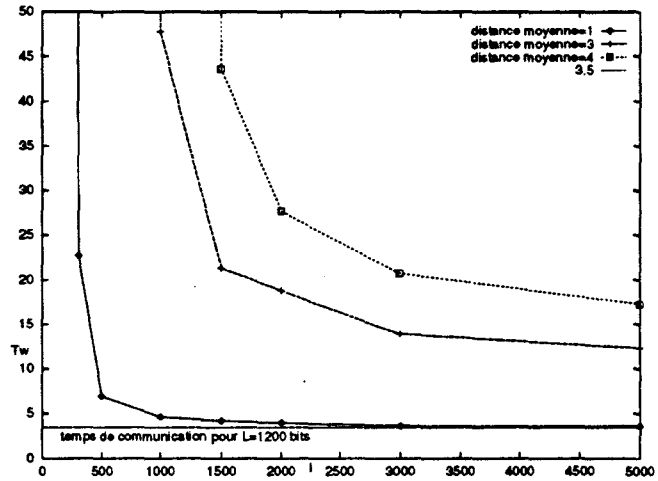


Figure 5.17 : Modèle à une unité de communication : temps de latence dans le réseau $8 \times 8 \times (2 \times 8)$=1K noeuds pour des communications très localisées ($\bar{d} = 1$), des communications avec $\bar{d} = \Delta$ et $\bar{d} = D$

- considérons un exemple de matrice à neuf blocs, les C_{ij} que l'on doit calculer sont :

$$\begin{array}{lcccccc}
C_{11} & = & (A_{11} \times B_{11}) & + & A_{12} \times B_{21} & + & A_{13} \times B_{31} \\
C_{12} & = & A_{11} \times B_{12} & + & (A_{12} \times B_{22}) & + & A_{13} \times B_{32} \\
C_{13} & = & A_{11} \times B_{13} & + & A_{12} \times B_{23} & + & (A_{13} \times B_{33}) \\
C_{21} & = & A_{21} \times B_{11} & + & (A_{22} \times B_{21}) & + & A_{23} \times B_{31} \\
C_{22} & = & A_{21} \times B_{12} & + & A_{22} \times B_{22} & + & (A_{23} \times B_{32}) \\
C_{23} & = & (A_{21} \times B_{13}) & + & A_{22} \times B_{23} & + & A_{23} \times B_{33}
\end{array}$$

En supposant qu'initialement les éléments A_{11} et B_{11} soient en place sur le processeur [1,1], le calcul du premier produit partiel C_{11} peut s'effectuer. Et comme A_{11} sert au calcul dans le processeur [1,1], on ne peut l'utiliser au même instant pour calculer C_{12}. Par contre A_{12} est disponible et doit rencontrer B_{22} **qu'il faudra avoir transféré au préalable**. Alors que A_{11} est libre pour la seconde étape, il peut être envoyé depuis [1,1] dans [1,3] où il devra rencontrer B_{13}. Un positionnement initial des blocs A_{ij} et B_{ij} est donc nécessaire, les C_{ij} ne changeant pas de place (cf figure 5.18). De manière générale, puisqu'on s'interdit *a priori* la duplication des éléments des matrices, le bloc A_{ik} (resp. B_{kj}) doit passer par tous les processeurs de la ligne i (resp. colone j) et seulement par ceux-ci. Dans l'expression précédente des C_{ij}, les calculs effectués à la première étape sont entre parenthèses. Et puisque nous avons $\sqrt{p}$ lignes ou colonnes, l'algorithme comporte nécessairement une boucle sur k et a la structure suivante :

{positionnement initial}
faire en parallèle
    envoi de A du processeur (i,j) vers le processeur (i, (j-i)modulo $\sqrt{p}$)
    envoi de B du processeur (i,j) vers le processeur ((i-j)modulo $\sqrt{p}$,j)
pour k ← 1 **jusqu'à** $\sqrt{p}$
    pour tous les processeurs (i,j) **faire en parallèle**
        C(i,j) ← C(i,j) + A(i,k)*B(k,j)
        décaler A vers son voisin de gauche
        décaler B vers son voisin de haut

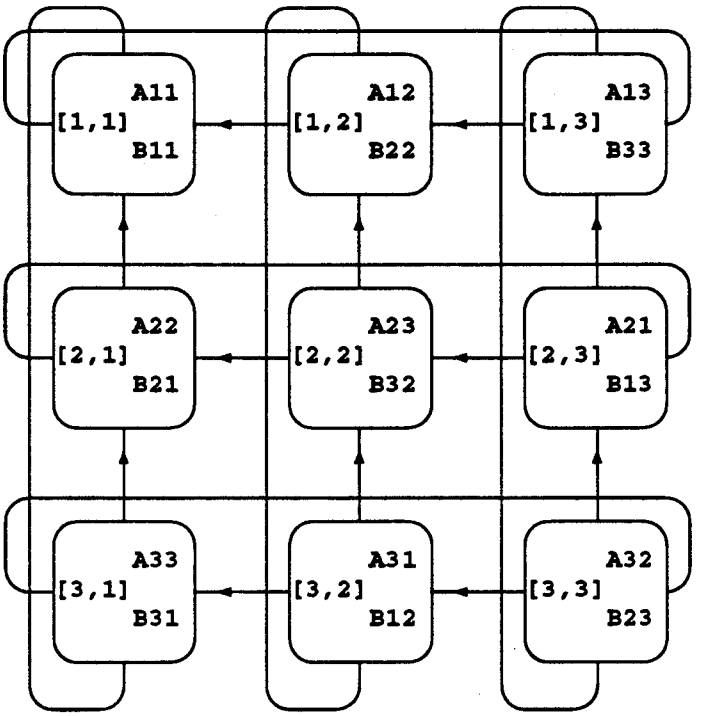


Figure 5.18 : Multiplication de matrices par blocs : positionnement initial et premier calcul partiel. Les blocs A_{ij} se déplacent ensuite sur les lignes et les blocs B_{ij} sur les colonnes.

Le coût de cet algorithme en termes de calculs locaux et de communication dans le réseau Hypercom est le suivant :

- une phase préliminaire de communication de blocs composés de n^2/p données simultanément sur chacune des deux dimensions. Dans le cas du réseau Hypercom, la position sur le même axe n'a pas d'influence.
- puis $\sqrt{p}$ étapes de calculs élémentaires de multiplication sur les blocs, soit $(n/\sqrt{p})^3$ multiplications et additions, avec à chaque étape, un transfert d'un bloc de taille n^2/p.

Le rapport communication/temps de calcul est de

$$\frac{(\sqrt{p}+1) \times \tau_{bloc}}{\sqrt{p}((n/\sqrt{p})^3 \times \tau_a)}$$

τ_{bloc} caractérisant le temps de transfert d'un bloc.

L'algorithme présenté ci-dessus est écrit de telle manière à éviter les conflits de destinataires, c'est à dire qu'à un moment donné, tous les destinataires de tous les processeurs sont différents. Ainsi, suivant le système utilisé (à deux ou une unité de communication par noeud), les temps de communications seront très différents :

- dans le premier système à 2 unités de communication, il n'y a aucun conflit de destinataire et l'algorithme s'exécute de façon déterministe ; aucun temps d'attente n'est observé, **le modèle étant alors proche d'une distribution optimale des requêtes de communications**.

- dans le second système, les noeuds ne possédant qu'une unité à la fois pour l'émission et pour la réception, les temps de communications τ_{bloc} sont totalement aléatoires puisque les conflits sont inévitables. Il ne s'agit pas ici de conflits pour l'accès à un destinataire donné mais d'un conflit sur l'utilisation de l'unité de communication pour l'émission ou (exclusivement) pour la réception. Nous sommes dans un cas où l'utilisation d'une unité de communication est très défavorable, puisque cette contention est répétée à chaque étape (on peut évidemment modifier l'algorithme pour effectuer l'émission et la réception en deux étapes dans le meilleur des cas).

L'algorithme décrit ci-dessus a été écrit pour une exécution sur un réseau mesh, en privilégiant les communications entre voisins directs. Mais de manière générale, tout algorithme adapté aux réseaux mesh doit pouvoir bien s'exécuter sur le réseau Hypercom par l'utilisation des liaisons *multicanaux* (si on considère l'algorithme ci-dessus sur un réseau spanning bus hypercube classique, chaque étape de communication serait exécutée en $\sqrt{p}$ étapes au lieu d'une seule dans le réseau hypercom).

5.8 Quelques réseaux à bus existants

En considérant des distances inférieures au mètre, les liaisons intermodules sont actuellement réalisées par :

1. des liaisons "cuivre sur époxy", avec des fréquences pouvant atteindre 60 Mhz. Les liaisons multipoints sont limitées par les caractéristiques d'adaptation électriques, notamment celles des interfaces : des drivers pour alimenter les bus et les receivers adaptés.
2. des liaisons par câble avec des supports à paire torsadée, différentielle. Ces liaisons permettent des fréquences de transmission pouvant atteindre 500 Mhz.

Les débits d'informations par canal sur un guide d'ondes sont de 320Mbits par seconde (pour 8 noeuds connectés). Ces valeurs sont nettement inférieures à celles obtenues à partir des technologies comme l'AsGa. Mais en considérant le débit global D_{guide} moyen sur un guide d'ondes qui atteint le gigabits par seconde ($D_{guide(1UC)} = \frac{\tau_{UC} \times D \times N}{2} = \frac{0.80 \times 320 \times 8}{2} = 1024 mbps$ avec τ_{UC} : taux d'occupation des unités de communication)[9], nous obtenons un premier critère de comparaison pour un réseau réduit à un guide d'ondes. Dans le cas où un noeud possède deux unités de communication, nous obtenons une bande passante de deux fois celle du modèle à une unité de communication, c'est-à dire 2 gigabits par seconde.

Il est évident que si nous comparons les communications en hyperfréquences dans un guide d'ondes avec les communications basse fréquence utilisant des composants CMOS, dont les fréquences de fonctionnement sont nettement inférieures au Giga hertz, les communications hyperfréquences sont meilleures. Mais en considérant les bus parallèles actuels (bus VME, MULTIBUS, NUBUS etc.) sur 32 ou 64 bits dont les performances dépassent le gigabits par seconde, nous obtenons des performances assez semblables mais avec un encombrement spacial nettement moindre dans le cas de l'utilisation de guides d'ondes : nous utilisons une seule ligne de communication au lieu de 64 dans le cas des bus parallèles.

La spécification du futurebus [79] prévoit des temps de transferts à 250 Mhz sur 256 bits [53], offrant ainsi un débit pouvant atteindre 8 Gigaoctets par secondes (64 Gigabits par secondes).

[9] le rapport par 2 s'explique par le fait qu'en un instant donné, tous les 8 noeuds ne sont pas en émission. La moitié est en émission et la moitié en réception.

Mais dans ce cas aussi, il s'agit de la mise en parallèle de 256 liaisons séries à 250 Mégabits par seconde comportant entre autre, des contraintes technologiques notamment la puissance des drivers et des receivers pour une telle largeur de bus.

Les technologies comme l'AsGa, l'ECL, l'HBT etc. permettent la réalisation de composants de communication dont les performances sont au moins d'un ordre de grandeur supérieur à celles des composants CMOS et BiCmos. Expériemtalement, ces circuits permettent d'atteindre des performances de l'ordre de 10 Gbps. Pour atteindre ces performances avec les hyperfréquences, il faut des VCO possédant une bande d'accord de plus de 20 Ghz, ce qui en notre connaissance n'existe pas, compte tenu du domaine d'utilisation de ce type de composant actuellement.

5.9 Conclusion

Nous avons effectué dans ce chapitre une étude des performances de réseaux pouvant être construits à base de communications en hyperfréquences avec l'utilisation des unités de communications hyperfréquences dont la bande passante est actuellement de 6Ghz si l'on se réfère aux VCO disponibles actuellement (cf chapitre3). Nous avons simulé le fonctionnement de noeuds connectés à un guide d'ondes, noeuds pouvant comporter soit une unité de communication (qui est alors sollicitée à la fois pour l'émission et pour la réception) ou deux unités de communications permettant de doubler la bande passante, ces choix dépendant essentiellement des exigences en communication des applications jous-jacentes. Nous avons en outre conclu qu'il existe un seuil des exigences en communication des noeuds (grain de parallélisme), pour lequel l'utilisation de deux unités n'offre plus un gain significatif en terme de temps d'attente pour l'accès au réseau (le gain n'étant pas linéaire) ; le seuil dépendant de manière générale au grain des applications. De plus, il est clair que compte tenu de la bande passante (par opposition au débit réel) assez élevée et du protocole de communication moins simple que celui des réseaux directs, le réseau est dédié aux applications gros grain.

Nous avons de plus étendu les performances d'un réseau réduit à un guide d'ondes aux grands réseaux comme le réseau maillé $8 \times 8 \times (2 \times 8)$=1K noeuds défini au chapitre 4. En considérant la technique de commutation (le *Store and Forward*) dans ce type de réseau où la distance moyenne (la localité des communications) a une très forte influence sur le temps de latence, nous avons donné des estimations des temps de latence, suivant que la distance moyenne dans le réseau est inférieure, égale ou supérieure au nombre de guides d'ondes auxquels sont connectés les noeuds, en l'occurence $\bar{d} = D$.

En ne considérant que les débits, une brève comparaison des guides d'ondes avec des réseaux à structure de bus a montré que si les communications utilisant les ondes hyperfréquences offrent des performances meilleures à celles des transmissions classiques à base de composants CMOS actuels, elles restent en deça de celles obtenues à partir des composants rapides de transmission comme l'AsGa ou l'ECL. Cependant, pour caractériser les performances globales d'un réseau, il faudra bien entendu considérer en plus du débit, la latence qui est dans le cas du réseau hypercom assez élevée, notamment pour les applications à grain fin.

Conclusion

L'utilisation de plusieurs technologies dans la conception de systèmes informatiques est actuellement étudiée par de nombreuses équipes de recherche, et pour une grande part, tendent vers la conception de systèmes intégrant les éléments de calcul classiques (les processeurs) avec un outil de communication réalisé à base d'une autre technologie, essentiellement optique [13].

Le projet *Hypercom* dont le but est l'étude de la contribution possible des hyperfréquences dans la conception des réseaux d'interconnexion des machines parallèles à mémoire distribuée est menée conjointement par deux laboratoires :

- le LIFL[10] dont la tâche est la spécification et l'étude des performances d'un réseau construit à partir des hyperfréquences. Cette thèse s'inscrit dans ce cadre et présente donc les premiers travaux dans ce domaine.
- le DHS[11] dont la tâche est essentiellement de montrer la faisabilité des réseaux spécifiés par le LIFL. Tous les tests de validation effectués sont donc reportés dans la thèse de P. VANGELUWE[80].

Après une étude des contraintes liées à l'utilisation des hyperfréquences, nous sommes arrivés à la conclusion que l'utilisation de guides d'ondes était possible pour un réseau destiné à connecter un grand nombre de processeurs. Quelques protocoles de communication ont donc été étudiés pour la communication entre les noeuds connectés à un guide d'ondes. L'atout principal des communications en hyperfréquences étant de permettre plusieurs communications simultanées et asynchrones dans un même support, un mode de fonctionnement MIMD est donc plus adapté à ce type de réseau.

Pour une meilleure extensibilité des réseaux à base de guides d'ondes, nous avons défini un réseau issu du spanning-bus hypercube avec de bonnes propriété d'extensibilité, tout en conservant les avantages du spanning-bus hypercube : un faible degré physiques des noeuds, un faible diamètre et un degré logique de connectivité des noeuds très élevé (un noeud connecté à Δ guides d'ondes peut adresser directement jusquà $\Delta \times W$ noeuds).

L'étude des performances pour un mode de fonctionnement MIMD a été effectuée pour les systèmes dans lesquels chaque noeud possède soit une ou deux unités de communication pour chaque connexion à un guide d'ondes. Il résulte de ces études que l'on peut atteindre les 80% de la bande passante totale dans le guide d'ondes. En considérant des VCO existants possédant une bande d'accord de 6Ghz, des débits de plus d'un gigabits par seconde peuvent donc être obtenus dans les systèmes à une unité de communication par noeud, et 2 gigabits par seconde dans les systèmes à deux unités de communication par noeud.

Notre étude a montré aussi, compte tenu des temps de latence élevés, qu'un réseau à base de guides d'ondes est plus dédié à un mode de programmation à gros grain pour réduire l'overhead lié à toute communication. Le réseau Hypercom semble donc plus proche du traitement distribué (à la manière de réseaux locaux rapides) que des réseaux d'interconnexions pour machines fortement couplés dont les réseaux tendent actuellement tout naturellement à permettre des latences très faibles pour de petits messages et des débits élevés pour de très longs messages.

[10] Laboratoire d'Informatique Fondamentale de Lille
[11] Département Hyperfréquences et Semiconducteurs de Lille

Les travaux futurs concerneront essentiellement une étude plus poussée des protocoles de communications pour permettre l'exploitation optimale de la bande passante des VCO, et surtout de réduire la latence pour les petits messages.

Annexe A

Les Hypergraphes et les Hypernets

A.1 Les Hypergraphes

La théorie des graphes a fourni d'excellents résultats dans la construction de topologies de réseaux d'interconnexion. Les hypergraphes encore peu exploités peuvent être un support intéressant pour l'analyse et la construction de topologies non plus de noeuds simples, mais de groupes de noeuds[6].

A.1.1 Quelques définitions

Un hypergraphe est un couple $H(X, E)$ où $X = \{x_1, x_2, \ldots, x_n\}$ est un ensemble fini et où $E = \{E_1, E_2, \ldots, E_m\}$ est une famille finie de sous ensembles non vides de X dont l'union est X c'est-à-dire :

- $E_i \neq \emptyset$, $i \in \{1, 2, \ldots, m\}$
- $\bigcup_{i=1}^{m} E_i = X$

Les éléments de E s'appellent les **arêtes** et les éléments de X les **sommets** de l'hypergraphe. L'**ordre** de l'hypergraphe étant le nombre de ses sommets.

H est un **hypergraphe simple** sur X si $E_i \subset E_j \Rightarrow i = j$. De ce fait, un graphe simple n'est qu'un hypergraphe simple dont toutes les arêtes sont de cardinalité 2.

A.1.2 Représentation des hypergraphes

Considérons un exemple simple :

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_9\}$

- $H = (E_1, E_2, E_3, E_4, E_5, E_6)$ avec
  - $E_1 = \{x_3, x_4, x_5\}$
  - $E_2 = \{x_5, x_8\}$
  - $E_3 = \{x_6, x_7, x_8\}$
  - $E_4 = \{x_2, x_3, x_7\}$
  - $E_5 = \{x_1, x_2\}$
  - $E_6 = \{x_7\}$

Pour chaque arête E_i, suivant son cardinal[1], elle est représentée par :

- une boucle si $|E_i| = 1$
- un trait continu joignant les deux éléments si $|E_i| = 2$
- un trait plein entourant ses éléments si $|E_i| > 2$

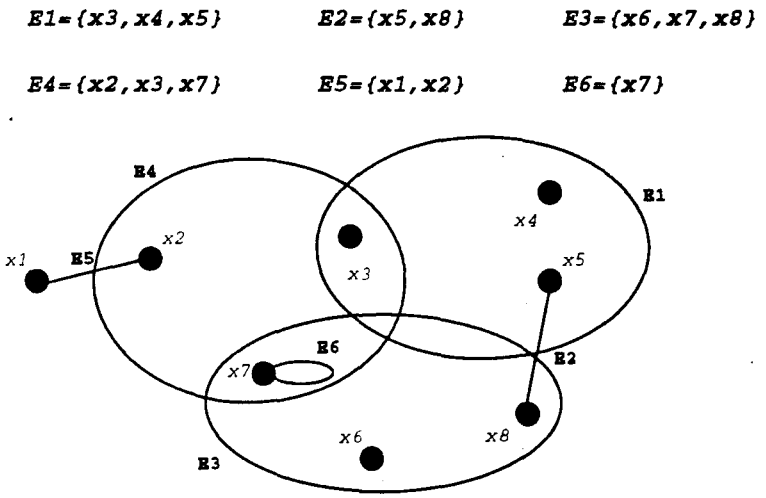


Figure A.1 : Représentation des hypergraphes

La figure A.1 montre une représentation graphique de l'exemple ci-dessus. Il est possible de représenter un hypergraphe par une matrice, appelée *matrice d'incidence* indicée en lignes par les sommets et en colonnes par les arêtes. La case (sommet, arête) contient un 1 si le sommet est dans l'arête, un 0 sinon. Le tableaau A.1 montre la représentation de l'exemple précédent sous forme matricielle.

A.1.2.1 Nombres associés à un hypergraphe

Nous donnons ci-dessous quelques définitions relatives aux hypergraphes.

- le rang d'un hypergraphe H (noté $r(h)$) est le cardinal de la plus grande arête de H, et l'antirang $s(H)$ est le cardinal de la plus petite arête de H. H est dit **uniforme** si $r(H) = s(h)$.
  Si H est simple et uniforme, alors, H est dit H **r-uniforme**.

[1] le nombre d'éléments $x_j \in E_i$

Tableau A.1 : Représentation matricielle des hypergraphes

	E_1	E_2	E_3	E_4	E_5	E_6
x_1	0	0	0	0	1	0
x_2	0	0	0	1	1	0
x_3	1	0	0	1	0	0
x_4	1	0	0	0	0	0
x_5	1	1	0	0	0	0
x_6	0	0	1	0	0	0
x_7	0	0	1	1	0	1
x_1	0	1	1	0	0	0

- pour x élément de X, l'**étoile centrée en** x noté $H(x)$ est l'hypergraphe partiel de H formé d'arêtes contenant x.
- le degré de x est le nombre d'arêtes de $H(x)$, et le degré d'un hypergraphe est le plus grand des degrés de ses sommets :
  - $d_H(x) = m(H(x))$
  - $\Delta(H) = max_{x \in X} d_H(x)$
- H est dit **régulier** si $d_H(x_i) = d_H(x_j), \forall (i \neq j)$
- $H = (E_1, E_2, \ldots, E_m)$ est **linéaire** si $|E_i \cap E_j| \leq 1, \forall (i \neq j)$
- H est dit **intersectant** si toutes les arêtes ont 2 à 2 une intersection non vide (exemple de l'étoile centrée en x).

A.1.3 Contribution des hypergraphes sur les topologies de groupes

A.1.4 Introduction

On veut construire un réseau d'interconnexion de noeuds par assemblage de plusieurs guides d'ondes ; ce réseau ayant :

- un faible diamètre,
- un faible degré pour le graphe **régulier**[2] créé, et
- un nombre maximal de noeuds.

En ne considérant uniquement que les connexions (donc négligeant les contraintes physiques : guides rigides, croisements de guides sur une surface plane etc.), il nous faut établire une projection de l'ensemble {noeuds, guides, connexion} vers l'ensemble {sommets, arêtes, appartenance}.

Pour cela, il nous suffit de montrer qu'il existe un isomorphisme entre les deux ensembles pour faire correspondre tout réseau à base de guides d'ondes à un hypergraphe.

Or Nous remarquons que :

[2] en ce sens où tous les noeuds ont le même nombre de ports physiques

- un noeud est identifié par :
  - une adresse unique dans le réseau R
- un sommet est identifié par :
  - un identifiant unique appartenant à X

De plus :

- un guide d'ondes est un *ensemble de noeuds*
- une arête est un *ensemble de sommets*

Enfin, les relations de *connexion* et d'*appartenance* sont les deux uniques relations respectivement dans l'ensemble *guide d'ondes* et l'ensemble *arête*.

De ces observations nous pouvons faire la correspondance de l'équation A.1 puis déterminer la topologie optimale à partir de contraintes sur Δ, d et N donnés.

$$(noeuds, guides, connexion) \Longleftrightarrow (sommets, aretes, appartenance) \tag{A.1}$$

La figure A.2 donne un exemple de correspondance entre un réseau à base de guides d'ondes et les hypergraphes.

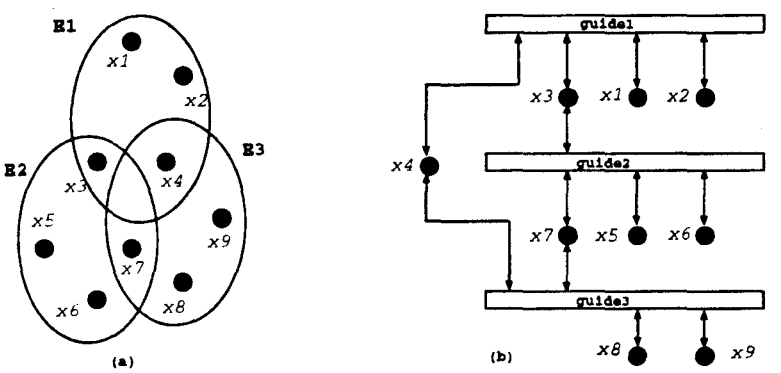


Figure A.2 : Représentation d'un réseau à base de guides d'ondes par des hypergraphes

A.1.4.1 Propriétés minimales

Pour qu'une topologie construite à l'aide des hypergraphes soit susceptible d'être matériellement réalisable, il faut que l'hypergraphe construit soit au moins :

- **r-uniforme**[3] : puisque les guides d'ondes sont de taille fixes.
- **régulier**[4] : les noeuds possèdent le même degré physique.

[3] toutes les arêtes possèdent exactement r sommets
[4] tout sommet appartient au même nombre d'arêtes

A.1.5 Conclusion

Une étude a été faite [55] pour la construction de grands réseaux de noeuds, communiquant au travers de guides d'ondes, en s'appuyant sur la théorie des hypergraphes définie ci-dessus. Les topologies obtenues se rapprochent beaucoup des hypernets étudiés au chapitre 4, Mais là encore, il ne s'agissait que d'une exploration d'un domaine qui à notre avis devrait fournir d'excellents résultats surtout dans des applications comme la communication par ondes radio-électriques.

A.2 Les hypernets

Les hypernets de Hwang [45] peuvent trouver une application dans la construction de la topologie d'une machine à base de guides d'ondes. C'est une topologie hiérarchique dont chaque niveau supérieur est construit par assemblage d'hypernets de niveau inférieur. Un noeud est un ensemble constitué d'un processeur, d'une mémoire locale et de quelques liens d'entrée/sortie (cf figure A.3 (a)) bidirectionnels reliés directement aux autres noeuds ou par l'intermédiaire d'un bus partagé. L'hypernet de premier niveau $h = 1$ appelé module de base est le réseau de base à partir duquel l'hypernet final (d'un niveau supérieur est construit). Un module de base appelé *s-cube* peut être construit en augmentant un hypercube d'un lien supplémentaire par noeud. La figure A.3(b) montre un s-cube de dimension d=3 parce que comportant $N = 2^d$ noeuds. Chaque s-cube a un noeud spécial, appelé *noeud d'E/S*, utilisée comme interface avec l'extérieur, son lien est utilisé comme lien d'entrée/sortie et ne sera pas utilisé pour la connexion avec d'autres noeuds. Les autre noeuds sont considérés comme des éléments de calcul, dont les liens externes servent soit d'interface avec le monde extérieur, soit à relier d'autre s-cubes lors de la construction de plus grands réseaux.

Le plus petit module de base pour la construction d'hypernet peut aussi être un arbre binaire (cf figure A.3(c)), un hypercube ou un bus simple (ou guide d'ondes) (cf figure A.3(b)) connectant N noeuds. Tous ces modules de base sont considérés comme des hypernets de niveau (ou hauteur) $h = 1$: un s-cube, un s-arbre et un s-bus, tous les trois de dimension $d = 3$ parce que comportant $2^d = 2^3 = 8$ noeuds.

Un lien utilisé pour connecter deux noeuds d'un même module est appelé *lien interne*, dans le cas contraire il est appelé *lien externe*. Quand un module de base est utilisé pour construire un plus grand réseau, les liens externes sont utilisés pour relier les modules entre eux, ou servent d'interface avec l'extérieur.

A.2.1 Construction d'hypernets

Un hypernet est caractérisé par un quadruplet (B, d, h, G). B représente l'ensemble des modules de bases utilisés pour la construction du réseau. Chaque module de base dans B a 2^d liens externes. h représente le nombre de niveaux dans la construction hiérarchique, et G caractérise la connectivité globale du réseau. Un choix particulier de B et de G fournit une famille de réseaux ; famille dans laquelle on peut identifier un réseau par le couple (d, h). Un hypernet plus petit avec $i < h$ niveaux formant une sous-structure du réseau (d, h) est appelé sous-réseau (d, i). De ce fait, le sous-réseau $(d, 1)$ n'est rien d'autre que le module de base avec 2^d liens externes.

Un hypernet est construit de la façon suivante : Un sous-réseau de niveau 2 $(d, 2)$ est réalisé par assemblage d'hypernets $(d, 1)$ tel que :

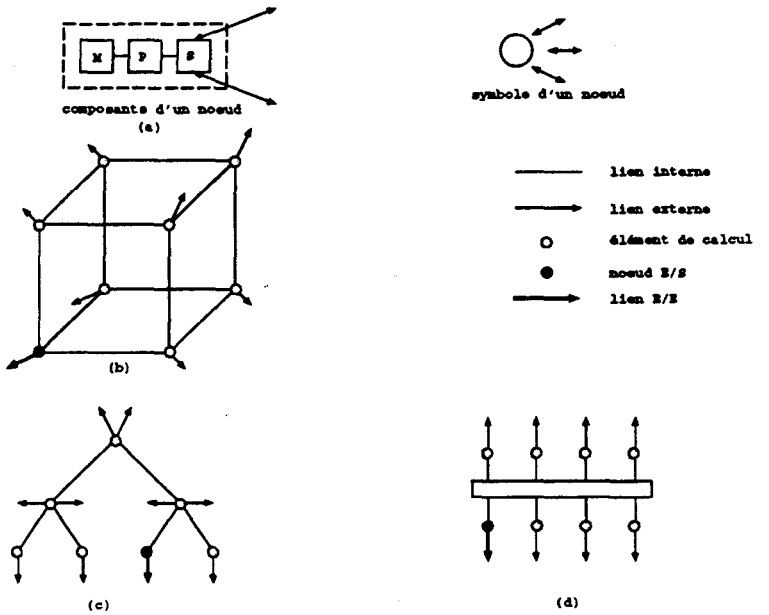


Figure A.3 : Plusieurs types de modules de base. (a) un noeud classique, (b) un s-cube de dimension 3 (de 2^3 noeuds), (c) un s-arbre et (d) un s-bus

1. Il y a G liens directs entre toute paire de modules de base. Ces liens sont appelés *liens de niveau-1* et représentés par λ_1.
2. Dans chaque module, G liens externes représentent les liens d'entrée/sortie et $2^{d-1} - G$ liens font partie des λ_1 nécessaires pour connecter les modules de base entre eux à l'étape précédente.

Des deux étapes ci-dessus, il en résulte qu'il faut $2^{d-1}/G$ modules de bases pour construire le réseau $(d, 2)$. Si chaque module de base $(d, 1)$ représente le sommet d'un graphe, alors le réseau $(d, 2)$ est un graphe de $2^{d-1}/G$ sommets complètement connectés (cf figure A.4(b)). De plus, exactement la moitié de liens restants sur chaque module de base sont disponibles pour la construction de réseaux de niveaux supérieurs. La même méthode avec quelques particularités suivant les modules de base est utilisée pour la construction de $(d, i), i \geq 3$.

Un réseau (d, h) est formé par l'interconnexion de sous réseaux $(d, h-1)$ tels qu'il existe exactement G liens entre toute paire de sous-réseaux $(d, h-1)$; ces liens étant appelés liens de niveau h-1 et représentés par λ_{h-1}. De plus à chaque niveau de construction, exactement la moitié des liens non utilisés sont disponibles pour la construction de niveaux supérieurs. D'autre part, en supposant que N_0 est le nombre de noeuds dans le module de base $(d, 1)$, le réseau (d, h) a les caractéristiques suivantes pour $G = 1$:

- N=(nombre de noeuds)=$N_0 2^{2^{h-1}(d-2)+h+1-d}$
- M=(nombre de sous-réseaux $(d, h-1)$)=$2^{2^{h-2}(d-2)+1}$
- C=(nombre de modules de base)=$\frac{N}{N_0} = 2^{2^{h-1}(d-2)+h+1-d}$
- I=(nombre de noeuds E/S)=$\sum_{i=1}^{i=h} 2^{((2d/2)-1)(2^h-2^i)+h-i}$

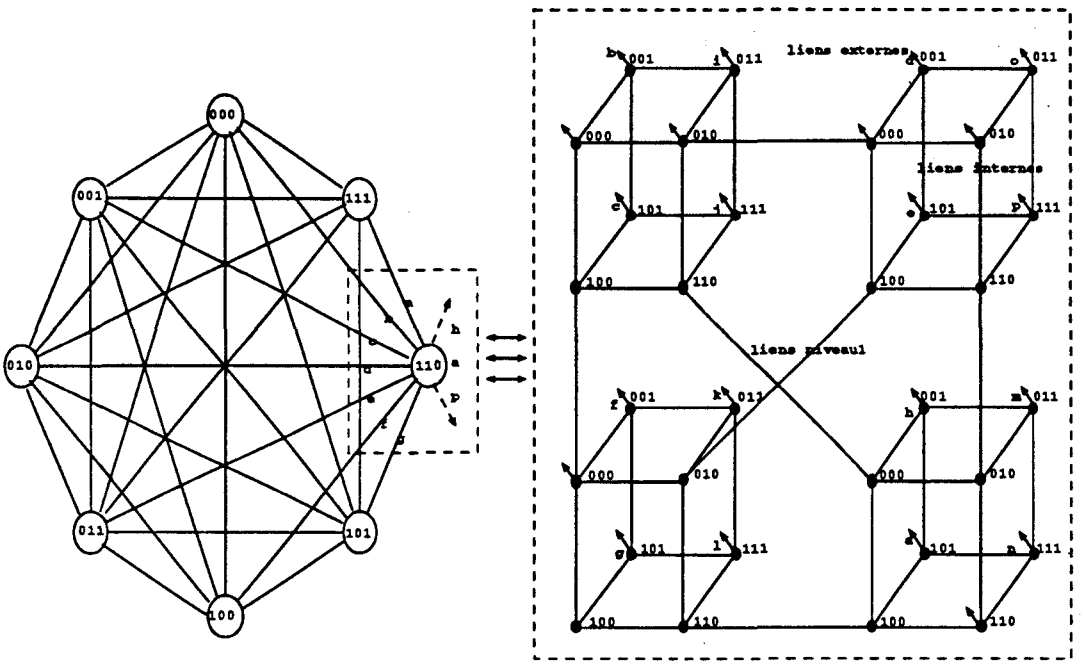


Figure A.4 : Exemple de construction d'un réseau hypernet (3,3) à partir de sous-réseaux (3,2)

- P=(nombre de processeurs de calcul)=$N - I$

La tableA.2 donne quelques chiffres quand $N_0 = 2^d$ et $G = 1$. On peut ainsi vérifier que la taille d'un réseau hypernet croît exponentiellement avec h. Par exemple, un réseau (4,1) a 16 noeuds, un réseau (4,2) a 128 noeuds, un réseau (4,3) a 4096 noeuds et le réseau (4,4) a plus de deux millions de noeuds.

A.2.2 Comparaison avec le Spanning-bus hypercube

Les deux classes de topologies *hypernets* et le *spanning bus hypercube*, comme plusieurs autres explorées au premier chapitre possèdent toutes des propriétés ayant suscité de l'intérêt. Néanmoins un facteur déterminant pour le choix d'une topologie lors de la conception d'une machine parallèle reste la difficulté de câblage ou de l'implantation lors de la réalisation physique.

L'étude faite sur les hypernets a montré une bonne extensibilité qui est exponentielle en fonction de la dimension d tout en conservant le degré physique (nombre de ports physiques) des noeuds constant. Cependant en observant la figure A.4 et la figure A.5, on constate un coût (exponentielle) croissant de croisement entre les liens lors des liaisons entre les modules. La conception physique d'un réseau hypernet à base de guide d'ondes serait pratiquement impossible (une des critiques que l'on pourrait apporter aux hypernets hors du cadre des hyperfréquences serait aussi leur complexité lors de la réalisation VLSI **à cause de leur densité d'interconnexion**), et de ce point de vue, le Spanning-bus hypercube, nettement plus simple est la topologie la plus adaptée pour la construction du réseau *Hypercom* à base de guides d'ondes.

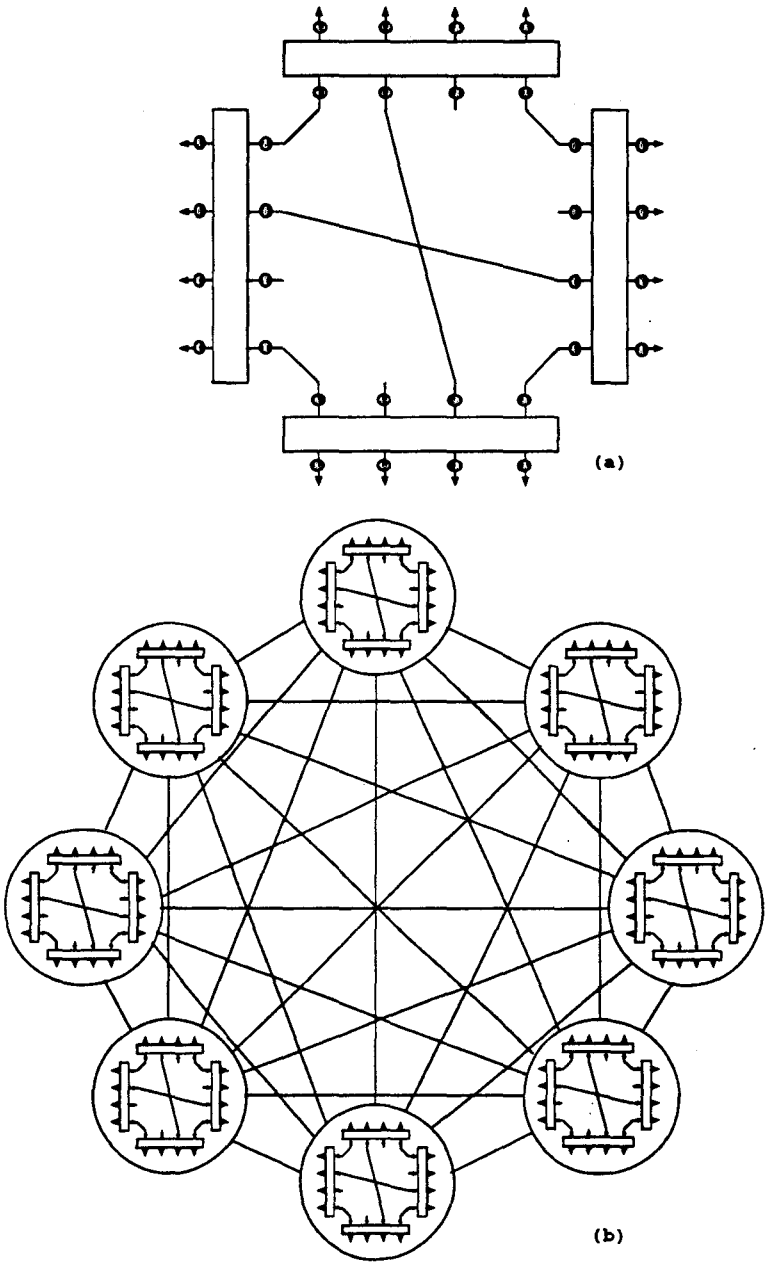


Figure A.5 : Construction de réseau hypernet (3,2) (a), et (3,3) (b) à base de guides d'ondes

Tableau A.2 : Récapitulation des composants formant les hypernets en fonction de d et h ; N_0 le nombre de noeuds par module de base=2^d et G le nombre liens E/S=1

d	h	N (noeuds)	M (s-réseau)	C (s-cube)	P (noeuds de calcul)	I (noeuds E/S)	L (liens externes)
2	2	8	2	2	6	2	4
2	3	16	2	4	10	6	4
3	2	32	4	4	28	4	16
3	3	256	8	32	216	40	64
3	4	8192	32	1024	6880	1312	1024
4	2	128	8	8	120	8	64
4	3	4096	32	256	1760	288	1024
4	4	2^{21}	512	2^{17}	1949184	147968	2^{18}
4	5	2^{38}	131072	2^{34}	$O(15 \times 2^{34})$	$O(2^{34})$	2^{34}
5	2	512	16	16	496	16	256
5	3	65536	128	2048	63360	2176	16384
5	4	2^{29}	8192	2^{24}	$O(31 \times 2^{24})$	$O(2^{24}$	2^{26}
5	5	2^{54}	2^{25}	2^{49}	$O(31 \times 2^{49})$	$O(2^{49}$	2^{50}
5	6	2^{103}	2^{49}	2^{98}	$O(31 \times 2^{98})$	$O(2^{98}$	2^{98}

Références

[1] N. ABRAMSON. Development of the alohanet. *IEEE trans. on Inform. Theory, vol.IT-31,pp. 119-123, Mars*, 1985.

[2] I. ALEKSANDER and H. MORTON. *An introduction to Neural Computing.* Chapman and Hall, 1990.

[3] O. ALLEN. Queuing models of computer systems. *IEEE Computer, No. 4, April*, 1980.

[4] W.C. ATHAS and C.L. SEITZ. Multicomputers: Message-passing concurrent computers. *Computer, vol. 21, No. 8, August*, 1988.

[5] V.E. BENÈS. On rearrangeable three-stage connecting network. *Bull System Technical Journal, XLI(5):1481-1492, September*, 1962.

[6] C. BERGE. *Combinatoires des ensembles finis.* Gauthier-Villars, 1987.

[7] L.N. BHUYAN, Q. YANG, and D.P. AGRAWAL. Performance of multiprocessor interconnection networks. *IEEE Computer, February*, 1989.

[8] I. BOND. Grands réseaux d'interconnexion. *Thèse, Université de Paris-Sud*, 1987.

[9] J. CAPETANAKIS. Generalized tdma : The multiaccessing tree protocol. *IEEE trans. Comm., vol. COM-27, pp. 1476-1484, Oct*, 1979.

[10] F. CAPPELLO. Ptah : étude d'une architecture massivement parallèle à ressources équilibrées et communications compilées. *Thèse, Université Paris Sud, Janvier*, 1994.

[11] P. CHAVEL and N. DE BEAUCOUDREY. *Technologies Matérielles Futures de l'Ordinateur.* Editions Frontières, 1992.

[12] C. CLOS. A study of non-blocking swicthing networks. *Bull System Technical Journal, 32:406-424, March*, 1953.

[13] D. COMTE, N. HIFDI, J.Y. ROUSSELOT, M. FRACÈS, S. KOCON, P. CHUROUX, and J.P. BOUZINAC. Oedip : Architecture optoélectronique massivement parallèle. *Journées Modèles d'exécution et architectures parallèles - Nouvelles technologies pour l'architecture, IRISA, Rennes, France (6-7), December*, 1993.

[14] M. COSNARD and Y. ROBERT. Algorithmique parallèle: une étude de complexité. *Technique et Science Informatique, vol. 6(2):115-125, June*, 1987.

[15] M. COSNARD and D. TYSTRAM. *Algorithmes et architectures parallèles.* InterEditions, 1993.

[16] W.J. DALLY. Performance analysis of k-ary *n*-cube interconnection network. *IEEE trans. on Computers, vol. 39 No. 6, June*, 1990.

[17] W.J. DALLY and C.L. SEITZ. Deadlock-free message routing in multiprocessor interconnexion networks. *IEEE Trans. on Computer, vol C36, No 5, May*, 1987.

[18] K. DEDA, C. LEGAREZ, and L. THOUILLIEZ. Les réseaux locaux sans fil. *Dossier Technique, EUDIL, December*, 1993.

[19] G.R. DESROCHERS. *Principles of Parallel and Multiprocessing, Chapter6: Performance Analysis.* Intertext Publications and McGraw-Hill Book Company, 1987.

[20] F. EL GUIBALY. Design and analysis of arbitration. *IEEE trans. on Computers, vol. 38 No. 2, February*, 1989.

[21] A.H. ESFAHANIAN and S.L. HAKIMI. Fault-tolerant routing in debruijn communication networks. *IEEE trans. on Computers, vol. C34, No. 9, September*, 1985.

[22] SOFT et MICRO. Wave lan, un réseau sur ondes radio. *Soft et Micro, No 98, Juillet-Août*, 1993.

[23] D. SITES et R. WITEK. Alpha architecture technical summary. -, 1993.

[24] D. ETIEMBLE. *Architecture des Processeurs RISC.* Armand Colin, 1991.

[25] D. ETIEMBLE. Impact de la technologie sur les architectures. *Technologies Matérielles futures de l'ordinateur, Mars*, 1992.

[26] T FEN. A survey of interconnection networks. *IEEE trans. on Computers, pp. 12-27 No. 12, December*, 1981.

[27] M.J. FLYNN. Some computer organizations and their effectiveness. *IEEE trans. on Computers, vol. C21, No. 9, September*, 1972.

[28] P. FRAIGNIAUD. Interblocage en routage wormhole. *Journées scientifiques communes, Campus d'Orsay, 9-11 Mars*, 1994.

[29] F. GARDIOL. *Traité d'électricité de l'Ecole Polytechnique de Lausanne.* vol. XIII, Hyperfréquences, Editions Georgi, 1987.

[30] C. GERMAIN, J.L. BÉCHENNEC, D. ETIEMBLE, and J.P. SANSONNET. A communication architecture for a massively parallel message-passing multicomputer. *Journal of Parallel and Distributed Computing, vol 19, pp 338-348*, 1993.

[31] C. GERMAIN-RENAUD. Etude des mécanismes de communication pour une machine massivement parallèles: Mega. *Thèse, Université de Paris-Sud*, 1989.

[32] J.P. GOEDGEBUER, N. BUTERLIN, and H. PORTE. Processeurs systoliques électro-optiques: Performances et limites. *Technologies Matérielles futures de l'ordinateur, Mars*, 1992.

[33] A. GOTTLIEB. Architectures for parallel supercomputing. -, 1992.

[34] P.E. GREEN. The future of fiber-optic computer networks. *Computer, No. 9, September*, 1991.

[35] W. GROPP. Early experiences with the IBM SP1 and the high-performance switch. *Technical Report ANL-93/41, Argone National Laboratory, November*, 1993.

[36] P.S. GUIFOYLE. Digital optical computing technology, performance and perspectives. *Technologies Matérielles futures de l'ordinateur, Mars*, 1992.

[37] J.L. HENNESSY and D. PATTERSON. *Computer Architecture - a quantitative approach.* Morgan Kauffmann Ed., 1990.

[38] J.L. HENNESSY and D.A. PATTERSON. *Architecture des Ordinateurs - Une approche quantitative.* Mc Graw-Hill; édition française par D. Etiemble et M. Israel, 1992.

[39] W.D. HILLIS. *La machine à connexions.* Edtion française par F. Lustman (ed Masson), 1988.

[40] R.W. HOCKNEY and C.R. JESSHOPE. *Parallel Computers.* Adam Hilger Ltd, Great Britain, 1981.

[41] R.W. HOCKNEY and C.R. JESSHOPE. *Parallel Computers-2.* Adam Hilger Ltd, Great Britain, 1988.

[42] T. HSU. Proposition d'une architecture de réseau d'interconnexion à reconfiguration dynamique et asynchrone. *Thèse, Université des Sciences et Technologies de Lille, Février*, 1993.

[43] K. HWANG. *Advanced Computer Architecture.* Mc GRAW-HILL, 1993.

[44] K. HWANG and F.A. BRIGGS. *Computer Architecture and Parallel Processing.* McGraw-Hill, New-York, 1984.

[45] K. HWANG and J. GHOSH. Hypernet: A communication-efficient architecture for constructing massively parallel computers. *IEEE trans. on Computers, vol. C-36, No. 12, December*, 1987.

[46] Performance Computing Industries. The CS-2 hardware architecture : A closer look. *PCI, April*, 1993.

[47] INTEL. The paragon xp/s system at a glance. *Intel*, 1992.

[48] KENDALL SQUARE RESEARCH. Technical summary. -, 1992.

[49] H. KUNG. Why sytolic architectures? *Computers, vol. C15, No. 1, January*, 1982.

[50] F.T. LEIGHTON. *Introduction to parallel algorithms and architectures.* Morgan Kauffmann Ed., 1992.

[51] D. LENOSKI, J. LAUDON, K. GHARACHORLOO, W. WEBER, A. GUPTA, J. HENNESSY, M. HOROWITZ, and M. S. LAM. The stanford dash multiprocessor. *IEEE Computer, March*, 1992.

[52] D. LITAIZE. Architectures multiprocesseurs à mémoire commune. *Deuxième symposium sur les Architectures Nouvelles des Machines PRC ANM - CNRS - MRT, September*, 1990.

[53] D. LITAIZE. La liaison à ultra haut débit: une (la?) solution pour les liaisons inter-module en environnement multiprocesseur. *Technologies Matérielles Futures de l'ordinateur, Mars*, 1992.

[54] A. LOURI. Three-dimentional optical architecture. *IEEE Micro, April*, 1991.

[55] K. MANE. Topologie de réseaux hypergraphes. *Mémoire de DEA, LIFL, Université des Sciences et Technologies de Lille*, 1993.

[56] E. McLellan. The alpha axp architecture and 21064 processor. *IEEE Micro, pp 37-47, June*, 1993.

[57] R. Melhem and D. Chiarulli. Optical computing and interconnection systems. *J.P.D.C. vol.17*, 1993.

[58] T.N. Mudge, J.P.Hayes, and D.C. Winsor. Multiple bus architectures. *IEEE Computer, pp. 42-48, Vol. 19 (6), June*, 1987.

[59] T. Muntean and P. Waille. L'architecture des machines supernode. *La lettre du transputer, 7:11-40, September*, 1990.

[60] K. Murakami, A. Fukuda, S. Mori, T. Sueyoshi, and S. Tomita. The kyushu university reconfigurable parallel processor-design of memory and intercommunication architectures. *In Proc. of ACM SIGARCH 1989 Int. Conf. On Supercomputing, pages 351-360, June*, 1989.

[61] L.M. Ni and P. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer, February*, 1993.

[62] J. R. Nickolls. The deseign of the maspar mp1: a cost effective massively parallel computer. *IEEE Digest of papers-CompCon*, 1990.

[63] W. Oed. The cray research massively parallel processor system cray t3d. *Cray Research GmbH, November 15*, 1993.

[64] J.J. Olsen. Control and reliability of optical networks in multiprocessors. *PhD Thesis, Massachusetts Institute of Technology, May*, 1993.

[65] C. Rochange. Evaluation des performances d'architectures multiprocesseurs à mémoire logiquement partagée. *Thèse, Université Paul Sabatier, Décembre*, 1993.

[66] Ecole Rumeur. *Chapitre1: Les machines parallèles actuelles.* Communication dans les réseaux de processeurs, 1992.

[67] P. Sainrat. Réseau d'interconnexion du multiprocesseur m3s - etude et mise en oeuvre. *Thèse, Université Paul Sabatier, Janvier*, 1991.

[68] A.A. Sawchuk, B.K. Jenkins, C.S. Raghavendra, and A. Varma. Optical crossbar networks. *IEEE Computer, June*, 1987.

[69] C.L. Seitz. Concurrent vlsi architectures. *IEEE trans. on Computers, vol. C33, No. 12, December*, 1984.

[70] C.L. Seitz. The cosmic cube. *Communications of the ACM, vol 28, Number 1, January*, 1985.

[71] A. Seznec, A.M. Kermarrec, and T. Vauleron. Etude comparée des architectures des microprocesseurs mips r4000, dec 21064 et t.i. supersparc. *IRISA*, 1992.

[72] A. Seznec and T. Vauleron. Etude comparative des architectures des microprocesseurs intel pentium et powerpc 601. *IRISA*, 1994.

[73] Simulog. Manuel de référence du langage qnap2. -, 1988.

[74] L. Snyder. Introduction to the reconfigurable, highly parallel computer. *IEEE Computers, vol. C15, No. 1, January*, 1982.

[75] H.S. STONE. *High Performance Computer Architecture.* Adison-Wesley, Reading, Mass., 1987.

[76] R. SUAYA and G. BIRTWISTLE. *VLSI and Parallel Computation.* Morgan Kauffman Ed., 1990.

[77] R.J. SWAN, S.H. Fuller, and D.P. Siewiorek. cm* - a modular multimicroprocessor. *proc. AFIPS 1977 Fall Jiont Computer Conference, pp.637-644*, 1977.

[78] A. TANENBAUM. *RESEAUX Architectures, protocoles, applications.* InterEditions, 1990.

[79] D.M. TAUB. Arbitration and control acquisition in the proposed ieee 896 futurebus. *IEEE Micro, pages 28-41, August 84*, 1984.

[80] P. VANGELUWE. Conception et réalisation d'un transactionnel hyperfréquence en gamme millimétrique, utilisé dans une machine massivement parallèle. *Thèse, Université des Sciences et Technologies de Lille, à paraître*, 1994.

[81] J.H.M. VEENDRICK. Short-circuit dissipation of static cmos circuitery and its impact on thes design of buffer circuits. *IEEE Jou. of Solid-State Circuis, vol. SC-19, No 4, August*, 1984.

[82] R.J. VETTER and D.H.C. DU. Distributed computing with high-speed optical networks. *Computer, February*, 1993.

[83] L.D. WITTIE. Communication structures for large networks of microcomputers. *IEEE trans. on Computers, vol. C30, No. 4, April*, 1981.

Liste des figures

Table des tableaux