



HAL
open science

Dimensionnement de réseaux virtuels de télécommunications

Sébastien Choplin

► **To cite this version:**

Sébastien Choplin. Dimensionnement de réseaux virtuels de télécommunications. Réseaux et télécommunications [cs.NI]. Université Nice Sophia Antipolis, 2002. Français. NNT : . tel-00505397

HAL Id: tel-00505397

<https://theses.hal.science/tel-00505397>

Submitted on 23 Jul 2010

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

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

THÈSE

présentée à

L'UNIVERSITÉ DE NICE – SOPHIA ANTIPOLIS
U.F.R. SCIENCES

pour obtenir le titre de

DOCTEUR EN SCIENCES

Spécialité : **INFORMATIQUE**

DIMENSIONNEMENT DE RÉSEAUX VIRTUELS DE TÉLÉCOMMUNICATIONS

par **Sébastien CHOPLIN**

Soutenue publiquement le 6 novembre 2002 devant le jury composé de :

M. Igor	LITOVSKY	Président
M. Jean-Claude	BERMOND	Directeur
M. Pierre	FRAIGNIAUD	Rapporteur
M. Jaroslav	OPATRNY	Rapporteur
M. David	PELEG	Rapporteur
M. Michel	KOSKAS	Examineur
M. Stéphane	PERENNES	Examineur

À tous ceux qui ont rendu cela possible.

Remerciements

Je tiens tout d'abord à remercier *Jean-Claude Bermond*, effectuer une thèse sous sa direction fût un plaisir, toujours prêt à consacrer du temps pour relire mes travaux malgré son emploi du temps plus que surchargé, il m'a permis d'aboutir dans ce dur labeur ; je ne sais pas si un scooter suffira pour le remercier. Toutes mes excuses à *Simone Bermond* qui a subi les heures de travail de Jean-Claude pour affiner les résultats et corriger les erreurs que je lui laissais.

Merci à *Igor Litovsky* d'avoir accepté de présider mon jury et en m'accompagnant jusqu'en haut de la côte. Je remercie également l'ensemble de mes rapporteurs pour m'avoir fait l'honneur de relire mon travail malgré des délais critiques : *Jaroslav Opatrny*, avec qui travailler fût un plaisir qui je l'espère sera renouvelé ; *Pierre Fraigniaud*, pour avoir remis de l'ordre et de la clarté dans le document ; *David Peleg*, pour avoir accepté d'évaluer mon travail à partir des traductions que lui ai envoyées ; ainsi que mes examinateurs : *Stéphane Pérennes* qui a co-dirigé cette thèse, la plupart des résultats sont issus de discussions, parfois houleuses, avec lui, il voit loin et ne regarde pas toujours où il met les pieds ; *Michel Koskas*, qui m'a initié aux mathématiques discrètes et a accepté de faire partie de mon jury.

Un grand merci à ceux qui m'ont aidé dans les tâches administratives : *Ephie Dérêche*, *Patricia Lachaume* et *Christian Raffaele*.

Mes compagnons de tous les jours, les membres et anciens du projet MAS-COTTE : *Bruno Beauquier* (alias *Bok*), qui a partagé mon bureau et l'a gardé toutes les nuits contre les rôdeurs ; *David Sagnol* avec qui j'ai épuisé les machines à café et qui m'a donné les clefs de Linux, j'en ferai un double pour *Eric Darrot* s'il en a besoin ; *Michel 6K*, sans qui cette thèse serait terminée depuis bien longtemps mais je n'aurais jamais connu Uzez, bien que n'y ayant jamais mis les pieds, j'en connais maintenant tous les bas fonds et les alentours ; *Philippe Mussi*, premier fournisseur de téléphone portable du projet Mascotte ; *Olivier Dalle*, représentant officieux et premier fournisseur RedHat de l'Université de Nice ; *Afonso Ferreira*, premier supporter de l'équipe de football du Brésil, participer à l'organisation de STACS avec lui fût très intéressant ; *Nausica Marlin*, qui m'a initié aux plaisirs du VPL ; *Aubin Jarry*, qui m'a apporté une preuve élégante (ainsi que le café du midi) ; *Frédéric Havet*, notre expert XML ; ainsi que nos collègues de France Telecom : *Alexandre Lauguier*, pour avoir relu la partie technologique pour vérifier que je ne disais pas trop de bêtises ; *Jérôme Galtier*, à qui je promets de terminer nos travaux en cours. Sans oublier les autres membres de l'équipe avec laquelle j'ai passé ces trois années enrichissantes : *David Coudert*, *Hervé Rivano*, *Jean-François Lalande* et *Bruno Bongiovanni*.

Merci à tous les enseignants que j'ai eu le plaisir d'avoir dans ma scolarité et qui m'ont donné le goût de l'enseignement et des mathématiques (*Mme Ely, Christian Blanquart, Mme Deneux et Pierre Capelle*).

Je remercie aussi ceux avec qui j'ai réussi à passer du temps en dehors de mon bureau : mes voisins et néanmoins amis *Karine et Olivier Ruatta, Arnaud Leleu* pour avoir réussi à m'emmener quelques fois refaire du kayak, *Alain et Aude Venisse* pour leur amitié.

Je remercie également tous les amis sans qui mes études auraient été bien tristes (*Sylvain, Elodie, Yohann, Céline, Zdob, Thierry, Arnaud, Lisa, Hervé, Jean-Claude, Stéphane* et tous les autres).

Merci à *Jean-Noël* pour tout ce qu'il m'a apporté.

Toute mon amitié à François et Mary.

Un grand merci à *mes parents* qui m'ont permis de faire des études et m'ont fait confiance en me laissant partir à l'autre bout de la France.

Et enfin merci à *Zaïre* de m'avoir suivi, accompagné et supporté jusque ici, encore quelques obstacles et nous arriverons au bout du parcours.

Et pardon à ceux que j'ai oublié.

Amicalement, *Sébastien*.

Table des matières

Introduction	1
1 Technologie des réseaux étudiés	5
1.1 Quelques notions de réseaux	6
1.1.1 Le multiplexage	6
1.1.2 Le routage	6
1.1.3 Les modes de transfert classiques	7
1.2 Le réseau de transport optique WDM	8
1.2.1 La transmission optique	8
1.2.2 La fibre optique	9
1.2.3 La commutation optique	9
1.2.4 Réseau de troisième génération	10
1.3 Le réseau de transmission numérique SONET/SDH	10
1.3.1 Transmission des données dans les réseaux SONET	11
1.3.2 Correspondance entre les réseaux SONET et SDH	11
1.3.3 Topologie des réseaux SONET/SDH	11
1.3.4 Routage dans une boucle SDH	12
1.4 Le réseau de commutation ATM	13
1.4.1 Inconvénients des modes de transfert classiques	13
1.4.2 Les connexions ATM	14
1.4.3 Transfert des données dans les réseaux ATM	16
1.4.4 Problématique des réseaux ATM	16
1.4.5 Nouvelle problématique	17
2 Modélisation des réseaux	19
2.1 Principes généraux	19
2.1.1 Requêtes	21
2.1.2 Routage	21
2.1.3 Temps de communication	22
2.1.4 Conditions sur les nœuds	22
2.1.5 Réseaux logiques/virtuels	23
2.1.6 Réseaux optiques	23
2.2 ATM et le positionnement de chemins virtuels (VPL)	24

2.2.1	Modélisation	25
2.2.2	Indice de transmission	29
2.2.3	Formulation des problèmes et leur complexité	31
2.2.4	Résultats connus sur les problèmes de VPL	33
3	Le VPL dans le cycle et la chaîne	35
3.1	La chaîne	35
3.1.1	La diffusion	35
3.1.2	L'échange total	39
3.2	Le cycle	44
4	Le VPL dans les arbres	55
4.1	Echange total en 1 sauts	55
4.2	Echange total en 2 sauts	55
4.3	Diffusion en 2 sauts	57
4.4	Diffusion en h sauts	68
4.5	2-to-Many en 2 sauts	76
4.6	Instance quelconque en h sauts	78
4.7	Echange total en 3 sauts	80
5	Extension optique du VPL	81
5.1	Le routage optique sans conversion	81
5.2	Le modèle du VPL optique	83
6	Réseaux hiérarchiques en anneaux	87
6.1	Définitions	87
6.2	Résultats	88
6.3	Idée de preuve	88
6.4	Construire un <i>HRN</i> optimal	90
6.4.1	Calcul de $e(d, D, w)$	91
6.4.2	Tables des valeurs de $e(d, D, w)$	95
6.5	Hierarchical Ring Network Design	99
7	Optimisation du coût d'une boucle SDH	117
7.1	Modélisation	117
7.2	Heuristiques classiques	120
7.2.1	Reroutage d'un chemin	120
7.2.2	Reroutage du flot sur un arc	121
7.3	Bornes inférieures et constructions	121
	Conclusion et perspectives	125

A	Notions élémentaires de théorie de graphes	135
A.1	Définitions	135
A.2	Graphes classiques	137
B	Algorithmes pour le VPL dans l'arbre	141
B.1	Instance <i>One – to – All</i> en h sauts	141
B.2	Programme linéaire pour une instance quelconque en h sauts . . .	144

Introduction

Les résultats obtenus dans cette thèse portent principalement sur le **dimensionnement de réseaux virtuels de télécommunications**.

Le **chapitre 1** décrit les aspects technologiques des réseaux étudiés, leurs enjeux et problématiques. Les réseaux haut-débit actuels consistent en une superposition de plusieurs réseaux. Le réseau de transport des données le plus utilisé aujourd'hui est le réseau à multiplexage en longueur d'onde (WDM) ; tirant partie de la fibre optique celui-ci permet de transmettre des données à très haut débit avec un taux d'erreur négligeable. De plus, le spectre optique d'une fibre pouvant être découpé en un nombre d'intervalles de longueur d'onde de plus en plus grand, ces réseaux permettent de faire du multiplexage et donc d'envoyer plusieurs flux de données simultanément. Généralement ces réseaux ne sont pas utilisés directement pour transmettre les données mais servent de support pour des réseaux virtuels, aussi appelés réseaux logiques (par exemple les réseaux SONET/SDH et ATM), plus génériques faisant ainsi abstraction du support physique. D'autres réseaux virtuels, permettant de gérer plus efficacement les connexions entre les utilisateurs, peuvent être établis sur ces premiers réseaux virtuels.

Les réseaux de type SONET/SDH (établis sur des réseaux WDM) permettent de regrouper le trafic en encapsulant les données de manière hiérarchique et en définissant une politique d'extraction ou d'insertion des données au niveau des nœuds. L'insertion et l'extraction est faite par des composants plus ou moins complexes suivant le nombre de niveaux traités. On peut ainsi chercher à regrouper le trafic intelligemment pour minimiser le nombre des composants les plus coûteux. Le regroupement du trafic entraînant généralement une charge supplémentaire dans le réseau et donc nécessitant des liens de plus grande capacité, on peut aussi chercher à minimiser le coût au niveau des liens.

Les réseaux de type ATM (réseau virtuel établi sur un réseau physique, SDH ou directement sur un réseau WDM) permettent d'établir des connexions rapides, appelées chemins virtuels. Un tel chemin emprunte une suite de liens physiques mais les données transmises ne sont pas analysées dans les nœuds intermédiaires.

Dans ce contexte d'empilement de réseaux, nous cherchons à déterminer des structures de réseau virtuel permettant de satisfaire les exigences des utilisateurs (par exemple en minimisant la distance entre les utilisateurs dans le réseau virtuel) et des opérateurs (en minimisant l'utilisation des ressources du réseau physique par le réseau virtuel).

Le **chapitre 2** décrit l'utilisation de la théorie des graphes pour la modélisation des réseaux de télécommunications. En effet on modélise les réseaux physiques ou virtuels par des graphes où les nœuds du réseau (les points d'accès ou les commutateurs) sont représentés par des sommets, les liens physiques ou virtuels par des arêtes (ou arcs). Les problèmes d'empilement se ramènent à des problèmes de plongement du graphe représentant le réseau virtuel dans le graphe correspondant au réseau physique.

La modélisation du réseau ATM, connue sous le nom de «problème du positionnement de chemins virtuels» (VPL), est introduite ainsi que les problèmes théoriques étudiés dans les chapitres suivants. Dans le problème du VPL, les données sont le réseau physique G et une instance de communications I (ensemble de requêtes à réaliser). Classiquement réaliser une requête revient à trouver un chemin dans le réseau G entre l'émetteur et le récepteur en satisfaisant diverses contraintes (par exemple de capacité ou de qualité de service) ou en optimisant des paramètres du réseau.

Ici le but est de concevoir un réseau virtuel H sur lequel va être réalisée l'instance I . A chaque requête on associe donc un chemin dans le réseau virtuel. Pour garantir un délai minimum de transmission des données entre les utilisateurs, nous fixons la longueur maximale (appelée nombre de sauts maximum) de ces chemins. Par exemple si l'instance de communication est un échange total (chaque nœud doit envoyer un message différent à chaque autre nœud), aussi appelée $All - To - All$, ceci revient à borner le diamètre du réseau virtuel H . Dans ce cas on cherchera un réseau H de diamètre au plus h . Le cas particulier où $h = 1$ correspond au problème classique où on réalise directement l'instance I sur G . Le réseau virtuel H est lui plongé dans le réseau physique en associant à chaque lien du réseau virtuel un chemin dans le réseau physique. La charge d'un lien physique correspond au nombre de liens du réseau virtuel qui utilisent ce lien physique. On cherche parmi les réseaux virtuels satisfaisant les contraintes sur le nombre de sauts celui qui minimise la charge maximale des liens physiques.

Enfin nous présentons les résultats connus sur ce problème se rapprochant le plus de ceux que nous allons étudier.

Le **chapitre 3** est consacré aux résultats obtenus sur le VPL dans le cas où le graphe physique est un cycle ou une chaîne. Pour les chaînes, nous donnons les

constructions optimales pour l'instance de communication d'échange total, ceci quels que soient la taille de la chaîne et le nombre de sauts maximal, ces résultats ont été publiés dans [Cho01]. Pour les cycles et pour l'instance de communication d'échange total, nous donnons les constructions optimales pour un diamètre égal à 2 et des bornes inférieures et supérieures sur la charge maximale pour un diamètre h supérieur à 2.

Toujours dans le cadre du VPL, le cas où le graphe physique est un arbre est traité dans le **chapitre 4**. Nous donnons des algorithmes polynomiaux permettant de résoudre le problème de minimisation de la charge et déterminer le réseau virtuel correspondant pour l'instance de communication Many-To-Many (on veut établir des communications entre deux groupes de nœuds), lorsque le nombre de nœuds de l'un des groupes est borné et avec un nombre maximum de sauts fixé.

Dans le **chapitre 5**, nous présentons une extension du problème du routage optique sans conversion dans les réseaux WDM appelée «problème du VPL optique», consistant à allouer des longueurs d'onde aux liens du réseau virtuel, deux liens virtuels utilisant le même lien physique ne pouvant pas utiliser la même longueur d'onde. Le critère de minimisation de la charge est alors remplacé par celui de la minimisation du nombre de longueurs d'onde. Ce problème apparaît comme difficile car nous avons construit des exemples de réseaux pour lesquels le nombre minimum de longueurs d'onde n'est pas égal à la charge minimum alors que c'est le cas si on réalise les requêtes directement sur le réseau physique.

Les problèmes abordés dans les chapitres 6 et 7 se sont pas liés au problème du VPL mais sont posés dans le cadre des réseaux SONET/SDH.

Le problème étudié dans le **chapitre 6** consiste à déterminer un réseau hiérarchique d'anneaux ayant un maximum de nœuds et un diamètre fixé. Un réseau hiérarchique d'anneaux est une structure définie de manière récursive : au niveau 1 c'est un anneau et au niveau d c'est une structure de niveau $d - 1$ sur laquelle ont été ajoutés des anneaux rattachés par des nœuds construits au niveau $d - 1$. Avec la contrainte d'un diamètre (la distance maximale entre deux nœuds du réseau) fixé, le but est de déterminer le nombre maximum de nœuds d'une structure de niveau d . En utilisant une structure similaire, nous parvenons à estimer ce nombre de nœuds en améliorant ainsi les résultats existant sur ce problème. Ces résultats ont été publiés, l'article est dans la partie 6.5.

Dans le **chapitre 7**, nous étudions un problème d'optimisation lié à la tarification d'une boucle SDH par un opérateur de télécommunications : dans ce contexte, le coût des liens est une fonction concave de leur capacité, il s'agit donc de déterminer un réseau de coût minimum satisfaisant les requêtes. Nous modélisons le problème et donnons des bornes inférieures et supérieures sur ce coût

pour une instance de communication d'échange total unitaire.

Chapitre 1

Technologie des réseaux étudiés

Introduction

Les communications modernes se diversifient et s'intensifient, rendant obsolètes les réseaux classiques offrant un service spécifique. Celles-ci demandent de plus en plus de ressources et de qualité de service. Les réseaux classiques ayant été conçus pour une utilisation particulière (la conversation, la messagerie, la consultation ou la distribution de données par exemple), ils sont peu flexibles et ont donc du mal à s'adapter aux nouveaux besoins.

L'avènement de la fibre optique dans les réseaux de télécommunications permet actuellement de transmettre des données sur des centaines de kilomètres, à des dizaines de Gbit/s, avec des taux d'erreur négligeables. L'apparition de ce type de moyen de transmission à haut débit offre la possibilité d'uniformiser le transport de l'information en utilisant le même support physique pour les différents services. Pour utiliser ce support avec les anciens et nouveaux services, il faut être capable de garantir les mêmes qualités de service que lors de l'utilisation d'un réseau dédié. Mais chaque service ayant des besoins spécifiques (délai garanti pour le téléphone, la vidéo-conférence ...), il faut trouver des protocoles de communication permettant la cohabitation de ces services sur des flux de données hétérogènes. Si les données peuvent transiter très rapidement sur une fibre optique, la commutation des données dans les noeuds entre les fibres peut être très coûteuse en temps (la plupart des commutateurs optiques effectuant une conversion électronique) et des protocoles aussi complexes que ceux implémentés dans les réseaux classiques ne peuvent actuellement être mis en place aux débits requis, à des coûts raisonnables. Aussi, la préoccupation principale du concepteur de réseau haut débit est de simplifier les protocoles pour permettre de réaliser les opérations électroniques suffisamment rapidement pour ne pas trop ralentir le débit offert par la fibre optique.

1.1 Quelques notions de réseaux

1.1.1 Le multiplexage

Nous appellerons *voie de communication* tout moyen par lequel l'information peut être transmise entre deux nœuds du réseau. Une voie de communication peut être réservée pour un seul utilisateur. Cela se justifie si cet utilisateur a besoin de l'essentiel de sa capacité (débit, ou bande passante). Ce n'est pas le cas de la majorité des applications. En outre, affecter toute la capacité à un utilisateur brime les autres utilisateurs. On appelle multiplexage l'opération qui permet le partage d'une voie de communication entre plusieurs voies. Il existe de nombreux moyens pour réaliser du multiplexage, voici les principaux utilisés.

Le multiplexage temporel consiste à partager la voie de communication en tranches de temps constantes. Ces tranches se répètent à l'identique dans le temps et sont elles-mêmes découpées en intervalles temporels ; chaque utilisateur ayant établi une connexion dispose d'un intervalle par tranche. Cette technique est utilisée intensivement dans les applications temps réel et en téléphonie. Par exemple la tranche de temps utilisée pour le réseau téléphonique est de $125\mu\text{s}$. Chaque tranche est découpée en 32 intervalles temporels de 1 octet.

Le multiplexage spatial consiste à séparer physiquement la voie de communication, comme par exemple un câble à brins multiples. Cette technique est peu extensible car souvent la séparation spatial du support est limitée et son évolution nécessite souvent une intervention physique au niveau de l'équipement.

Le multiplexage en fréquence (qui peut être considéré comme un multiplexage spatial) consiste à découper la bande passante d'un canal en plusieurs sous-bandes, chaque sous-bande étant affectée à une voie de communication. Cette technique est utilisée dans les réseaux optiques, où le spectre optique de chaque fibre est découpé en longueurs d'onde. La partie 1.2 décrit plus en détail ces réseaux.

Le multiplexage statistique consiste à multiplexer plusieurs communications sur la même voie en considérant que les communications n'utilisent pas la totalité des ressources qui leur sont allouées en même temps. Cette technique ne peut s'appliquer qu'en cas de flux sporadique (variant beaucoup dans le temps).

1.1.2 Le routage

Les réseaux sont construits à partir d'un maillage de voies interconnectées. En général, les interconnexions (carrefours) relient plus de deux voies. On appelle

nœud de commutation ou commutateur ou routeur un composant qui interconnecte deux voies ou plus. Ces nœuds de commutation assurent l'aiguillage des messages arrivant sur une voie vers la voie de sortie qui permettra au message reçu d'atteindre sa destination. Les nœuds de commutation assurent le choix des chemins à un carrefour.

Le routage est la fonction exécutée dans chaque routeur qui permet d'acheminer localement un message vers sa destination. Le routage utilise pour cela des informations d'acheminement qui sont à sa disposition. Ces informations peuvent être des tables, des vecteurs, des listes, . . . , de routage.

L'algorithme de routage décrit les règles qui permettent d'assurer cette fonction d'acheminement des messages dans le réseau. Il permet de trouver les chemins qui vont d'un point à un autre et a pour but de renseigner les tables, vecteurs, listes, . . . , de routage.

Le routage statique consiste à fixer de manière permanente (au moins pour une période de temps assez longue) les tables de routage. Les routeurs statiques ont besoin de l'intervention du gestionnaire du réseau pour mettre à jour les tables de routage lorsqu'une modification a lieu ou lorsqu'un lien tombe en panne. Ce type de routage permet une optimisation fine lorsque l'on connaît précisément les besoins des utilisateurs mais n'est pas envisageable dans le cas d'un réseau dont la topologie est susceptible de subir des changements imprévisibles ou si le trafic varie beaucoup.

Le routage dynamique ne fixe pas les tables de routage à l'avance. Les routeurs dynamiques utilisent des protocoles de routage spécifiques qui régissent les échanges d'informations de routage à travers le réseau.

1.1.3 Les modes de transfert classiques

Le mode de transfert est la technique utilisée dans un réseau pour effectuer la transmission, la commutation et éventuellement le multiplexage des données. On distingue deux type de modes de transfert : le mode connecté (utilisé dans les réseaux téléphoniques par exemple) et le mode non-connecté (utilisé dans le réseau Internet par exemple).

Le mode connecté. Lorsqu'on veut réaliser une communication, une liaison est établie entre les correspondants avant que ne commence l'échange des informations et cette liaison est détruite à la fin de l'échange par une action explicite (en raccrochant le combiné téléphonique pour l'exemple du téléphone). Cette liaison est fixée pendant tout l'échange, sa capacité est constante et les ressources qu'elle

réserve ne peuvent pas être utilisées par une autre connexion, même lorsque le flux de données transmis est inférieur à la capacité réservée. Ce mode évite le stockage de l'information dans les nœuds intermédiaires du réseau et permet de réduire son délai d'acheminement.

Le mode non-connecté. Les messages issus du flux de communication avancent dans le réseau vers leurs destinations en transitant par les nœuds intermédiaires où ils sont stockés entièrement avant d'être réémis. À chaque étape, le canal emprunté est aussitôt libéré. Cette technique nécessite l'implémentation de protocoles complexes au niveau de chaque commutateur pour stocker l'information et la router car aucune route n'est préétablie pour acheminer le message.

REMARQUE. Lorsque le message est découpé en plusieurs morceaux, il peut arriver que les morceaux utilisent des routes différentes pour aller à la destination et peuvent donc ne pas arriver dans le même ordre que celui dans lequel ils sont partis ; un réordonnement de morceaux est alors nécessaire pour la reconstruction du message original.

REMARQUE. Cependant certains protocoles comme MPLS (MultiProtocol Label Switching) utilisent le mode non-connecté tout en imposant l'utilisation d'une seule route pour une communication.

1.2 Le réseau de transport optique à multiplexage en longueurs d'onde WDM

La technologie de multiplexage en longueurs d'onde (*Wavelength-Division Multiplexing*, WDM en anglais) permet le multiplexage en longueur d'onde du signal optique passant dans une fibre optique en plusieurs canaux WDM. Le spectre optique est découpé en un certain nombre d'intervalles de longueurs d'onde (ou fréquences) distincts, de telle sorte que chaque intervalle supporte un canal de communication transmettant au débit désiré.

1.2.1 La transmission optique

En quoi réside l'intérêt des liaisons optiques et quelle est la technologie nécessaire à leur utilisation ? Elle comprend une source de lumière laser, qui est une diode laser analogue à celle des lecteurs de disques compacts, fonctionnant près de l'infrarouge (à des longueurs d'onde allant de 1,2 à 1,6 μm). La lumière émise est modulée par un transmetteur, système commandé par le signal électrique qui apporte l'information. Ces impulsions lumineuses sont envoyées dans la fibre et récupérées à l'autre extrémité par un récepteur (une photodiode) qui reconvertit

le signal optique en signal électrique, lequel est enfin transformé en son, image ou texte, dans le téléphone, la télévision ou l'écran d'ordinateur.

Comme dans tout système de communication numérique, l'information est codée sous la forme d'une succession de *bits* (de *binary digit*, pour désigner 0 ou 1). Dans une fibre optique, les bits sont acheminés physiquement par une onde lumineuse dont on module l'intensité : le temps est divisé en créneaux de même durée, pendant lesquels le bit 1 est codé par une impulsion lumineuse et le bit 0 par une absence de lumière, comme pour la lecture des disques laser compacts.

1.2.2 La fibre optique

La fibre optique possède un grand nombre de propriétés remarquables qui en font un support physique excellent pour les télécommunications : son débit, une faible atténuation, une faible dispersion et un taux d'erreur très faible. Le débit est directement lié la vitesse de la lumière, soit 300 000km/sec. La faible atténuation du signal permet d'avoir des fibres optiques de l'ordre de la centaine de kilomètres sans avoir besoin de régénérer le signal. Un signal trop faible ne se distinguera pas du bruit de fond et engendrera des erreurs. La dispersion représente la dépendance de la vitesse de propagation par rapport à la fréquence, les différentes longueurs d'onde n'ayant pas la même vitesse de transmission, une dispersion importante engendrera des chevauchements entre les différentes impulsions provoquant des erreurs. Les fibres optiques sont classées par catégorie appelées **OC** (*Optical Carrier*) suivant leur niveau de multiplexage. Ainsi une fibre optique de type OC- n correspond au multiplexage en n longueurs d'ondes.

1.2.3 La commutation optique

La plupart des réseaux actuels procèdent électroniquement au traitement des données et utilisent la fibre optique seulement comme un support de transmission. Les opérations de commutation et de routage sont réalisées en convertissant les signaux optiques sous leur forme électronique originelle. Il en résulte une grande flexibilité, cependant le débit de l'électronique n'est pas compatible avec l'énorme bande passante de la fibre optique.

De plus, la conversion opto-électronique introduit un délai supplémentaire dans le routage des flots d'information. Pour s'affranchir de ces limitations des composants de commutation optiques ont été développés, capables de traiter des signaux optiques à très haut débit sans conversion opto-électronique.

Pour cette raison, les réseaux utilisant cette technologie de commutation sont appelés **réseaux tout-optiques**. Dans l'état actuel, le contrôle des fonctions de

commutation demeure cependant réalisé de façon électronique.

1.2.4 Réseau de troisième génération

On distingue trois générations de réseau :

- la première génération de réseaux ne fait aucune utilisation de la fibre optique,
- la deuxième génération de réseaux, les transmissions électriques des données sont remplacées par des systèmes de transmission optiques ; l'architecture traditionnelle des réseaux est cependant conservée,
- la troisième correspond aux réseaux tout-optiques.

1.3 Le réseau de transmission numérique SONET/SDH

Le système de transmission définit la manière dont sont acheminées les données sur le support physique. Nous décrivons ici l'un des principaux systèmes utilisés aujourd'hui : SDH(Synchronous Digital Hierarchy).

Pendant les années 1960-70, les progrès dans le domaine des circuits de semi-conducteurs et la demande sans cesse croissante des besoins de téléphonie ont fait apparaître une nouvelle méthode de transmission appelée PCM(Pulse Code Modulation). PCM permettait le multiplexage en temps sur un lien et donc son utilisation pour plusieurs flux de communication. Le débit nécessaire pour une ligne téléphonique est de 64 kilobits par secondes. Sur un même lien, 30 lignes étaient combinées pour obtenir un flux de 2048 kbits/s (Aux Etats-Unis, au Canada et au Japon, le standard était de 24 lignes et donc un flux de 1544kbits/s).

L'augmentation de la demande en bande passante nécessitait de combiner plus de lignes. Comme l'insertion ou l'extraction d'une ligne dans un lien nécessite des équipements complexes, une notion de hiérarchie fut introduite ; un niveau de la hiérarchie étant une combinaison des flux du niveau inférieur. Ainsi l'extraction ou l'insertion se fait par niveau et évite d'avoir à traiter tous les flux de base individuellement.

Ces réseaux de deuxième génération sont apparus à la fin des années 1980 et baptisés Synchronous Optical Network (SONET) et Synchronous Digital Hierarchy (SDH). Ils constituent aujourd'hui la quasi-totalité des réseaux physiques basés sur la fibre optique. Ils utilisent la technologie WDM pour la transmission optique, mais la conversion ce fait la plupart du temps en opto-électronique. Ils permettent l'unification de réseaux hétérogènes (autres que les réseaux téléphoniques) pour un coût et une gestion réduite du fait de leur structure hiérarchique.

De plus, cette technique peut s'adapter aux demandes futures en ajoutant des niveaux de hiérarchie.

1.3.1 Transmission des données dans les réseaux SONET

Le système SONET est synchrone, c'est-à-dire commandé par une horloge qui rythme la transmission des bits à intervalle de temps très précis. La transmission de l'information se fait dans des trames de 810 octets (contenant une partie de contrôle et une partie de données). Les trames de base sont émises en permanence toutes les $125 \mu s$ (8000 fois par secondes), qu'il y ait ou non des données à transmettre. La cadence de transmission de 8000 trames par seconde correspond exactement à la vitesse d'échantillonnage des systèmes téléphoniques numériques. Cette structure constitue le canal SONET de base, appelé STS-1 (Synchronous Transport Signal - 1). Toute artère de communication SONET est un multiple de STS-1. Les données transmises sont encapsulées dans des containers virtuel (VC) qui sont multiplexés en temps dans les trames.

1.3.2 Correspondance entre les réseaux SONET et SDH

Sur leur principe fonctionnel, SONET (utilisé aux Etats-Unis, au Canada et au Japon) et SDH sont assez semblables, et ne diffèrent que dans la hiérarchie numérique. Il existe une correspondance entre les deux systèmes pour permettre leur interopérabilité. Le premier niveau de SDH, STM-1 (Synchronous Transport Module-0), est équivalent à STS-3 de SONET, les niveaux supérieurs étant compatibles avec ceux de SONET. Le tableau de la figure 1.1 donne les correspondances entre les deux systèmes hiérarchiques.

SONET	SDH	Débit (Mbits/s)
STS-1	-	51,84
STS-3	STM-1	155,52
STS-12	STM-4	622,08
STS-48	STM-16	2 488,32
STS-192	STM-64	9 953,28

FIG. 1.1 – Correspondances SONET-SDH

1.3.3 Topologie des réseaux SONET/SDH

Une topologie décrit comment le réseau est construit, c'est-à-dire comment les voies de communication relient les nœuds les uns aux autres. Un réseau SDH est composé de liens optiques sur lesquels circulent des containers virtuels VC, les nœuds connectant ces liens sont des Multiplexeurs par Insertion-Extraction

(MIP, Add-Drop Multiplexor (ADM) en anglais) qui permettent l'extraction ou l'insertion de données entre les trames STM. La topologie d'un réseau SDH peut être quelconque, mais est souvent constituée d'une boucle où d'une interconnexion de boucles. Cette structure en boucle est justifiée par son faible coût de mise en œuvre et sa simplicité de routage et de reroutage en cas de panne.

Nous décrivons ici la boucle unidirectionnelle composée de deux anneaux dans laquelle les informations circulent toujours dans le même sens (sauf en cas panne). Dans l'exemple de la figure 1.2, si le lien entre les nœuds A et B est coupé (en fait le lien de chaque anneau car le plus souvent ces liens passent par le même endroit géographique et la défaillance d'un lien est généralement causée par une destruction physique du lien), toutes les connexions peuvent être reroutées en utilisant l'anneau secondaire (par exemple : avant la panne, pour aller du nœud D au nœud A, un VC passait par C puis B tandis qu'après la panne du lien entre A et B, il passera par C puis B pour revenir en C puis en D pour arriver en A).

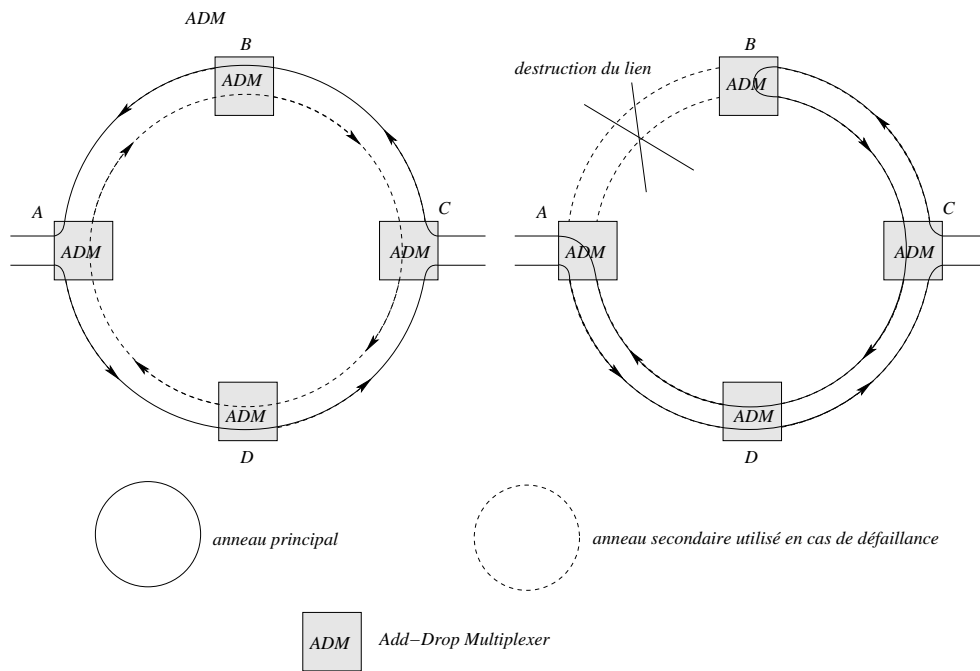


FIG. 1.2 – Boucle SDH unidirectionnelle subissant une défaillance

1.3.4 Routage dans une boucle SDH

Les données qui circulent dans une trame STM peuvent soit être interceptées par le commutateur soit ne pas être interceptées et continuer leur route jusqu'au

prochain commutateur. La politique de routage visant à décider si les trames doivent être extraites ou non est fixée par l'opérateur et non par les données qui circulent dans la trame (il peut par exemple décider d'extraire les trames à intervalles réguliers, on voit ici l'intérêt d'avoir un système de transmission synchrone). Lorsque des données sont extraites, c'est toute la trame qui doit être extraite. Ainsi la politique de routage au niveau de la boucle SDH définit un réseau logique où les nœuds sont les commutateurs et les liens sont définis entre deux nœuds par le fait que les données d'une trame sont insérées dans le premier nœud et extraites dans le second. Le routage des données des utilisateurs se fait donc dans ce réseau logique. Lorsqu'un lien dans ce réseau logique est établi par l'opérateur entre le nœud x et le nœud y , il est établi généralement un lien utilisant l'autre partie de la boucle entre le nœud y et le nœud x , ceci pour des raisons de contrôle du trafic et donc ce deuxième lien ne peut pas être utilisé pour transporter d'autres données. Par habitude la plupart des opérateurs établissent les liens de manière symétrique et font en sorte que les routages soient eux aussi symétriques.

Dans ce cadre, nous étudierons dans le chapitre 7 le dimensionnement des liens d'une boucle SDH en fonction d'un trafic donné cherchant à minimiser le coût d'établissement d'un tel réseau, ce problème ayant été posé par France Telecom R&D dans le cadre de la CTI «Dimensionnement pour les services SDH aux entreprises». D'un point de vue plus général, nous nous intéresserons dans la même partie au dimensionnement d'un réseau hiérarchique d'anneaux avec une distance (nombre de liens) maximale entre les nœuds, ce problème pouvant être rattaché à l'interconnexion de boucles SDH.

1.4 Le réseau de commutation ATM

La solution la plus répandue pour gérer l'hétérogénéité des communications dans les réseaux est le *Mode de Transfert Asynchrone* (ATM), longuement décrit dans la littérature [Tan95, KG98, SB99]. La technologie ATM peut être utilisée sur n'importe quel système de transmission, le principal utilisé étant SDH décrit dans la partie 1.3.

1.4.1 Inconvénients des modes de transfert classiques

La commutation de circuits avait l'avantage de transmettre les informations de manière transparente sur le réseau (c'est-à-dire sans inversion dans l'ordre des données reçues, en effet si le message est découpé en paquets il se peut que ces paquets n'arrivent pas dans le même ordre que celui dans lequel ils sont partis s'ils n'empruntent pas les mêmes routes dans le réseau) et de minimiser les fonctions des commutateurs et des multiplexeurs qui pouvaient donc fonctionner

à haut débit. L'inconvénient de ce mode de transfert est son manque de flexibilité. Il n'est pas possible de l'implémenter efficacement pour transporter des trafics différents. Le multiplexage statistique est quasiment impossible. Pour répondre à ce problème, ce mode a évolué progressivement. D'abord a été proposé le mode *commutation de circuits multi-débit* dans lequel une connexion peut réserver plusieurs voie de communication. Mais une connexion de débit élevé doit alors réserver des milliers de voies ; cela est difficile à gérer. De plus, la capacité réservée pour une connexion est constante, ce qui empêche encore le multiplexage statistique. Pour palier ce dernier problème, a été mis en place un mode appelé *commutation de circuits rapide* qui permet aux sources d'annoncer et de modifier le débit dont elles ont besoin. Mais ce système complexifie trop les protocoles effectués au niveau électronique et limite le débit.

Le mode commutation de paquets quant à lui présentait l'avantage de la flexibilité puisque quand une source ne transmet rien, les ressources qu'elle n'utilise pas sont récupérées pour d'autres communications. Même lorsque le trafic est sporadique, les ressources sont correctement utilisées. L'inconvénient de ce mode est la complexité des protocoles à traiter. En simplifiant les protocoles de détections d'erreur (moins utiles à présent car la fibre optique est un support de transmission très fiable) et les protocoles de contrôle du flux, on obtient un mode plus simple appelé *relais de trame* qui permet des débits plus élevés mais n'atteignant pourtant pas la centaine de Mbit/s dont certaines applications ont besoin.

La construction de réseaux haut débit nécessite donc une évolution des modes de transfert des données.

Les idées développées pour faire évoluer les modes commutation de circuits et commutation de paquets les ont finalement fait converger vers un mode de transfert à la fois simple et flexible qui permet le transfert à haut débit de flux de données hétérogènes et sporadiques. C'est le mode de transfert asynchrone (*Asynchronous Transfer Mode*, ATM).

1.4.2 Les connexions ATM

Les connexions ATM sont des liaisons point-à-point entre un ordinateur et un commutateur ATM ou entre deux commutateurs ATM, ce qui implique que les nœuds intermédiaires ne peuvent pas intercepter les données entre la source et la destination (contrairement aux réseaux locaux qui acceptent le plus souvent plusieurs équipements source et destinataire sur un même support). On distingue deux type de connexions :

Conduit virtuel. Chaque connexion physique est divisée logiquement en un certain nombre de conduits virtuels (Virtual Path, noté VP). Une connexion de

conduit(s) virtuel(s) (Virtual Path Connexion, noté VPC) se compose d'un VP ou de la concaténation de plusieurs VP.

Voie virtuelle. La capacité de chaque VP est divisée logiquement en un certain nombre de liens ATM appelés voies virtuelles (Virtual Channel, VC). Une connexion de voie virtuelle (Virtual Channel Connexion, VCC) se compose d'une VC ou de la concaténation de plusieurs VC. Un VC ne peut commencer et finir qu'en un point de connexion de VPC.

La figure 1.3 schématise les VPs et les VCs d'un lien physique.

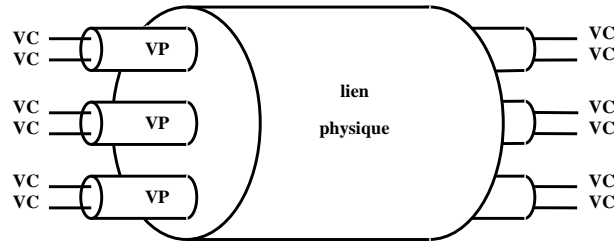


FIG. 1.3 – Schématisation des VPs d'un lien physique

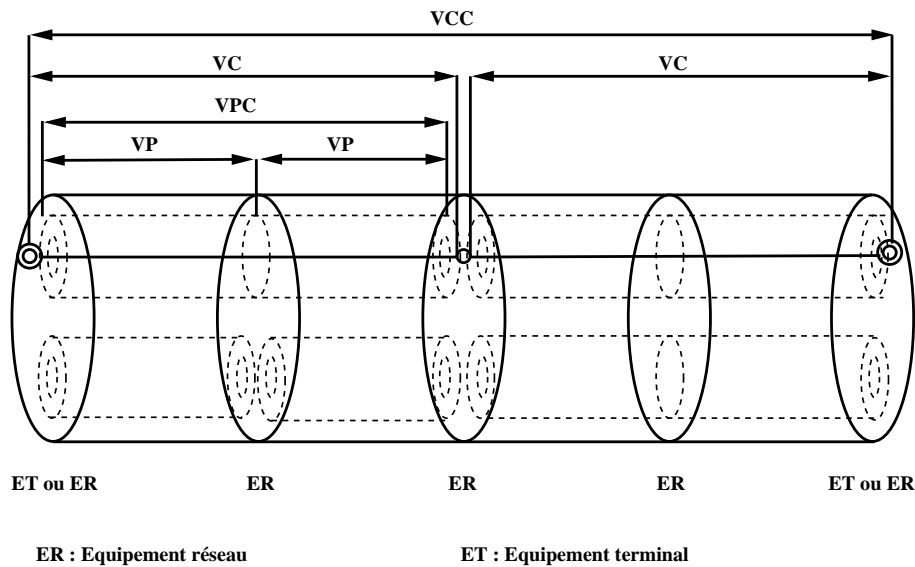


FIG. 1.4 – Connexion ATM - VPC et VCC

Un réseau ATM s'appuie donc sur un réseau de liaisons dit *réseau physique* et sur un réseau de liaisons virtuelles (VPC). Une communication entre deux nœuds

est réalisée via une connexion de voie virtuelle (VCC).

1.4.3 Transfert des données dans les réseaux ATM

Le transfert des données dans les réseaux ATM est basé sur des paquets de taille fixée (53 octets), appelés *cellules*. En comparaison avec SDH, le transfert des cellules ATM est fait de manière asynchrone : il n'y a pas d'exigence particulière dans le flux binaire sur la régularité d'apparition des cellules. Des espacements variables peuvent apparaître entre les cellules de données, ils sont comblés par le système par des cellules spéciales de 'bourrage'. Le routage du réseau virtuel des VPC dans le réseau physique est statique, il est donc établi pour une période relativement longue et géré directement par l'opérateur. Les données qui traversent un VPC transitent de manière transparente, sans être interceptées, de la source vers la destination. La commutation de circuit est utilisée pour transmettre les cellules sur les VCC dans le réseau des VPC, pour se faire, chaque cellule possède un identifiant précisant sur quel VCC elle est acheminée et chaque commutateur ATM dispose d'une table pour l'envoyer dans le VPC suivant. Le multiplexage statistique est généralement utilisé au niveau des VCC, mais c'est une partie que nous n'étudierons pas.

1.4.4 Problématique des réseaux ATM

Le concepteur d'un réseau ATM devra donc établir le réseau des VPCs (y compris leur routage dans le réseau physique) de manière à pouvoir écouler le trafic dans des VCC. Un lien physique ne peut avoir qu'un nombre limité de VPCs le traversant. Comme nous l'avons déjà fait remarquer, les cellules traversent un VPC de manière transparente ; par contre lorsqu'elles changent de VPC, les cellules sont traitées par le commutateur (qui met à jour leur entête), ce qui ralentit leur vitesse de transmission. Pour garantir un délai à l'utilisateur, il faut donc que la VCC qu'il aura établi utilise un nombre limité de VPCs.

Le concepteur d'un réseau ATM devra donc prendre en compte ce paramètre lors de l'établissement des VPCs sans perdre de vue que les liaisons physiques ne peuvent accueillir qu'un nombre limité de VPs. Il devra donc trouver réseau virtuel dans lequel les nœuds désirant communiquer sont proches (dans le réseau des VPs) tout en utilisant au mieux les ressources du réseau physique. Le problème consistant à déterminer le réseau des VPCs porte le nom de «Positionnement de Chemins Virtuels» (Virtual Path Layout, VPL en anglais). Les thèses de P. Chanas [Cha98] et N. Marlin [Mar00] (dont celle-ci est le prolongement) traitent directement de ce problème d'un point de vue théorique.

1.4.5 Nouvelle problématique

Dans le chapitre 5, nous introduirons un nouveau problème consistant à prendre en compte le niveau optique dans les réseau ATM : l'un des paramètres supplémentaires étant le nombre de conversions au niveau optique des chemins virtuels. En effet, ceux-ci traversent les commutateurs ATM mais traversent aussi (dans la couche optique) des commutateurs optiques. La conception du réseau optique sous-jacent peut être optimisée si le nombre de fois que les chemins virtuels empruntent des longueurs d'ondes différentes (c'est-à-dire sont commutés au niveau optique) est minimisé.

Chapitre 2

Modélisation des réseaux

Introduction

Pour étudier les réseaux de télécommunications et concevoir des algorithmes de communication, il convient de définir un modèle théorique. Nous allons décrire quelques types des modèles, certains simples ne prenant en compte qu'un nombre limité de paramètres, d'autres complexes collant de plus près à la réalité. Les modèles complexes sont souvent lourds à manipuler ; au contraire, les modèles plus simples s'étudient plus aisément, sans signifier que les problèmes à résoudre soient simples. Il convient donc, lorsque l'on veut choisir un modèle, de trouver un compromis entre son pouvoir d'expression et sa complexité.

Les réseaux de télécommunications peuvent être modélisés par des graphes, les notions élémentaires de théorie des graphes nécessaires à la compréhension sont fournies dans l'annexe A. Pour plus de détails, le lecteur pourra trouver les notions non rappelées dans le livre de l'action RUMEUR [dR94], dans les livres de théorie des graphes de Berge [Ber83], de Bondy et Murty [BM76].

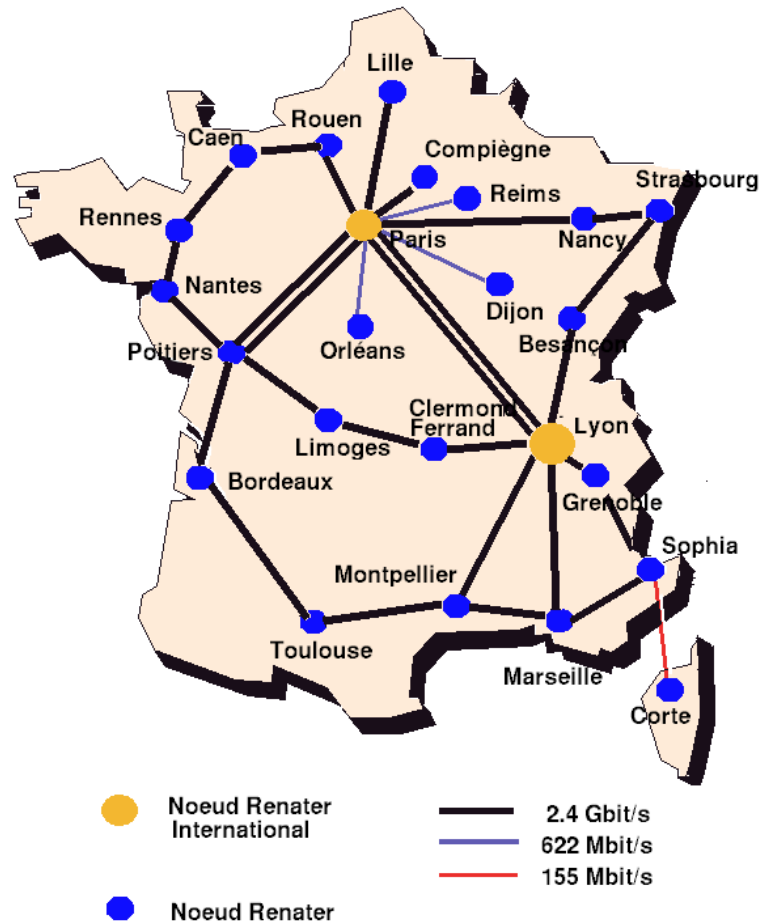
En règle générale les nœuds du réseau (des points d'accès ou des commutateurs) sont représentés par des sommets, les liens physiques unidirectionnels par des arcs et les liens physiques bidirectionnels par des arêtes.

2.1 Principes généraux

Le réseau peut être modélisé par un graphe orienté ou non-orienté, suivant que les liens seront considérés comme unidirectionnels ou bidirectionnels, les sommets du graphe représentant les nœuds du réseau et les arcs ou les arêtes les liaisons physiques entre ces nœuds. On peut attribuer des valeurs particulières aux sommets et aux arcs (ou arêtes) comme la capacité (ou débit) d'un lien, sa charge, ...

La figure 2.1, issue du site web de RENATER¹ donne une modélisation du réseau RENATER en France métropolitaine par un graphe non orienté. Les sommets représentent les villes et les arêtes représentent les liaisons haut débit entre ces villes. A chaque arête du graphe est associé le débit (en Mbits/sec) du lien correspondant. Le réseau RENATER s'appuie sur la technologie ATM ainsi que sur les technologies optiques.

LE RESEAU RENATER-3



Renater

Paris, 27 juin 2002

RENATER-3

FIG. 2.1 – Le réseau RENATER

¹Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche - <http://www.renater.fr>

2.1.1 Requêtes

On appellera requête une connexion à établir dans le réseau entre deux nœuds. Une requête est donc la donnée d'un couple de nœuds du réseau. Les requêtes peuvent être connues à l'avance par le concepteur du réseau ou être données de manière dynamique une fois le réseau établi. Généralement ces requêtes sont pondérées par une valeur représentant le débit requis entre les deux nœuds, ou la taille du message à envoyer.

Instances de communication

On appelle instance de communication une famille de requêtes. Les plus classiques sont :

- **La diffusion** (*one-to-all* ou *broadcasting*) consiste à envoyer un message à tous les nœuds du réseau à partir d'un initiateur unique. Dans la modélisation, l'ensemble des requêtes sera alors $\{(x, y) | y \in V - x\}$ si x représente l'initiateur et V l'ensemble des nœuds du réseau.
- **Le multicast** (*multicast*), consiste à envoyer un message à un groupe de nœuds du réseau. L'ensemble de requête sera $\{(x, y), y \in \text{groupe}\}$.
- **L'échange total** (*all-to-all*, *total exchange* ou *gossiping*) consiste à effectuer une diffusion à partir de tous les nœuds simultanément. Dans la modélisation, l'ensemble des requêtes sera alors $\{(x, y) | (x, y) \in V \times V, x \neq y\}$ si V représente l'ensemble des nœuds du réseau.
- **La distribution** (diffusion personnalisée, *personalized one-to-all*, *distributing* ou *scattering*) consiste, pour un initiateur unique, à envoyer un message différent à chacun des autres nœuds.

Dans cette thèse, nous étudierons principalement les instances d'échange total et de diffusion.

2.1.2 Routage

Les routeurs/commutateurs réalisent de façon distribuée un algorithme de routage qui spécifie le chemin à suivre dans le réseau pour se rendre de tout nœud x vers tout nœud y . Dans le cadre de la modélisation, on appelle **routage** d'une requête entre les nœuds x et y la donnée d'une chaîne si le graphe est non orienté, d'un chemin si le graphe est orienté, reliant x à y .

REMARQUE. Si le réseau est modélisé par un graphe orienté symétrique, que les requêtes sont symétriques (si on a une requête du nœud x vers le nœud y alors on a aussi la requête du nœud y vers le nœud x) et qu'on souhaite router ces

requêtes en utilisant des chemins symétriques, on peut modéliser le réseau par un graphe non orienté et considérer chaque paire de requêtes symétriques (x, y) et (y, x) comme une seule requête correspondant à la paire de nœuds $[x, y]$. Ainsi lorsqu'on cherche un routage dans le réseau réalisant la requête $[x, y]$, on cherche une chaîne entre x et y dans un graphe non-orienté.

2.1.3 Temps de communication

Pour concevoir des algorithmes de communications, il convient de définir un modèle théorique prenant en compte le temps de transfert d'un message entre deux nœuds du réseau. Voici quelques modèles plus ou moins simples pouvant prendre en compte la longueur du message, la longueur des liens ou encore le temps d'initialisation avant d'envoyer le message.

Modèle temps constant Dans ce modèle, on considère qu'une communication entre deux nœuds voisins prend une unité de temps et ne dépend pas de la longueur du message envoyé.

Modèle temps linéaire Ici le temps de transmission d'un message de longueur L entre deux nœuds voisins est dépendant d'un temps d'initialisation β et d'un temps de propagation $L\gamma$ proportionnel à la taille L du message. On peut définir une variante de ce modèle, où les paramètres β et γ sont différents pour chaque équipement.

Modèles plus sophistiqués. D'autres modèles collant plus à la réalité peuvent être définis, mais plus le modèle est complexe, plus il est difficile à manipuler d'un point de vue analytique. Le lecteur désirant plus de précisions à sujet pourra se référer à [dR94].

Dans cette thèse nous utiliserons le modèle le plus simple, celui en temps constant. Ce choix se justifie par la nature des réseaux étudiés : de grands réseaux (WAN,MAN) haut débits ; en effet, les débits offerts par la fibre optique réduisent l'importance de la longueur des messages envoyés par rapport aux ressources monopolisées et dans la plupart des problèmes que nous étudierons, les messages seront considérés de même taille.

2.1.4 Conditions sur les nœuds

Les différents équipements réseaux ont des fonctionnalités différentes. Une modélisation cohérente doit donc prendre en compte leurs contraintes technologiques. Un point particulier est leur capacité à émettre et transmettre simultanément plusieurs messages sur leurs ports d'entrée et de sortie. Si un nœud ne peut gérer à un

instant donné qu'un seul de ses ports en entrée et en sortie, on parlera de modèle $1 - port$. S'il peut en gérer simultanément k , on parlera de modèle $k - port$. Et enfin s'il est capable de gérer tous ses liens simultanément, on parlera de modèle $\Delta - port$. Nous considérerons ici uniquement le modèle $\Delta - port$.

2.1.5 Réseaux logiques/virtuels

Dans beaucoup de réseaux (en particulier les réseaux ATM), les communications ne sont pas établies directement dans le réseau physique mais dans un réseau logique (ou virtuel) défini au dessus du réseau physique. Les nœuds de ce réseau logique sont généralement un sous-ensemble de ceux du réseau physique, mais les liens logiques correspondent à des chemins dans le réseau physique.

Dans la modélisation, l'association entre graphe représentant le réseau logique et le graphe représentant le réseau physique est défini par une application qu'on appelle *plongement* dont voici une définition pour les graphes non-orientés :

Définition 2.1.1 [dR94] *Soient G et H deux graphes simples non orientés. Un plongement du graphe H dans le graphe G est défini par la donnée d'une application f de l'ensemble des sommets de H dans l'ensemble des sommets de G et d'une application P_f de l'ensemble des arêtes de H dans l'ensemble des chaînes de G .*

Nous ne considérerons que le cas où f est injective pour les sommets, c'est à dire le cas où G a plus de sommets que H et à chaque sommet de G est associé au plus un sommet de H .

2.1.6 Réseaux optiques

En général, les problèmes se posant dans les réseaux WDM tout-optiques consistent à satisfaire simultanément une instance des requêtes de connexions formée de couples ou paires de nœuds. Pour chaque requête, il s'agit d'attribuer un chemin dans le réseau et d'allouer une longueur d'onde (qui sera représenté par une couleur), soit tout le long du chemin, soit sur chaque lien formant le chemin lorsque la conversion en longueurs d'onde dans les commutateurs est possible, de manière à ce que deux chemins distincts n'utilisent pas la même longueur d'onde sur le même lien. L'objectif dans ce cadre est de minimiser l'utilisation des ressources optiques, donc par exemple le nombre total de longueurs d'onde permettant de satisfaire l'instance donnée. Ces problèmes se ramènent généralement à des problèmes de coloration de graphes.

Les premières études dans la littérature ont représenté les réseaux optiques par des graphes non-orientés. Cette modélisation ne correspond pas précisément

à la réalité physique, puisque les communications optiques sont unidirectionnelles (les informations sur une fibre optique ne peuvent circuler que dans un seul sens). Mais généralement lorsque une fibre optique est installée d'un nœud x vers un nœud y , une autre fibre optique est installée du nœud y vers le nœud x . Le réseau optique est donc très souvent modélisé par un (multi-)graphe orienté symétrique où chaque arc correspondra à un lien où un groupe de liens unidirectionnels.

Comme il est précisé dans la remarque 2.1.2, si le réseau optique est modélisé par un graphe orienté symétrique, si les requêtes sont symétriques et si on cherche un routage symétrique, on peut modéliser le réseau par un graphe non-orienté avec la condition supplémentaire que les chemins symétriques doivent utiliser la même longueur d'onde.

2.2 ATM et le positionnement de chemins virtuels (VPL)

Comme nous l'avons vu dans le chapitre 1, dans un réseau ATM, les connexions (les VCs) sont réalisées dans un réseau logique en utilisant des chemins virtuels (les VPs). Les connexions que nous souhaitons établir sont bidirectionnelles et symétriques, lorsqu'on veut établir une connexion entre x et y , on établit deux connexions de type VCs (utilisant de manière symétrique les mêmes VPs), une entre x et y et une entre y et x .

Les deux caractéristiques qui nous intéressent sont :

- **Le nombre de VPs utilisés par chaque connexion.** Chaque fois qu'une connexion passe d'un VP à un autre, le routeur doit effectuer un traitement électronique ; pour limiter le nombre de ces traitements il est donc intéressant de limiter le nombre de VPs (nombre de sauts) qu'utilise chaque connexion.
- **Le nombre de VP qui partagent un même lien physique,** ce nombre étant limité par la capacité des liens physiques.

En fonction des objectifs, on pourra fixer l'un de ces paramètres et chercher à optimiser l'autre. Les deux paramètres ne peuvent être optimisés en même temps car plus on diminue la charge, plus on risque d'augmenter le nombre de VPs par connexion et réciproquement.

Ce type de problème, connu sous le nom de *positionnement de chemin virtuel*

(en anglais : *VPL : Virtual Path Layout*), a été introduit d'un point de vue théorique dans [GZ94a] puis [CGZ94]. Un état de l'art sur ce problème a été écrit par Zaks [Zak97]. Il a été étudié sous diverses formes dans [BCG97], [Cha98], [BCG99], [Mar00], [BG97a], [Sta], [SV96], [GZ94b], [BMPP99], [BMPP98], [EFZ98], [FZ97].

2.2.1 Modélisation

Comme nous l'avons expliqué dans le chapitre 2, un réseau optique est généralement modélisé par un graphe orienté symétrique. Considérant que les réseaux ATM sont souvent basés sur des réseaux optiques, considérant aussi que nous souhaitons établir des requêtes et connexions symétriques et d'après la remarque 2.1.2, nous modéliserons tous nos réseaux par des graphes non-orientés. Le réseau physique sera modélisé un graphe non-orienté $G = (V, E)$ (où V est l'ensemble des sommets et E l'ensemble des arêtes) appelé **graphe physique** et l'ensemble des chemins virtuels par un graphe non-orienté $H = (V, E')$ appelé **graphe virtuel** où chaque arête $e' = \{x, y\}$ de E' correspond aux deux chemins virtuels (x, y) et (y, x) ayant un routage symétrique (ce qui revient à associer à chaque arête de H une chaîne dans G). L'ensemble des routages est modélisé par une fonction de routage $P : E' \rightarrow \mathcal{P}(G)$ où $\mathcal{P}(G)$ est l'ensemble des chaînes élémentaires (chaînes sans boucle) de G . On se restreint aux chaînes élémentaires car une boucle dans une chaîne ne ferait qu'augmenter la charge sur les arêtes qu'elle utilise et n'apporterait rien aux communications. Une instance de communication sera la donnée d'un ensemble de requêtes correspondant à des paires de sommets.

Définitions

Nous donnons maintenant la définition d'un **positionnement de chemins virtuels** abrégé en **VPL** (*Virtual Path Layout*) qui permet de caractériser la manière dont est plongé le réseau virtuel dans le réseau physique, c'est à dire la manière dont les liens virtuels sont routés dans le réseau physique :

Définition 2.2.1 *Soit $G = (V, E)$ un graphe non-orienté, $H = (V, E')$ un graphe non-orienté et $P : E' \rightarrow \mathcal{P}(G)$ une fonction de l'ensemble des arêtes E' de H dans l'ensemble des chaînes élémentaires de G , le couple (H, P) est appelé **VPL** sur G .*

REMARQUE. L'application P d'un VPL (H, P) sur le graphe G définit un plongement de H dans G .

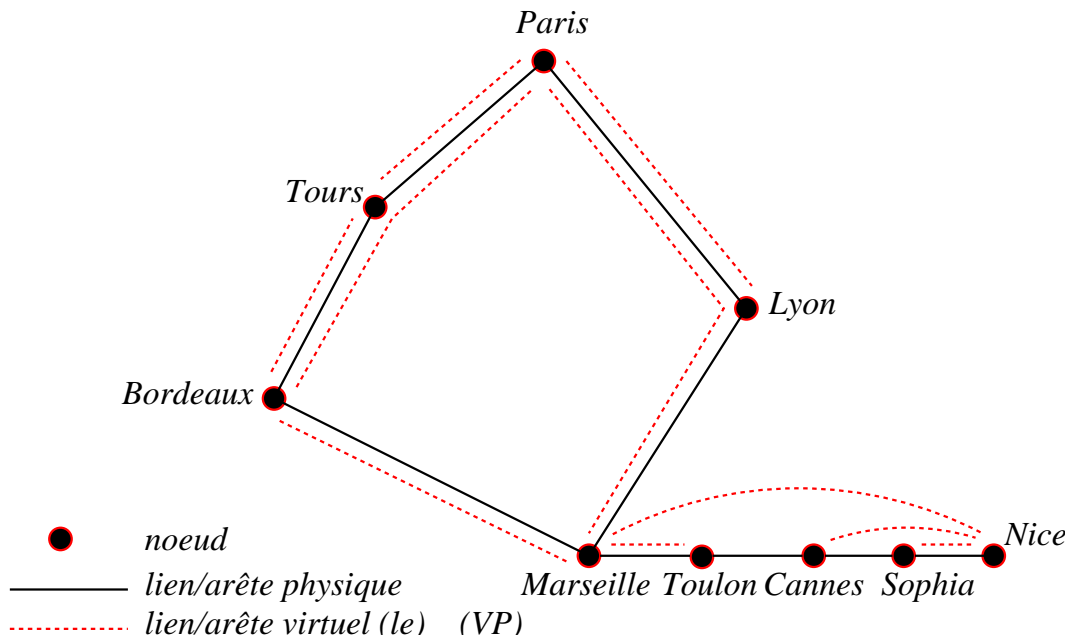


FIG. 2.2 – Un exemple de réseau virtuel

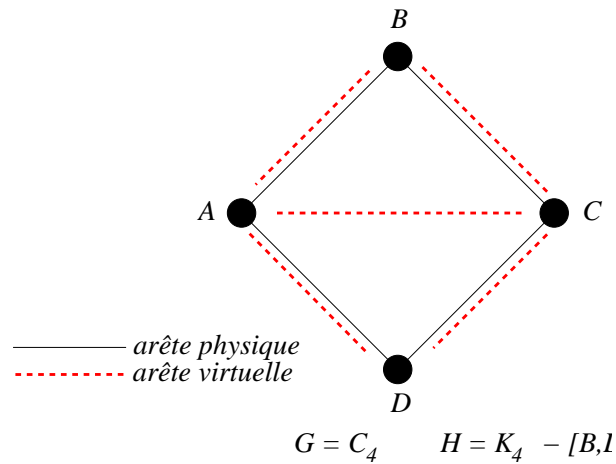


FIG. 2.3 – Un autre exemple de réseau virtuel

Dans les exemples des figures 2.2 et 2.3, les arêtes des graphes physiques sont représentées en noir, les arêtes des graphes virtuels en pointillés rouges et la chaîne associée à une arête d'un graphe virtuel est décrite par sa représentation graphique lorsqu'il n'y a pas d'ambiguïté (pour la figure 2.3, on peut associer l'arête $[A, C]$ soit à la chaîne $([A, B], [B, C])$ soit à la chaîne $([A, D], [D, C])$).

REMARQUE. Si le graphe physique G est un arbre, le plongement d'un graphe virtuel (i.e. : l'application P) est unique car dans un arbre il n'existe qu'une seule chaîne élémentaire entre deux sommets distincts. Dans ce cas, le graphe virtuel suffit à définir un VPL.

Définition 2.2.2 *Étant donné H un graphe et une requête entre les nœuds x et y , on appelle **nombre de sauts** entre x et y (hop count en anglais), noté $h(x, y)$, la distance entre ces nœuds dans le graphe H .*

$$h(x, y) = \text{distance}_H(x, y)$$

Dans l'exemple de la figure 2.2, le nombre de sauts entre *Paris* et *Toulon* est 2, les arêtes virtuelles utilisées étant $[Paris, Marseille]$ et $[Marseille, Toulon]$.

Définition 2.2.3 *Étant donné un graphe virtuel H et une instance communication I , on note $D(H, I)$ le **nombre de sauts maximal du graphe H pour l'instance I** la distance maximale dans le graphe virtuel entre deux sommets d'une requête de connexion de I .*

$$D(H, I) = \max\{h(x, y) \mid [x, y] \in I\}$$

Proposition 2.2.1 *Instance particulière :*

- Dans le cas où l'instance de communication I est un échange total (noté AA), le nombre de sauts maximal correspond au diamètre du graphe virtuel.
- Dans le cas où I est une diffusion (notée OA), le nombre de sauts maximal correspond à l'excentricité dans H de l'initiateur de cette diffusion.

Dans l'exemple de la figure 2.2, si l'instance de communication est la diffusion depuis *Paris*, le nombre de sauts maximal est 3 qui est atteint pour les requêtes $[Paris-Cannes]$ et $[Paris-Sophia]$. Dans l'exemple de la figure 2.3, si l'instance de communication est la diffusion depuis A , le nombre maximal de sauts est 1, mais si c'est l'échange total, le nombre de sauts maximal est 2 (atteint pour la requête $[B, D]$).

Définition 2.2.4 *La charge d'une arête e du graphe physique $G = (V, E)$ pour le VPL $(H = (V, E'), P)$, notée $\pi(G, H, P, e)$, correspond au nombre de chaînes du VPL utilisant cette arête e .*

$$\pi(G, H, P, e) = |\{e' \in E' \mid e \in P(e')\}|$$

REMARQUE. Le notion de charge d'un arête physique ne concerne pas le trafic observé sur le lien physique correspondant, mais uniquement le nombre de liens virtuels qui utilisent ce lien physique. Cette définition est celle utilisée dans la

littérature, il aurait peut être été plus judicieux de la qualifier de *charge virtuelle*.

Dans l'exemple de la figure 2.2, la charge de l'arête physique $[Paris, Lyon]$ est 2 car il y a deux arêtes virtuelles ($[Paris, Lyon]$ et $[Paris, Marseille]$) dont la chaîne associée passe par cette arête physique. Dans l'exemple de la figure 2.3, la charge de l'arête $[A, B]$ est 2 si $[A, C]$ est associée à la chaîne ($[A, B], [B, C]$) et 1 sinon.

Définition 2.2.5 *On appelle charge du graphe $G = (V, E)$ pour le VPL (H, P) sur G , la charge maximale sur l'ensemble des arêtes de G .*

$$\pi(G, H, P) = \max_{e \in E} \{\pi(G, H, P, e)\}$$

Définition 2.2.6 *On appelle charge du graphe G pour le graphe H la charge minimum du graphe G sur tous les plongements de H dans G (on ne considère ici que les plongements de la définition 2.2.1).*

$$\pi(G, H) = \min_P \{\pi(G, H, P)\}$$

Dans l'exemple de la figure 2.2, la charge du graphe physique pour le VPL est 3, elle est atteinte sur l'arête physique $[Sophia, Nice]$. La charge du graphe physique pour le graphe virtuel est aussi 3 car les arêtes virtuelles incidentes à *Nice* qui induisent la charge maximale ne peuvent pas être routées autrement. Dans l'exemple de la figure 2.3, on a $\pi(C_4, H) = 2$.

Cas particulier où $D(H, I) = 1$. Dans ce cas, le graphe virtuel H est le même que le graphe des requêtes (ce qui revient à réaliser directement I sur le graphe physique G), on retrouve la notion classique de charge d'un réseau pour une instance de communication I (notée $\pi(G, I)$) et on a dans ce cas $\pi(G, I) = \pi(G, H)$. Le paramètre $\pi(G, I)$ a été étudié par de nombreux auteurs, en particulier pour des instances $I = AA$ ou $I = OA$ ou pour des réseaux particuliers.

Cas orienté

La formulation du modèle orienté est très similaire à celle du modèle non-orienté. G et H sont des graphes orientés, l'instance de communication I est formée de couples de sommets et P est une application de E' l'ensemble de arcs de H dans l'ensemble des chemins élémentaire de G . Le nombre de sauts de la requête (x, y) de x à y est la distance "orientée" $\vec{h}(x, y)$ dans H de x à y . On note, pour une instance I , $\vec{D}(H, I) = \max \left\{ \vec{h}(x, y), (x, y) \in I \right\}$. De même on définit la charge $\vec{\pi}(G, H, P, e)$ d'un arc e pour un VPL (H, P) , $\vec{\pi}(G, H, P)$ la charge de G pour le VPL (H, P) et $\vec{\pi}(G, H)$ la charge de G pour H .

REMARQUE. Si G^* (resp. H^*) est le graphe orienté symétrique issu de G (resp. H) (voir def A.1), on a $\vec{\pi}(G^*, H^*) \leq \pi(G, H)$. En effet, $\pi(G, H)$ peut être vu comme le minimum des $\pi(G^*, H^*, P)$ restreint aux routages P symétriques. Cette inégalité peut être stricte, comme le montre la figure 2.5.

2.2.2 Indice de transmission

La notion d'*indice de transmission* d'un graphe a été introduite par Chung, Coffman, Reiman et Simon en 1987 [CCRS87]. Outre les nombreux problèmes théoriques soulevés par son étude, ce paramètre peut servir à juger des performances de routage d'un réseau d'interconnexion. Deux types d'indice de transmission sont distingués : le *sommet-indice de transmission* (introduit dans [CCRS87]) et l'*arête-indice de transmission* (introduit par Heydemann, Meyer et Sotteau en 1989 dans [HMS89]). Nous ne présenterons que l'arête-indice de transmission car c'est celui qui se rapproche le plus du problème que nous étudions.

La définition donnée dans [HMS89] de l'arête-indice de transmission (et reprise dans les articles suivants) apparaît purement théorique et n'est reliée à aucun problème appliqué. En effet, les auteurs considèrent un graphe physique non-orienté mais une instance de communication formée de tous les couples de sommets et à chaque requête (x, y) (de x à y) est associée une chaîne dans G . Les requêtes (x, y) et (y, x) ne sont donc pas routées par la même chaîne. L'ensemble des chaînes associées aux requêtes forme un routage.

Définition 2.2.7 *La charge d'une arête e d'un graphe G pour un routage R de toutes les requêtes (x, y) , notée $\pi(G, R, e)$, est le nombre de chaînes de R passant par e .*

L'arête-indice de transmission d'un graphe G muni du routage R (de toutes les requêtes (x, y)), noté $\pi(G, R)$, est le maximum de l'ensemble des charges de toutes les arêtes.

$$\pi(G, R) = \max_{e \in E} \pi(G, R, e)$$

L'arête-indice de transmission d'un graphe G (edge-forwarding index en anglais) est l'arête-indice de transmission minimum parmi tous les routages possibles de l'ensemble des requêtes de la forme (x, y) dans le graphe G .

$$\pi(G) = \min_R \pi(G, R)$$

REMARQUE. Si le routage R est symétrique, c'est à dire qu'aux requêtes (x, y) et (y, x) sont associées deux chaînes identiques, on a

$$\min_{R \text{ routage symétrique}} \pi(G, R) = 2\pi(G, K_n)$$

Donc $\pi(G) \leq 2\pi(G, K_n)$. Clairement on a $\pi(G, K_n) \leq \pi(G)$.

On peut aussi considérer un routage de tous les couples ($H = K_n^*$ est le graphe complet orienté symétrique) sur le graphe physique orienté symétrique G^* . La charge de l'arête $e = [x, y]$, $\vec{\pi}(G, R, e)$ est alors égale à la somme des charges des arcs (x, y) et (y, x) et donc $\pi(G) \leq 2\vec{\pi}(G^*, K_n^*)$.

En résumé on a :

$$\vec{\pi}(G^*, K_n^*) \leq \pi(G, K_n) \leq \pi(G) \leq 2\vec{\pi}(G^*, K_n^*) \leq 2\pi(G, K_n)$$

Les exemples ci-après montrent qu'il n'y a pas forcément égalité.

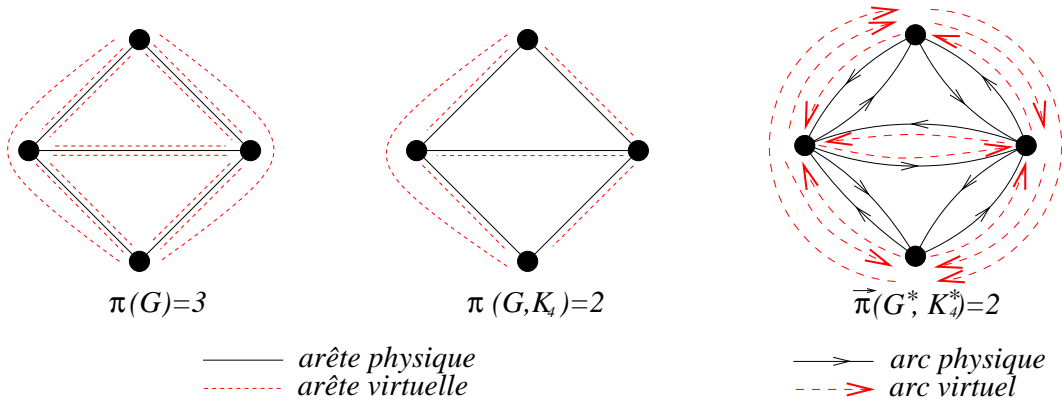


FIG. 2.4 – Il existe G tel que $\pi(G) < 2\vec{\pi}(G^*, K_n^*)$.

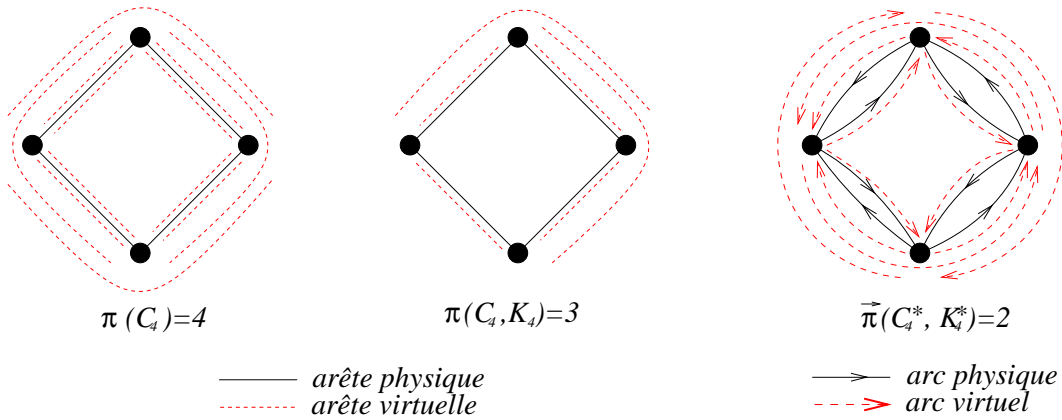


FIG. 2.5 – Il existe G tel que $\vec{\pi}(G^*, K_n^*) < \pi(G, K_n) < \pi(G)$

Le facteur d'étirement

Le **facteur d'étirement** (en anglais : *stretch factor*) d'une connexion est le rapport entre la longueur dans le graphe physique des chaînes empruntées par la connexion entre deux sommets dans le réseau virtuel et la distance dans le réseau physique entre ces deux sommets. Certains travaux (comme par exemple [CGZ94], [FNP97] ou [GWZ98]) s'imposent la contrainte de construire des VPL dont le facteur d'étirement vaut 1. Fixer le facteur d'étirement à 1 impose que les connexions se fassent le long des plus courts chemins dans le graphe physique.

Cette contrainte supplémentaire permet de simplifier le problème en imposant la forme des connexions. Un problème \mathcal{NP} -complet peut devenir polynomial lorsqu'on impose un facteur d'étirement borné. De plus le type de réseaux physiques utilisés (principalement constitués de fibre optique) permettent d'établir des connexions de longueur importante sans que cela ne pose de problèmes, c'est pourquoi nous ne considérons pas cette notion dans les travaux présentés.

Dans ce modèle on suppose que les arêtes du graphe virtuel ont une capacité infinie (considérant que la limite du réseau n'est pas la taille de la bande passante allouée aux VPs mais la topologie du réseau des VPs et le nombre de VPs à gérer par lien physique). On pourra donc faire passer un nombre illimité de connexion VCs par le même VP. Contrairement au cas du facteur d'étirement fixé, ce qui est important dans ce modèle c'est la topologie du graphe virtuel et non le routage dans ce graphe virtuel.

2.2.3 Formulation des problèmes et leur complexité

Nous reprenons ici les formulations données dans [Mar00].

Définition 2.2.8 *Étant donné un graphe G , une instance de communications I , un nombre de sauts maximal h , une charge maximale c , un VPL (H, P) sur G est dit (h, c) -réalisable si $D(H, I) \leq h$ et $\pi(G, H, P) \leq c$.*

PROBLÈME A, de l'existence d'un VPL

Données : un graphe G , une instance de communications I dans G , h et c deux entiers.

Question : existe-t-il un VPL (h, c) -réalisable sur le graphe G avec l'instance de communications I ?

Cette formulation induit deux problèmes d'optimisation dits *duaux* :

PROBLÈME B, du nombre de sauts minimal : $D(G, I, c_0)$

Entrée : un graphe G , une instance I dans G et un entier c_0 .

Sortie : un VPL (h, c_0) -réalisable sur le graphe G avec l'instance I .

Objectif : minimiser h .

On note $D(G, I, c_0)$ la valeur optimale du problème B pour le graphe G , l'instance I et la charge c_0 .

PROBLÈME C, de la charge minimale : $\pi(G, I, h_0)$

Entrée : un graphe G , une instance I dans G et un entier h_0 .

Sortie : un VPL (h_0, c) -réalisable sur le graphe G avec l'instance I .

Objectif : minimiser c .

On note $\pi(G, I, h_0)$ la valeur optimale du problème C pour le graphe G , l'instance I et le nombre de sauts h_0 .

REMARQUE. Si $h_0 = 1$ on a $\pi(G, I, 1) = \pi(G, I)$.

PROBLÈME D, de l'existence d'un VPL avec un facteur d'étirement égal à 1

Données : un graphe G , une instance de communications I dans G et h et c deux entiers.

Question : existe-t-il un VPL (h, c) -réalisable sur le graphe G avec l'instance de communications I et de facteur d'étirement égal à 1 ?

Le problème A est \mathcal{NP} -complet en général, et même dans le cas $h = 1$ où il est équivalent au problème du multi-flot entier [GJ79].

Il est également \mathcal{NP} -complet lorsque l'instance de communication est de type *One – to – Many* [GZ94a], il peut se réduire dans ce cas au problème des k chemins disjoints (ND40) [GJ79].

Il est également \mathcal{NP} -complet lorsque l'instance de communication est de type *Many – to – Many* [GZ94a].

Le problème A est polynomial si $h = 1$ avec l'instance de communication *One – to – Many* car il se ramène à un problème de simple flot où la source est l'initiateur de la diffusion et le puits un sommet supplémentaire connecté à tous

les sommets de l'ensemble à atteindre par la source. Il suffit ensuite d'affecter comme capacité sur les arêtes la valeur de c et de résoudre ce problème de flot qui est polynomial [JF56].

Si le problème A est polynomial pour un h_0 donné, alors le problème C est lui aussi polynomial pour h_0 en résolvant le problème A plusieurs fois en faisant varier c de manière dichotomique (ce qui fait au plus $\log_2(|V|)$ problèmes A à résoudre si on prend comme bornes sur c : 0 et $|V|$).

Dans [EFZ98], il est montré que le problème D est \mathcal{NP} -complet pour une instance de communication de type diffusion sauf pour les cas $h = 1$ et $(h, c) = (2, 1)$ pour lesquels il est polynomial.

2.2.4 Résultats connus sur les problèmes de VPL

Cas général

Lorsqu'aucune information sur le graphe n'est donnée, il est difficile de répondre aux questions posées par les problèmes A, B et C .

Il existe cependant divers résultats dans la littérature mettant en relation ces paramètres.

Relations entre les paramètres π et D

Théorème 2.2.2 ([Mar00])

$$\pi(G, AA, 2h_0) \leq \pi(G, OA, h_0)$$

$$D(G, AA, c_0) \leq 2D(G, OA, c_0)$$

Théorème 2.2.3 ([Mar00]) *Soit $G = (V, E)$ un graphe et OA une instance du type one-to-all,*

$$\pi(G, OA, 2) \leq 2 \cdot \left\lceil \sqrt{\pi(G, OA, 1)} \right\rceil - 1$$

PREUVE. Ce résultat s'obtient directement en remarquant qu'à partir d'une diffusion depuis une source en h_0 sauts, on peut réaliser un échange total en $2h_0$ sauts en passant par cette source pour réaliser toutes les requêtes. \square

Théorème 2.2.4 ([Mar00]) *Soit $G = (V, E)$ un graphe et OM une instance du type one-to-many,*

$$\pi(G, OM, h) \leq h\pi(G, OM, 1)^{\frac{1}{h}} - h + 1$$

Théorème 2.2.5 ([Mar00]) Soit $G = (V, E)$ un graphe à $n \geq 3$ sommets.

$$\pi(G, AA, 2) \leq \max \left\{ 2\sqrt{6}e \cdot \sqrt{\frac{\ln n}{n}} \cdot \pi(G), \frac{2}{\ln 2} \cdot \ln n \right\}$$

Bornes inférieures

Théorème 2.2.6 ([Mar00]) Soit $G = (V, E)$ un graphe de degré maximal Δ , I une instance de connexion et h un entier non nul, on a :

$$(\Delta \pi(G, I))^{\frac{1}{h}} \leq 2\Delta \pi(G, I, h)$$

Théorème 2.2.7 ([BG97b, BBGG00]) Soient G un graphe non orienté de degré maximal Δ , une coupe-arête C et N_c le nombre de requêtes de connexion de I séparées par C .

$$\left(\frac{\Delta}{4h} \frac{N_c}{|C|} \right)^{\frac{1}{h}} \leq \Delta \pi(G, I, h)$$

Facteur d'étirement fixé

Constructions

Dans [CGZ94], les auteurs donnent un algorithme polynomial permettant de construire un VPL sur un arbre avec un facteur d'étirement égal à 1 et de charge minimum, réalisant une instance de communication de type *One-To-Many* en h sauts. Utilisant ce résultat, ils en déduisent des constructions sur différents type de graphes de la manière suivante :

- trouver un ensemble d'arbres qui couvrent le graphe
- déterminer sur chacun de ces arbres un VPL de charge minimum avec au plus $\lfloor \frac{h}{2} \rfloor$ sauts
- connecter ces arbres de manière à avoir un VPL en au plus h sauts.

Ils obtiennent ainsi les résultats suivants :

Définition 2.2.9 Un graphe est dit K -séparable s'il existe un ensemble de K sommets qui déconnecte le graphe.

Théorème 2.2.8 ([CGZ94]) Si $G = (V, E)$ est K -séparable, MM est une instance de communication de type *Many-to-Many*,

$$\pi_1(G, MM, h) \leq \frac{Kh}{2(1.5^2/h - 1)} n^{2/h}$$

Théorème 2.2.9 ([CGZ94]) Soit MM est une instance de communication de type *Many-to-Many* sur G ,

$$\pi_1(G, MM, h) \leq \sqrt{hn}^{1+1/h}$$

Chapitre 3

Le VPL dans le cycle et la chaîne

Dans tous les problèmes qui suivent, nous considérons que le facteur d'étirement n'est pas borné (autrement dit, nous ne nous occupons pas de ce paramètre).

Etant donné un réseau physique, nous allons chercher un réseau virtuel minimisant la charge avec un nombre de sauts fixé pour une instance de communication donnée.

Définition 3.0.10 *Étant donné un graphe virtuel $H = (V, E')$ sur $G = (V, E)$ et un ensemble de sommets S de G , on dira d'un sommet s de l'ensemble S est **sortant** s'il est adjacent dans H à un sommet de $V \setminus S$. Un ensemble de sommets S est dit **sortant** si tout sommet s de S est sortant.*

3.1 La chaîne (ou chemin)

Le graphe considéré dans cette partie est la chaîne, appelée aussi chemin par abus de langage.



FIG. 3.1 – La chaîne à 6 sommets

3.1.1 La diffusion

Ce problème a été résolu par Zaks et Al dans [GWZ95a]. Nous redonnons ici une démonstration de leur résultat, il sera utilisé afin de construire des solutions quasi-optimales pour le cas de l'échange total dans la chaîne dans la section suivante.

L'instance de communication est ici la diffusion, la contrainte du nombre de sauts maximal correspond donc à l'excentricité de la source de cette diffusion dans le graphe virtuel. Nous considérons que la source est une des extrémités de la chaîne, nous la noterons s .

Définition 3.1.1 Dans la chaîne P_n , on dit que deux arêtes virtuelles $[a, c]$ et $[b, d]$ se croisent si a ou (exclusif) c se trouve (strictement) entre b et d .

Lemme 3.1.1 (sans croisement) Étant donné un H un graphe virtuel tel que s est d'excentricité au plus h , si deux arêtes se croisent, on peut toujours trouver un autre graphe virtuel H' avec s d'excentricité au plus h tel que $\pi(G, H') \leq \pi(G, H)$ et les arêtes ne se croisent pas.

PREUVE. Supposons que deux arêtes se croisent comme dans la figure 3.2.

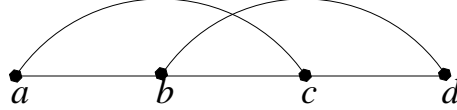


FIG. 3.2 – Deux arêtes se croisent.

- Si $d(s, b) = d(s, c)$, la construction 3.3 ne change ni la charge ni l'excentricité de s .



FIG. 3.3 – Si $d(s, b) = d(s, c)$.

- Si $d(s, b) < d(s, c)$, la construction 3.4 ne change ni la charge ni l'excentricité de s .

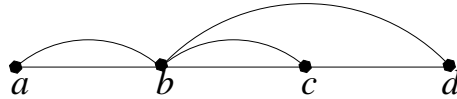


FIG. 3.4 – Si $d(s, b) < d(s, c)$.

- Si $d(s, b) > d(s, c)$, on utilise la construction 3.4 en permutant b et c .

□

Définition 3.1.2 Nous notons $n(h, c)$ le nombre maximum de sommets d'une chaîne P tel que $\pi(P, OA, h) \leq c$ lorsque la source est une extrémité de la chaîne.

Proposition 3.1.2 (d'après [FZ97, GWZ95b])

$$n(h, c) = n(h, c - 1) + n(h - 1, c - 1) + n(h - 1, c) - 1$$

PREUVE. La figure 3.5 illustre la preuve. Considérons un graphe virtuel à $n(h, c)$ sommets. Soit v le voisin de s dans le graphe virtuel qui est le plus éloigné de s dans la chaîne, alors, comme les arêtes virtuelles ne se croisent pas d'après 3.1.1, le nombre maximum de sommet qui peuvent être :

- entre s et v du côté de s (i.e. : tel que le chemin pour aller en s n'utilise pas l'arête $[s, v]$) est $n(h, c - 1)$, en effet, la distance à s est h et la charge à ne pas dépasser est $c - 1$ car l'arête virtuelle entre s et v contribue pour 1 sur la charge de toutes les arêtes physiques entre s et v .
- entre s et v du côté de v est $n(h - 1, c - 1)$, en effet, contrairement au cas précédent, les sommets de cette partie devront être à distance au plus $h - 1$ de v pour être à distance au plus h de s .
- en dehors de la partie entre s et v est $n(h - 1, c)$, car les sommets doivent être à distance au plus $h - 1$ de v et la charge à ne pas dépasser est c .

Le -1 permet de ne pas compter deux fois le sommet v .

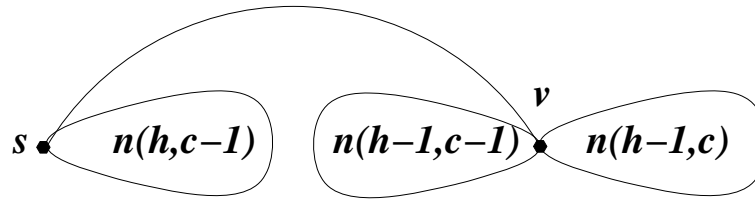


FIG. 3.5 – Récurrence sur le nombre de sommets.

□

REMARQUE. Dans le cas où le facteur d'étirement vaut 1, la longueur des chaînes associées aux arêtes virtuelles entre deux sommets ne devant pas dépasser la distance entre ces sommets dans le graphe physique, la récurrence devient : $n'(h, c) = n'(h, c - 1) + n'(h - 1, c)$.

Les valeurs initiales de $n(h, c)$ sont $n(0, c) = 1$ et $n(h, 0) = 1$. Le calcul de

$n(h, c)$ peut alors être fait à l'aide de la série génératrice associée :

$$\begin{aligned}
\mathcal{N}_h(z) &= \sum_{c \geq 0} n(h, c) z^c \\
&= 1 + \sum_{c \geq 1} [n(h, c-1) + n(h-1, c-1) + n(h-1, c) - 1] z^c \\
&= z\mathcal{N}_h(z) + (1+z)\mathcal{N}_{h-1}(z) - \frac{1}{1-z} + 1 \\
&= \frac{1+z}{1-z} \mathcal{N}_{h-1}(z) - \frac{z}{(1-z)^2} \\
&= \left(\frac{1+z}{1-z} \right)^h \mathcal{N}_0(z) - \frac{z}{(1-z)^2} \sum_{i=0}^{h-1} \left(\frac{1+z}{1-z} \right)^i \\
&= \frac{1}{2(1-z)} \left[1 + \left(\frac{1+z}{1-z} \right)^h \right]
\end{aligned}$$

Notons que le terme d'ordre i de la série génératrice $\frac{1}{1-z} \left(\frac{1+z}{1-z} \right)^h$ est le nombre de points à coordonnées entières dans une sphère de rayon i (pour la distance l_1) et de dimension h :

$$\frac{1}{1-z} \left(\frac{1+z}{1-z} \right)^h = \sum_{c \geq 0} ball(h, c) z^c$$

La fonction $ball(h, c)$ a aussi été étudiée dans ce cadre par Dinitz, Feigelson et Zaks dans [DFZ97a]). On peut exprimer le terme générique de cette série puisque

$$ball(h, c) = \sum_{i=0}^{\min(c, h)} 2^i \binom{h}{i} \binom{\pi}{i} = \frac{2^h c^h}{h!} + o(c^h)$$

Nous avons donc montré que $n(h, c) = \frac{1}{2} ball(h, c) + \frac{1}{2} = \frac{1}{2} \frac{2^h c^h}{h!} + o(c^h)$, et pouvons en déduire le corollaire suivant :

Corollaire 3.1.3

$$\pi(P_n, OA, h) = \frac{(2h!)^{1/h}}{2} n^{1/h} + o(n^{1/h})$$

si OA est l'instance de diffusion avec la source correspondant à une extrémité de la chaîne.

La figure 3.6 donne un exemple de graphe virtuel établi sur la chaîne à 20 sommets.

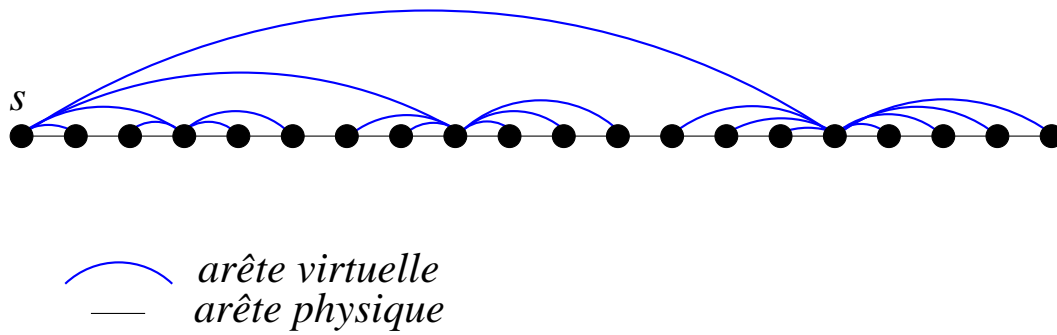


FIG. 3.6 – Graphe virtuel optimal sur la chaîne pour la diffusion, le nombre maximal de sauts est 2, la charge maximale est 4.

3.1.2 L'échange total

Nous considérons maintenant l'instance de communication d'échange total : tout nœud du réseau doit envoyer un message aux autres nœuds, le nombre de sauts correspond alors au diamètre du graphe virtuel. Nous allons déterminer la charge optimale en h sauts $\pi(P_n, AA, h)$ pour cette instance AA d'échange total dans la chaîne P_n à n sommets.

Proposition 3.1.4

$$\pi(P_n, AA, 1) = \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil$$

PREUVE. Si on note $V = \{0, 1, \dots, n-1\}$ les sommets de la chaîne P_n , la charge induite par le graphe virtuel complet sur l'arête physique $(i-1, i)$ est $i(n-i)$ pour $1 \leq i \leq n-1$ (illustration par la figure 3.7.)

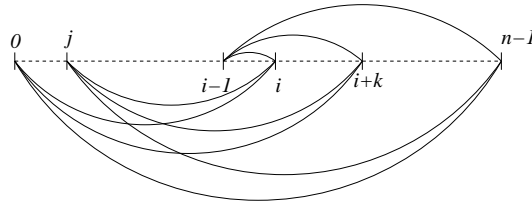


FIG. 3.7 – La charge sur l'arête $(i-1, i)$ est $i(n-i)$.

On a donc $\pi(P_n, AA, 1) = \max_{1 \leq i \leq n-1} i(n-i) = \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil$ □

Proposition 3.1.5

$$\pi(P_n, AA, 2) = \left\lfloor \frac{n}{2} \right\rfloor$$

PREUVE. Soit $I_i = \{0, 1, \dots, i-1\}$ une partie de V . Si I_i est non sortant, il existe un sommet de I_i qui ne sort pas et comme le diamètre est 2, $\overline{I_i}$ (le complémentaire de I_i est sortant. Donc la charge sur l'arête $(i-1, i)$ est supérieure à $n-i$. Si au contraire I_i est sortant, la charge sur $(i-1, i)$ est supérieure à i . La charge est donc supérieure à $\min\{i, n-i\} = \lfloor \frac{n}{2} \rfloor$. La construction s'obtient en reliant le sommet $\lfloor \frac{n}{2} \rfloor$ à tous les autres. On a donc $\pi(P_n, AA, 2) = \lfloor \frac{n}{2} \rfloor$. \square

La figure 3.8 nous montre un exemple de graphe virtuel de diamètre 2.

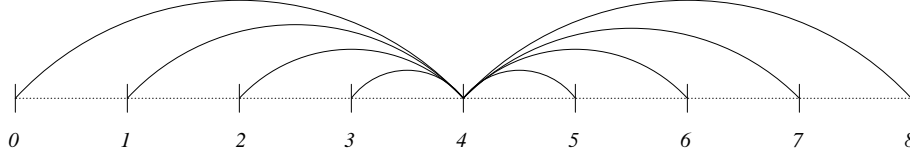


FIG. 3.8 – Exemple de graphe virtuel pour $h = 2$ et $n = 9$, la charge est 4.

Théorème 3.1.6

$$\pi(P_n, AA, h) = \begin{cases} \frac{(p!)^{2/h}}{2} n^{2/h} + o(n^{2/h}) & \text{pour } h = 2p \\ \frac{(hp!)^{2/h}}{2^{1+1/h}} n^{2/h} + o(n^{2/h}) & \text{pour } h = 2p + 1 \end{cases}$$

PREUVE. L'idée principale est d'établir une construction hiérarchique constituée d'un ensemble de k sommets reliés par un graphe virtuel correspondant à la solution du même problème avec k sommets pour un nombre de sauts maximal de $h-2$ et de relier les sommets restants à l'un de ces k sommets; on obtient ainsi un graphe virtuel de diamètre h . On peut montrer que cette construction est optimale. Ces résultats ont fait l'objet d'une publication [Cho01].

Au lieu de calculer la charge $\pi(P_n, h)$, nous allons procéder comme pour la diffusion (voir section 3.1.1) et calculer le nombre maximum de sommets de la chaîne tel qu'il existe un graphe virtuel H de diamètre inférieur ou égal à h avec $\pi(P_n, H) \leq \pi$, où π est la charge maximale donnée; nous notons ce paramètre $n_{AA}(\pi, h)$.

$$n_{AA}(\pi, h) = \max \{n \mid \pi(P_n, AA, h) \leq \pi\}$$

REMARQUE. π est alors une capacité car c'est la limite de charge qu'on ne doit pas dépasser.

Borne supérieure sur $n_{AA}(\pi, h)$

Soit H un graphe virtuel de charge minimale sur P_n , on note $\{a_1, a_2, \dots, a_k\}$ la suite de sommets de P_n telle que

$$- a_1 < a_2 < \dots < a_k$$

- $\forall 0 \leq i \leq k : [a_i + 1, a_{i+1} - 1]$ est sortant (tous les sommets ont un voisin en dehors).
 - $\forall 0 \leq i \leq k - 1$, il existe un sommet non-sortant dans $[a_i + 1, a_{i+1}]$
 - $[a_k + 1, n]$ est sortant
- (avec $a_0 = -1$ et $a_{k+1} = n$). REMARQUE. Cette suite est unique car est

construite directement ainsi : on commence avec un intervalle réduit au premier sommet de la chaîne et tant que cet intervalle est sortant on y ajoute le sommet suivant. On obtient ainsi le premier intervalle (qui est alors non sortant) et on construit l'intervalle suivant en partant du prochain sommet. Quand on arrive à la fin de la chaîne, on obtient un dernier intervalle où tous les sommets sont sortants.

Pour $0 \leq i \leq k - 1$, on note J_i (appelé *bloc de niveau 1* de H) l'intervalle $[a_i + 1, a_{i+1}]$ et I_i l'intervalle $[a_i + 1, a_{i+1} - 1]$. La propriété importante de blocs étant qu'ils contiennent au moins un sommet non-sortant. On note J_k l'intervalle $[a_k + 1, a_{k+1}]$, cet intervalle ne contient pas nécessairement de sommet non-sortant et ne sera donc pas appelé bloc.

Le graphe H_1 des blocs de niveau 1 (le graphe quotient des blocs) de H peut être considéré comme un graphe virtuel sur la chaîne à k sommets. Le diamètre de H_1 est au plus $h - 2$ car chaque bloc de niveau 1 a au moins un sommet non-sortant et le diamètre de H est au plus h (illustration par la figure 3.9).

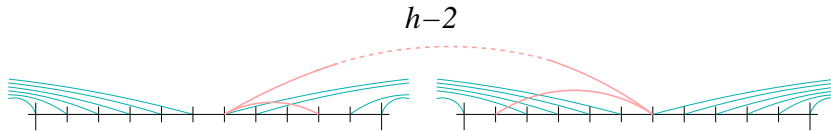
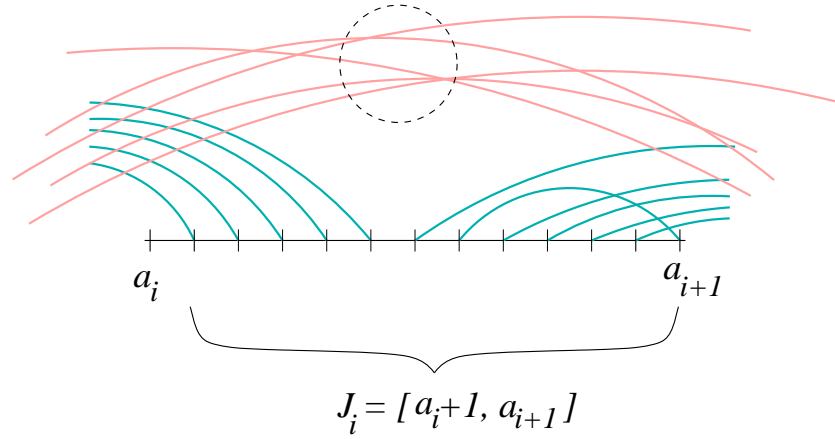


FIG. 3.9 – La distance entre deux blocs de niveau 1 est au plus $h - 2$.

De même on définit des blocs de niveau l correspondant aux blocs de niveau 1 du graphe H_{l-1} . Ainsi le diamètre de H_2 est au plus $h - 4$ et plus généralement, pour $i \leq \frac{h}{2} - 1$, le diamètre de H_i est au plus $h - 2i$.

On appelle *capacité disponible* d'un bloc J la capacité totale π moins la charge des arêtes virtuelles entre les blocs différents de J .

Le nombre de sommets d'un bloc de capacité disponible p est au plus $2p$ car par définition un bloc est sortant, chaque sommet induit donc une charge d'au moins 1 sur l'une des deux arêtes physique délimitant le bloc.



NOTATION. On note $B(l, c)$ le nombre maximum de sommets dans un bloc de niveau l avec une capacité disponible c .

Un bloc de niveau l de capacité disponible c est constitué d'un certain nombre de blocs de niveau $l - 1$ et tous (sauf peut-être un) induisent une arête qui sort du bloc (voir la figure 3.10). Ainsi, ce bloc de niveau l est constitué d'au plus $2c$ blocs de niveau $l - 1$ (2 blocs de capacité disponible $c - i$ pour $0 \leq i \leq c - 1$) auxquels on doit adjoindre un bloc de niveau $l - 1$ (qui lui n'est pas sortant).

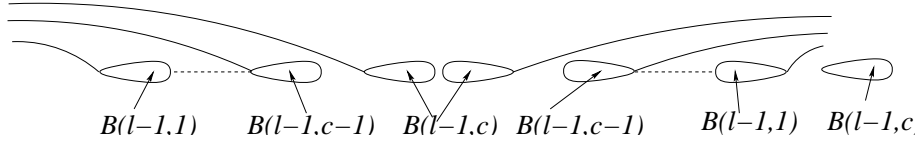


FIG. 3.10 – Un bloc extrême de niveau l et de charge disponible c .

Si on note $\bar{B}(l, c)$ le majorant de $B(l, c)$ correspondant à cette estimation, on a :

$$B(l, c) \leq \bar{B}(l, c) = 2 [\bar{B}(l - 1, 1) + \bar{B}(l - 1, 2) + \dots + \bar{B}(l - 1, c)] + \bar{B}(l - 1, c)$$

avec $\bar{B}(0, c) = 1$. La valeur de $\bar{B}(l, c)$ peut être calculée à l'aide de la série génératrice associée $\mathcal{B}_l(z) = \sum_{c \geq 0} \bar{B}(l, c) z^c = \frac{1-z}{(1-z)^2}$, et on obtient

$$B(l, c) \leq \frac{2^l c^l}{l!} + o(c^l)$$

- Cas $h = 2p + 1$ impair

Soit H un graphe virtuel de diamètre $h = 2p + 1$ tel que $\pi(P_n, H) = \pi$. Le graphe H_p (le graphe quotient des blocs de niveau p) est un graphe complet, notons k le nombre de sommets de ce graphe. Soient $\{J_i\}_{0 \leq i \leq k-1}$

l'ensemble des blocs de niveau p , ils induisent une charge de $i(k - i - 1)$ ($k - i - 1$ car nous ne comptons pas la charge induite par J_i) sur toutes les arêtes physiques de J_i (voir la figure 3.7 du calcul de $\pi(P_n, AA, 1)$). On a donc $|J_i| \leq B(p, \pi - i(k - i - 1))$ pour $i \leq k - 1$ et $|J_k| \leq B(p, \pi)$, donc :

$$n_{AA}(\pi, 2p + 1) \leq B(p, \pi) + \sum_{i=0}^{k-1} B(p, \pi - i(k - i - 1)), \text{ soit :}$$

$$n_{AA}(\pi, 2p + 1) \leq B(p, \pi) + \max_{k \leq 2\sqrt{\pi}+1} \left\{ \sum_{i=0}^{k-1} B(p, \pi - i(k - i - 1)) \right\}$$

- Cas $h = 2p$ pair

Le principe est le même, la différence est que les blocs de niveau $p - 1$ forment un graphe de diamètre au plus 2. On obtient alors :

$$n_{AA}(\pi, 2p) \leq B(p - 1, \pi) + \max_{k \leq 2\pi} \left\{ 2 \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} B(p - 1, \pi - i) \right\}$$

Borne inférieure sur $n_{AA}(\pi, h)$

Pour construire une solution, nous allons déterminer des sommets particuliers qui seront reliés par un graphe complet si h est impair et par un graphe de diamètre 2 si h est pair. Ensuite nous enracinons des arbres d'excentricité p en ces sommets (les mêmes que ceux décrits par Gerstel, Wool et Zaks dans [GWZ95a] pour réaliser la diffusion, voir section 3.1.1), la taille des arbres sera conditionnée par la charge induite par le graphe reliant les sommets particuliers pour utiliser au mieux la capacité disponible. Si on a k sommets particuliers, on aura :

- cas $h = 2p + 1$ impair : $\max_{k \leq 2\sqrt{\pi}} \left\{ \sum_{i=0}^{k-1} ball(p, \pi - i(k - i)) \right\}$ sommets au total
- cas $h = 2p$ pair : $\max_{k \leq 2\pi} \left\{ 2 \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} ball(p - 1, \pi - i) \right\}$ sommets au total,

où $ball(p, c) = \sum_{i=0}^{\min\{p, c\}} 2^i \binom{l}{i} \binom{r}{i}$ est, comme dans la section 3.1.1, le nombre de points entiers dans la sphère de dimension p et de rayon c avec la norme l_1 .

$$ball(p, c) = \frac{2^p c^p}{p!} + o(c^p) = B(p, c) + o(c^p)$$

Cette construction a donc $n_{AA}(\pi, h) + o(\pi^p)$ sommets.

Calcul de $n_{AA}(\pi, h)$

- cas $h = 2p + 1$ impair :

$$n_{AA}(\pi, 2p + 1) = \frac{2^p}{p!} \max_{k \leq 2\sqrt{\pi}} \left\{ \sum_{i=0}^k (\pi - i(k - i))^p \right\} + o(\pi^p)$$

$\sum_{i=0}^k (\pi - i(k - i))^p = \sum_{j=0}^p \binom{p}{j} \pi^{p-j} (-1)^j \sum_{i=0}^k i^j (k - i)^j$ et $\sum_{i=0}^k i^j (k - i)^j = k^{2j+1} \sum_{i=0}^j \binom{j}{i} \frac{(-1)^i}{j+i+1} + o(k^{2j+1}) = k^{2j+1} \frac{(j!)^2}{(2j+1)!} + o(k^{2j+1})$. Le maximum étant atteint pour $k = 2\sqrt{\pi}$, on a donc :

$$n_{AA}(\pi, 2p + 1) = \frac{2^{p+1}}{(2p + 1)p!} \pi^{p+1/2} + o(\pi^{p+1/2})$$

- cas $h = 2p$ pair :

$$n_{AA}(\pi, 2p) = \max_{k \leq 2\pi} \left\{ 2 \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \frac{2^{p-1} (\pi - i)^{p-1}}{(p-1)!} \right\} = \frac{2^p \pi^p}{p!} + o(\pi^p)$$

Il ne reste plus qu'à renverser les équations pour obtenir une estimation de $\pi(P_n, AA, h)$. \square

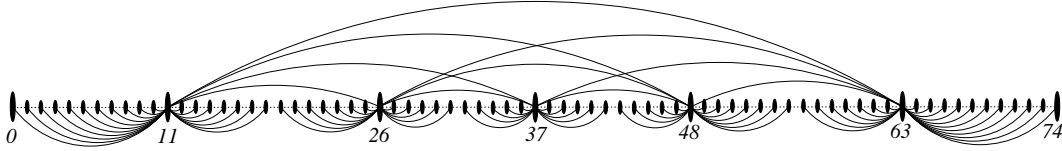
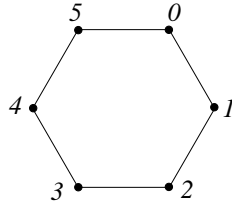


FIG. 3.11 – Un graphe virtuel optimal pour l'instance d'échange total sur la chaîne à 75 sommets avec $h = 3$, $\pi(P_{75}, AA, 3) = 11$.

3.2 Le cycle

Contrairement aux arbres, dans un cycle il n'existe pas un seul chemin élémentaire entre deux sommets mais deux. Lorsque on établit une arête virtuelle entre deux sommets, il faut donc aussi donner son routage dans le graphe physique qu'est le cycle.



On note $V = \{0, 1, \dots, n-1\}$ les sommets du cycle C_n . Rappelons que $\pi(C_n, AA, h_0)$ est la charge minimum sur tous les graphes virtuels établis sur C_n , de diamètre $D \leq h_0$.

FIG. 3.12 – Le cycle à 6 sommets

Théorème 3.2.1

$$\pi(C_n, AA, 1) = \begin{cases} \left\lceil \frac{n^2+4}{8} \right\rceil & \text{si } n \text{ est pair} \\ \frac{n^2-1}{8} & \text{si } n \text{ est impair} \end{cases}$$

PREUVE. Le nombre maximal de sauts est 1, donc le graphe virtuel H est le graphe complet, il n'y a qu'à déterminer le plongement de charge minimum du graphe complet dans le cycle.

Pour calculer la charge minimum induite par H , nous allons utiliser un critère de coupe : si on partitionne l'ensemble des sommets en deux ensembles (connexes dans le cycle) de cardinal $\lfloor \frac{n}{2} \rfloor$ et $\lceil \frac{n}{2} \rceil$ comme l'illustre la figure 3.13, on obtient deux arêtes de coupes dont la somme des charges est au moins le produit des deux cardinaux : $\lfloor \frac{n}{2} \rfloor \lceil \frac{n}{2} \rceil$. On en déduit donc que la charge de l'une des deux arêtes est au moins $\left\lceil \frac{\lfloor \frac{n}{2} \rfloor \lceil \frac{n}{2} \rceil}{2} \right\rceil$. Si n est impair, on a donc $\pi(C_n, AA, 1) \geq \frac{n^2-1}{8}$.

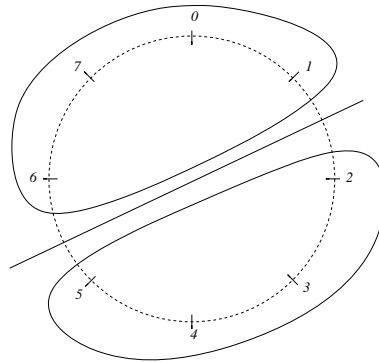


FIG. 3.13 – Utilisation du critère de coupe pour $\pi(C_n, AA, 1)$.

Mais cette borne inférieure n'est pas fine quand le cycle a un nombre de sommets multiple de 4 : $\pi(C_4, AA, 1) = 3 \neq 2$.

Considérons que le nombre de sommets n du cycle est multiple de 4. Si on avait $\pi = \frac{n^2}{8}$, alors toutes les arêtes auraient la même charge ; mais les sommets ont chacun $n - 1$ voisins et $n - 1$ est impair. Donc pour tout sommet du cycle, l'une de ses deux arêtes virtuelles incidente a une charge strictement supérieure à celle de l'autre arête virtuelle. Donc quand n est multiple de 4, la charge maximum est strictement supérieure à la charge moyenne.

Soit :

$$\pi(C_n, AA, 1) \geq \begin{cases} \frac{n^2+4}{8} & \text{si } n \equiv 2[4] \\ \frac{n^2}{8} + 1 & \text{si } n \equiv 0[4] \\ \frac{n^2-1}{8} & \text{si } n \equiv 1[2] \end{cases}$$

Pour trouver un plongement réalisant cette charge, il suffit d'associer à chaque arête virtuelle le plus court chemin dans le cycle lorsqu'il n'y en a qu'un (i.e. : n est impair), ce qui donne une charge uniforme de $\sum_{i=1}^{\lceil n/2 \rceil - 1} i$, qui vaut $\frac{n^2-1}{8}$ si n est impair et $\frac{n^2}{8} - \frac{n}{4}$ si n est pair. Si n est pair on répartit les arêtes de longueurs $n/2$ en alternant le sens : on relie $2i$ à $2i + n/2$ en routant l'arête dans un sens et $2i + 1$ à $2i + 1 + n/2$ en routant l'arête dans l'autre sens ; ce qui donne une charge supplémentaire de $\lfloor \frac{n}{4} + 1 \rfloor$, pour obtenir une charge égale la borne inférieure. La figure 3.14 représente le VPL optimal sur le cycle à huit sommets, en bleue sont représentée les arêtes virtuelles adjacentes au sommet 6 dont la longueur du chemin associé est strictement inférieure à $\frac{n}{2} = 4$ et en rouge les arêtes dont les chemins associés sont de longueur $\frac{n}{2}$.

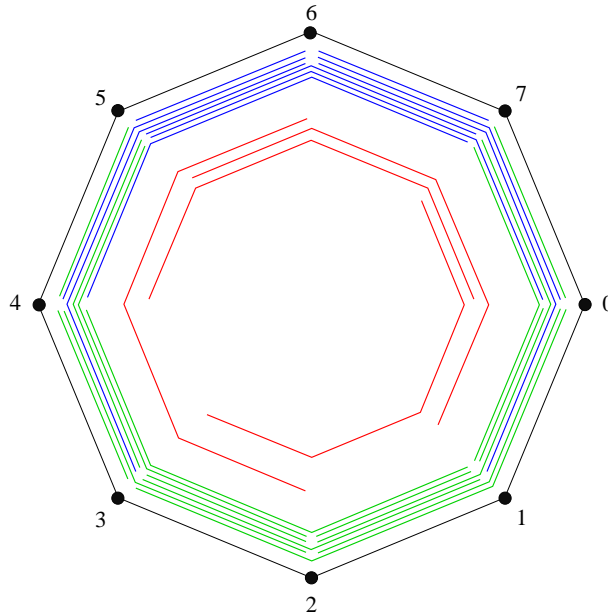


FIG. 3.14 - $\pi(C_8, AA, 1) = 9$

□

Théorème 3.2.2 (Choplin, Jarry, Pérennes)

$$\left\lceil \frac{n}{3} \right\rceil - 1 \leq \pi(C_n, AA, 2) \leq \left\lfloor \frac{n+1}{3} \right\rfloor$$

PREUVE.

Borne supérieure (construction) : $\pi(C_n, AA, 2) \leq \left\lfloor \frac{n+1}{3} \right\rfloor$

On va construire un graphe H de diamètre 2 tel que $\pi(C_n, H) = \left\lfloor \frac{n+1}{3} \right\rfloor$.
Considérons les intervalles suivants :

$$\begin{aligned} I_1 &= \{1, \dots, \lfloor n/3 \rfloor - 1\} \\ I_2 &= \{\lfloor n/3 \rfloor + 1, \dots, n - \lfloor n/3 \rfloor - 1\} \\ I_3 &= \{n - \lfloor n/3 \rfloor + 1, \dots, n - 1\} \end{aligned}$$

On place les arêtes ainsi, en utilisant un plongement des plus courts chemins :

- $[0, \lfloor n/3 \rfloor]$ et $[0, n - \lfloor n/3 \rfloor]$
- Pour tout x dans I_1 , on ajoute les arêtes $[0, x]$ et $[x, \lfloor n/3 \rfloor]$.
- Pour tout x dans I_2 , on ajoute les arêtes $[\lfloor n/3 \rfloor, x]$ et $[x, n - \lfloor n/3 \rfloor]$.
- Pour tout x dans I_3 , on ajoute les arêtes $[n - \lfloor n/3 \rfloor, x]$ et $[x, 0]$.

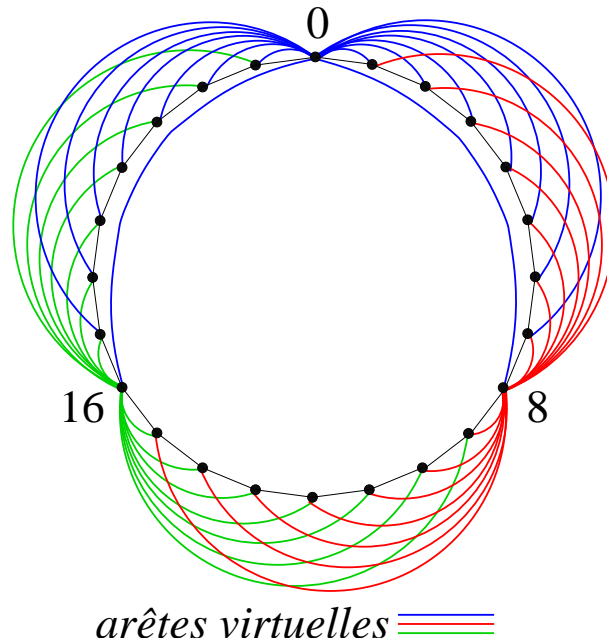


FIG. 3.15 – Exemple de graphe virtuel pour $D = 2$ et $n = 24$, le plongement des arêtes virtuelles est effectué en utilisant les plus courts chemins dans le cycle. La charge est 8.

Le graphe ainsi obtenu est de diamètre 2. La charge dans les intervalles I_1 et I_3 est uniforme et vaut $\lfloor n/3 \rfloor$, la charge dans l'intervalle I_2 est uniforme et égale au nombre de sommets de I_2 soit $n - 1 - 2 \lfloor n/3 \rfloor$; la charge maximale est donc $\lfloor (n + 1)/3 \rfloor$. La figure 3.15 nous montre un exemple de graphe virtuel sur C_n de diamètre 2.

Borne inférieure : $\lfloor \frac{n}{3} \rfloor - 1 \leq \pi(C_n, AA, 2)$

Considérons un VPL (H, \mathcal{P}) de charge optimale π sur le cycle C_n tel que H soit de diamètre 2. On partitionne l'ensemble des sommets du cycle en trois ensembles connexes (dans C_n) P_0, P_1, P_2 de cardinalité supérieure ou égale à $\lfloor \frac{n}{3} \rfloor$, les indices étant pris modulo 3, on notera $P_3 = P_0$. On appelle *GR* le *graphe résiduel*, le graphe dont l'ensemble des sommets est V et l'ensemble des arêtes est constitué des arêtes entre les parties P_i dans H .

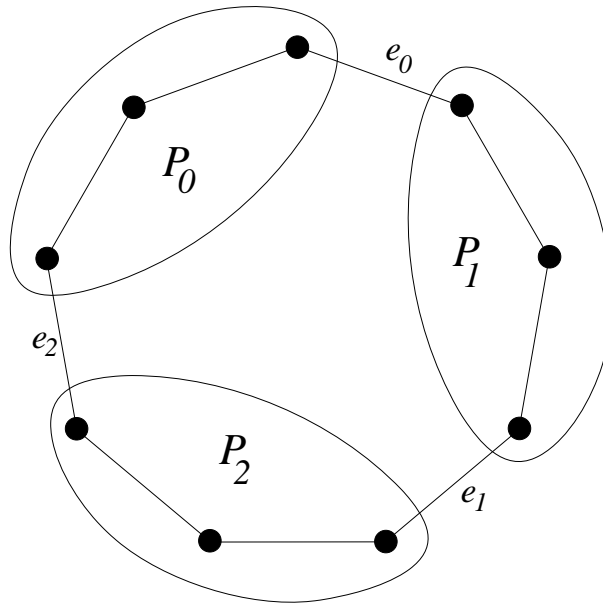


FIG. 3.16 – Partition du cycle en 3 ensemble de cardinaux voisins.

NOTATION. On note e_i l'arête physique entre P_i et P_{i+1} (avec i modulo 3).

NOTATION. On note $e(P_i, P_j)$ le nombre d'arêtes entre P_i et P_j dans *GR*.

REMARQUE. On n'a pas nécessairement $\pi \geq e(P_i, P_j)$ car les arêtes virtuelles entre P_i et P_j peuvent être réparties sur les trois arêtes (physiques) de coupes.

Les arêtes de *GR* issues d'un ensemble P_i utilisent les arêtes physiques e_i ou e_{i-1} , comme la charge sur ces arêtes est au plus π , on a

$$\forall i, j, k \text{ distincts, } e(P_i, P_j) + e(P_i, P_k) \leq 2\pi \quad (3.1)$$

Donc

$$e(P_0, P_1) + e(P_1, P_2) + e(P_2, P_0) \leq 3\pi \quad (3.2)$$

Cas 1 : Il existe un sommet de degré 0 dans GR .

Quelque soit la partie P_i , s'il existe un sommet de P_i de degré 0 (dans GR) alors tous les sommets de P_{i-1} et P_{i+1} doivent avoir un voisin dans P_i (car le diamètre du graphe virtuel est 2) et on a donc $e(P_i, P_{i+1}) + e(P_i, P_{i-1}) \geq |P_{i+1}| + |P_{i-1}|$. Si $n = 3p$ ou $3p + 1$, on a d'après (3.1) $\pi \geq p$ et si $n = 3p + 2$, on a $\pi \geq p + 1$, donc dans tous les cas on a

$$\pi \geq \left\lfloor \frac{n+1}{3} \right\rfloor$$

Cas 2 : Il existe une partie P_i qui ne contient que des sommets de degré au moins 2 dans GR .

Quelque soit la partie P_i , si tous les sommets de P_i sont de degré au moins 2 (dans GR), on a $e(P_i, P_{i+1}) + e(P_i, P_{i-1}) \geq 2|P_i|$. On a donc d'après (3.1) :

$$\pi \geq \left\lfloor \frac{n}{3} \right\rfloor \geq \left\lceil \frac{n}{3} \right\rceil - 1$$

Cas 3 : Toute partie P_i contient au moins un sommet de degré 1 et des sommets de degré au moins 2.

NOTATION. On note $\{P_i \rightarrow P_j\}$ l'ensemble des sommets de P_i qui ne sont adjacents (dans GR) qu'à des sommets de P_j (donc non adjacents à un sommet de P_k pour $k \neq i, j$). Il existe ici six ensembles de ce type.

REMARQUE. Tout sommet de degré 1 de P_i appartient à $\{P_i \rightarrow P_j\}$ ou à $\{P_i \rightarrow P_k\}$.

Supposons qu'il existe x dans $\{P_j \rightarrow P_i\}$ et y dans $\{P_k \rightarrow P_i\}$. x n'ayant pas de voisin dans P_k et y aucun voisin dans P_j , pour que ces sommets soient à distance 2 dans le graphe virtuel, il est nécessaire qu'ils aient un voisin commun dans P_i dans le graphe virtuel et donc aussi dans le graphe GR . Donc ils sont dans la même composante connexe dans GR :

$$\left. \begin{array}{l} \{P_j \rightarrow P_i\} \neq \emptyset \\ \{P_k \rightarrow P_i\} \neq \emptyset \end{array} \right\} \Rightarrow \left. \begin{array}{l} \text{les sommets de } \{P_j \rightarrow P_i\} \text{ et } \{P_k \rightarrow P_i\} \\ \text{appartiennent à une même composante} \\ \text{connexe de } GR. \end{array} \right\} (3.3)$$

Cas 3.1 : Les 6 ensembles $\{P_i \rightarrow P_j\}$ sont non vides.

D'après (3.3), on a au plus 3 composantes connexes contenant les 6 ensembles et donc tous les sommets de degré 1. Ces 3 composantes connexes ont un nombre d'arêtes au moins égal à leur nombre de sommets moins 1. Toutes les autres composantes connexes de GR n'ayant que des sommets de degré au moins 2 ont un nombre d'arêtes supérieur à leur nombre de sommets. Donc le graphe résiduel GR a au moins $n - 3$ arêtes et donc d'après 3.2, $3\pi \geq n - 3$ et donc

$$\pi \geq \left\lceil \frac{n}{3} \right\rceil - 1$$

Cas 3.2 : Il existe un ensemble $\{P_i \rightarrow P_j\}$ vide.

Alors tout sommet de P_i est adjacent à au moins un sommet de P_k et $e(P_i, P_k) \geq |P_i|$. Comme il existe au moins un sommet de degré 1 dans P_i par hypothèse, on a $\{P_i \rightarrow P_k\}$ qui est non vide.

Si $\{P_j \rightarrow P_k\}$ est vide alors tout sommet de P_j est adjacent à au moins un sommet de P_i et $e(P_i, P_k) + e(P_j, P_i) \geq |P_i| + |P_j|$ et d'après (3.1), $\pi \geq \left\lfloor \frac{n+1}{3} \right\rfloor$.

Si $\{P_j \rightarrow P_i\}$ est vide alors tout sommet de P_j est adjacent à au moins un sommet de P_k et $e(P_i, P_k) + e(P_j, P_k) \geq |P_i| + |P_j|$ et d'après (3.1), $\pi \geq \left\lfloor \frac{n+1}{3} \right\rfloor$.

Si $\{P_k \rightarrow P_i\}$ est vide alors tout sommet de P_k est adjacent à au moins un sommet de P_j et $e(P_i, P_k) + e(P_j, P_k) \geq |P_i| + |P_k|$. et d'après (3.1), $\pi \geq \left\lfloor \frac{n+1}{3} \right\rfloor$.

Comme $\{P_i \rightarrow P_k\}, \{P_j \rightarrow P_k\}, \{P_j \rightarrow P_i\}$ et $\{P_k \rightarrow P_i\}$ sont tous non vides, d'après (3.3), $\{P_j \rightarrow P_i\}, \{P_k \rightarrow P_i\}$ et $\{P_i \rightarrow P_k\}, \{P_j \rightarrow P_k\}$ sont dans au plus deux composantes connexes de GR .

Cas 3.2a : Si $\{P_k \rightarrow P_j\}$ ne contient pas de sommets de degré 1 alors on a deux composantes connexes contenant les sommets de degré 1 et toutes les autres composantes connexes ne contiennent que des sommets de degré au moins 2; donc GR a au moins $n - 2$ arêtes et $\pi \geq \left\lceil \frac{n+1}{3} \right\rceil - 1$.

Cas 3.2b : S'il existe un sommet x de degré 1 dans $\{P_k \rightarrow P_j\}$, on appelle y , son voisin dans P_j . Si y n'a pas de voisin dans P_i alors $y \in \{P_j \rightarrow P_k\}$ et x appartient à la même composante connexe que $\{P_j \rightarrow P_k\}$.

Si y a un voisin z dans P_i , comme $z \in \{P_i \rightarrow P_j\}$ est vide, z a un voisin t dans P_k . Si t n'a pas de voisin dans P_j alors $t \in \{P_k \rightarrow P_i\}$ et t et donc x appartiennent à la même composante connexe que $\{P_k \rightarrow P_i\}$, sinon t a un voisin u dans P_j . En continuant ainsi, soit x appartient à l'une des deux composantes connexes soit x appartient à un cycle mais alors la composante connexe de x a au moins autant d'arêtes que de sommets. Donc dans tous les cas on a un nombre d'arêtes supérieur ou égal à $n - 2$. et $\pi \geq \left\lceil \frac{n+1}{3} \right\rceil - 1$.

Valeur de n	Borne supérieure	Borne inférieure			
		Cas 1	Cas 2	Cas 3.1	Cas 3.2
$n = 3p$	p	p	p	$p - 1$	p
$n = 3p + 1$	p	p	p	p	p
$n = 3p + 2$	$p + 1$	$p + 1$	p	p	p

FIG. 3.17 – Récapitulatif des bornes inférieures en fonctions de n et des différents cas

La borne inférieure correspond donc à la borne supérieure dans le cas $n = 3p + 1$. Nous conjecturons que la valeur optimale est atteinte pour $\pi = \left\lfloor \frac{n+1}{3} \right\rfloor$ si $n \geq 6$, c'est à dire que la construction est optimale. Cette conjecture ne peut s'appliquer que pour $n \geq 6$ car pour $n = 5$, le cycle à 5 sommets étant lui-même de diamètre 2, le graphe virtuel optimal est aussi le cycle à 5 sommets et sa charge est alors 1 et non $\left\lfloor \frac{6}{3} \right\rfloor = 2$.

□

Conjecture 3.2.3

$$\pi(C_n, AA, 2) = \left\lfloor \frac{n+1}{3} \right\rfloor$$

Théorème 3.2.4 (Borne inférieure)

$$\frac{1}{2} \left(\frac{1}{2} \right)^{2/h} n^{2/h} \leq \pi(C_n, AA, h)$$

PREUVE.

Nous allons prouver la borne inférieure par récurrence sur h . L'inégalité est vérifiée pour $h = 1$ et $h = 2$. Soit k un entier inférieur à n . On partitionne l'ensemble des sommets en k intervalles de tailles quasi-égale :

$$I_i = \left[i \times \frac{n}{k}, \dots, (i+1) \times \frac{n}{k} - 1 \right] \text{ pour } 1 \leq i \leq k-1$$

- S'il existe i tel que I_i est sortant alors la charge est au moins $\frac{n}{2k}$ (on peut répartir les arêtes virtuelles sortantes sur deux arêtes physiques).
- Si au contraire, pour tout i , I_i est non sortant alors le graphe quotienté aux intervalles I_i est de diamètre $h - 2$ donc la charge est supérieure à celle du graphe virtuel de diamètre $h - 2$ à k sommets.

On a donc

$$\pi(C_n, AA, h) \geq \max_k \left\{ \min \left\{ \frac{n}{2k}, \frac{1}{2} \left(\frac{1}{2} \right)^{2/(h-2)} k^{2/(h-2)} \right\} \right\}$$

Donc on cherche k tel que $\frac{n}{2k} = \left(\frac{1}{2} \right)^{\frac{h}{h-2}} k^{\frac{2}{h-2}}$.

On a alors $k_{max} = 2^{\frac{2}{h}} n^{\frac{h-2}{h}}$ et donc, $\frac{n}{2k_{max}} = \frac{1}{2} \left(\frac{1}{2} \right)^{\frac{2}{h}} n^{\frac{2}{h}}$ d'où

$$\pi(C_n, AA, h) \geq \frac{1}{2} \left(\frac{1}{2} \right)^{2/h} n^{2/h}$$

□

Théorème 3.2.5 (Borne supérieure)

$$\pi(C_n, AA, h) \leq \begin{cases} \frac{p}{2(p-1)} \left(\frac{2}{3}(p-1)!(p-1) \right)^{1/p} n^{2/2p} + o(n^{2/h}) & \text{si } h = 2p \\ \frac{2p+1}{4p} (p!)^{2/(2p+1)} \left(\frac{p}{2} \right)^{1/(2p+1)} n^{2/(2p+1)} + o(n^{2/h}) & \text{si } h = 2p+1 \end{cases}$$

PREUVE.

Pour obtenir une construction en h sauts sur le cycle, on peut découper le cycle en plusieurs chemins, construire des VPL optimaux pour la diffusion dans ces chemins en h_0 sauts et construire le meilleur VPL que l'on connaisse pour le cycle entre les sommets initiateurs des diffusions en h_1 sauts tels que $2h_0 + h_1 \leq h$.

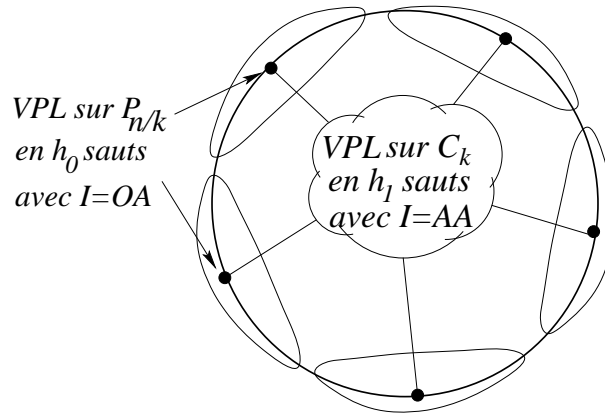


FIG. 3.18 – Construction pour un VPL sur le cycle en h sauts pour l'échange total avec $h = 2h_0 + h_1$.

Nous savons d'après [DFZ97b] que si l'initiateur de la diffusion dans le chemin est au centre du chemin, nous avons $n_{OA}(P, h, c) = B(h, c)$ où $n_{OA}(P, h, c)$ est le nombre maximum de sommets que l'on peut avoir dans un chemin pour lequel il existe un VPL de charge c et de nombre de saut maximal h et $B(h, c)$ est le nombre de sommets entiers dans la boule de dimension c et de rayon h pour la distance l_1 (longueur du plus court chemin dans la grille). On en déduit que quand OA est une instance de diffusion avec la source au centre du chemin :

$$\frac{1}{2}(h!n)^{1/h} - h/2 \leq \pi(P_n, OA, h) \leq \frac{1}{2}(h!n)^{1/h} + 1$$

On peut donc utiliser la valeur de la borne supérieure pour effectuer les calculs (cette borne correspondant à une construction). Comme nous savons d'après 2.2.6 que $\pi(C_n, AA, h) \geq \frac{2^{1/h}}{4}\pi(C_n, AA, 1)^{1/h}$ et comme $\pi(C_n, AA, h) \leq \pi(P_n, AA, h) = O(n^{2/h})$, nous avons : $\pi(C_n, AA, h) = O(n^{2/h})$, nous effectuerons donc les calculs en $O(n^{2/h})$ pour calculer le facteur du terme de plus au degré. Donc si on partitionne le cycle en k sous-ensembles connexes de cardinal $\frac{n}{k}$, la charge induite par les VPLs sur les chemins est $\frac{1}{2}(h_0! \frac{n}{k})^{1/h_0} + 1$. La charge induite par le VPL sur le cycle est $\pi(C_k, AA, h_1)$.

Pour $h = 3$, nous n'avons pas le choix pour h_0 et h_1 , nécessairement $h_0 = h_1 = 1$. Ce qui donne une charge de $\frac{k^2}{8} + \frac{1}{2}\frac{n}{k}$ qui est minimum pour $k = (2n)^{1/3}$ est vaut dans ce cas $\frac{3}{8}2^{2/3}n^{2/3}$. La figure 3.19 donne les premières valeurs des constructions obtenues ainsi. Pour un h donné, il semblerait qu'il est toujours plus intéressant de prendre h_1 le plus petit possible.

h	h_0	h_1	$\pi(P_{n/k}, OA, h_0)$	Charge pour le VPL du cycle	Valeur optimale $+o(n^{2/h})$
3	1	1	$\frac{1}{2}\frac{n}{k}$	$\frac{k^2}{8}$	$\frac{3}{8}2^{2/3}n^{2/3}$
4	1	2	$\frac{1}{2}\frac{n}{k}$	$\frac{k}{3}$	$\frac{6^{1/2}}{3}n^{1/2}$
5	1	3	$\frac{1}{2}\frac{n}{k}$	$\frac{3}{8}2^{2/3}k^{2/3}$	$\frac{5}{8}2^{4/5}n^{2/5}$
	2	1	$\frac{1}{2}\left(\frac{2n}{k}\right)^{1/2}$	$\frac{k^2}{8}$	$\frac{5}{8}2^{2/5}n^{2/5}$
6	1	4	$\frac{1}{2}\frac{n}{k}$	$\frac{6^{1/2}}{3}k^{1/2}$	$\left(\frac{3}{2}\right)^{2/3}n^{1/3}$
	2	2	$\frac{1}{2}\left(\frac{2n}{k}\right)^{1/2}$	$\frac{k}{3}$	$\frac{3^{2/3}}{2}n^{1/3}$
7	1	5	$\frac{1}{2}\frac{n}{k}$	$\frac{5}{8}2^{2/5}k^{2/5}$	$\frac{7}{8}2^{4/7}n^{2/7}$
	2	3	$\frac{1}{2}\left(\frac{2n}{k}\right)^{1/2}$	$\frac{3}{2^{7/3}}k^{2/3}$	$\frac{7}{8}2^{4/7}n^{2/7}$
	3	1	$\frac{1}{2}\left(\frac{6n}{k}\right)^{1/3}$	$\frac{k^2}{8}$	$\frac{7}{12}2^{1/7}3^{3/7}n^{2/7}$
...

FIG. 3.19 – Valeurs de la charge minimale des constructions obtenues pour le VPL sur le cycle en échange total.

$$\pi(C_n, AA, h) \leq \min_k \begin{cases} \pi(P_{n/k}, AA, p-1) + \pi(C_k, AA, 2) & \text{si } h = 2p \text{ est pair} \\ \pi(P_{n/k}, AA, p) + \pi(C_k, AA, 1) & \text{si } h = 2p+1 \text{ est impair} \end{cases}$$

$$\pi(P_{n/k}, AA, p-1) + \pi(C_k, AA, 2) = \frac{1}{2} \left(\frac{(p-1)!n}{k} \right)^{1/(p-1)} + \frac{k}{3}$$

$$\pi(P_{n/k}, AA, p) + \pi(C_k, AA, 1) = \frac{1}{2} \left(\frac{p!n}{k} \right)^{1/p} + \frac{k^2}{8}$$

Si $h = 2p$ est pair, le minimum est atteint pour $k = \left(\frac{3}{2(p-1)} \right)^{p-1/p} ((p-1)!n)^{1/p}$

Si $h = 2p+1$ est impair, le minimum est atteint pour $k = \left(\frac{2}{p} \right)^{p/(2p+1)} (p!n)^{1/(2p+1)}$

On a donc :

$$\pi(C_n, AA, h) \leq \begin{cases} \frac{p}{2(p-1)} \left(\frac{2}{3} (p-1)! (p-1) \right)^{1/p} n^{2/2p} + o(n^{2/h}) & \text{si } h = 2p \\ \frac{2p+1}{4p} (p!)^{2/(2p+1)} \left(\frac{2}{p} \right)^{1/(2p+1)} n^{2/(2p+1)} + o(n^{2/h}) & \text{si } h = 2p+1 \end{cases}$$

□

Chapitre 4

Le VPL dans les arbres

4.1 Echange total en 1 sauts

Théorème 4.1.1 *Soit $T = (V, E)$ un arbre, u et v deux sommets de T , $T_{[u,v],u}$ la composante connexe de $T - [u, v]$ contenant u ,*

$$\pi(T, AA, 1) = \max_{[u,v] \in E} \{|T_{[u,v],u}| \times |T_{[u,v],v}|\}$$

PREUVE. Le graphe virtuel est le graphe complet, donc pour toute arête physique le nombre d'arêtes virtuelles l'utilisant est le produit des cardinaux de deux composantes connexes qu'elle engendre quand on la supprime. \square

4.2 Echange total en 2 sauts

NOTATION. Étant donné un arbre T , x un sommet de T , on notera $\{T_{x,i}\}_i$ l'ensemble des composantes connexes de $T - x$.

Définition 4.2.1 *Soit T un arbre, on appelle **sommet central** de T tout sommet c tel que chacune des composantes connexes qu'il engendre quand on l'enlève est de cardinal inférieur ou égal à la moitié du nombre de sommets, i.e. :*

$$\forall i, |T_{c,i}| \leq \sum_{j \neq i} |T_{c,j}| + 1$$

REMARQUE. Un sommet central peut être différent d'un centre qui par définition est un sommet d'excentricité minimale.

Proposition 4.2.1 *Pour tout arbre, il existe au moins un sommet central*

PREUVE. Pour trouver un tel sommet, il suffit de partir de n'importe quel sommet, si la condition n'est pas vérifiée de considérer son voisin qui fait partie de la composante de cardinal le plus élevé et de recommencer (voir l'algorithme de la figure 1) jusqu'à ce que la condition soit vérifiée..

Algorithme 1 Renvoie le sommet central de l'arbre

```

Trouve_Sommet_central(Arbre  $T$ , Sommet  $x$  de  $T$ )
  si il existe  $i$  tel que  $|T_{x,i}| \geq \sum_{j \neq i} |T_{x,j}| + 2$  alors
     $x \leftarrow$  Voisin de  $x$  dans  $T_{x,i}$ 
    Renvoie(Trouve_Sommet_central( $T, x$ ))
  sinon
    Renvoie  $x$ 
  fin si

```

Supposons que l'algorithme 1 ne se termine pas, cela signifie qu'il va sélectionner un sommet y en venant d'un sommet x puis revenir en x . Soit $T_{x,i}$ la composante connexe contenant y , on a $|T_{x,i}| \geq \sum_{j \neq i} |T_{x,j}| + 2$ et comme une fois en y il retourne en x c'est que $\sum_{j \neq i} |T_{x,j}| + 1 \geq |T_{x,i}| - 1 + 2$ et donc $|T_{x,i}| - 1 \geq |T_{x,i}| + 1$ ce qui est impossible. La figure 4.1 donne un exemple de sommet central trouvé par l'algorithme 1.

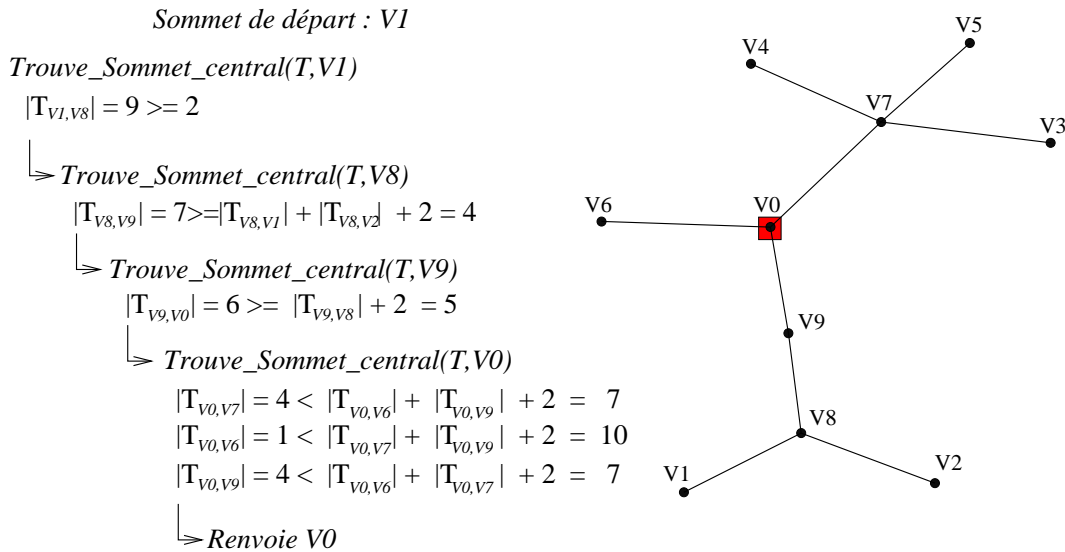


FIG. 4.1 – Sommet central de l'arbre.

□

Théorème 4.2.2 Soit $T = (V, E)$ un arbre,

$$\pi(T, AA, 2) = \min_{x \in V} \{ \max_i \{ |T_{x,i}| \} \}$$

PREUVE. Pour tout sommet x de T et toute composante $T_{x,i}$, la charge sur l'arête entre x et $T_{x,i}$ est au moins $\min\{|T_{x,i}|, \sum_{j \neq i} |T_{x,j}| + 1\}$ (le nombre de sauts étant 2, c'est $|T_{x,i}|$ si tous les sommets de $T_{x,i}$ sont sortants (de $T_{x,i}$) et $\sum_{j \neq i} |T_{x,j}| + 1$ s'il existe un sommet de $|T_{x,i}|$ non sortant ce qui implique que tous les autres sommets doivent avoir un voisin dans $|T_{x,i}|$ (le +1 vient du sommet x); la figure 4.2 illustre cet argument.).

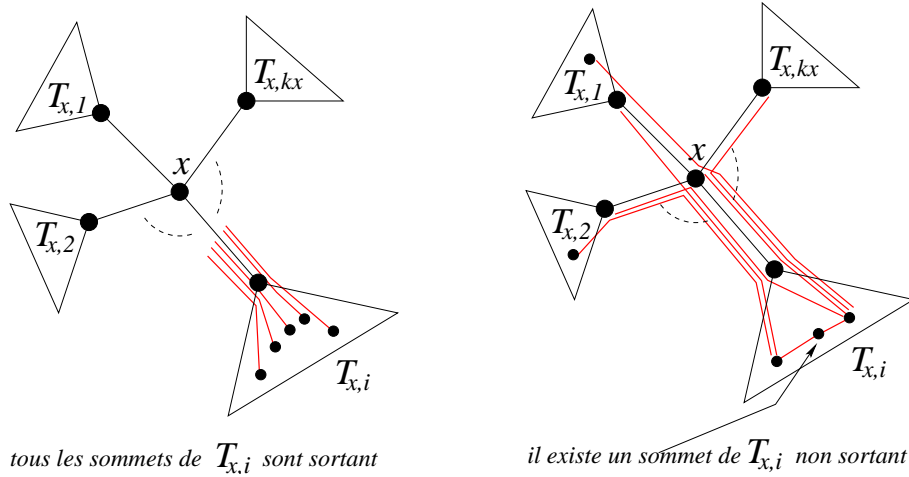


FIG. 4.2 – Pour tout sommet x , $\pi(T, AA, 2) \geq \min\{|T_{x,i}|, \sum_{j \neq i} |T_{x,j}| + 1\}$.

Pour un sommet central c de l'arbre, on a $|T_{c,i}| \leq \sum_{j \neq i} |T_{c,j}| + 1$ et donc $\pi(T, AA, 2) \geq \max_i \{|T_{c,i}|\}$ soit $\pi(T, AA, 2) \geq \min_{x \in V} \{\max_i \{|T_{x,i}|\}\}$.

La construction permettant de réaliser ce VPL est simple : il suffit de connecter tous les sommets à l'un des sommets centraux de l'arbre. \square

4.3 Diffusion en 2 sauts

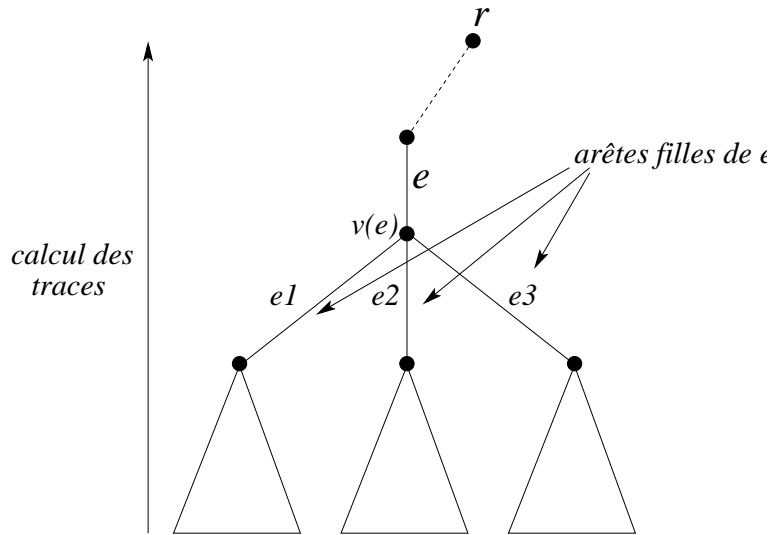
Dans cette partie nous montrons que le problème de l'existence et la construction d'un VPL dans un arbre quelconque avec un nombre de sauts maximal donné et une charge maximale donnée pour une instance de communication de type diffusion peut être résolu en temps polynomial. Dans [GCZ96], Gerstel, Cidon et Zaks ont montré que le même problème était polynomial dans le cas où le facteur d'étirement était limité à 1 et qu'il était NP -complet pour les graphes quelconque avec un facteur d'étirement non borné.

Définition 4.3.1 Dans un arbre T enraciné, on appelle *sous-arbre d'une arête* e , noté $T(e)$, la composante connexe de $T - e$ qui ne contient pas la racine.

Proposition 4.3.1 *Il existe un algorithme linéaire en temps (en fonction du nombre de sommets de l'arbre) et en mémoire pour la résolution du problème de décision de l'existence d'un VPL (problème A) avec $h = 2$ et c fixé dans les arbres de degré borné pour toute instance de communication de type diffusion.*

PREUVE. Nous donnons ici un algorithme utilisant la programmation dynamique et permettant de décider s'il existe un VPL de charge donnée avec un nombre maximal de sauts égal à 2 dans un arbre.

Notons r le sommet initiateur de la diffusion et enracinons l'arbre en ce sommet. Le principe de l'algorithme est le suivant : en remontant l'arbre à partir de feuilles vers la racine r , en considérant pour chaque arête des traces de trois paramètres : le nombre de sommets connectés à la source (on appellera ces sommets des *serveurs* qui sont dans le sous-arbre, le nombre de sommets du sous-arbre qui doivent être connectés à un serveur extérieur au sous-arbre et le nombre maximum de sommets extérieurs au sous-arbre que l'on peut connecter aux serveurs du sous-arbre. Il faut mémoriser toutes les combinaisons possibles de ces trois paramètres pour pouvoir progresser vers la racine en ne conservant que celles qui respectent la capacité. Une solution existe s'il existe une combinaison pour chaque arête fille de la racine qui respecte la charge. Pour toute arête e , on note $v(e)$ le sommet incident à e du côté des feuilles de l'arbre.



A chaque arête e , on associe une table $Traces(e)$ des traces, chaque trace t contient les entrées suivantes $(t.p, t.q, t.s)$:

- $t.s$ est le nombre de serveurs dans le sous-arbre de e .
- $t.p$ est le nombre maximum de sommets extérieurs au sous-arbre qui peuvent être connectés aux serveurs du sous-arbre.

- $t.q$ est le nombre de sommets du sous-arbre qui doivent être connectés à un serveur à l'extérieur du sous-arbre.

Une trace est dite **valide** si elle respecte la charge, c'est à dire si $t.s + t.q \leq c$. Le nombre $t.p$ vaut 0 si $t.s = 0$ sinon $t.p \leq c - t.s - t.q$. On ne gardera dans $Traces(e)$ que des traces valides et pour un nombre de serveurs donné on gardera la trace minimisant $t.q$ et si $t.q = 0$, on gardera celle maximisant $t.p$. En effet, supposons qu'il existe une solution globale au problème avec un certain $t.q \geq 1$ et $t.p \geq 1$ sur une arête e . Ceci signifie qu'au moins un sommet $x \in T(e)$ est relié à un serveur $s \notin T(e)$; comme $t.p \geq 1$ on peut le relier à un serveur s' de $T(e)$. Si s' était relié à un sommet $x' \notin T(e)$, on relie x' à s et on obtient une solution qui n'augmente la charge d'aucune arête (et au contraire la diminue sur certaines arêtes, en particulier e). La solution avec $t.q' = t.q - 1$ et $t.p' = t.p - 1$ est donc aussi une solution (la figure 4.3 illustre la transformation).

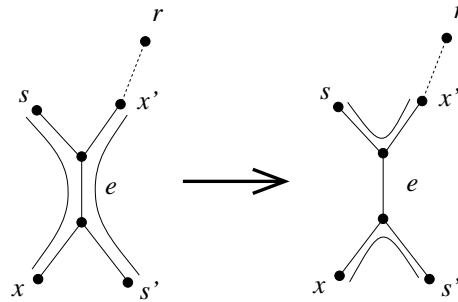


FIG. 4.3 – On peut trouver une trace meilleure que la trace t si $t.p \geq 1$ et $t.q \geq 1$.

Ainsi on peut ne garder que des traces ayant soit $t.p = 0$ soit $t.q = 0$. Si on a une solution globale avec $t.q \geq 1$ et $t.p = 0$ sur e et que la table de e contient une autre trace t' avec $1 \leq t'.q \leq t.q$, $t'.p = 0$ et $t'.s = t.s$ alors cette trace doit aussi permettre de générer une solution globale vu qu'elle nécessite moins de connexion que t . De même si on a une solution globale avec $t.q = 0$ et $t.p \geq 1$ sur e et que la table de e contient une autre trace t' avec $1 \leq t'.p \leq t.p$, $t'.q = 0$ et $t'.s = t.s$ alors cette trace doit aussi permettre de générer une solution globale vu qu'elle permet de connecter plus de sommets que t .

A chaque étape de l'algorithme, on ne gardera donc au plus que $c + 1$ traces par table (une par nombre possible de serveurs).

Génération d'une trace t à partir des traces des arêtes filles t_i

Pour générer les traces d'une arête e , il faut d'abord générer les traces des arêtes filles et ensuite faire des combinaisons de ces traces avec deux cas différents :

- le sommet $v(e)$ est un serveur,
- le sommet $v(e)$ n'est pas un serveur.

Lorsque la trace d'une arête fille $e1$ permet de connecter des sommets ($t.p \geq 1$) et que la trace d'une autre arête fille $e2$ doit connecter des sommets ($t.q \geq 1$), on gardera une trace qui connecte au maximum les sommets du sous-arbre de $e2$ dans le sous-arbre de $e1$. Quand on génère les traces d'une arête e à partir d'une combinaison de traces (t_1, t_2, \dots, t_i) de ses arêtes filles, on a donc :

- Si $v(e)$ n'est pas serveur :
 - le nombre de serveurs est la somme des serveurs des traces filles. Soit :

$$t.s = \sum_i t_i.s$$

- Si $\sum_i t_i.q + 1 \leq \sum_i t_i.p$ cela signifie que tous les sommets de $T(e)$ qui ne sont pas serveurs peuvent être connecté à des serveurs de $T(e)$, et donc $t.q = 0$. Si $\sum_i t_i.q + 1 > \sum_i t_i.p$ cela signifie qu'il y a des sommets de $T(e)$ qui ne peuvent pas être reliés à des serveurs de $T(e)$, ils devront donc être comptés dans la trace de t comme des sommets à connecter à l'extérieur de $T(e)$.

Soit :

$$t.q = \max \left\{ 0, \sum_i (t_i.q - t_i.p) + 1 \right\}$$

- Comme précédemment, si $\sum_i t_i.q + 1 \leq \sum_i t_i.p$ alors de serveurs de $T(e)$ peuvent être connecté à des sommets non serveurs en dehors de $T(e)$ sinon on a $t.p = 0$. Mais la contrainte de la charge impose que le nombre de sommets qu'on peut connecter plus le nombre de serveurs du sous-arbre soit inférieur à la charge donc si $t.p \geq 1$, $t.p$ est borné par $c - t.s$.

Soit :

$$t.p = \max \left\{ 0, \min \left\{ \sum_i (t_i.p - t_i.q) - 1, c - t.s \right\} \right\}$$

- Si $v(e)$ est serveur
 - le nombre de serveurs est la somme des serveurs des traces filles plus un pour $v(e)$. Soit :

$$t.s = 1 + \sum_i t_i.s$$

- il n'y a plus de sommets à connecter à l'intérieur du sous-arbre vu qu'on peut connecter à $v(e)$ tous ceux qui devait être connectés à un serveur.

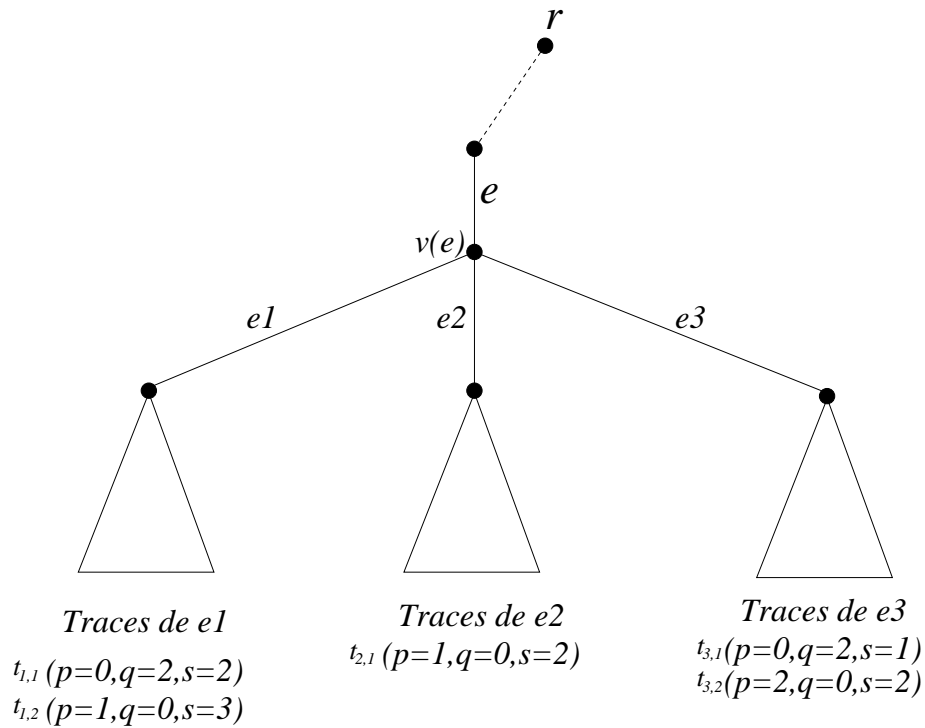
Soit

$$t.q = 0$$

- le nombre de sommets qu'on peut connecter est la capacité moins le nombre de serveurs (étant directement reliés à la source, ils induisent chacun une charge de 1 sur l'arête physique) :

$$t.p = c - t.s$$

La figure 4.4 donne un exemple de résultat de calcul des traces d'une arête. Les traces conservées (une par nombre de serveurs) seront $t_{1,1}t_{2,1}t_{3,1} = (p = 1, q = 0, s = 6)$ et $t_{1,1}t_{2,1}t_{3,2} = (p = 0, q = 0, s = 7)$ (ou $t_{1,2}t_{2,1}t_{3,2}$). On peut remarquer que la trace à 6 serveurs est «meilleure» que celle à 7 serveurs, dans l'algorithme nous garderons les deux, mais dans un soucis d'optimisation on pourra comparer les traces ayant un nombre de serveurs différent pour réduire la taille des tables.



Traces de e pour une charge max $c=7$

<i>Si $v(e)$ est serveur</i>	<i>Si $v(e)$ n'est pas serveur</i>
$t_{1,1}t_{2,1}t_{3,1} (p=1, q=0, s=6)$	$t_{1,1}t_{2,1}t_{3,1} (p=0, q=4, s=5)^*$
$t_{1,1}t_{2,1}t_{3,2} (p=0, q=0, s=7)$	$t_{1,1}t_{2,1}t_{3,2} (p=0, q=0, s=6)$
$t_{1,2}t_{2,1}t_{3,1} (p=0, q=0, s=7)$	$t_{1,2}t_{2,1}t_{3,1} (p=0, q=1, s=6)$
$t_{1,2}t_{2,1}t_{3,1} (p=0, q=0, s=8)^*$	$t_{1,2}t_{2,1}t_{3,1} (p=0, q=0, s=7)$

* *trace invalide*

FIG. 4.4 – Calcul des traces d'une arête .

Initialisation

Les traces initiales sont celles dont les arêtes sont incidentes à une feuille f et sont : $(c - 1, 0, 1)$ et $(0, 1, 0)$.

REMARQUE. Dans les algorithmes, on notera $\max(t1, t2)$ la trace la plus intéressante entre $t1$ et $t2$ si elles ont le même nombre de serveurs.

L'algorithme 2 résume le calcul des traces.

Algorithme 2 Calcule les traces, version 1

Calculer Trace(Arbre T enraciné en r, e une arête de l'arbre T, c une charge maximale)

$Traces(e)$ est un tableau de c traces, $Traces(e)[i]$ représente la meilleure trace à i serveurs.

```

si  $e$  est incidente à une feuille alors
   $Traces(e) \leftarrow \{(c - 1, 0, 1), (0, 1, 0)\}$ 
sinon
  pour tout arête fille  $e'$  de  $e$  dont la table des traces n'est pas calculée faire
    Calculer Trace( $T, e', c$ )
  fin pour
   $Traces(e) \leftarrow \emptyset$ 
  pour tout combinaisons de traces filles  $t_1, t_2, \dots, t_k$  faire
    /* Cas où le sommet  $v(e)$  est un serveur */
     $s \leftarrow 1 + \sum_{i=1}^{i=k} t_i.s$ 
    si  $s \leq c$  /* si la charge est respectée */ alors
       $Traces(e)[s] \leftarrow \max \{Traces(e)[s], (c - s, 0, s)\}$ 
    fin si
    /* Cas où le sommet  $v(e)$  n'est pas un serveur */
     $s \leftarrow \sum_{i=1}^{i=k} t_i.s$ 
     $p \leftarrow \max \left\{ 0, \min \left\{ \sum_{i=1}^{i=k} (t_i.p - t_i.q) - 1, c - s \right\} \right\}$ 
     $q \leftarrow \max \left\{ 0, \sum_{i=1}^{i=k} (t_i.q - t_i.p) + 1 \right\}$ 
    si  $s + q \leq c$  /* la charge est respectée */ alors
       $Traces(e)[s] \leftarrow \max \{Traces(e)[s], (p, q, s)\}$ 
    sinon
      /* cette combinaison de traces ne permet pas de construire une trace
      valide */
    fin si
  fin pour
fin si

```

Pour résoudre le problème de décision de l'existence d'un VPL, il suffit de calculer la table des traces de chacune des arêtes incidentes à r (elles peuvent

être calculées simultanément, il est en effet inutile qu'une arête virtuelle soit routée sur un chemin contenant r si r n'est pas une extrémité de ce chemin et donc chacun des sous-arbres peut être considéré de manière indépendante). Si pour chacune de ces tables il existe une trace valide c'est qu'il existe une solution, sinon c'est impossible avec la charge c donnée.

Etude de l'algorithme 2.

Complexité en temps

Le nombre maximum de traces par arêtes est majoré par le nombre maximum de serveur (lui-même majoré par la charge). Si chaque arête a au plus c traces, le nombre de combinaisons des traces filles est au plus $c^{\Delta-1}$ si Δ est le degré maximum des sommets de l'arbre. L'algorithme 2 va s'appeler de manière recursive au plus $|V|$ fois, la complexité en temps est donc de l'ordre de

$$O(c^{\Delta-1}|V|)$$

Complexité en espace

Pour générer les traces d'une arête, il suffit d'avoir en mémoire les traces des arêtes filles. Si $f(i)$ est la complexité en espace pour une arête fille e tel que le sommet $v(e)$ est à distance i de la source, il faut avoir en mémoire les tables de ses premières arêtes filles soit au plus $(\Delta - 2)c$ tables et calculer les tables de la dernière soit au plus $f(i + 1)$, si p est la profondeur de l'arbre, $f(p)$ est constant et on a donc une complexité en espace en

$$O(p\Delta c)$$

p étant au pire de l'ordre de $|V|$, la complexité en temps est linéaire en le nombre de sommets.

□

Comment construire la solution

L'algorithme 2 permet uniquement de répondre au problème de l'existence ou non d'une solution. Si une solution existe, pour la construire il suffit de modifier un peu l'algorithme en conservant plus d'information, ce qui aura pour effet d'augmenter la complexité en espace.

Trouver les serveurs

Pour trouver les serveurs, il suffit de conserver un ensemble de serveurs possibles S pour chaque valeur de $t.s$, ce qui a pour effet d'augmenter la taille des traces

de c et dont la taille des tables sera de l'ordre de c^2 .

Si on connaît les serveurs, la résolution d'un problème de simple flot peut suffire à construire le graphe virtuel : il suffit de relier tous les sommets "serveurs" à r et de résoudre un problème de simple flot sur le graphe $G' = (V', E')$ orienté symétrique où $V' = V \cup \{Source, Puits\}$ et $E' = \{(x, y) \mid [x, y] \in E\} \cup (Source, r) \cup \{Source\} \times S \cup \{T\} \times (V \setminus (S \cup \{r\}))$. Les capacités sur les arcs de E' étant la charge disponible sur l'arête correspondante une fois que l'on a placé les arêtes virtuelles entre les serveurs et r .

Trouver un graphe virtuel On peut aussi modifier l'algorithme pour trouver directement un graphe virtuel correspondant à une solution. On va associer à chaque trace un ensemble d'arêtes virtuelles E . Comme précédemment, on conserve pour chaque nombre de serveurs possible un ensemble de sommets S représentatif de ces serveurs, on a donc dans E toutes les arêtes entre ces serveurs et la racine. Si q est positif, on conserve un ensemble de sommets Q à connecter et si p est positif, on conserve un multi-ensemble de sommets P (qui sont des serveurs), chaque serveur apparaissant autant de fois dans l'ensemble P que le nombre de sommets qu'il permet de connecter. Ainsi lorsqu'on peut connecter des sommets d'un sous arbre T_1 dans un sous-arbre T_2 avec les traces t_1 et t_2 , il suffit de choisir un sommet s_1 de l'ensemble $t_1.Q$ (qui n'est pas encore connecté) et un serveur s_2 de $t_2.P$ et d'ajouter à E l'arête $[s_1, s_2]$, il n'y a plus qu'à retirer les éléments s_1 et s_2 des ensembles P et Q . Ainsi à la fin de l'algorithme, chaque combinaison des traces filles valides de la racine fournit un graphe virtuel vérifiant les conditions de nombre de sauts et de charge.

Avec cette méthode, il faut conserver pour chaque trace deux ensembles de cardinal au plus c (on ne peut pas avoir p positif et q positif, ce qui fait qu'on garde S et P ou S et Q) et un ensemble d'arêtes de cardinal au plus $|V|$. La complexité en espace devient donc :

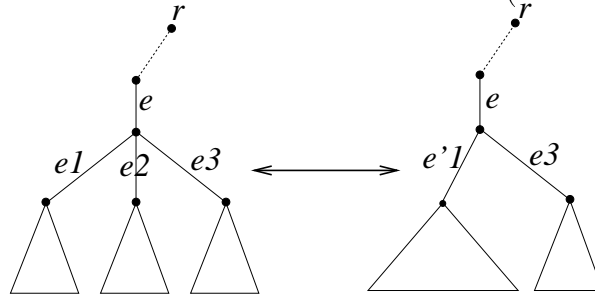
$$O(p\Delta c^2|V|)$$

La complexité en temps ne change pas car il n'y a aucune recherche à effectuer sur les ensembles, il suffit toujours de prendre le premier élément de chaque ensemble.

Proposition 4.3.2 *Il existe un algorithme linéaire en temps et en mémoire pour la résolution du problème de décision de l'existence d'un VPL (problème A) et la construction d'une solution optimale si elle existe avec $h = 2$ et c fixé dans les arbres pour toute instance de communication de type diffusion.*

PREUVE. La génération des traces dans l'algorithme 2 peut être faite de manière plus efficace : au lieu de faire toutes les combinaisons possibles des tables filles, on peut calculer des tables intermédiaires en ne considérant que deux tables filles à la fois. tout en ne conservant qu'au plus c traces (toujours une par nombre de

serveurs) avant de continuer avec la table fille suivante (illustration par la figure 4.5).



$$\text{Traces}(e'1) = \text{combinaisons de Traces}(e1) \text{ et Traces}(e2)$$

FIG. 4.5 – On peut combiner les traces deux à deux .

Ainsi au lieu de faire les combinaisons de Δ tables à c éléments, on peut faire Δ combinaisons de deux tables à c éléments. Les algorithmes 3 et 4 implémentent cette amélioration ainsi que la construction du graphe virtuel décrite précédemment. Dans les algorithmes 3 et 4, chaque trace contient les entrées (P, Q, S, E) où P est un multi-ensemble de serveurs du sous-arbre qui peuvent être reliés à d'autres sommets dans le graphe virtuel (chaque serveur apparaissant autant de fois qu'il peut connecter de sommets), Q est un ensemble de sommets du sous-arbre qui doivent être reliés dans le graphe virtuel à un serveur qui se trouve à l'extérieur du sous-arbre, S est l'ensemble des serveurs du sous arbre, E est l'ensemble d'arêtes du graphe virtuel qui sont déjà constituées. La fonction $\text{pop}(L)$ retire et renvoie le premier élément d'un ensemble L . La fonction $\text{cut}(P, c)$ retourne les c premiers éléments de P (pour que le nombre de sommets de P ne soit pas supérieur à la charge disponible c).

Algorithme 3 Combine deux tables de traces

Combine_Tables(Table T_1 , Table T_2) renvoie une table.

```

T_Resultat  $\leftarrow \emptyset$ 
pour tout combinaisons  $(t_1, t_2)$  de traces de  $T_1$  et  $T_2$  faire
   $S \leftarrow t_1.S \cup t_2.S$ 
   $P \leftarrow t_1.P \cup t_2.P$ 
   $Q \leftarrow t_1.Q \cup t_2.Q$ 
   $E \leftarrow t_1.E \cup t_2.E$ 
  tant que  $P \neq \emptyset$  et  $Q \neq \emptyset$  faire
     $sp = \text{pop}(P)$ 
     $sq = \text{pop}(Q)$ 
     $E \leftarrow E \cup [sp, sq]$ 
  fin tant que
  si  $|S| \leq c$  (si la charge est respectée)
    alors  $T\_Resultat[|S|] \leftarrow \max \{T\_Resultat[|S|], (P, Q, S, E)\}$  fin si
fin pour
Renvoie  $T\_Resultat$ 

```

Algorithme 4 Calcule les traces, version 2

 Calcule_Traces_2(Arbre T enraciné en r, e arête de T, c charge maximale)

```

si  $e$  est incidente à une feuille alors
   $Traces(e) = \{(\emptyset, \{v(e)\}, \emptyset, \emptyset), (\{v(e)(c - 1 \text{ fois})\}, \emptyset, \{[r, v(e)]\})\}$ 
sinon
  pour tout arête fille  $e'$  de  $e$  dont la table des traces n'est pas calculée faire
    Calcule_Trace_2( $T, e', c$ )
  fin pour
   $Traces\_temp(e) = \emptyset$ 
  pour tout arête fille  $e'$  de  $e$  faire
     $Traces\_temp(e) \leftarrow \text{Combine\_Tables}(Traces\_temp(e), Traces(e'))$ 
  fin pour
  pour tout Traces  $t$  de  $Traces\_temp$  faire
    /* Cas où le sommet  $v(e)$  est un serveur */
    si  $|t.S| + 1 \leq c$  (si la charge est respectée) alors
       $S \leftarrow t.S \cup \{v(e)\}; E \leftarrow t.E \cup \{[r, v(e)]\}$ 
      pour tout  $s \in t.Q$  faire  $E \leftarrow t.E \cup \{[s, v(e)]\}$  fin pour
       $Traces(e)[|S|] \leftarrow \max \{Traces(e)[|S|], (\{v(e)(c - |S| \text{ fois})\}, \emptyset, S, E)\}$ 
    fin si
    /* Cas où le sommet  $v(e)$  n'est pas un serveur */
     $P \leftarrow \emptyset$ 
     $Q \leftarrow \emptyset$ 
    si  $|t.P| \geq 1$  (On peut connecter le sommet dans le sous-arbre) alors
       $sp = pop(t.P)$ 
       $E \leftarrow t.E \cup \{[sp, v(e)]\}$ 
       $P \leftarrow cut(t.P, c - |t.S|)$ 
    sinon
       $Q \leftarrow t.Q \cup \{v(e)\}$ 
    fin si
    si  $|t.S| + |Q| \leq c$  (si la charge est respectée)
      alors  $Traces(e)[|t.S|] \leftarrow \max \{Traces(e)[|t.S|], (P, Q, t.S, E)\}$  fin si
  fin pour
fin si

```

Exemple

Dans l'exemple de la figure 4.6, l'algorithme 4 trouve une solution réalisable pour $c = 5$. On peut remarquer que certains voisins de la racine r ne sont pas marqués comme serveur, cela est dû au fait que l'algorithme 4 doit être appelé pour chacune des arêtes filles de la racine et ne résout donc pas la dernière étape qui consiste à connecter les sommets restants dans les ensembles Q de chacune

des traces valides à la racine r .

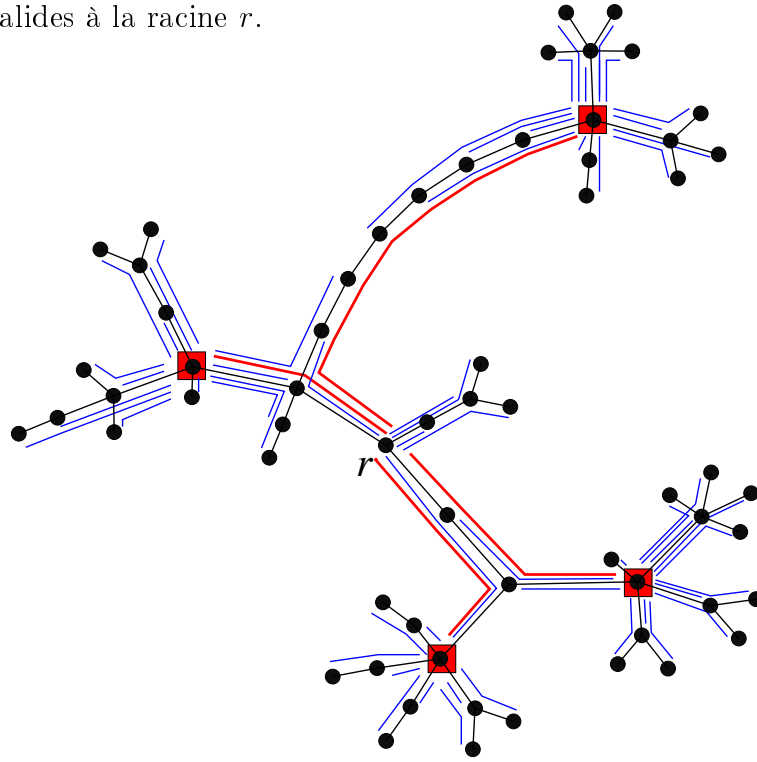


FIG. 4.6 – L’algorithme 4 a trouvé qu’une solution existait pour $c = 5$, les serveurs sont en rouge.

Etude de l’algorithme 4.

Complexité en temps

La complexité en temps de la combinaison de deux tables par l’algorithme 3 se fait en $O(c^2)$, la génération de la table intermédiaire se fait donc en $O(\Delta c^2)$. La partie de l’algorithme 4 consistant à décider si le sommet est serveur ou non s’effectue en $O(c)$ à cause de la fonction $cut(P, c - |S|)$. L’algorithme 4 s’appelant récursivement $|V|$ fois, la complexité total en temps est donc en $O(\Delta c^2 |V|)$. En fait on peut améliorer l’estimation de la complexité car le nombre de combinaisons de tables (de taille c) dans la totalité de l’algorithme correspond au nombre d’arêtes et donc à $|V| - 1$ et non $\Delta |V|$, ce qui fait une complexité de

$$O(c^2 |V|)$$

Complexité en espace

La combinaison des tables deux à deux permet aussi de réduire la complexité

en espace. Chaque table temporaire (de taille au plus c) est combinée avec une table fille pour générer une nouvelle table temporaire. Si p est la profondeur de l'arbre (l'excentricité de la source dans l'arbre), $f(i)$ la complexité en espace pour calculer une table de profondeur i , la complexité en espace pour une table de profondeur $i + 1$ est $c|V| + f(i) + c|V|$ ($c|V|$ pour la table temporaire où $|V|$ est le nombre maximum d'arêtes dans une trace, $f(i)$ pour la table du fils, $c|V|$ pour la combinaison des deux tables). Ce qui fait une complexité en

$$O(pc|V|)$$

□

4.4 Diffusion en h sauts

Théorème 4.4.1 *Il existe un algorithme polynomial en temps et en mémoire pour la résolution du problème de minimisation de la charge (problème C) et la construction d'une solution optimale avec $h = 2$ pour les arbres pour toute instance de communication de type One-To-Many (diffusion depuis une source à un sous-ensemble de V).*

PREUVE. Il est facile d'adapter l'algorithme présenté dans la preuve de la proposition 4.3.2 pour l'instance de communication *One – To – Many* : lors de la génération d'une trace, un sommet pouvait être de deux types différents (à distance 1 de la source ou à distance 2), si ce sommet n'est pas dans l'ensemble destination des requêtes il suffit de considérer un cas supplémentaire : le sommet n'est pas dans le graphe virtuel.

D'après le théorème 2.2.3, nous savons que $\pi(T, OM, 2) \leq 2\sqrt{\lceil \pi(T, OA, 1) \rceil} - 1 \leq 2\sqrt{|V|}$. La complexité en temps de l'algorithme pour la diffusion était $O(\Delta c^2 |V|)$, pour déterminer π , il suffit donc d'exécuter au plus $\log(2\sqrt{|V|})$ fois l'algorithme (en procédant de manière dichotomique sur la charge). Ce qui donne une complexité de $O\left(\Delta \log(2\sqrt{|V|})|V|^2\right) = O(\Delta \log(|V|)|V|^2)$. Le degré maximal Δ étant lui aussi inférieur au nombre de sommets, on obtient un algorithme polynomial en $|V|$. □

Proposition 4.4.2 *Le problème de décision de l'existence d'un VPL avec le nombre maximal de sauts h fixé et la charge maximale c fixée est polynomial en temps et en mémoire dans les arbres pour toute instance de communication de type diffusion.*

PREUVE. Comme dans le cas où h est égal à 2, nous donnons un algorithme basé sur la programmation dynamique résolvant ce problème. Dans le cas de

deux sauts la solution dans un sous-arbre était caractérisée par le nombre de sommets du sous-arbre adjacents à la source, le nombre de sommets qui n'était pas encore connectés (précédemment noté q) et le nombre de sommets extérieurs au sous-arbre qui pouvaient être connectés dans le sous-arbre tout en respectant la contrainte de charge (précédemment noté p).

On peut remarquer que les variables p et q ne sont pas vraiment indépendantes, vu que l'on privilégiait toujours une trace qui connectait d'abord les sommets de son sous-arbre dans les sous-arbres des arêtes filles, on a donc soit p positif ou nul, soit q positif ou nul. En utilisant un codage différent pour les traces, il n'est pas non plus nécessaire d'enraciner l'arbre en la source, il peut être enraciné en n'importe quel sommet.

Ainsi le codage pour le cas $h = 2$ serait :

- un flot f_1 dont le signe dépend de la présence de la source dans le sous-arbre. Si la source est dans le sous-arbre, f_1 représente le nombre de sommets qui peuvent être reliés à la source tout en respectant la charge. Si la source n'est pas dans le sous-arbre, f_1 représente le nombre de sommets reliés à la source dans le sous-arbre.
- Un flot f_2 comptant soit combien de sommets sortent du sous-arbre et doivent être connectés à des sommets à distance 1 de la source (f_2 sera négatif est correspondra à $-q$ dans le codage précédent), soit combien de sommets à l'extérieur du sous-arbre peuvent être connectés dans le sous-arbre à un sommet à distance 1 de la source (f_2 sera positif est correspondra à p dans le codage précédent) et cela tout en respectant la charge maximale. La présence de la source dans le sous-arbre n'influe pas sur cette variable.

On appellera la variable f_d le *flot à distance d* .

Dans le cas de h sauts nous codons une trace en utilisant h variables f_1, \dots, f_h , où f_d correspond au flot à distance d de la source dans le sous-arbre. Lorsque f_d est positif cela signifie qu'on doit connecter $|f_d|$ sommets extérieurs au sous-arbre qui sont à distance d de la source à un sommet à distance $d - 1$ dans le sous-arbre. Lorsque f_d est négatif cela signifie qu'il existe $|f_d|$ sommets dans le sous-arbre qui sont à distance d de la source et qui doivent être connectés à un sommet à distance $d - 1$ à l'extérieur du sous-arbre. On peut déjà remarquer que la charge sur l'arête e est définie grâce à ces variables, c'est simplement la somme des flots $\sum_{1 \leq d \leq h} |f_d|$ (en valeur absolue car on étudie ici le cas où les arêtes virtuelles sont non orientées).

REMARQUE. Au lieu d'affecter les traces à une arête e , nous les affecterons au sommet $v(e)$ correspondant (pour pouvoir remonter jusqu'à la racine de l'arbre qui n'est plus nécessairement la source de la diffusion) et on notera $e(v)$ l'arête e

si elle existe (si v n'est pas la racine de l'arbre).

Le calcul d'une trace t d'un sommet v au moyen des traces de ses fils s'effectue ainsi : la distance du sommet v à la source peut prendre les valeurs de 0 (0 si c'est la source) jusqu'à h , pour chacune de ces valeurs d_0 , on génère alors toutes les traces t vérifiant les contraintes suivantes à partir des combinaisons (t_1, t_2, \dots, t_k) des traces des fils :

- la contrainte de capacité sur l'arête $e(v)$ impose que la somme des flots qui passent sur l'arête est inférieur à la charge maximale c :

$$\sum_{1 \leq d \leq h} |f_d| \leq c \quad (4.1)$$

- le sommet v ne permet pas de connecter plus de sommets à distance inférieure à d_0 que n'en permettraient les sous-arbres fils :

$$\forall 1 \leq d \leq d_0 - 1, t.f_d \leq \sum_i t_i.f_d \quad (4.2)$$

- si le sommet v n'est pas la source, par hypothèse du choix de d_0 , au flot à distance d_0 issu des sous-arbres fils s'ajoute une unité de flot issue du sommet v :

$$t.f_{d_0} \leq \sum_i t_i.f_{d_0} - 1 \quad (4.3)$$

- si le sommet v n'est pas la source, étant à distance d_0 de la source, il peut absorber tous les flots des sommets à distance supérieure ou égale à $d_0 + 1$ qui sont dans le sous-arbre :

$$\forall d \geq d_0 + 1, 0 \leq t.f_d \quad (4.4)$$

- si le sommet v est la source, tous les sommets du sous-arbre qui devaient être connectés peuvent être connectés à la source :

$$t.f_1 = c \text{ et } \forall 2 \leq d \leq h, t.f_d = 0 \quad (4.5)$$

L'initialisation se fait au niveau des arêtes incidentes aux feuilles de l'arbre, si la feuille est la source, on a pour $2 \leq d \leq h, t.f_d = 0$ et $t.f_1 = c - 1$ et si la feuille est à distance d_0 de la source, on a $t.f_{d_0} = -1, t.f_{d_0+1} = c - 1$ et pour $d \neq d_0, d \neq d_0 + 1, t.f_d = 0$, ce qui fait au plus une trace par valeur de d soit h traces.

L'existence d'une trace ne nécessitant pas de connecter des sommets (c'est à dire $f_d \geq 0$ pour $0 \leq d \leq h$) pour la racine de l'arbre est nécessaire est suffisante pour qu'un VPL en h sauts et de charge maximale c existe. Notons aussi que comme dans le cas de 2 sauts il est possible de combiner les traces filles deux à deux, afin de diminuer la complexité.

Étude de l'algorithme

Complexité en temps

Tout d'abord nous allons combiner les tables comme lors de la démonstration de la Proposition 4.3.2, en combinant le traces des tables des deux premiers fils pour obtenir une table temporaire que l'on combine avec la table du troisième fils et ainsi de suite.

Soit t le nombre maximal de traces vérifiant les contraintes (4.1), (4.2), (4.3) et (4.4) dans une table. Lorsqu'on combine deux tables filles de v , on effectue au plus t^2 combinaisons pour générer une table temporaire, mais dans cette table il se peut que les traces ne vérifient pas les contraintes (par exemple si le flot de chacune des traces combinées est égal à c mais sera absorbée par d'autres traces filles). Nous allons donc simplement majorer t par le nombre de h -uplets de variables comprises entre $-c$ et c , c'est à dire $(2c+1)^h$ (que nous approximerons par $(2c)^h$ pour les calculs).

Lors de la première combinaison, la valeur de chacune des variables d'un trace obtenue est comprise entre $-2c$ et $2c$ ce qui fait au plus $(2 \times 2c)^h$ traces distinctes. Il faut ensuite combiner ces traces avec celle du troisième fils de v ce qui fait au plus $(4c)^h \times (2c)^h$ combinaisons et $(2 \times 3c)^h$ traces distinctes. Ainsi de suite, lors du traitement du i -ème fils, on combine un table de $(2 \times (i-1)c)^h$ traces avec une trace de $(2c)^h$ traces ce qui fait $(i-1)^h (2c)^{2h}$ combinaisons pour obtenir une table de $(2ic)^h$ traces. La complexité en temps de la combinaison des traces filles de cette manière est au plus de $(2c)^{2h} \sum_{i=2}^{\Delta-1} (i-1)^h = (\Delta^h (2c)^{2h})$. On obtient une table ayant au plus $(2(\Delta-1)c)^h$ traces.

Pour chacune de traces obtenues, il faut décider de la distance du sommet v (h possibilités) et pour chacune de ces valeurs, il faut générer t traces. Nous allons majorer t par le nombre de combinaisons respectant la contrainte (4.1). On peut majorer le nombre de h -uplets d'entiers compris entre $-c$ et c vérifiant (4.1) ainsi : sur une ligne à $c+h$ points, on en choisit h qui délimitent les valeurs des différentes variables, le nombre de points qui ne sont pas des délimiteurs et qui sont avant le dernier délimiteur et bien inférieur à c quelle que soit la combinaison choisie. Chaque variable pouvant prendre une valeur négative ou positive, chaque combinaison en engendre 2^h (si on considère que 0 peut être positif ou négatif), ce qui fait au total $2^h \binom{c+h}{c}$ combinaisons. Ce qui fait pour chaque trace traitée une complexité dans le pire des cas en $O(h 2^h \binom{c+h}{c})$.

La complexité en temps de l'algorithme est donc dans le pire des cas en

$$\begin{aligned}
 & O \left(|V| \left(\Delta^h (2c)^{2h} + (2(\Delta-1)c)^h h 2^h \frac{(c+h)^h}{h!} \right) \right) = \\
 & O \left(|V| \Delta^h 2^{2h} c^h \left(c^h + h \frac{(c+h)^h}{h!} \right) \right)
 \end{aligned}$$

ou si on majore simplement t par $(2c)^h$,

$$O(|V|(\Delta^h(2c)^{2h} + (2(\Delta - 1)c)^h h(2c)^h)) = O(|V|\Delta^h h(2c)^{2h})$$

Complexité en espace

Comme dans le cas de 2 sauts, si $f(i)$ est la complexité en espace pour calculer la table d'un sommet à distance i de r , il suffit d'avoir en mémoire la combinaison des premières tables filles (de taille au plus $(2(\Delta - 1)c)^h$) et calculer la table de la dernière, soit une complexité de $(2(\Delta - 1)c)^h + f(i - 1)$. Si D est le diamètre de l'arbre, la racine étant choisie d'excentricité minimale, la complexité en espace de l'algorithme est donc dans le pire des cas en

$$O(D(2\Delta c)^h)$$

□

Comment construire une solution

Comme dans le cas où $h = 2$, en conservant plus d'information dans les traces, il est possible de construire une solution optimale si elle existe. Il suffit de conserver h ensembles de la forme P_i où P_i représente l'ensemble des sommets à distance i d'une traces qui doivent connecter des sommets à distance $i + 1$ et h ensembles Q_i où Q_i représente l'ensemble des sommets à distance i du sous-arbre qu'il faut connecter à des sommets à distance $i - 1$ à l'extérieur du sous-arbre. Avec les notations précédentes, f_i correspond donc à $|P_{i-1}|$ si f_i est positif et $-|Q_i|$ si f_i est négatif. Chaque trace sera donc de la forme $(P_0, \dots, P_{h-1}, Q_1, \dots, Q_h, E)$ donc de taille de l'ordre de $O(2ch + |V|)$. L'algorithme 7 décrit la combinaison de deux tables avec ce codage.

L'algorithme permettant la résolution du problème ainsi que la construction de la solution est donné de manière explicite en annexes.

REMARQUE. Comme dans le cas de 2 sauts, l'algorithme utilisé pour résoudre le problème de la charge pour la diffusion dans l'arbre peut s'adapter facilement pour l'instance de communication *One - To - Many* en ajoutant la contrainte suivante lors du calcul de la trace d'un sommet s'il n'est pas destination d'une requête : le sommet peut être à distance d avec d entre 1 et h ou à distance ∞ auquel cas il ne sera pas dans le graphe virtuel.

Théorème 4.4.3 *Il existe un algorithme polynomial en temps et en mémoire pour la résolution du problème de minimisation de la charge avec h fixé (problème C) et la construction d'une solution optimale pour les arbres pour toute instance de communication de type One-To-Many.*

PREUVE. Nous savons d'après 2.2.4 que $\pi(T, OM, h) \leq h\pi(T, OM, 1)^{\frac{1}{h}} - h + 1$. La complexité de l'algorithme pour la résolution du problème de la charge avec c et h donné étant $O(|V|\Delta^h h(2c)^{2h})$, en l'appliquant au plus $\log(h|V|^{1/h})$ (en faisant varier c de manière dichotomique), on obtient une complexité en $O(\log(h|V|^{1/h})|V|\Delta^h h(2h|V|^{1/h})^{2h}) = O(|V|^3 \log(h|V|^{1/h})\Delta^h h(2h)^{2h})$ et donc un algorithme polynomial en $|V|$. \square

Utilisation de la programmation linéaire pour la résolution du problème de minimisation de la charge dans l'arbre avec une source s et au plus h sauts.

Bien que ce problème soit polynomial et que nous avons un algorithme pour le résoudre, nous donnons une formulation en programme linéaire en nombre entier (dont la résolution n'est à priori pas effectuée en temps polynomial) pour pouvoir introduire le cas d'une instance quelconque dont la résolution sera une extension de cette formulation.

Le programme linéaire 1 détermine un graphe $H = (V, E')$ dont le plongement (unique car le graphe physique est un arbre) donne une charge minimale. Rappelons que lorsqu'on note une paire (ensemble de deux éléments non ordonnés) $[x, y]$ et un couple (ensemble de deux éléments ordonnés) (x, y) . On note $\mathcal{P}([u, v])$ le chemin dans le graphe physique correspondant à l'arête virtuelle $[u, v]$.

Programme Linéaire 1 Résolution du problème de minimisation de la charge dans l'arbre avec 1 source et h sauts

Description des variables :

$$\begin{aligned}
X[a, b] &= \begin{cases} \text{indicateur de présence d'une arête virtuelle} \\ 1 \text{ si } [a, b] \in E' \\ 0 \text{ sinon} \end{cases} \\
X^i(a, b) &= \begin{cases} \text{propagation du flot issu de la source} \\ > 0 \text{ si } [a, b] \in E', \text{ } dist_H(s, b) = i \text{ et } dist_H(s, a) = i - 1 \text{ avec } s \in S \\ = 0 \text{ sinon} \end{cases} \\
L_{[u, v]} &= \text{charge sur l'arête physique } [u, v] \\
L_{max} &= \text{charge maximale}
\end{aligned}$$

Minimiser : L_{max}

Avec les contraintes :

$$X^i(a, b) \leq \sum_{x \in V \setminus \{b\}} X^{i-1}(x, a), \forall (a, b) \in V \times V, \forall 2 \leq i \leq h \quad (4.6)$$

$$1 \leq \sum_{a \in V \setminus \{b\}, 1 \leq i \leq h} X^i(a, b), \forall b \in V \setminus \{s\} \quad (4.7)$$

$$X[a, b] = \sum_{1 \leq i \leq h} (X^i(a, b) + X^i(b, a)), \forall [a, b] \in V \times V. \quad (4.8)$$

$$L_{[u, v]} \geq \sum_{[a, b] \in V \times V \mid [u, v] \in \mathcal{P}([a, b])} X[a, b], \forall [u, v] \in E \quad (4.9)$$

$$L_{max} \geq L_{[u, v]} \forall [u, v] \in E \quad (4.10)$$

$$X^1(a, b) = 0, \forall a \neq s, \forall b \in V \quad (4.11)$$

$$X[a, b] \in \{0, 1\} \quad (4.12)$$

Nombre de variables :

de type $X[a, b]$: $\frac{|V|(|V|-1)}{2}$

de type $X^i(a, b)$: $h|V|(|V|-1)$

de type $L[u, v]$: $|E|$

Soit $O(h|V|^2)$ variables fractionnaires et $O(|V|^2)$ variables binaires.

Nombre de contraintes :

de type 4.8 : $\frac{|V|(|V|-1)}{2}$

de type 4.6 : $(h-1)|V|(|V|-1)$

de type 4.7 : $|V|$

de type 4.9 : $\frac{|V|(|V|-1)}{2}$

de type 4.10 : $|E|$

Soit $O(h|V|^2)$ contraintes.

Les contraintes 4.6 garantissent que si un sommet est à distance i de la source, il a bien un prédecesseur à distance $i - 1$ de la source. Les contraintes 4.7 garantissent que les sommets seront à distance au plus h de la source. Les contraintes 4.8 permettent de déterminer si l'arête $[a, b]$ existe ou non. Les contraintes 4.9 permettent de comptabiliser la charge sur les arêtes physiques en ne comptant que les arêtes virtuelles donc le chemin associé passe par l'arête physique en question. Les contraintes 4.10 permettent de déterminer la charge maximale.

La figure 4.7 représente un arbre physique et la figure 4.8 un graphe virtuel sur l'arbre de charge minimale (2) trouvé par le programme linéaire 1 avec un nombre de sauts maximal égal à 3. On peut remarquer que le graphe virtuel contient des cycles, ce qui est inutile vu qu'un arbre de diffusion suffirait, mais les arêtes qui forment ces cycles n'induisent pas de charge supplémentaire.

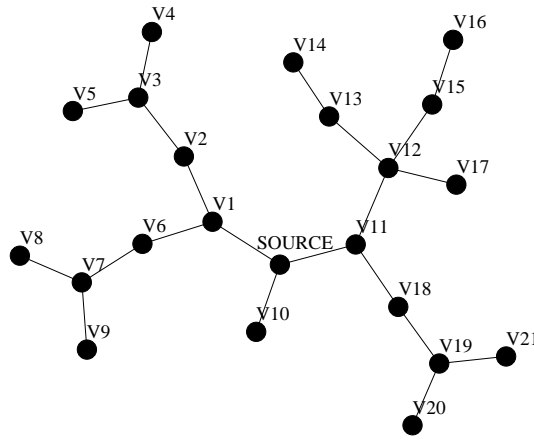


FIG. 4.7 – Un arbre dont on cherche un VPL de charge minimale avec un nombre de sauts maximal égal à 3

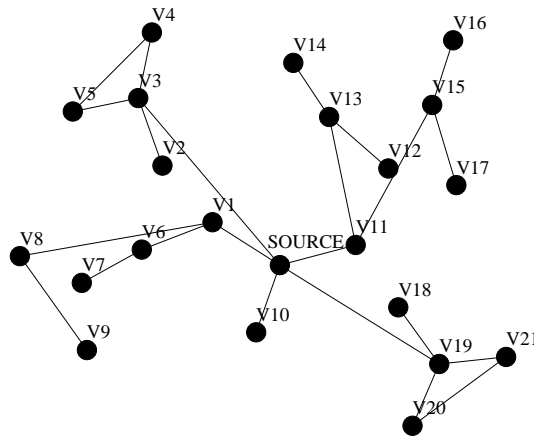


FIG. 4.8 – Exemple de solution trouvée par le programme linéaire 1

Comparaison de l'algorithme avec la programmation linéaire

Le tableau de la figure 4.4 donne une idée du temps de calcul nécessaire pour calculer la charge optimale d'une diffusion en h sauts dans divers arbres ; $|V|$ est le nombre de sommets, Δ le degré du maximal de l'arbre. Ces calculs ont été réalisés sur des Pentium III 933Mhz (512 Mo de mémoire RAM), le logiciel utilisé pour résoudre les programmes linéaire était CPLEX 7.5 (ILOG).

$ V $	Δ	profondeur	h	Tps CPU PL	Tps CPU Algo	π
12 (chaîne)	2	11	3	1.33s	0.24s	2
20 (chaîne)	2	19	3	87s	0.56s	3
39	6	5	2	1123s	0.32s	5
3071	7	6	4	?? siècles	611s	4

FIG. 4.9 – Comparaison programmation linéaire / algorithme

Dans le programme linéaire, le problème de flot à résoudre est dans le graphe déterminé par les variables binaires (qui correspondent à la présence ou non des liens virtuels), ce qui explique pourquoi le programme linéaire n'est pas performant.

4.5 2-to-Many en 2 sauts

Nous commençons par donner la solution du problème pour une instance de communication de type 2-to-All, une instance avec deux sources, et un nombre maximum de sauts égal à 2, ce qui impose que chacune des sources soit d'excentricité au plus 2 dans le graphe virtuel.

Théorème 4.5.1 *Le problème de décision de l'existence d'un VPL pour un nombre de sauts maximal $h = 2$ et une charge c fixée est polynomial en temps et en mémoire dans les arbres pour toute instance de communication de type 2-to-Many.*

PREUVE. Nous utilisons toujours la technique de la programmation dynamique. Cependant le fait d'avoir deux sources complexifie le problème, notamment les notions sur les ensembles de type P et Q . Tout d'abord classifions les sommets en différents ensembles en fonction de leur distance aux sources notées s_1 et s_2 : On note V_{d_1, d_2} l'ensemble de sommet à distance d_1 de s_1 dans le graphe virtuel H et à distance d_2 de s_2 dans H :

$$V_{d_1, d_2} = \{x \in V \mid d_H(x, s_1) = d_1, d_H(x, s_2) = d_2\}$$

Si on veut calculer les traces comme nous le faisons précédemment, nous allons garder en mémoire un certain nombre de sommets qui permettent d'établir une

connexion dans le sous-arbre, mais pour les sommets nécessitant d'être connectés il faut faire attention car un sommet de $V_{2,2}$ pourra être connecté de deux manière différentes, en le connectant à un sommet de $V_{1,1}$ ou à deux sommets différents, l'un dans $V_{1,2}$ et l'autre dans $V_{2,1}$. Donc caractériser les sommets qu'il reste à connecter avec les ensembles $V_{i,j}$ ne permet pas de déterminer la charge imposée par les connexions à établir, nous procéderons un peu différemment.

Pour chaque trace, on notera f_{d_1,d_2} le flot allant vers un sommet à distance d_1 de s_1 et à distance d_2 de s_2 . f_{d_1,d_2} sera positif si le flot est sortant du sous-arbre (si le sommet à distance d_1 et d_2 est dans le sous-arbre) et négatif sinon.

La combinaison des tables des fils de v (avant le choix des distances sur v) est similaire au cas d'une source. Comme il y a 9 variables f ($f_{0,1}$, $f_{0,2}$, $f_{1,0}$, $f_{2,0}$, $f_{1,1}$, $f_{1,2}$, $f_{2,1}$, $f_{0,0}$ et $f_{2,2}$) et que chacune des variables est comprise entre $-c$ et c , le nombre de traces distinctes pour un sommet est au plus $(2c)^9$.

Lors de la combinaison des tables des fils, comme dans le cas de une source (preuve de la proposition 4.4.2), la taille de la table temporaire augmente au fur et à mesure de la progression vers le dernier fils. On obtient de manière similaire une complexité en temps de $O(\Delta^9(2c)^{18}) = O(\Delta^9c^{18})$ pour la combinaison des tables. On obtient une table temporaire de $(2\Delta c)^9$ traces.

Une fois que les tables des fils de v sont combinées en une table temporaire ayant au plus $(2\Delta c)^9$ traces, nous considérerons les cas suivants :

- si v est la source s_1 (idem pour s_2), il peut être à distance 1 ou 2 de s_2 , on génère alors deux types de traces :
 - $f_{1,0} \leftarrow f_{1,0} - 1$ si on est dans le cas $d(v, s_2) = 1$,
 - $f_{1,1} \leftarrow f_{1,1} - 1$ si on est dans le cas $d(v, s_2) = 2$,
- si v est à distance 1 de s_1 et 1 de s_2 , il faut aussi distinguer deux cas : si s_1 et s_2 sont reliés ou non, ce qui fait deux types de traces :
 - $f_{1,0} \leftarrow f_{1,0} - 1$ et $f_{0,1} \leftarrow f_{0,1} - 1$,
 - $f_{2,0} \leftarrow f_{2,0} - 1$ et $f_{0,2} \leftarrow f_{0,2} - 1$,
- si v est à distance 1 de s_1 et 2 de s_2 , soit v est relié à un sommet de $V_{0,1}$ soit il est relié à deux sommets : un de $V_{0,2}$ et un de $V_{1,1}$, on génère donc deux types de traces
 - $f_{0,1} \leftarrow f_{0,1} - 1$ contient v ,
 - $f_{0,2} \leftarrow f_{0,2} - 1$ et $f_{1,1} \leftarrow f_{1,1} - 1$,
- si v est à distance 2 de s_1 et 2 de s_2 , soit v est voisin d'un sommet de $V_{1,1}$ soit il est voisin de deux sommets l'un de $V_{1,2}$ et l'autre de $V_{2,1}$. On génère donc deux type de traces :

- $f_{1,1} \leftarrow f_{1,1} - 1$,
- $f_{1,2} \leftarrow f_{1,2} - 1$ et $f_{2,1} \leftarrow f_{2,1} - 1$.

Il y a donc 8 types de traces engendrées. Pour chaque trace générée, si i (resp. j) est la distance de v à s_1 (resp. s_2), on pourra connecter à v tous les sommets du sous-arbre qui doivent être connecter à un sommet de ce type et v pourra connecter autant de sommet que la charge le permet :

$$f_{i,j} \leftarrow c - \sum_{0 \leq d_1, d_2 \leq 2 | f_{d_1, d_2} < 0} |f_{d_1, d_2}|$$

La contrainte de la charge sur l'arête dont v est le fils s'exprime ainsi :

$$\sum_{0 \leq d_1, d_2 \leq 2} |f_{d_1, d_2}| \leq c$$

Comme dans le cas de h sauts, pour chaque type de trace il faudra considérer toutes les valeurs possibles des variables de type p qui verifient cette contrainte. Il y a 9 variables de types f , pour chaque type de traces le nombre de traces générées sera au maximum le nombre de combinaisons d'entiers $(x_i)_{1 \leq i \leq 9}$ telles que $0 \leq x_i \leq c$ et $\sum_i x_i \leq c$ soit $2^9 \binom{c+9}{9}$. Finalement pour une trace issue des combinaisons des tables des fils de v , le nombre des traces générées pour v sera au pire $8 \times 2^9 \binom{c+9}{9} = O(c^9)$.

La complexité totale en temps est donc

$$O(|V|(\Delta^9 c^{18} + (2\Delta c)^9 c^9)) = O(|V|\Delta^9 c^{18})$$

□

4.6 Instance quelconque en h sauts

Définition 4.6.1 Soit $G = (V, E)$ un graphe non orienté, un **vertex cover** est un sous-ensemble S de V tel que toute arête de E est incidente à un sommet de S .

Théorème 4.6.1 Le problème de décision de l'existence d'un VPL pour un nombre de sauts maximal h et une charge c fixés est polynomial en temps et en mémoire dans les arbres de degré borné pour toute instance de communication de type *Many – to – Many* si la taille du vertex cover du graphe des requêtes est borné.

PREUVE. Soit S un sous ensemble de V tel que chacune des arêtes du graphe des requêtes défini par l'instance de communication est incidente à un sommet

S . Prendre S de taille minimal correspond à résoudre le problème de «Minimal Vertex cover» qui est \mathcal{NP} -complet [GJ79]. Il existe un algorithme polynomial simple pour trouver une 2-approximation de ce problème : on prend une arête quelconque et on met les deux sommets incidents dans S , ensuite on enlève toutes les arêtes couvertes par l'un de ces deux sommets et on recommence jusqu'à ce qu'il n'y ai plus d'arêtes.

On peut procéder de la même manière que pour l'instance $2 - To - Many$, les ensembles de type V_{d_1, d_2} du cas $2 - to - Many$ seront de la forme $V_{d_1, \dots, d_{|S|}}$, on a donc $|S|^{h+1}$ variables de la forme $f_{d_1, \dots, d_{|S|}}$, chacune étant comprise entre $-c$ et c , le nombre de traces distinctes est au plus $(2c)^{|S|^{h+1}}$. La combinaison des tables des fils s'effectue donc en $O\left((2\Delta c)^{|S|^{h+1}}\right)$ qui est indépendant de $|V|$. Le choix des distances d'un sommet lors du calcul de ses traces va engendrer, pour une trace donnée, un nombre borné par une fonction de c et h , indépendante de $|V|$.

Le calcul des traces d'un sommet reste donc une fonction uniquement dépendante de c et h , la résolution du problème peut donc être fait grâce à un algorithme polynomial en $|V|$ si $|S|$ est borné.

Pour la complexité en mémoire, c'est toujours polynomial car nous ne gardons en mémoire que les tables des fils d'un sommet, donc la complexité en mémoire est liée au degré maximal de l'arbre, au nombre de sources et au nombre de sauts et non aux nombre de sommets total. \square

Utilisation de la programmation linéaire pour la résolution du problème de minimisation de la charge dans l'arbre pour une instance de communication quelconque et au plus h sauts.

Soit I une instance de communications quelconque. Soit S un sous ensemble de V correspondant à une 2-approximation du problème du «minimum vertex cover» sur le graphe des requêtes. On peut utiliser la programmation linéaire pour résoudre le problème de manière similaire à celui donné dans la section 4.4. Le programme est donné dans l'annexe B.2. Rappelons que ce programme ne peut fonctionner que dans les arbres car le plongement d'une arête virtuelle étant déterminé, le problème de décision se réduit à la présence ou non des arêtes virtuelles.

4.7 Echange total en 3 sauts

Conjecture 4.7.1 *Dans le cas où le nombre de sauts maximal est 3, nous conjecturons qu'il existe une solution ayant une charge d'au plus $\pi(T, AA, 3) + O(1)$ formée par un graphe virtuel contenant une clique (i.e. : un sous-graphe complet, nous appellerons les sommets de cette clique des serveurs), et des arêtes reliant tout sommet qui n'est pas un serveur à un serveur.*

Si cette conjecture est vraie, la construction de ce graphe virtuel peut se faire en temps polynomial.

Chapitre 5

Extension optique du VPL

Dans le cadre des réseaux WDM, étant donné un ensemble de requêtes, on cherche généralement à les réaliser en utilisant des chemins dans le réseau, chacun de ces chemins utilisant une longueur d'onde particulière et deux chemins utilisant le même lien ne pouvant pas utiliser la même longueur d'onde. Ce problème est appelé *problème du routage optique*. Les commutateurs tout-optiques n'étant pas encore sur le marché et afin d'éviter de convertir les signaux optiques en signaux électroniques (ce qui entraîne une dégradation importante du débit de communication), on peut se restreindre au cas où les conversions optiques sont interdites dans le réseau. Sous ces conditions, chaque communication ne peut utiliser qu'une seule longueur d'onde. La thèse de Bruno Beauquier [Bea00] étudie en détails ce problème. On parlera alors de routage optique sans conversion.

5.1 Le routage optique sans conversion

PROBLÈME du Routage Optique sans conversion[Bea00]

Entrée : un (multi)graphe G et une instance I de requêtes dans G

Sortie : une allocation de chemins et de couleurs aux requêtes, telle que deux chemins utilisant le même lien ont deux couleurs différentes

Objectif : minimiser le nombre de couleurs utilisées

On note $w(G, I)$ le nombre de couleurs d'une solution optimale si le graphe G est non-orienté, et $\vec{w}(G, I)$ si le graphe G est orienté.

De manière évidente, la charge optimale d'un routage réalisant une instance donnée minore le nombre optimal de longueurs d'onde pour le problème du routage optique sans conversion appliqué à la même instance. Ceci résulte immédiatement du fait que le problème de la charge est un problème d'allocation de chemins et de longueurs d'onde, avec la liberté supplémentaire de la conversion.

Propriété 5.1.1 *Pour tout graphe G et instance I , on a $w(G, I) \geq \pi(G, I)$, .*

La propriété analogue est vraie dans le cas orienté. En général il n'y a pas égalité, mais il existe des réseaux particuliers avec des instances particulières pour lesquels $w(G, I) = \pi(G, I)$.

Théorème 5.1.2 (Beauquier, Hell et Pérennes [BHP98]) *Pour tout graphe orienté G et toute instance de multicast I_M , le problème de routage optique (G, I_M) peut être résolu de façon optimale en temps polynômial, et de plus,*

$$\vec{w}(G, I_M) = \vec{\pi}(G, I_M) .$$

Le cas non orienté se déduit de ce théorème :

Corollaire 5.1.3 *Pour tout graphe non-orienté G et toute instance de multicast I_M , le problème de routage optique (G, I_M) peut être résolu de façon optimale en temps polynômial, et de plus,*

$$w(G, I_M) = \pi(G, I_M)$$

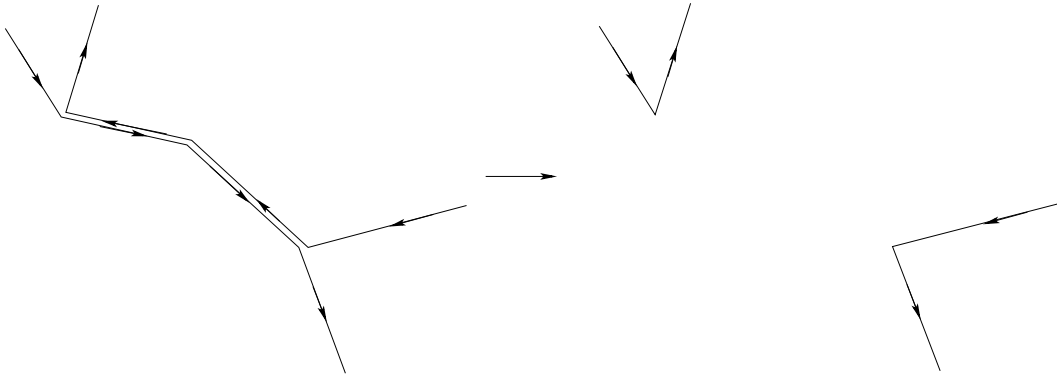
PREUVE. Dans le cas orienté, une solution avec des chemins qui se croisent peut toujours être transformée en une autre solution où ces chemins ne se croisent pas : supposons que le croisement entre le chemin

$$P = (x_0, \dots, x_{i-1}, u, x_{i+1}, \dots, x_{j-1}, v, x_{j+1}, \dots, x_n)$$

et le chemin

$$P' = (x'_0, \dots, x'_{i'-1}, v, x'_{i'+1}, \dots, x'_{j'-1}, u, x'_{j'+1}, \dots, x'_{n'})$$

commence en u et se termine en v , on peut remplacer P par le chemin $(x_0, \dots, x_{i-1}, u, x'_{i'+1}, \dots, x'_{n'})$ et P' par le chemin $(x'_0, \dots, x'_{i'-1}, v, x_{j+1}, \dots, x_n)$.



Ainsi le théorème 5.1.2 est aussi valable dans le cas non orienté. Une solution pour le cas non orienté consistera à prendre une solution (ne comportant pas deux chemins qui se croisent) pour la même instance sur le graphe orienté symétrique obtenu en remplaçant les arêtes par deux arcs symétriques. \square

Pour l'instance *All – to – All** (un requête pour chaque paire de sommets), on ne sait toujours pas si $\vec{w}(G^*, AA^*) = \vec{\pi}(G^*, AA^*)$, pour de nombreux réseaux il y a égalité, par exemple $\vec{w}(C_n^*, AA^*) = \vec{\pi}(C_n^*, AA^*)$ d'après [BGP⁺00]. Dans le cas non orienté, on n'a pas, en général, égalité, par exemple pour le graphe G de la figure 5.1, $w(G, AA) \neq \pi(G, AA)$; pour le cycle en revanche on a égalité : $w(C_n, AA) = \pi(C_n, AA)$ d'après [BCML02, BCCT01].

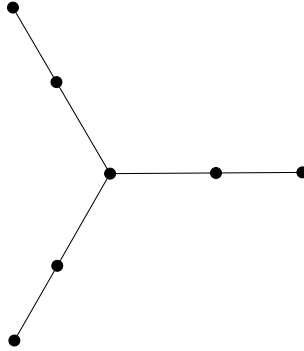


FIG. 5.1 – Exemple de graphe pour lequel $w(G, AA) \neq \pi(G, AA)$

5.2 Le modèle du VPL optique

Nous proposons ici un modèle mélangeant le VPL et le routage optique : étant donné un réseau physique, une instance de communication et un nombre de sauts maximal, le but est de trouver un graphe virtuel et un plongement optique sans conversion (à chaque arête du graphe virtuel est associé un chemin dans le réseau physique et une longueur d'onde) de ce graphe virtuel minimisant le nombre de longueurs d'onde utilisées par ce plongement. Nous pouvons dériver des définitions et problèmes posés dans le cadre du VPL les problèmes suivants :

Définition 5.2.1 Soit $G = (V, E)$, $H = (V, E')$ et $P : E' \rightarrow \mathcal{P}(G) \times \mathcal{C}$, où \mathcal{C} est un ensemble de longueurs d'onde, le couple (H, P) est appelé **VPL optique** sur G .

Définition 5.2.2 On appelle **nombre chromatique du graphe** $G = (V, E)$ pour le VPL (H, P) sur G , le nombre de longueurs d'onde utilisées par P , on le note $w(G, H, P)$.

Définition 5.2.3 *Étant donné un graphe G , une instance de communication I dans G , un nombre de sauts maximal h et un nombre de longueurs d'onde maximal w , un VPL optique (H, P) sur G est dit (h, w) -réalisable si $D(G, H, I) \leq h$ et $w(G, H, P) \leq w$.*

Le problème de l'existence d'un VPL optique est donc :

PROBLÈME de l'existence d'un VPL optique

Données : un graphe G , une instance de communication I dans G , h et w deux entiers.

Question : existe-t-il un VPL optique (h, w) -réalisable sur le graphe G pour l'instance de communication I ?

Comme dans le cas du VPL classique, ce problème induit les deux problèmes duaux que sont la minimisation du nombre de sauts pour un nombre de longueurs d'onde donné et la minimisation du nombre de longueurs d'onde pour un nombre maximal de sauts donné.

PROBLÈME du Nombre de Sauts Minimal $Dw_0(G, H, I)$

Entrée : un graphe G , une instance I dans G et un entier w_0 .

Sortie : un VPL optique (H, P) (h, w_0) -réalisable sur le graphe G avec l'instance I .

Objectif : minimiser h .

PROBLÈME du nombre minimal de longueur d'onde $w(G, I, h_0)$

Entrée : un graphe G , une instance I dans G et un entier h_0 .

Sortie : un VPL optique (H, P) (h_0, w) -réalisable sur le graphe G avec l'instance I .

Objectif : minimiser w .

Propriété 5.2.1 *Comme dans le cas du routage optique sans conversion, on a :*

$$w(G, I, h) \geq \pi(G, I, h)$$

Comme dans le cas du routage optique sans conversion, il peut être intéressant de comparer la charge minimale avec le nombre minimal de longueurs d'onde.

Nous mettons ici en évidence deux contre-exemples, un montrant que le théorème 5.1.2 ne s'applique pas dans ce modèle et un autre confirmant que dans ce modèle, la charge minimale n'est en générale pas égale au nombre minimal de couleur pour une instance de communication de type échange total.

Proposition 5.2.2 *Il existe un arbre T tel que*

$$w(T, OA, 2) > \pi(T, OA, 2)$$

PREUVE. Dans le graphe de la figure 5.2, pour ne pas dépasser une charge de 3, il est nécessaire de connecter les sommets V_i à la source S (ou un de leur fils, mais autant connecter V_i). Il n'y a qu'une possibilité pour établir les arêtes virtuelles connectant les sommets restants en particulier le sommet x doit être connecté au sommet V_1 et donc les arêtes $[S, V_1]$, $[S, V_2]$, $[S, V_3]$ et $[x, V_1]$ ne peuvent pas être colorées avec trois couleurs sans conflits.

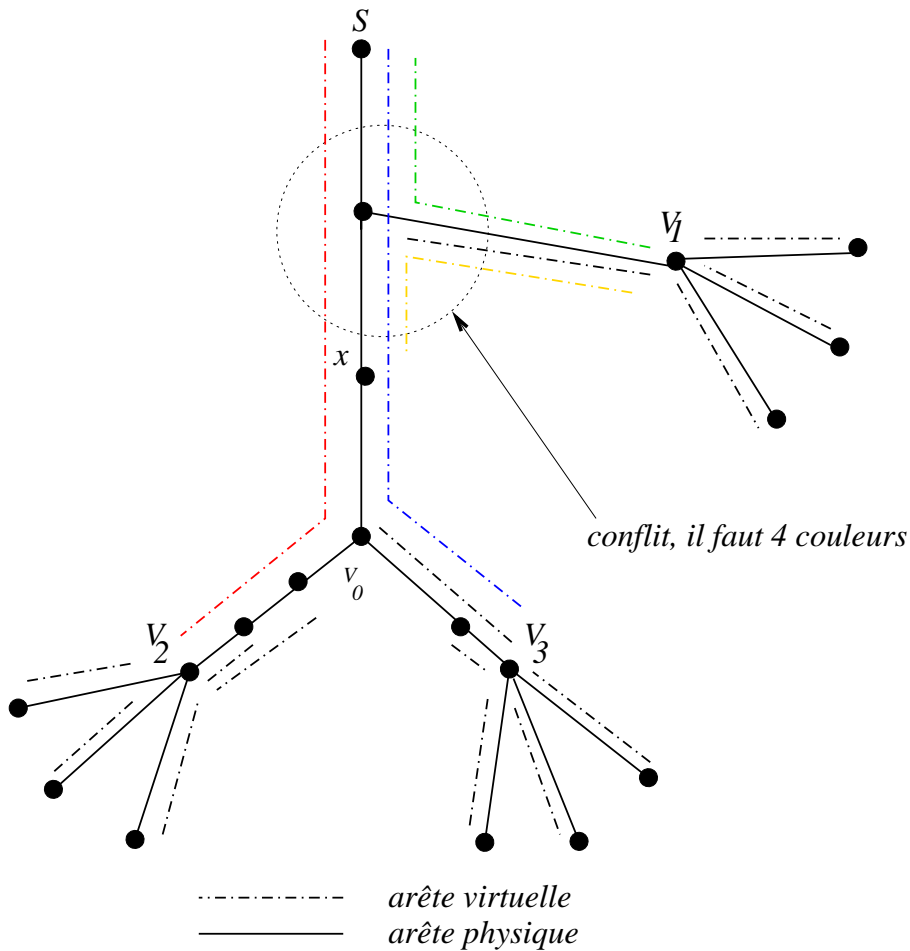


FIG. 5.2 – Contre exemple One-To-All dans l'arbre, $h = 2$, $\pi = 3$, $w = 4$

□

Proposition 5.2.3 *Il existe un arbre T tel que*

$$w(T, AA, 3) > \pi(T, AA, 3)$$

PREUVE. Comme dans la preuve de la proposition précédente, la construction du graphe virtuel sur le graphe de la figure 5.3 est forcée et induit une charge de $2k^2 + 1$ et un nombre de couleurs minimum de $3k^2$.

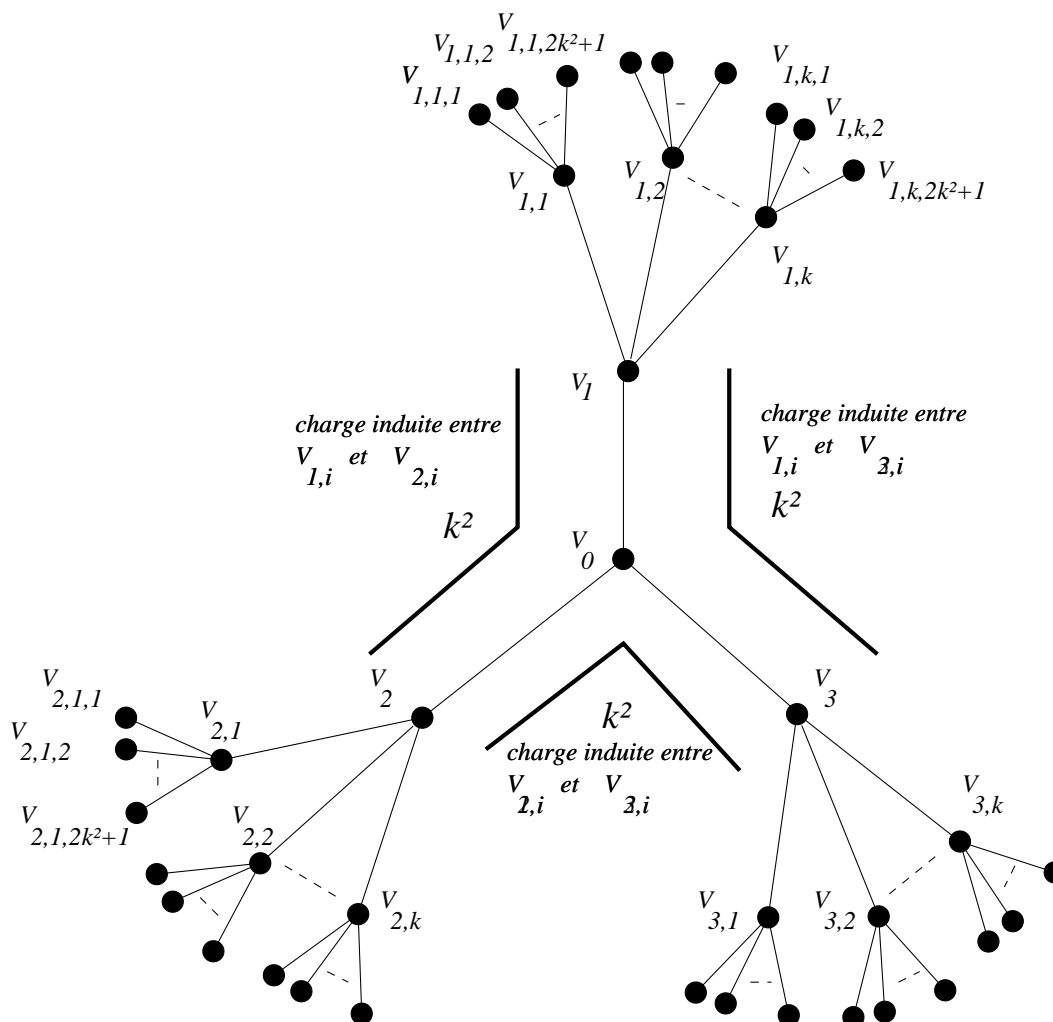


FIG. 5.3 – Contre exemple All-To-All, $h = 3$, $\pi = 2k^2 + 1$, $w = 3k^2$

□

Chapitre 6

Dimensionnement de réseaux hiérarchiques en anneaux

Nous nous intéressons ici à une topologie de réseau particulière pouvant être utilisée pour l'interconnexion de boucles SDH : les arbres de cycles hiérarchiques. Nous cherchons de telles structures ayant un grand nombre de sommets et un faible diamètre assurant ainsi un délai de transmission borné avec un grand nombre d'utilisateurs. Nous donnons une amélioration sur les résultats concernant la relation entre le diamètre minimum et le nombre de nœuds d'une telle structure.

6.1 Définitions

Définition 6.1.1 [ABC⁺ 99] *Un réseau hiérarchique de cycles, noté HRN (Hierarchical Ring Network), est obtenu à partir d'un cycle en ajoutant au plus un cycle subsidiaire à chaque nœud du cycle et, récursivement, à chaque nœud de chaque cycle subsidiaire ; chaque nœud ne pouvant appartenir qu'à au plus deux cycles.*

Définition 6.1.2 *La profondeur d'un HRN notée d est le niveau de récursion du HRN.*

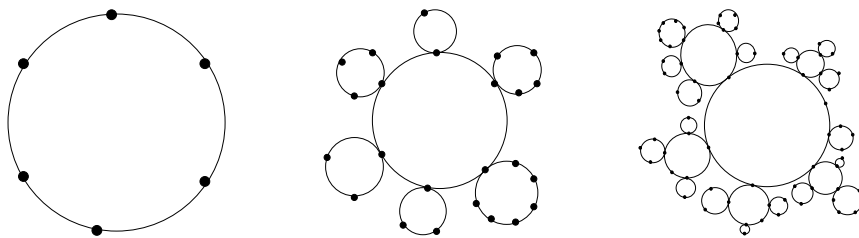


FIG. 6.1 – Exemples de HRN de profondeur 1, 2 et 3.

REMARQUE. Dans un HRN , les sommets sont dans au plus deux cycles élémentaires.

6.2 Résultats

Ce type de structure à été étudié par Aiello, Bhatt, Chung, Rosenberg et Sitaraman dans [ABC⁺99] qui ont donné une estimation du diamètre minimal $D_d^{HRN}(N)$ par rapport au nombre de sommets N et à la profondeur d :

$$\frac{1}{2} (N \times d!)^{1/d} (1 - o(1)) \leq D_d^{HRN}(N) \leq (N \times d!)^{1/d} (1 + o(1))$$

ils ont depuis amélioré ce résultat (qui nous a été communiqué par les auteurs) :

$$\left(\frac{1}{2}\right)^{2/d} (N \times d!)^{1/d} (1 - o(1)) \leq D_d^{HRN}(N) \leq (N \times d!)^{1/d} (1 + o(1))$$

Ces bornes sur le diamètre paraissent 'presque' serrées (vu que lorsque de d tend vers l'infini, $(1/2)^{2/d}$ tend vers 1), mais si on cherche à maximiser le nombre de sommets pour un diamètre donné, en renversant ces encadrements on obtient :

$$\begin{aligned} \frac{D^d}{d!} &\leq N_d(D) \leq 2^d \frac{D^d}{d!} \\ \frac{D^d}{d!} &\leq N_d(D) \leq 4 \frac{D^d}{d!} \end{aligned}$$

Dans [BCP02], nous avons cherché à améliorer ce résultat et nous avons obtenu un encadrement plus précis :

Théorème 6.2.1 (Bermond, Choplin, Pérennes)

$$N_d^{HRN}(D) = \frac{D^d}{d!} (1 + o(1))$$

Que l'on peut transformer en :

$$D_d^{HRN}(N) = (N \times d!)^{1/d} (1 + o(1))$$

6.3 Idée de preuve

Lorsqu'on construit un HRN de profondeur d à partir d'un HRN de profondeur $d - 1$, on ne peut pas ajouter des cycles que sur les sommets qui ont été ajoutés au niveau $d - 1$; cette contrainte rend le comptage des sommets difficile. Pour pouvoir déterminer le nombre de sommets d'une telle structure, nous introduisons un structure «plus récursive» que nous appelons HBN (pour *Hierarchical*

Bubble Network. Un *HBN* est similaire à un *HRN* avec la liberté supplémentaire de pouvoir ajouter des cycles en n'importe quel sommet d'un *HBN* de profondeur $d - 1$ pour construire un *HBN* de profondeur d .

Définition 6.3.1 On peut aussi définir un *HBN* ainsi :

- un *HBN* de profondeur 0 *HBN* est un sommet.
- un *HBN* de profondeur $d \geq 1$ est composé d'un cycle dit cycle principal et d'un ensemble fini de *HBN* (deux à deux disjoints) de profondeur $d - 1$ ayant un sommet de leur propre cycle principal commun avec le cycle principal.

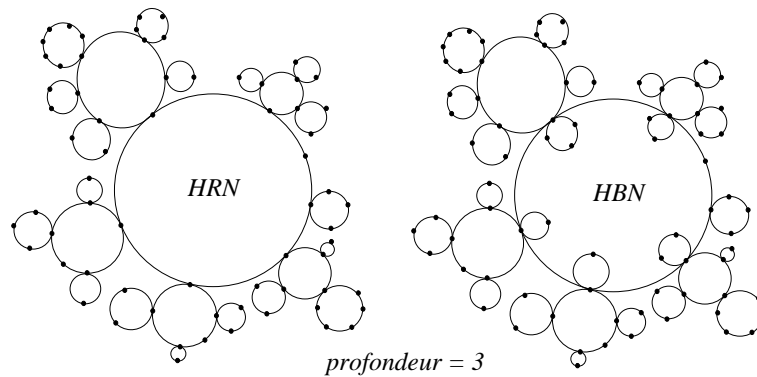


FIG. 6.2 – Pour une profondeur supérieure à 3, la structure d'un *HBN* peut être différente de celle d'un *HRN*.

REMARQUE. A la différence d'un *HRN*, dans un *HBN*, un sommet peut être dans plusieurs cycles élémentaires (voir la figure 6.2).

Nous parvenons à estimer le nombre de sommets maximum d'un *HBN* en fonction de la profondeur et du diamètre en déterminant la structure des solutions optimales. Nous montrons ensuite que ce nombre de sommets est du même ordre que le nombre maximum de sommets d'un *HRN* de même profondeur et même diamètre, ce qui nous permet d'améliorer le résultat de [ABC⁺99]. Nous n'avons cependant pas réussi à dégager une structure optimale pour les *HRN* qui d'après les expériences sont beaucoup moins régulières que celle des *HBN*.

6.4 Construire un *HRN* ayant un nombre maximal de sommets avec une profondeur et un diamètre donnés.

Pour construire un *HRN* optimal, nous allons le décomposer en deux structures de même type que nous appellerons *oreilles*.

Définition 6.4.1 On appelle une *oreille* de profondeur 0 un sommet et une oreille de profondeur $d \geq 1$ un sommet appelé **base de l'oreille** et un ensemble fini d'oreilles de profondeur $d - 1$ connectées en leur base par un cycle.

Un *HRN* de profondeur d peut être défini comme l'interconnexion d'oreilles de profondeur $d - 1$ en leur base par un cycle ou encore comme la connexion d'une oreille de profondeur d et une oreille de profondeur $d - 1$ en leur base.

Définition 6.4.2 On appelle **poids d'une oreille** l'excentricité de la base dans l'oreille.

Ainsi un *HRN* de diamètre D et de profondeur d peut être décomposé pour chaque sommet x du cycle principal en deux oreilles :

- une de base x , de poids e_1 et de profondeur d
- et une de base x , de poids e_2 et de profondeur $d - 1$

telles que $e_1 + e_2 \leq D$.

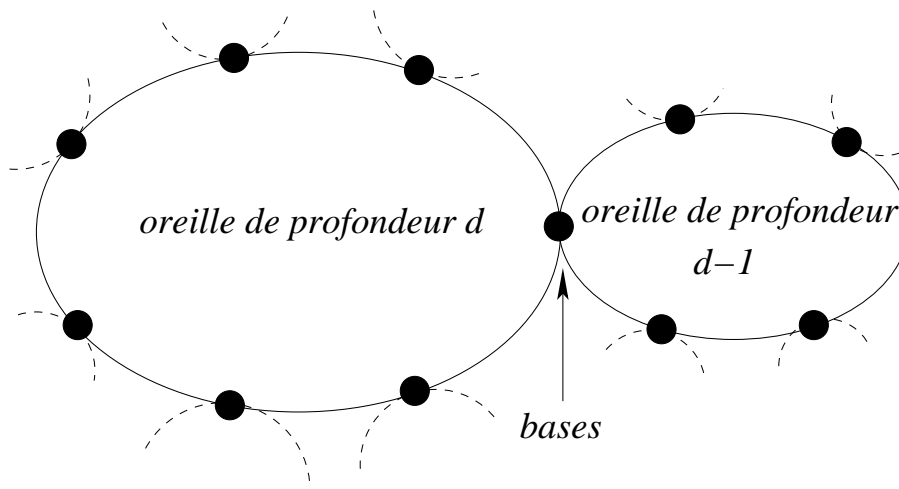


FIG. 6.3 – Décomposition d'un HRN en deux oreilles

NOTATION. On note $e(d, D, w)$ le nombre maximum de sommets d'une oreille de profondeur d de diamètre au plus D et de poids w .

On a alors

$$N_d(D) = \max_{0 \leq w \leq D} \{e(d, D, w) + e(d-1, D, D-w)\} - 1$$

6.4.1 Calcul de $e(d, D, w)$

Dans une oreille de profondeur d , les contraintes à satisfaire sont uniquement liées au diamètre et à l'excentricité de la base. Comme tout chemin entre deux sommets de deux oreilles différentes passe nécessairement par les bases de ces oreilles, on peut considérer les contraintes uniquement au niveau des bases des oreilles de profondeur $d-1$.

Les contraintes :

- La distance entre deux sommets i et j doit être inférieure ou égale à D , si on note B_i (resp B_j) l'oreille de profondeur $d-1$ qui contient i (resp. j) et w_i son poids (resp. w_j), on a la contrainte suivante :

$$w_i + w_j + d(i, j) \leq D \quad (6.1)$$

où $d(i, j)$ est la distance dans le cycle principal entre i et j (illustration par la figure 6.4).

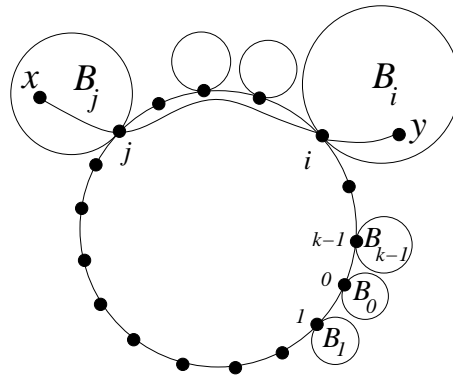


FIG. 6.4 – Contrainte de distance entre les oreilles de profondeur $d-1$

- Si l'excentricité de la base est w , pour tout sommet i du cycle principal, on a la contrainte suivante :

$$w_i + d(i, base) \leq w \quad (6.2)$$

Calculer $e(d, D, w)$ revient donc à résoudre le problème d'optimisation suivante :

$$\max_{0 \leq k \leq 2D} \left\{ \max_{(w_1, \dots, w_k) \in [0, D-w]^k} \left\{ 1 + \sum_{1 \leq i \leq k} e(d-1, D, w_i) \right\} \right\} \quad (6.3)$$

où k correspond aux nombres d'oreilles de profondeur $d-1$ et les w_i vérifient les contraintes (6.1) et (6.2).

Calcul de $e(d, D, w)$ si $w \leq \lfloor \frac{D}{2} \rfloor$. Le calcul de $e(d, D, w)$ si $w \leq \lfloor \frac{D}{2} \rfloor$ se fait aisément car dans ce cas toutes les oreilles de profondeur $d-1$ ont aussi un poids inférieur à $\lfloor \frac{D}{2} \rfloor$, la fonction $w \rightarrow e(d, D, w)$ étant croissante, il suffit de maximiser le poids des oreilles de profondeur $d-1$ en ne considérant que les contraintes (6.2) qui sont indépendantes les unes des autres, les contraintes (6.1) étant alors toujours vérifiées. Ceci étant valable quelque soit la longueur k du cycle principal, en prenant $k = 2D$ on obtient la valeur maximale :

$$\begin{aligned} \forall d \geq 0, e(d, D, 0) &= 1 \\ \forall 0 \leq w \leq D, e(0, D, w) &= 1 \\ \forall d \geq 1, \forall 1 \leq w \leq \left\lfloor \frac{D}{2} \right\rfloor, e(d, D, w) &= 1 + 2 \sum_{i=0}^{w-1} e(d-1, D, i) \\ &= e(d, D, w-1) + 2e(d-1, D, w-1) \end{aligned}$$

La figure 6.5 décrit la composition d'une oreille de diamètre D , de profondeur d et de poids $w \leq \lfloor \frac{D}{2} \rfloor$.

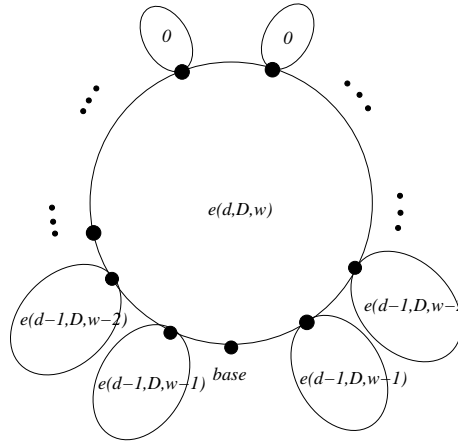


FIG. 6.5 – Construction d'une oreille de poids $w \leq \lfloor \frac{D}{2} \rfloor$

Calcul de $e(d, D, w)$ si $w \geq \lfloor \frac{D}{2} \rfloor$. En utilisant la programmation dynamique, pour calculer $e(d, D, w)$ connaissant toutes les valeurs de $e(d-1, D, w)$ pour

$w \leq D$, nous pourrions énumérer toutes les constructions satisfaisant les contraintes (6.1) et (6.2) pour trouver celle qui correspond à la solution du problème de maximisation (6.3) ce qui serait quelque peu fastidieux et certainement pas polynomial en la taille des entrées que sont d , D et w . Pour trouver les constructions optimales, nous formulons les contraintes sous forme d'un programme linéaire en nombres entiers (dont le résolution n'est pas polynomiale) pour une longueur du cycle principal donnée.

En entrée nous avons :

- d la profondeur de l'oreille
- e son poids (l'excentricité de la base dans l'oreille)
- D le diamètre de l'oreille
- les valeurs de $e(d-1, D, v)$ pour $v \leq D$
- k la longueur du cycle principal

On numérote les sommets du cycle principal de 0 à k , le sommet indexé par 0 correspondant à la base de l'oreille. On note $d(i, j)$ la distance entre le sommet i et le sommet j dans le cycle principal.

Les variables sont :

- $X_{i,v} \in \{0, 1\}$ qui vaut 1 si le sommet i est la base d'une oreille de profondeur $d-1$ et de poids v et vaut 0 sinon.
- w_i le poids de l'oreille de profondeur $d-1$ dont la base est le sommet i .
- e_i le nombre de sommets de l'oreille dont la base est le sommet i .

Le programme linéaire :

$$\begin{aligned}
 & \text{Maximize} && 1 + \sum_{1 \leq i \leq k-1} e_i \\
 \forall 1 \leq i \leq k-1 & , && \sum_{0 \leq v \leq D} X_{i,v} = 1 \\
 \forall 1 \leq i \leq k-1 & , && e_i = \sum_{0 \leq v \leq D} e(d-1, D, v) \times X_{i,v} \\
 \forall 1 \leq i \leq k-1 & , && w_i = \sum_{0 \leq v \leq D} v \times X_{i,v} \\
 \forall 1 \leq i, j \leq k-1 & , && w_i + w_j + d(i, j) \leq D \\
 \forall 1 \leq i \leq k-1 & , && w_i + d(0, i) \leq w \\
 \forall 1 \leq i \leq k-1, \forall 0 \leq v \leq D & , && X_{i,v} \in \{0, 1\}
 \end{aligned}$$

On a ainsi pour chaque valeur de k entre 0 et $2D+1$ une construction optimale, il suffit de prendre la valeur k_0 pour laquelle le nombre de sommets est maximal. Les valeurs initiales, pour $d=1$, s'obtiennent aisément vu que lorsque d vaut

1, la structure est un cycle et le nombre maximum de sommets que contient un cycle avec un sommet d'excentricité w vaut $2w + 1$.

Le tableau 6.6 donne les valeurs de w pour lesquelles $N_d(D) = e(d, D, w) + e(d - 1, D, D - w) - 1$. et le tableau 6.7 donne les valeurs de $N_d(D)$ pour $d \leq 9$, $D \leq 14$. Les valeurs intermédiaires de $e(d, D, w)$ pour $d \leq 9$ et $D \leq 14$ sont données dans la section 6.4.2.

$d \setminus D$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0	0	0	2	3	4	4	5	6	7	7	8	9	10	10
3	0	0	0	0	2	3	3	4	4	6	5	6	6	7	7
4	0	0	0	0	0	0	3	4	4	5	5	6	6	7	7
5	0	0	0	0	0	0	0	0	4	5	5	10	6	7	7
6	0	0	0	0	0	0	0	0	0	0	5	6	6	7	7
7	0	0	0	0	0	0	0	0	0	0	0	0	6	7	7
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIG. 6.6 – Valeurs de w tel que $N_d(D) = e(d, D, w) + e(d - 1, D, D - w) - 1$

$d \setminus D$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2	1	3	5	9	15	21	27	35	45	55	65	77	91	105	119
3	1	3	5	9	17	27	45	65	97	133	181	233	305	379	477
4	1	3	5	9	17	27	53	81	145	211	341	473	705	939	1317
5	1	3	5	9	17	27	53	81	161	243	453	665	1137	1611	2549
6	1	3	5	9	17	27	53	81	161	243	485	729	1393	2059	3669
7	1	3	5	9	17	27	53	81	161	243	485	729	1457	2187	4245
8	1	3	5	9	17	27	53	81	161	243	485	729	1457	2187	4373
9	1	3	5	9	17	27	53	81	161	243	485	729	1457	2187	4373

FIG. 6.7 – Valeurs de $N_d(D)$

6.4.2 Tables des valeurs de $e(d, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1														
1	1	3													
2	1	3	5												
3	1	3	5	7											
4	1	3	5	7	9										
5	1	3	5	7	9	11									
6	1	3	5	7	9	11	13								
7	1	3	5	7	9	11	13	15							
8	1	3	5	7	9	11	13	15	17						
9	1	3	5	7	9	11	13	15	17	19					
10	1	3	5	7	9	11	13	15	17	19	21				
11	1	3	5	7	9	11	13	15	17	19	21	23			
12	1	3	5	7	9	11	13	15	17	19	21	23	25		
13	1	3	5	7	9	11	13	15	17	19	21	23	25	27	
14	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29

FIG. 6.8 – Valeurs de $e(1, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1														
1	1	3													
2	1	3	5												
3	1	3	7	9											
4	1	3	9	13	13										
5	1	3	9	15	19	19									
6	1	3	9	19	23	25	25								
7	1	3	9	19	27	31	33	33							
8	1	3	9	19	33	37	41	41	41						
9	1	3	9	19	33	43	47	51	51	51					
10	1	3	9	19	33	51	55	59	61	61	61				
11	1	3	9	19	33	51	63	67	71	73	73	73			
12	1	3	9	19	33	51	73	77	81	85	85	85	85		
13	1	3	9	19	33	51	73	87	91	95	99	99	99	99	
14	1	3	9	19	33	51	73	99	103	107	111	113	113	113	113

FIG. 6.9 – Valeurs de $e(2, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1														
1	1	3													
2	1	3	5												
3	1	3	7	9											
4	1	3	9	15	17										
5	1	3	9	19	25	27									
6	1	3	9	27	37	43	45								
7	1	3	9	27	47	57	63	65							
8	1	3	9	27	65	79	89	95	97						
9	1	3	9	27	65	99	115	123	129	131					
10	1	3	9	27	65	131	149	163	173	179	181				
11	1	3	9	27	65	131	183	201	215	225	231	233			
12	1	3	9	27	65	131	233	255	273	287	297	303	305		
13	1	3	9	27	65	131	233	307	329	347	361	371	377	379	
14	1	3	9	27	65	131	233	379	404	426	444	458	468	474	476

FIG. 6.10 – Valeurs de $e(3, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1													
1	1	3												
2	1	3	5											
3	1	3	7	9										
4	1	3	9	15	17									
5	1	3	9	19	25	27								
6	1	3	9	27	45	50	53							
7	1	3	9	27	55	73	79	81						
8	1	3	9	27	81	119	137	143	145					
9	1	3	9	27	81	147	185	203	209	211				
10	1	3	9	27	81	211	277	315	333	339	341			
11	1	3	9	27	81	211	343	409	447	465	471	473		
12	1	3	9	27	81	211	473	575	640	678	697	702	705	
13	1	3	9	27	81	211	473	707	809	875	913	931	937	938
14	1	3	9	27	81	211	473	939	1084	1187	1253	1291	1309	1314

FIG. 6.11 – Valeurs de $e(4, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1	1	3											
2	1	3	5										
3	1	3	7	9									
4	1	3	9	15	17								
5	1	3	9	19	25	27							
6	1	3	9	27	45	50	53						
7	1	3	9	27	55	73	79	81					
8	1	3	9	27	81	135	153	159	161				
9	1	3	9	27	81	163	217	235	241	243			
10	1	3	9	27	81	243	372	426	444	450	452		
11	1	3	9	27	81	243	454	584	638	656	663	665	
12	1	3	9	27	81	243	665	926	1057	1110	1129	1135	1136
13	1	3	9	27	81	243	665	1139	1401	1531	1585	1603	1609
14	1	3	9	27	81	243	665	1611	2077	2338	2468	2522	2541

FIG. 6.12 – Valeurs de $e(5, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1	1	3											
2	1	3	5										
3	1	3	7	9									
4	1	3	9	15	17								
5	1	3	9	19	25	27							
6	1	3	9	27	45	50	53						
7	1	3	9	27	55	73	79	81					
8	1	3	9	27	81	135	153	159	161				
9	1	3	9	27	81	163	217	235	241	243			
10	1	3	9	27	81	243	405	459	476	482	485		
11	1	3	9	27	81	243	487	649	703	721	727	729	
12	1	3	9	27	81	243	729	1150	1313	1367	1384	1391	1393
13	1	3	9	27	81	243	729	1395	1817	1979	2033	2051	2057
14	1	3	9	27	81	243	729	2059	3005	3426	3589	3642	3661

FIG. 6.13 – Valeurs de $e(6, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1	1	3											
2	1	3	5										
3	1	3	7	9									
4	1	3	9	15	17								
5	1	3	9	19	25	27							
6	1	3	9	27	45	50	53						
7	1	3	9	27	55	73	79	81					
8	1	3	9	27	81	135	153	159	161				
9	1	3	9	27	81	163	217	235	241	243			
10	1	3	9	27	81	243	405	459	476	482	485		
11	1	3	9	27	81	243	487	649	703	721	727	729	
12	1	3	9	27	81	243	729	1215	1377	1430	1448	1455	1457
13	1	3	9	27	81	243	729	1459	1945	2107	2161	2179	2185
14	1	3	9	27	81	243	729	2187	3517	4002	4164	4218	4237

FIG. 6.14 – Valeurs de $e(7, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1	1	3											
2	1	3	5										
3	1	3	7	9									
4	1	3	9	15	17								
5	1	3	9	19	25	27							
6	1	3	9	27	45	50	53						
7	1	3	9	27	55	73	79	81					
8	1	3	9	27	81	135	153	159	161				
9	1	3	9	27	81	163	217	235	241	243			
10	1	3	9	27	81	243	405	459	476	482	485		
11	1	3	9	27	81	243	487	649	703	721	727	729	
12	1	3	9	27	81	243	729	1215	1377	1430	1448	1455	1457
13	1	3	9	27	81	243	729	1459	1945	2107	2161	2179	2185
14	1	3	9	27	81	243	729	2187	3645	4130	4292	4347	4364

FIG. 6.15 – Valeurs de $e(8, D, w)$

$D \setminus w$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1	1	3											
2	1	3	5										
3	1	3	7	9									
4	1	3	9	15	17								
5	1	3	9	19	25	27							
6	1	3	9	27	45	50	53						
7	1	3	9	27	55	73	79	81					
8	1	3	9	27	81	135	153	159	161				
9	1	3	9	27	81	163	217	235	241	243			
10	1	3	9	27	81	243	405	459	476	482	485		
11	1	3	9	27	81	243	487	649	703	721	727	729	
12	1	3	9	27	81	243	729	1215	1377	1430	1448	1455	1457
13	1	3	9	27	81	243	729	1459	1945	2107	2161	2179	2185
14	1	3	9	27	81	243	729	2187	3645	4130	4292	4347	4364

FIG. 6.16 – Valeurs de $e(9, D, w)$

6.5 Publication : *Hierarchical Ring Network Design*

Ces travaux ont fait l'objet d'une publication [BCP02], incluse ci-après, contenant les arguments et calculs permettant d'obtenir le résultat annoncé dans le théorème 6.2.1.

Hierarchical Ring Network design

J-C. BERMOND

Projet MASCOTTE I3S-CNRS/INRIA/UNSA, France

S. CHOPLIN

Projet MASCOTTE I3S-CNRS/INRIA/UNSA, France

S. PÉRENNES

Projet MASCOTTE I3S-CNRS/INRIA/UNSA, France

Abstract

A *Hierarchical Ring Network* is obtained from a ring network by appending at most one subsidiary ring to each *node* of the ring and, recursively, to each node of each subsidiary ring. The depth d is the number of levels of the recursive appending of subsidiary rings. There are different definitions according rings are appended to nodes created at the preceding level (called *HRN*) or to any node (called here *HBN* for Hierarchical Bubble Network). The case of *HRN* was considered by Aiello, Bhatt, Chung, Rosenberg and Sitaraman who give bounds (not tight) on the diameter of such *HRN* in function of the depth and the number of nodes. Here we determine the exact order of the diameter both for *HRN* and *HBN*. In fact we consider the optimization problem of maximizing the number of nodes of an *HBN* (or *HRN*) of given depth d and diameter D . We reduce the problem to a system of equation with a complex objective function. Solving this system enables us to determine precisely the structure of optimal *HBN* and to show that the maximum number of nodes is of order $D^d/d!$.

Keywords

Combinatorial Optimization, Network Design, Graph and Network algorithms

1 Introduction

Hierarchical networks are important structure in telecommunications ; indeed some of the current telecommunication networks are built in connecting rings in a hierarchical fashion. Two parameters are of importance : the number of levels of the hierarchy (going from one level to another implies changes in the transmission mode and need interface routers) and the number of hops needed to establish a connection (whose minimization corresponds to minimize the transmission time).

In [1], the authors define a "Hierarchical Ring Network" as follows : "A *Hierarchical Ring Network (HRN)*, for short) is obtained from a ring network by appending at most one subsidiary ring to each *node* of the ring and, recursively, to each node of each subsidiary ring. (Each node of an *HRN* thus belong to either one or two rings) . . .". The depth d of the *HRN* is the number of levels of the recursive appending of subsidiary rings. A *HRN* of depth 1 is nothing else than a cycle. A *HRN* of depth 2 consists in cycles interconnected via a cycle. More formally the sets of nodes V_i of level i , $1 \leq i \leq d$ consists in the disjoint union of chains $P_{i,1}, \dots, P_{i,j}, \dots, P_{i,k_i}$ $1 \leq j \leq k_i$, such that each $P_{i,j}$ is associated to a node v_j of V_{i-1} (that is of level $i-1$) and such that the subgraph generated by $P_{i,j} \cup v_j$ is a cycle (with $v_j \neq v_i$ for $j \neq i$).

In [1], the authors show the relation of the model and other structures (Multi-Rings, Chordal-Rings and Express-Rings) and give the following bounds on the diameter $D_d^{HRN}(N)$ of a N -node depth- d *HRN* :

$$\frac{1}{2}(N \times d!)^{1/d}(1 - o(1)) \leq D_d^{HRN}(N) \leq (N \times d!)^{1/d}(1 + o(1))$$

From this result we get the following bounds on the number $N_d^{HRN}(D)$ of nodes of a diameter- D depth- d *HRN* : $\frac{D^d}{d!} \leq N_d^{HRN}(D) \leq 2^d \frac{D^d}{d!}$. Recently A. Rosenberg informed us that they improve their results and showed :

$$\left(\frac{1}{2}\right)^{2/d} (N \times d!)^{1/d}(1 - o(1)) \leq D_d^{HRN}(N) \leq (N \times d!)^{1/d}(1 + o(1))$$

but the bounds are still not tight. Motivated by [1] we prove here the following results which give a tight estimation of $N_d^{HRN}(D)$ and so of $D_d^{HRN}(N)$:

Theorem 1 $\forall d \geq 2, N_d^{HRN}(D) = \frac{D^d}{d!}(1 + o(1))$

Corollary 2 $\forall d \geq 2, D_d^{HRN}(N) = (N \times d!)^{1/d}(1 + o(1))$

In order to prove the Theorem 1 we introduce a different kind of hierarchical ring network that we call *Hierarchical Bubble Network (HBN)*, for short). The difference between *HBN* and *HRN* is that the cycle appended at level d can be appended to any node of lower level (and not only to a node of level $d-1$). We propose here an equivalent definition which emphasize the recursive structure :

Definition 3 Recursive definition of *HBN*

- a depth-1 *HBN* is a ring.
- for $d \geq 2$, a depth- d *HBN* consists in a ring called *principal ring* and of a finite set of depth- $(d-1)$ *HBN* connected to nodes of the principal ring by a node of their own principal ring.

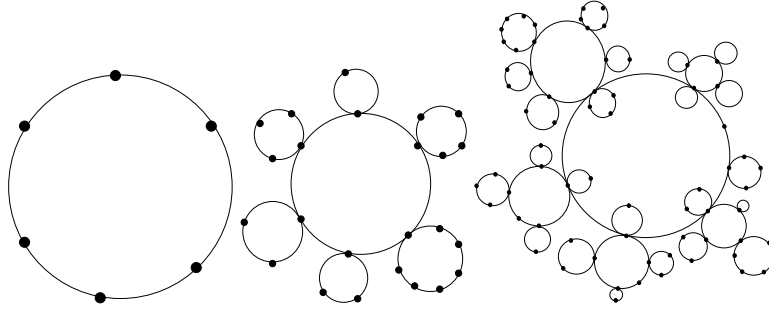


Figure 1: Examples of depth-1 *HRN/HBN*, depth-2 *HRN/HBN* and depth-3 *HBN*.

Remark 4 We can remark that for depth-1 and depth-2, there is no difference between a *HRN* and a *HBN* (see the figure 1 for examples). We can not use such a recursive definition for the *HRN* because the structure appended on each node of the *principal ring* of the *HRN* is not a *HRN*. That is the reason why we use the *HBN* structure.

In this paper we determine the maximum number of nodes $N_d^{HBN}(D)$ of nodes of a diameter- D depth- d *HBN* :

Theorem 5 $\forall d \geq 2, N_d^{HBN}(D) = \frac{D^d}{d!}(1 + o(1))$.

Then we prove that $N_d^{HRN}(D) = N_d^{HBN}(D) + O(D^{d-1}) = \frac{D^d}{d!} + O(D^{d-1})$ implying Theorem 1.

The paper is organized as follows : we first reduce the problem to the solution of a system of equations with a complex objective function. Solving partially this system enables us to determine the structure of optimal *HBN* and to prove Theorem 5. Then we prove that for all given $d \geq 2, N_d^{HRN}(D) = \frac{D^d}{d!}(1 + o(1))$.

2 Characterization of the Optimal *HBN*

Definition 6 The *weight* of a *HBN* B , denoted by $w(B)$, is defined as the minimum eccentricity of the nodes of the principal ring of B .

Definition 7 We denote by $n(d, D, w)$ the maximum number of nodes of a depth- d diameter- D *HBN* of weight w .

Note that the maximum number of nodes of a depth- d diameter- D *HBN*:

$$N_d^{HBN}(D) = \max_{0 \leq w \leq D} n(d, D, w)$$

2.1 Another Formulation of the Problem

Proposition 8 Determining $N_d^{HBN}(D)$ is equivalent to the following problem :

$$\text{Maximize} \quad \sum_{0 \leq i \leq k-1} n(d-1, D, w(B_i)) \quad (1)$$

$$\text{where} \quad k \in \mathbb{N}, k \leq 2D+1 \quad (2)$$

$$\forall 0 \leq i \leq k-1, \quad B_i \text{ is a depth-}(d-1) \text{ diameter } D \text{ HBN} \quad (3)$$

$$\forall 0 \leq i, j \leq k-1, \quad w(B_i) + w(B_j) + d(i, j) \leq D \quad (4)$$

with $d(i, j) = \min\{i - j \bmod k, j - i \bmod k\}$.

Proof. Consider an optimal depth- d diameter- D HBN. It satisfies (2,3), indeed it consists in a principal ring of some size k . To each node i is attached a depth- $(d-1)$, diameter- D HBN B_i with some weight $w(B_i)$. The number of nodes of B_i is at most $n(d-1, D, w(B_i))$. To prove (4) consider x in B_i (resp. $y \in B_j$) at distances $w(B_i)$ (resp. $w(B_j)$) from i (resp. j). Such vertices exist as i (resp. j) has eccentricity at least $w(B_i)$ (resp. $w(B_j)$) (which is the minimum of the eccentricities). Then $d(x, y) = w(B_i) + w(B_j) + d(i, j)$ where $d(i, j)$ is the distance between i and j in the principal ring. As $d(x, y) \leq D$ that implies (4) (see figure 2). So $N_d^{HBN}(D)$ is less than or equal to the maximum solution of the problem.

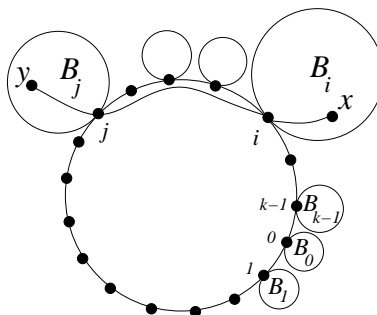


Figure 2: $\forall 0 \leq i, j \leq k-1, w(B_i) + w(B_j) + d(i, j) \leq D$

Conversely, consider a solution for which the maximum is attained and let k and $\{B_i\}_{0 \leq i \leq k-1}$ be an optimal solution. Consider the HBN obtained by taking a principal ring of size k and attaching to the node i a depth- $(d-1)$ diameter- D HBN B_i by its node minimum eccentricity $w(B_i)$. We obtained a depth- d HBN whose diameter is at most D . Indeed the distance between any pair of nodes x and y is at most D if they belong to the same B_i (the diameter of B_i being at most D) and otherwise if $x \in B_i$ and $y \in B_j$, $d(x, y) \leq d(x, i) + d(i, j) + d(y, j)$ but $d(x, i) \leq ecc_{B_i}(i) = w(B_i)$ and $d(j, y) \leq ecc_{B_j}(j) = w(B_j)$. So $d(x, y) \leq w(B_i) + w(B_j) + d(i, j) \leq D$. \square

Proposition 9 Initial values of $n(d, D, w)$.

$$\forall D \geq 0, \forall 0 \leq w \leq D, n(1, D, w) = 2w + 1 \quad (5)$$

$$\forall d \geq 1, \forall D \geq 0, n(d, D, 0) = 1 \quad (6)$$

Proof.

(5) The number of vertices of a ring with a node of eccentricity w is $2w + 1$.

(6) There can be only one node in a graph of minimal eccentricity 0. \square

2.2 Determination of $n(d, D, w)$ for $w \leq \lfloor D/2 \rfloor$

Proposition 10 $\forall d \geq 2, \forall 1 \leq w \leq \lfloor D/2 \rfloor$,

$$n(d, D, w) = n(d-1, D, w) + 2 \sum_{p=0}^{w-1} n(d-1, D, p) \quad (7)$$

$$= \sum_{p=0}^{\min(d, w)} 2^p \binom{d}{p} \binom{w}{p} = \sum_{p=0}^d \binom{d}{p} \binom{d+w-p}{p} \quad (8)$$

Proof.

(7) Let x be a node of the principal ring whose eccentricity is w , the weight of the depth- $(d-1)$ HBN based in x is at most w , by definition of the eccentricity. The weight of the depth- $(d-1)$ HBN at distance i is at most $w-i$; so we deduce that $n(d, D, w) \leq n(d-1, D, w) + 2 \sum_{p=0}^{w-1} n(d-1, D, p)$. Conversely consider the HBN obtained by taking a ring of size $2w+1$ and appending a depth- $(d-1)$ diameter D HBN B_i to node i with weight $(w-i)$ for $i=0, \dots, w$ and weight $i-(w+1)$ for $i=w+1, \dots, 2w$. Such a construction satisfies the constraint (4) and so we obtain the equality (7).

(8) The equality (7) can also be written as follows :

$$\forall d \geq 2, \forall 1 \leq w \leq \lfloor D/2 \rfloor,$$

$$n(d, D, w) - n(d, D, w-1) = n(d-1, D, w) + n(d-1, D, w-1) \quad (9)$$

Then (8) can be computed using generating series. It represents the number of integer points of a radius- w dimensional- D sphere ; some other results on this function can be found in [2]. \square

Remark 11 For all $d \geq 2$, the function $i \rightarrow n(d, D, i)$ is strictly increasing when i increases from 0 to $\lfloor D/2 \rfloor$.

2.3 First Case : $\forall 0 \leq i \leq k-1, w(B_i) \leq \lfloor D/2 \rfloor$.

In this part, we consider that the weight of each depth- $(d-1)$ HBN is less or equal to $\lfloor D/2 \rfloor$, then using the Proposition 10, we will be able to determine the maximum number of nodes :

Corollary 12 *Determining $N_d^{HBN}(D)$ is equivalent to the following optimization problem :*

$$\begin{aligned} \text{Maximize} \quad & \text{val}(S) = \sum_{0 \leq i \leq k-1} n(d-1, D, w(i)) \\ \text{where } n(d-1, D, w) &= \sum_{p=0}^{d-1} \binom{d-1}{p} \binom{d-1+w-p}{d} \end{aligned}$$

and where S is the following system :

$$\forall 0 \leq i, j \leq k-1 \quad , \quad w(i) + w(j) + d(i, j) \leq D \quad (10)$$

$$k \in \mathbb{N}, 1 \leq k \leq 2D+1 \quad (11)$$

$$\forall 0 \leq i \leq k-1 \quad , \quad 0 \leq w(i) \leq \lfloor D/2 \rfloor \quad (12)$$

Proof. $n(d-1, D, i)$ used in the Proposition 8 is given by the Proposition 10 as $w(i) \leq \lfloor D/2 \rfloor$. Then all the parameters of the system to optimize are known. \square

A solution satisfying the inequalities (10, 11,12) will be said to be feasible. Among the feasible solutions, one which attains the maximum of $\text{val}(S)$ will be said optimal.

Lemma 13 *Consider an optimal solution of S , then for each i , there exists a j such that $w(i) + w(j) + d(i, j) = D$.*

Proof. Suppose that for all i and j , $w(i) + w(j) + d(i, j) \leq D-1$. If $w(i) < \lfloor D/2 \rfloor$, then $w(i)$ could be increased without violating any constraint, and $\text{val}(S)$ will not be maximum. If $w(i) = \lfloor D/2 \rfloor$, then we have for all j_1 and j_2 , $w(j_1) + w(j_2) + d(j_1, j_2) \leq 2D-2-2w(i) < D$ as $d(j_1, j_2) \leq d(i, j_1) + d(i, j_2)$ so one $w(j)$ for $j \neq i$ could be increased without violating constraints. \square

Lemma 14 *Given an optimal solution (w, k) of the numerical system, the difference between two consecutive weights is at most 1, i.e.: $\forall 0 \leq i \leq k-1, |w(i+1) - w(i)| \leq 1$.*

Proof. Suppose that there exists some i such that $w(i) \leq w(i+\varepsilon) - 2$, with $\varepsilon = 1$ or -1 , the inequality (10) applied with $i + \varepsilon$ gives $\forall j, w(i + \varepsilon) + w(j) + d(i + \varepsilon, j) \leq D$, so we deduce $w(i) + w(j) + d(i + \varepsilon, j) \leq D - 2$, then $w(i) + w(j) + d(i, j) \leq D - 1$, contradicting Lemma 13. See figure 3 for an example with $\varepsilon = 1$. \square

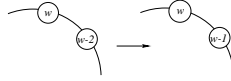


Figure 3: Example with $\varepsilon = 1$.

Notation 15 Let $M = \max_{0 \leq i \leq k-1} \{w(i)\}$, $m = \min_{0 \leq i \leq k-1} \{w(i)\}$ and $h = \lfloor \frac{k}{2} \rfloor$.

Lemma 16 $M + m + h = D$ and if $w(i) = M$ then $w(i+h) = w(i-h) = m$.

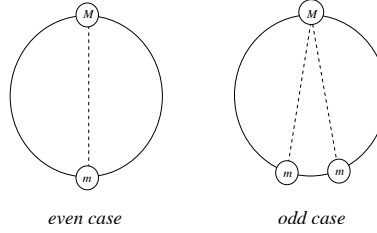


Figure 4: M is antipodal to m .

Proof. Let i_0 be such that $w(i_0) = M$. Consider one antipodal of i_0 , $i_0 + h$. By definition of m , $w(i_0 + h) = m + \alpha$ with $\alpha \geq 0$. $w(i_0) + w(i_0 + h) + d(i_0, i_0 + h) = M + m + \alpha + h$. So by (10), $M + m + h \leq D - \alpha$. Now let i_1 , be such that $w(i_1) = m$. $w(i_1) + w(j) + d(i_1, j) \leq m + M + h \leq D - \alpha$. So if $\alpha > 0$, we have a contradiction with Lemma 13 for $i = i_1$. So $\alpha = 0$ and $M + m + h = D$ and $w(i_0 + h) = m$. If h is odd, i_0 has an other antipodal $i_0 - h$ and we can prove in the same way that $w(i_0 - h) = m$. See figure 4. \square

Lemma 17 Let $d \geq 2$ and $\alpha \leq w \leq w' \leq \lfloor D/2 \rfloor - \alpha$, then

$$n(d, D, w - \alpha) + n(d, D, w' + \alpha) > n(d, D, w) + n(d, D, w')$$

Proof. It is sufficient to prove it for $\alpha = 1$, then repeated application will give the Lemma. By equation (9) (in the proof of the Proposition 10), $n(d, D, w - 1) + n(d, D, w' + 1) = n(d, D, w) + n(d, D, w') + \varepsilon$ where $\varepsilon = n(d - 1, D, w' + 1) + n(d - 1, D, w') - n(d - 1, D, w) - n(d - 1, D, w - 1)$ and $\varepsilon \geq 0$ by Remark 11. \square

Proposition 18 (Structure of an optimal solution in the first case) In an optimal solution of S , we have :

- k is odd, $k = 2h + 1$

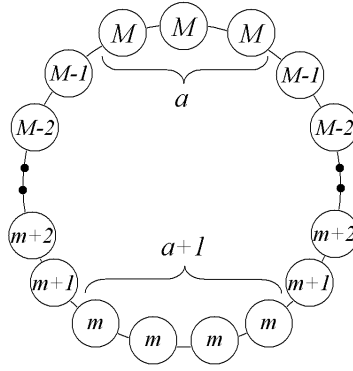


Figure 5: Structure of an optimal solution in the first case if $m < M$.

- $M + m + h = D$, (recall that $M = \max_i\{w(i)\}$ and $m = \min_i\{w(i)\}$)
- either $M = m$ and there are $k = 2h + 1$ values $w(i) = M$
- either $M > m$ and we have (up to a cyclic shift of the indices) :
 - for $0 \leq i \leq a - 1, w(i) = M$
 - for $a \leq i \leq h - 1, w(i) = M - 1 - (i - a)$
 - for $h \leq i \leq h + a, w(i) = m$
 - for $h + a + 1 \leq i \leq 2h, w(i) = M - 1 - 2h + i$

where a satisfies $2M + a - 1 = D$ or $a = h + 1 - M + m$.

Proof. First let us show that a solution of the kind described above, denoted by S' , is feasible. That is the case if $M = m$ as $w(i) + w(j) + d(i, j) \leq 2M + h = D$. If $M \neq m$ we will show by induction on i that if $i < j \leq i + h$, $w(i) + w(j) + d(i, j) \leq 2M + a - 1 = D$. That is true for $i = 0$ as for $0 < j \leq k$, $w(0) + w(j) + d(0, j) \leq D$. Now we suppose by induction that it is true till i . For $i + 1 < j \leq i + h$, $w(i + 1) + w(j) + d(i + 1, j) \leq w(i) + 1 + w(j) + d(i, j) - 1 \leq D$ by induction hypothesis and $d(i + 1, i + 1 + h) \leq D$. Furthermore $w(i + 1) + w(i + 1 + h) + d(i + 1, i + 1 + h)$ is $\leq M + m + h = D$ if $0 \leq i + 1 \leq h$ or $h \leq i + 1 \leq h + a$ or $\leq 2M + a - 1$ otherwise.

Now we will show that $val(S')$ is optimum.

Suppose that we have an optimal solution S_{opt} with value $val(S_{opt})$. If $m = M$ then it contains $2h + 1$ weights M with $2M + h = D$. So now let us suppose $m < M$. Consider one i_0 such that $w(i_0) = M$ so by Lemma 16, $w(i_0 + h) = w(i_0 - h) = m$. By Lemma 14 there should exist between i_0 and $i_0 + h$ a sequence $I : i_0 < i_1 < i_2 < i_\alpha < i_0 + h$ with $w(i_\alpha) = M - \alpha$ ($w(i_1) = M - 1, w(i_2) = M - 2, \dots, w(i_{M-m-1}) =$

$m + 1$). Now note that for $0 \leq i \leq i + h$, if $w(i) = M - \alpha$ then $w(i - h) \leq m + \alpha$ as $w(i) + w(i - h) + d(i, i - h) = M - \alpha + w(i - h) + h \leq M + m + h = D - w(i - h)$. If we replace $w(i - h)$ by $m + \alpha$, we increase the value of S_{opt} by remark 11 and so obtain a solution S_1 (not necessarily feasible) with $val(S_1) \geq val(S_{opt})$. Now for $0 < i < i + h, i \notin I (i \neq i_0, i \neq i_1, \dots, i_{M-m+1})$, we replace $w(i)$ by M and $w(i - h)$ by m we obtain a solution S_2 (not necessarily feasible). If $w(i) = M - \alpha$, $w(i - h) = m + \alpha$.

If $M - \alpha \geq m + \alpha$, Lemma 17 applied with $w = m + \alpha$ and $w' = M - \alpha$ shows that the solution S_2 where $w(i) = M$ and $w(i - h) = m$ is better. If $M - \alpha \leq m + \alpha$, Lemma 17 applied with $w = M - \alpha$ and $w' = m + \alpha$ shows that S_2 is better so $val(S_2) \geq val(S_1) \geq val(S_{opt})$.

Furthermore if there exists $i, i_0 < i < i_0 + h, i \notin I$ for which $w(i) = M - \alpha$ with $\alpha > 0$, then $val(S_2) > val(S_{opt})$. If k is even, the sequence S_2 consists in a indices with weight M , a indices with weight m and exactly two with weight $M - \alpha$, $1 \leq \alpha \leq M - m - 1$ where $2M + a - 1 = D$. In that case we can add one weight m obtaining a solution S_3 with the same weight sequence as S' of the theorem. In summary as

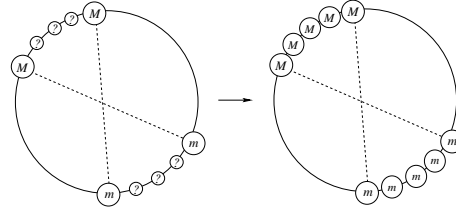


Figure 6: Transformation from S_{opt} to S_3 .

S' is feasible, an optimal solution contains an odd number of indices so k is odd, $k = 2h + 1$, and there are exactly a weight M , $a + 1$ weights m and exactly two weights $M - \alpha$, $1 \leq \alpha \leq M - m - 1$. Now in a feasible solution, the distance between two indices with weight M is at most $a - 1$ as $2M + a - 1 = D$ so the a indices of weight M are consecutive so for $i_0 \leq i \leq i_0 + a - 1, w(i) = M$. Furthermore as $w(i_0 + h) = m$ and between $i_0 + a$ and $i_0 + h - 1$ there are exactly $M - m - 1 = h - a$ indices $w(a - 1 + \alpha) = M - \alpha$ for $1 \leq \alpha \leq M - m - 1$ and similarly as $w(i_0 + a - 1 - h) = w(h + a) = m$ we have for $h + a + 1 \leq i \leq 2h, w(i) = M - 1 - 2h + i$.

So an optimal solution can be obtain with this structure. \square

Now to find the optimal solution we only have to compare a small number of structures. Given D , we have to find the better couple (M, m) with $0 \leq m \leq M \leq \lfloor D/2 \rfloor$ where $M + m + h = D$. The value is

- if $m = M : val(S) = (2h + 1)n(d - 1, D, M)$ with $2M + h = D$.

- if $m < M$: $val(S) = a n(d-1, D, M) + 2 \sum_{\alpha=1}^{M-m-1} n(d-1, D, M-\alpha) + (a+1)n(d-1, D, m)$ with $2M+a-1 = D$.

2.3.1 Case $d = 2$

Proposition 19 *If $d = 2$, an optimal solution of the first case is obtained with $M = m = \lfloor \frac{D+1}{4} \rfloor$. It is unique except for $D \equiv 2$ or $3[4]$ where the solution with $M = m + 1 = \lfloor \frac{D+3}{4} \rfloor$ is also optimal.*

Proof. First suppose there exists an optimal solution S_{opt} with $m \leq M - 2$. If we consider the solution obtained with the parameters $M - 1$ instead of M and $m + 1$ as minimum (which is also feasible) that corresponds to replace $(a - 2)$ indices M by $(a - 2)$ indices $M - 1$ and $(a - 1)$ indices m by $(a - 1)$ indices $m + 1$ but as $n(d-1, D, w) = 2w + 1$ for $d = 2$ the value is increased by 2 so the solution was not optimal.

If $M = m$, $val(S) = (2h + 1)(2M + 1)$ with $2M + h = D$ so $val(S) = -8M^2 + (2M + 1)(2D - 1) + 2$. A study of this function shows that the optimum is attained for $M = \lfloor \frac{D+1}{4} \rfloor$. Indeed in real number, the maximum is attained for $M = \frac{D}{4} - \frac{1}{8}$; then it is sufficient to compare for $D = 4q + 1 + \varepsilon$ the value for $M = q$ and $M = q + 1$. If $\varepsilon = 0$ or 1 : the best is $M = q = \lfloor \frac{D+1}{4} \rfloor$. For $\varepsilon = 2$ or 3 : the best is $M = q + 1 = \lfloor \frac{D+1}{4} \rfloor$.

For $M = m + 1$, $val(S') = a(2M + 1) + (a + 1)(2M - 1)$ with $2M + a - 1 = D$ so $val(S') = -8M^2 + 2M(2D + 3) - 1$. By comparing the values we always have $val(S') \leq val(S)$ and equality is attained for $D = 4q + 2$ for which the value for $M = q + 1, m = q, val(S') = 8q^2 + 14q + 5$ that is the same as $val(S)$ for $M = m = q$ and for $D = 4q + 3$ for which the value for $M = q + 1, m = q, val(S') = 8q^2 + 18q + 9$ is the same as $val(S)$ for $M = m = q + 1$. \square

2.3.2 Case $d \geq 3$

Proposition 20 *In an optimal solution of the first case, if $d \geq 3$, then $a \leq 2$ except for $d = 3$ and $D = 4, 6, 7, 9$.*

Proof. We will use the following equality : $\forall d \geq 2, D, w \leq \lfloor D/2 \rfloor$,

$$\begin{aligned} n(d, D, w) - n(d, D, w - 1) &= \sum_{p=0}^d \binom{d}{p} \left(\binom{d+w-p}{d} - \binom{d+w-1-p}{d} \right) \\ &= \sum_{p=0}^d \binom{d}{p} \binom{d+w-1-p}{d-1} \end{aligned}$$

Consider an optimal solution with a indices of weight M where $2M + a - 1 = D$. Remember that if $m < M$ there are $(a + 1)$ indices of weight m and if $m = M$, $(a - 1)$ indices will be considered of weight m .

Claim 1: In an optimal solution, $m \leq \frac{(a+1)(d-1)}{2}$.

Either $m \leq d-1$ and it is true as $a \geq 1$, otherwise consider the transformation T_1 which consists in replacing $(a-1)$ indices of weight m by $(a+1)$ indices of weight $m-1$. The solution obtained is feasible and should have value less or equal to that of the optimal solution. So

$$\begin{aligned} T_1(a, d, D, m) &= (a+1)n(d-1, D, m-1) - (a-1)n(d-1, D, m) \leq 0 \\ \sum_{p=0}^{d-1} \binom{d-1}{p} \left[(a+1) \binom{d-1+m-1-p}{d-1} - (a-1) \binom{d-1+m-p}{d-1} \right] &\leq 0 \\ \text{If } m \geq d-1, \sum_{p=0}^{d-1} \binom{d-1}{p} \frac{(d-1+m-1-p)!}{(d-1)!(m-p)!} P(p, a, d, m) &\leq 0 \end{aligned}$$

where $P(p, a, d, m) = (a+1)(m-p) - (a-1)(d-1+m-p)$. $P(p, a, d, m)$ is decreasing in p , so we should have $P(d-1, a, d, m) \leq 0$, that is $(a+1)(m-d+1) - (a-1)m \leq 0$ or $m \leq \frac{(a+1)(d-1)}{2}$.

Claim 2: In an optimal solution, $m \geq (a-2)(d-1) - (a-1)$.

We suppose $a \geq 3$, otherwise the formula gives $m \geq -1$. Consider the transformation T_2 which consists in replacing $(a-2)$ indices of weight M by $(a-2)$ indices of weight $M+1$ and $(a-1)$ indices of weight m by $(a-1)$ indices of weight $m-1$. The solution obtained is feasible and should have a value less or equal to that of the optimal solution. Let $T_2(a, d, D, M, m) =$

$$\begin{aligned} &(a-2)[n(d-1, D, M+1) - n(d-1, D, M)] \\ &- (a-1)[n(d-1, D, m) - n(d-1, D, m-1)] \end{aligned}$$

, then $T_2(a, d, D, m, m) \leq T_2(a, d, D, M, m) \leq 0$

$$T_2(a, d, D, m, m) = \sum_{p=0}^{d-1} \binom{d-1}{p} \frac{(d-2+m-p)!}{(d-2)!(m-p+1)!} Q(p, a, d, m) \leq 0$$

where $Q(p, a, d, m) = (a-2)(d-1+m-p) - (a-1)(m-p+1)$. $Q(p, a, d, m)$ is increasing in p , so we should have $Q(0, a, d, m) \leq 0$, that is $m \geq (a-2)(d-1) - (a-1)$.

Claim 3: If $d \geq 4$ then $a \leq 5 + \frac{8}{d-3}$.

Combining claim 1 with claim 2, we obtain : $(a-2)(d-1) - (a-1) \leq m \leq (a+1)\frac{d-1}{2}$ that is $(d-3)a \leq 5d-7$ or if $d \neq 3$, $a \leq 5 + \frac{8}{d-3}$.

Claim 4: If $d \geq 12$, then $a \leq 2$.

Consider the transformation T_3 which consists in replacing $(a-2)$ indices of

weight M by $(a-2)$ indices of weight $M+1$ and $(a-1)$ indices of weight m by 2 indices of weight $m-1$ and $(a-1)$ indices of weight $m-2$. We will show that this transformation increase the value val if $d \geq 12$ and $a \geq 3$, so giving a contradiction.

Let $T_3(a, d, D, M, m) = (a-2)[n(d-1, D, M+1) - n(d-1, D, M)] - (a-1)[n(d-1, D, m) - n(d-1, D, m-2)] + 2n(d-1, D, m-1)$, one can prove (see detailed version) that for $a \geq 3, d \geq 12, T_3(a, d, D, M, m) > 0$. Then, in an optimal solution, if $d \geq 12$ then $a \leq 2$.

Claim 5: If $4 \leq d \leq 11$ then $a \leq 2$.

Now we can compute all other cases : for $4 \leq d \leq 11, 3 \leq a \leq 5 + \frac{8}{d-3}$ (implied by claim 2), $1 \leq m \leq (a+1)(d-1)/2$ (implied by claim 1), $T_3(a, d, D, m, m) > 0$ or $T_2(a, d, D, m, m) > 0$.

Claim 6: In an optimal solution, if $d = 3$ then $a \leq 2$ unless for $D = 4, 6, 7, 9$. There exist optimal solutions for $d = 3, D = 4, 7$ such that $a \leq 2$.

If $m = 1, T_2(a, 3, D, M, 1) = 4((a-2)M - 1)$, which is positive for $a \geq 3, M \geq 2$ and for $a \geq 4, M \geq 1$ then the only case to study is $a = 3, M = m = 1$, i.e. $D = 4$. In this case, there are two optimal solutions: one with $h = 2, M = m = 1$ and one with $a = 1, M = 2, m = 0$.

If $m \geq 2$, using T_1 or T_2^2 (twice the transformation T_2), we can conclude (see detailed version) in all cases except for $D = 6$ (where the optimal solution satisfies $h = 2, m = M = 2$), $D = 7$ (where there are two optimal solutions: one with $a = 4, m = M = 2$ and one with $a = 2, m = 1$ and $M = 3$) and $D = 9$ (where the optimal solution satisfies $h = 3$ and $m = M = 3$). \square

2.4 Second Case : $\exists 0 \leq i \leq k-1, w(B_i) > \lfloor D/2 \rfloor$.

Suppose that there exists a solution S_α for d with some $w(B_i) = \lfloor D/2 \rfloor + \alpha, \alpha \geq 1$. Then condition 10 $w(B_i) + w(B_j) + d(i, j) \leq D$ implies that the two B_j at distance p of B_i have weight at most $\lfloor D/2 \rfloor - \alpha - p$ so $val(S_\alpha) \leq n(d-1, D, \lfloor D/2 \rfloor + \alpha) + 2 \sum_{w=0}^{\lfloor D/2 \rfloor - \alpha - 1} n(d-1, D, w)$. We will show by induction on d that this value is less than the value of a system S' of the previous case with $M = \lfloor D/2 \rfloor, m = 0$ and $a = 1$ or $a = 2$, according D is even or odd, implying that a solution of the second case can not be optimal.

2.4.1 Case $D = 2q$

It is true for $d = 2$. Indeed $val(S_\alpha) = 2(q + \alpha) + 1 + 2 \sum_{w=0}^{q-\alpha-1} (2w+1)$ and for $M = q$ and $a = 1, val(S') = n(1, D, q) + 2 \sum_{w=0}^{q-1} n(1, D, w) = 2q + 1 + 2 \sum_{w=0}^{q-1} (2w+1)$. So $val(S') - val(S_\alpha) = -2\alpha + 2 \sum_{w=q-\alpha}^{q-1} (2w+1)$. As $\alpha < q$ ($\alpha = q$ will imply a unique B_i and so a depth $d-1$) and $q > 1, val(S') - val(S_\alpha) \geq -2\alpha +$

$2(2q - 1) > 0$. Now suppose it is true till $d - 1$, then by induction hypothesis, $n(d - 1, D, q + \alpha) \leq n(d - 1, D, q)$. So $val(S_\alpha) \leq n(d - 1, D, q) + 2 \sum_{w=0}^{q-\alpha-1} n(d - 1, D, w)$. As $val(S') = n(d - 1, D, q) + 2 \sum_{w=0}^{q-1} n(d - 1, D, w)$, $val(S') - val(S_\alpha) \geq 2n(d - 1, D, q - 1) > 0$.

2.4.2 Case $D = 2q + 1$

It is true for $d = 2$. Indeed $val(S_\alpha) = 2(q + \alpha) + 1 + 2 \sum_{w=0}^{q-\alpha} (2w + 1)$ and for $M = q, a = 2$, $val(S') = 2(2q + 1) + 2 \sum_{w=0}^{q-1} (2w + 1)$. So $val(S') - val(S_\alpha) \geq 2q - 2\alpha + 1$. As $\alpha \leq q$, $2q - 2\alpha + 1 > 0$ and $val(S') - val(S_\alpha) > 0$. Now suppose it is true till $d - 1$, then by induction hypothesis, $val(S_\alpha) \leq n(d - 1, D, q) + 2 \sum_{w=0}^{q-\alpha-1} n(d - 1, D, w)$. $val(S') = 2n(d - 1, D, q) + 2 \sum_{w=0}^{q-1} n(d - 1, D, w)$, $val(S') - val(S_\alpha) \geq n(d - 1, D, q) > 0$.

2.5 Results

Now we can state the final results:

Theorem 21 For $d = 2$, the optimal solutions are obtained with $M = m = \lfloor \frac{D+1}{4} \rfloor$ plus for $D \equiv 2$ or $3[4]$, $M = m + 1 = \lfloor \frac{D+3}{4} \rfloor$.

For D even, $d \geq 3$, the unique optimal solution satisfies $a = 1, M = D/2, m = 0$ except for $d = 3, D = 4$ where there is another optimal solution with 5 nodes of weight $M = m = 1$ and $d = 3, D = 6$ where the optimal solution consists in 5 nodes of weight $M = m = 2$.

For D odd, $d \geq 3$, the unique optimal solution satisfies $a = 2, M = \frac{D-1}{2}$ except for $d = 3, D = 7$ where there is another solution with 7 nodes of weight 2 and $d = 3, D = 9$ where the optimal solution consists in 9 nodes of weight 3.

Proof. As we have seen in section 2.4 the optimal solution is always obtained in the first case (all $w(B_i) \leq \lfloor D/2 \rfloor$). So the theorem follow for $d = 2$ from Proposition 19 and for $d \geq 3$ from Proposition 20 by noting that if D is even (resp. D odd), $a = D + 1 - 2M$ being odd (resp. even), $a \leq 2$ implies $a = 1$ for D even and $a = 2$ for D odd. As we are in the first case, by Proposition 20, if D is even, we have, except for $d = 3, D \in \{4, 6\}$ that $a \leq 2$ but $a = D + 1 - 2M$ being odd, $a = 1$, $M = D/2$ and if m was positive, we could add 2 nodes of weight $m - 1$ obtaining a better solution. \square

Thus we have characterized the optimal solution for D even and almost characterized for D odd where we propose the following conjecture :

Conjecture 22 The optimal value of m in the case D odd, $d \geq 3$ is $m = M = \frac{D-1}{2}$ for $D \leq 2 \lfloor \frac{3(d-1)}{4} \rfloor + 1$ and $m = \lfloor \frac{3(d-1)}{4} \rfloor$ for $D \geq 2 \lfloor \frac{3(d-1)}{4} \rfloor + 1$ (except for $d = 3, D \in \{7, 9\}$).

However the results of theorem 21 are widely sufficient to give the asymptotic order of $N_d^{HBN}(D)$ and prove the Theorem 5 :

Theorem 5 For given $d \geq 2$, $N_d^{HBN}(D) = \frac{D^d}{d!} + O(D^{d-1})$.

Proof. If $d = 2$, recall that an optimal solution consists in $2D + 1 - 4 \lfloor \frac{D+1}{4} \rfloor$ nodes of weight $M = \lfloor \frac{D+1}{4} \rfloor$ and its values is $-8M^2 + (2M + 1)(2D - 1) + 2 = \frac{D^2}{2} + O(D)$. If $d \geq 3$ and $D \geq 2d$, let us prove the theorem by induction on d .

For D even, the optimal value is $n(d-1, D, D/2) + 2 \sum_{p=0}^{D/2-1} n(d-1, D, p) = n(d, D, D/2) = \sum_{p=0}^d 2^p \binom{d}{p} \binom{D/2}{p}$. But $\binom{D/2}{p} = \frac{D^p}{2^p p!} + O(D^{p-1})$, so $\sum_{p=0}^{d-1} 2^p \binom{d}{p} = O(D^{d-1})$ and $n(d, D, D/2) = 2^d \binom{d}{d} \frac{D^d}{2^d d!} + O(D^{d-1}) = \frac{D^d}{d!} + O(D^{d-1})$.

For $D = 2q + 1$ odd, the optimal value is $2 \sum_{p=m}^q n(d-1, D, p) + n(d-1, D, m) = n(d, D, q) + n(d-1, D, q) + n(d-1, D, m) - 2 \sum_{p=0}^{m-1} n(d-1, D, p) = n(d, D, q) + n(d-1, D, q) + n(d-1, D, m) - n(d, D, m) + n(d-1, D, m)$. But by claim 1 of Theorem 20, $m \leq \frac{3d}{2}$ and so $n(d, D, m)$ is bounded by a value independent of D and $n(d, D, q) = \frac{D^d}{d!} + O(D^{d-1})$. \square

3 Relation with HRN

Now we come back to the first model :

Definition 23 We call depth-0 *ear* a node and for $d \leq 1$ a depth- d *ear* with $d \geq 1$ of base x a set of depth- $(d-1)$ ears and a node x interconnected by a ring called *principal ring*.

Definition 24 Then a *HRN* can be defined as follows : A depth- d *HRN* is a set of depth- $(d-1)$ ears were the bases are interconnected by a ring.

We denote by $e(d, D, w)$ the maximum number of nodes that can be contained in a depth- d diameter- D ear whose base has an eccentricity w .

Proposition 25

$$\begin{aligned} \forall d \geq 0, e(d, D, 0) &= 1 \\ \forall 0 \leq w \leq D, e(0, D, w) &= 1 \\ \forall d \geq 1, \forall 1 \leq w \leq \lfloor D/2 \rfloor, e(d, D, w) &= 1 + 2 \sum_{i=0}^{w-1} e(d-1, D, i) \\ &= e(d, D, w-1) + 2e(d-1, D, w-1) \end{aligned}$$

Proof. Same as the Proposition 10. \square

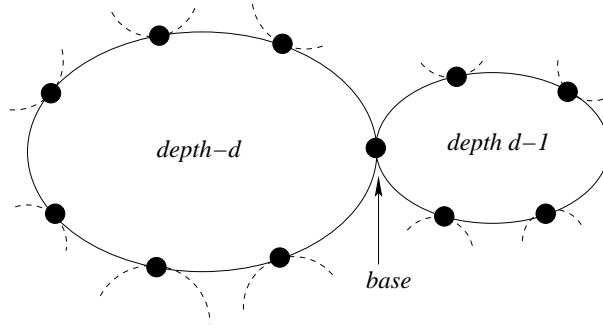


Figure 7: A HRN is the union of two ears

Proposition 26 *The maximum number of nodes of a depth- d diameter- D HRN is*

$$N_d^{HRN}(D) = \max_{0 \leq w \leq D} \{e(d, D, w) + e(d-1, D, D-w)\} - 1$$

Proof. We can decompose every HRN in two ears : given a node x of the principal ring, we can extract a ear of depth d and another of depth $d-1$, then the number of nodes is the sum of the number of the two ears minus one as x is counted twice. In the other hand, any HRN can be construct with two ears of this kind (see figure 7). \square

Remark 27 $\forall D \geq 0, N_1^{HRN}(D) = N_1^{HBN}(D) = 2D + 1$, in this case, the two constructions are the same.

Corollary 28 $e(d, D, \lfloor D/2 \rfloor) = n(d, D, \lfloor D/2 \rfloor) + O(D^{d-1})$

Proof. We prove it by induction on d . It's true for $d = 1$.

Suppose that it's true for $d-1$, then $n(d, D, w) - e(d, D, w) =$

$$\begin{aligned} & n(d-1, D, w) + n(d-1, D, w-1) + n(d, D, w-1) - e(d, D, w-1) \\ & \qquad \qquad \qquad - 2e(d-1, D, w-1) = \\ & n(d-1, D, w) - e(d-1, D, w) + n(d-1, D, w-1) - e(d-1, D, w-1) \\ & + n(d, D, w-1) - e(d, D, w-1) + e(d-1, D, w) - e(d-1, D, w-1) \end{aligned}$$

So $n(d, D, w) - e(d, D, w) =$

$$n(d, D, w-1) - e(d, D, w-1) + e(d-1, D, w) - e(d-1, D, w-1) + O(D^{d-2})$$

Applying this w times, we obtain

$$n(d, D, w) - e(d, D, w) = e(d-1, D, w) + wO(D^{d-2})$$

therefore $n(d, D, \lfloor D/2 \rfloor) - e(d, D, \lfloor D/2 \rfloor) = e(d-1, D, \lfloor D/2 \rfloor) + O(D^{d-1})$
 but $e(d-1, D, \lfloor D/2 \rfloor) \leq n(d-1, D, \lfloor D/2 \rfloor)$
 and so $e(d-1, D, \lfloor D/2 \rfloor) = O(D^{d-1})$. \square

Theorem 1 For a given $d \geq 2$, $N_d^{HRN}(D) = \frac{D^d}{d!} + O(D^{d-1})$.

Proof. By Proposition 26,

$$\begin{aligned} N_d^{HRN}(D) &\leq \max_{0 \leq w \leq D} \{n(d, D, w) + n(d-1, D, D-w)\} - 1 \\ &\leq N_d^{HBN}(D) + N_{d-1}^{HBN}(D) - 1 \end{aligned}$$

and by Theorem 5, $N_d^{HBN}(D) + N_{d-1}^{HBN}(D) - 1 = \frac{D^d}{d!} + O(D^{d-1})$.

On the other side $e(d, D, \lfloor D/2 \rfloor) \leq N_d^{HRN}(D)$, and by Corollary 28,

$$e(d, D, \lfloor D/2 \rfloor) = n(d, D, \lfloor D/2 \rfloor) + O(D^{d-1}) = \frac{D^d}{d!} + O(D^{d-1})$$

$$\text{Then } N_d^{HRN}(D) = \frac{D^d}{d!} + O(D^{d-1}). \quad \square$$

4 Conclusion

In this paper we have improved a result from Aiello, Bhatt, Chung, Rosenberg and Sitaraman by giving tight bounds on the diameter of a Hierarchical Ring Network according to the number of nodes. To do that we have defined another kind of Hierarchical Ring Network for which we give the optimal design according to depth and diameter constraints. This network due to its recursive structure might be more interesting for applications.

References

- [1] W. Aiello, S. N. Bhatt, F. R. K. Chung, A. L. Rosenberg, R. K. Sitaraman. Augmented Ring Networks. In *Proc of The 6th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 1–16, 1996.
- [2] Y. Dinitz, M. Feigelson, S. Zaks. On Optimal Graphs Embedded into Paths and Rings, with Analysis Using 11-Spheres. In *Lecture Notes in Computer Science*, 1335:171–183, 1997

Jean-Claude Bermond, Sébastien Choplin and Stéphane Pérennes are in the MASCOTTE Project (I3S-CNRS/INRIA/UNSA)
 INRIA, 2004 route des lucioles, BP93 06902 Sophia Antipolis, France.
 E-mail: {jean-claude.bermond,sebastien.choplin,stephane.perennes}@sophia.inria.fr

Chapitre 7

Optimisation du coût d'une boucle SDH

Les boucles SDH sont utilisées par la plupart des opérateurs de télécommunications comme support de transmission entre leurs clients. Comme nous l'avons vu dans le chapitre 1, le transport de l'information dans les réseaux SDH se fait par l'intermédiaire de containers encapsulés dans des trames STM de différentes capacités.

Le coût de la mise en place de ces liaisons dépend de plusieurs facteurs, donc les principaux sont les travaux à mettre en œuvre pour les installer, et la capacité des liaisons. La facturation faite par l'opérateur à ses clients dépend donc du coût de cette installation. Le coût d'une liaison n'est donc pas une fonction linéaire de sa capacité, ainsi lorsque l'opérateur souhaite installer un nouveau réseau et s'il a établi des prévisions sur la demande de ces clients potentiels, il lui sera intéressant de regrouper le trafic des différents clients pour qu'ils utilisent quelques liaisons de capacité importante plutôt que beaucoup de liaisons de faible capacité.

7.1 Modélisation

Dans ce contexte nous allons étudier le cas où les requêtes des utilisateurs sont symétriques (lorsqu'un client demande une liaison entre un nœud x et un nœud y avec un débit donné, il demande également une liaison entre y et x avec le même débit).

Nous modélisons le réseau logique par un graphe orienté où les nœuds sont les commutateurs et les liens sont les liaisons logiques définies par la politique de routage de l'opérateur. Etant donnée une instance de communication, nous chercherons à établir un tel réseau logique de coût minimal qui puissent satisfaire cette instance où le coût d'un lien sera défini par une fonction du type c^α où c est la capacité du lien et α une constante entre 0 et 1. A priori nous ne savons rien sur la topologie du graphe logique que nous voulons définir et dimensionner, il

peut donc être considéré comme le graphe complet sur lequel nous devons définir les capacités des arcs.

L'exemple la figure 7.1 décrit quatre graphes orientés permettant de réaliser l'instance de communication d'échange total unitaire pour 5 sommets. La valeur sur chaque arc correspond à sa capacité, seul le routage des requêtes issues de D sont illustrées (en pointillés bleus). Si on prend $\alpha = \frac{1}{2}$ le graphe $G1$ coûte $8 \times \sqrt{(4)} = 16$, le graphe $G2$ coûte $6 \times \sqrt{(7)} = 15,87$, le graphe $G3$ coûte $5 \times \sqrt{(10)} = 15,81$ et le graphe $G4$ coûte $20 \times \sqrt{(1)} = 20$; suivant les valeurs de α , une des solutions sera meilleure que les autres, les courbes de la figure 7.2 représentent le coût des exemples de constructions de la figure 7.1 en fonction de α .

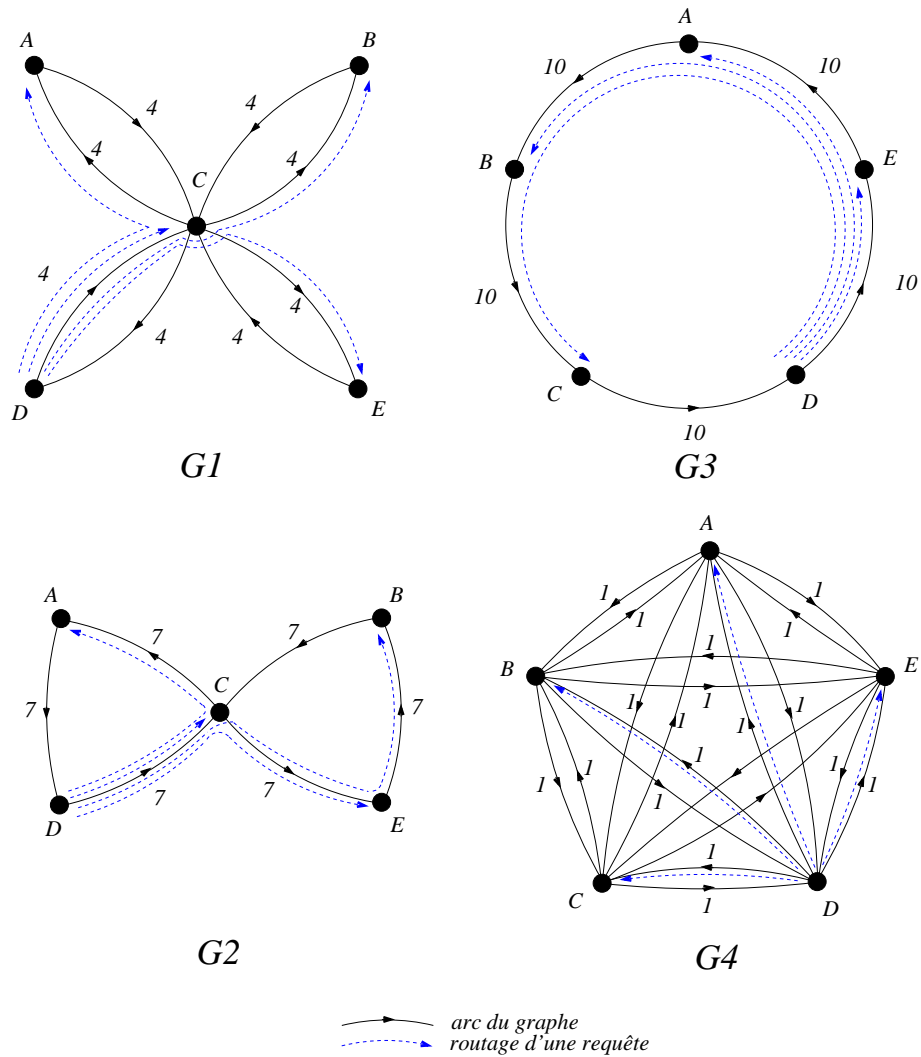
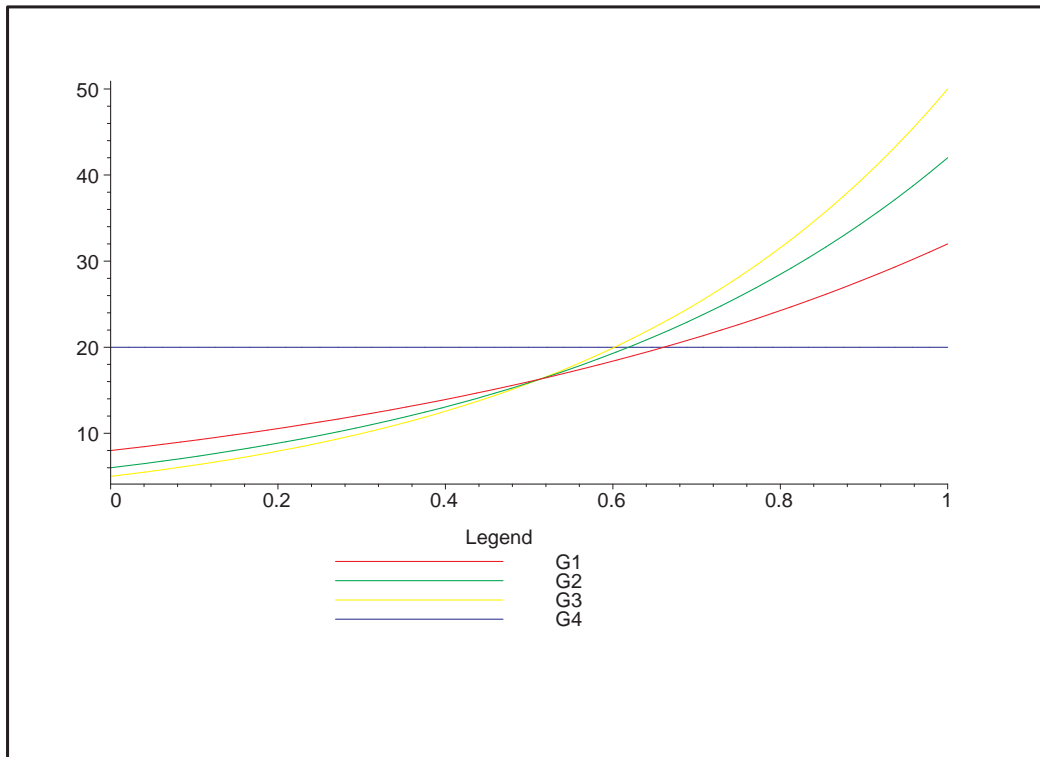


FIG. 7.1 – Exemple de graphe permettant de réaliser l'échange total entre 5 sommets.

FIG. 7.2 – Coût des graphes de la figure 7.1 en fonction de α .

On note :

- K_n^* le graphe complet orienté à n sommets.
- $I = \{R_1, R_2, \dots, R_k\}$ l'ensemble des requêtes,
- r_i le débit associé à la requête R_i ,
- \mathcal{C} l'ensemble des valeurs que peut prendre la capacité d'un arc.
- $c : A \rightarrow \mathcal{C}$ une application qui à chaque arc lui associe une capacité.
- \mathcal{P} l'ensemble des chemins élémentaires de K_n^* ,
- \mathcal{P}_i l'ensemble des chemins élémentaires de K_n^* entre les deux sommets de la requête R_i ,
- $f : \mathcal{P} \rightarrow \mathbb{R}^+$ une application qui à chaque chemin de \mathcal{P} lui associe une quantité de flot,
- $\pi(a)$ la charge de l'arc a comme étant la somme des flots des chemins qui passent par $a : \pi(a) = \sum_{P \in \mathcal{P} | a \in P} f_P$,
- le coût du réseau $\mathcal{K}_\alpha(I)$ le coût minimal du réseau permettant d'établir les requêtes de l'instance I pour le paramètre de coût α .

Le but est donc de trouver la valeur optimale $\mathcal{K}_\alpha(I)$ du système suivant :

$$\begin{array}{ll}
\text{Minimize} & \sum_{a \in A} c^\alpha(a) \\
\text{Avec les contraintes :} & \\
& \sum_{P \in \mathcal{P}_i} f(P) = r_i \quad \forall i \in \{1, \dots, k\} \\
\pi(a) = \sum_{P \in \mathcal{P} | a \in P} f(P) & \forall a \in A \\
c(a) \geq \pi(a) & \forall a \in A \\
f(P) \geq 0 & \forall P \in \mathcal{P}
\end{array}$$

REMARQUE. Si on considère que les capacités sont réelles ($\mathcal{C} = \mathbb{R}^+$), dans une solution optimale on aura toujours la capacité d'un arc égale au flot qui circule sur cette arc ($c(a) = \pi(a)$).

REMARQUE.

- Si α est égal à 1, le coût est linéaire en fonction de la capacité, si les capacités sont réelles il n'y a donc aucun intérêt à regrouper le trafic, le graphe logique optimal correspond au graphe des requêtes et $\mathcal{K}_1(I) = \sum_{1 \leq i \leq k} r_i$.
- Si α est égal à 1 et que les capacités sont à valeurs dans un ensemble discret, la solution optimale ne correspond pas en général au graphe des requêtes.
- Si α est égal à 0, la fonction de coût ne dépend pas de la capacité et vaut toujours 1 si l'arc existe, le coût du réseau correspond donc au nombre d'arcs. Le sous-graphe orienté fortement connexe du graphe complet ayant le plus petit nombre d'arcs étant le cycle, le coût du réseau est n .

Il existe des travaux traitant l'optimisation de coût des réseaux avec des fonctions de coût du même type (continues ou en escalier) [Min89], mais le but est en général de dimensionner les liens d'un réseau existant. Ici nous cherchons à définir un réseau en même temps que son dimensionnement.

7.2 Heuristiques classiques

Les heuristiques classiques consistant à effectuer des transformations locales sur une solution réalisable permettent de trouver des optimum locaux mais en général n'aboutissent pas à l'optimal global, ceci étant dû au caractère concave de la fonction de coût c^α .

7.2.1 Reroutage d'un chemin

Etant donnée une solution réalisable du problème, on peut chercher à rerouter un des chemins P d'une requête de manière à diminuer le coût, c'est à dire trouver

un autre chemin P' tel que $f(P) = f(P')$ et

$$\sum_{a \in P} (c(a) - f(P))^\alpha + \sum_{a \in P'} (c(a) + f(P'))^\alpha \leq \sum_{a \in P} c^\alpha(a) + \sum_{a \in P'} c^\alpha(a)$$

ce qui revient à trouver un plus court chemin dans le graphe avec un poids sur les arcs défini par :

$$w(x, y) = \begin{cases} c^\alpha((x, y)) & \text{si } (x, y) \in P \\ (c((x, y)) + f(P))^\alpha & \text{si } (x, y) \notin P \end{cases}$$

7.2.2 Reroutage du flot sur un arc

Au lieu de rerouter uniquement un chemin, on peut rerouter tous les flots passant par un arc a donné sur un autre chemin pour diminuer le coût global, c'est à dire trouver un plus court chemin dans le graphe avec un poids sur les arcs défini par :

$$w(x, y) = \begin{cases} \infty & \text{si } a = (x, y) \\ (c((x, y)) + c(a))^\alpha & \text{si } a \neq (x, y) \end{cases}$$

7.3 Bornes inférieures et constructions dans le cas des capacités réelles

Lemme 7.3.1 *Si $\alpha < 1$ et que $\mathcal{C} = \mathbb{R}^+$ alors il existe une solution optimale pour laquelle le sous-graphe induit par les chemins issus d'un sommet s est un arbre.*

PREUVE. Soit s un sommet du graphe et supposons que le sous-graphe induit par les chemins issus de s ne soit pas un arbre. Il existe donc un sommet t et deux chemins arc-disjoints p_1 et p_2 entre s et t . Considérons un chemin p'_1 (resp. p'_2) issu de s qui contient p_1 (reps. p_2).

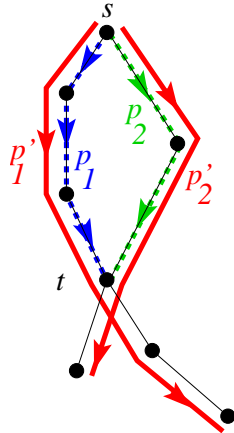


FIG. 7.3 – Cycle dans le graphe induit par les chemins issus de s .

Si la solution est optimale, il n'est pas intéressant de rerouter p'_1 (resp. p'_2) sur p_2 (resp. p_1) et donc :

$$\sum_{e \in p_1 \cup p_2} f(e)^\alpha = \sum_{e \in p_1} f(e)^\alpha + \sum_{e \in p_2} f(e)^\alpha \quad (7.1)$$

$$\leq \sum_{e \in p_1} (f(e) + f(p'_1))^\alpha + \sum_{e \in p_2} (f(e) - f(p'_2))^\alpha \quad (7.2)$$

$$\leq \sum_{e \in p_1} (f(e) - f(p'_1))^\alpha + \sum_{e \in p_2} (f(e) + f(p'_2))^\alpha \quad (7.3)$$

or l'une de ces deux inéquations est toujours fautive à cause de la concavité de la fonction c^α car $0 < \alpha < 1$. \square

Corollaire 7.3.2 *Si $\alpha < 1$ et que $\mathcal{C} = \mathbb{R}^+$ alors il existe une solution optimale où chaque requête est routée sur un seul chemin.*

Lemme 7.3.3 *Soit I_n une instance de l'échange total unitaire dans le graphe complet à n sommets,*

$$n(n-1)^\alpha \leq \mathcal{K}_\alpha(I_n) \leq \frac{1}{1-\alpha} \left(\frac{1-\alpha}{\alpha} \right)^\alpha n^{1+\alpha} + o(n^{1+\alpha})$$

PREUVE.

Borne Inférieure

Dans le cas de l'échange total, chaque sommet doit envoyer $n-1$ unité de flots, sur au moins un arc, le coût des arcs sortant de ce sommet est donc au moins $(n-1)^\alpha$. Le coût total du réseau est donc minoré par $n(n-1)^\alpha$.

Borne Supérieure

Une construction composée de cycles de même longueur (ou quasiment de même longueur) tous connectés en un même sommet semble donner une bonne construction. Dans cette construction, le routage par des chemins élémentaires est unique, il est donc facile de calculer le coût d'un tel réseau : si on note $k+1$ la longueur des cycles et supposons que $n-1$ est multiple de k , on a une charge uniforme sur les cycles de $nk - k(k+1)/2$; chaque cycle coûte donc $(k+1)(nk - k(k+1)/2)^\alpha$. Le nombre de cycles étant $(n-1)/k$, le coût total du réseau est

$$(n-1) \frac{k+1}{k} \left(nk - \frac{k(k+1)}{2} \right)^\alpha$$

Ce coût est minimal lorsque $1/\alpha = (k+1)(1 - \frac{k}{2n-k-1})$, la construction de coût minimum est obtenue pour $k = 1/\alpha - 1$, ce qui donne : $(n-1) \frac{1/\alpha}{1/\alpha-1} (n(1/\alpha-1) - 1/\alpha(1/\alpha-1)/2)^\alpha = (n-1) \frac{1}{1-\alpha} \left(\frac{1-\alpha}{\alpha} \right)^\alpha \left(n - \frac{1}{2\alpha} \right)^\alpha = \frac{1}{1-\alpha} \left(\frac{1-\alpha}{\alpha} \right)^\alpha n^{1+\alpha} + o(n^{1+\alpha})$ \square

Proposition 7.3.4 (Meilleure borne inférieure) *Soit I_n une instance de l'échange total unitaire dans le graphe complet à n sommets, si $n \geq 2^{\left(\frac{\alpha}{1-\alpha}\right)}$,*

$$2^\alpha n^{1+\alpha} + o(n^{1+\alpha}) \leq \mathcal{K}_\alpha(I_n)$$

PREUVE. Considérons l'ensemble des chemins d'une solution optimale. Pour compter la charge minimale induite par ces chemins, nous n'allons prendre en compte que la charge induite par le premier et le dernier arc de chaque chemin. L'instance de communication étant l'échange total unitaire, chaque sommet doit donc être adjacent à des arcs entrants (resp. sortant) dont la somme des charges est au moins $(n-1)$; autrement dit il doit y avoir $(n-1)$ extrémités de chemins entrant (resp. sortant) en chacun des sommets.

Notons $c(a)$ la charge induite sur l'arc $a = (u, v)$ par les extrémités des chemins. Nous avons :

$$\mathcal{K}_\alpha(I_n) \geq \sum_{a \in A} c^\alpha(a)$$

Le nombre de chemins sortant de u plus le nombre de chemins entrant en v en utilisant l'arc a est exactement $c(a) + 1$ (plus 1 pour le chemin entre u et v qui passe nécessairement par a si a est de capacité non nulle, d'après le Lemme 7.3.1). On a déjà : $2n(n-1) = \sum_{a \in A} (c(a) + 1)$.

Notons $eff(c(a))$ (l'efficacité d'un arc de capacité a) la valeur $\frac{c(a)+1}{c^\alpha(a)}$.

Nous avons

$$\begin{aligned} \mathcal{K}_\alpha(I_n) &\geq \sum_{a \in A} \frac{c(a) + 1}{eff(a)} \\ &\geq \min_{a \in A} \left\{ \frac{1}{eff(c(a))} \right\} \sum_{a \in A} (c(a) + 1) \\ &\geq \min_{a \in A} \left\{ \frac{1}{eff(c(a))} \right\} 2n(n-1) \end{aligned}$$

La valeur de $c(a)$ ne peut pas dépasser $2(n-1) - 1$ (le nombre de chemins qui peuvent partir de u plus le nombre de chemins qui peuvent arriver en v moins celui qui va de u à v), donc

$$\mathcal{K}_\alpha(I_n) \geq \frac{2n(n-1)}{\max_{1 \leq c \leq 2(n-1)-1} eff(c)}$$

La fonction eff décroît sur l'intervalle $\left[1, \frac{1}{1-\alpha} \left(\frac{1-\alpha}{\alpha}\right)^\alpha\right]$ puis croît sur l'intervalle $\left[\frac{1}{1-\alpha} \left(\frac{1-\alpha}{\alpha}\right)^\alpha, +\infty\right]$. Le maximum de la fonction eff est donc atteint en 1 ou en $2(n-1) - 1$. Donc

$$\mathcal{K}_\alpha(I_n) \geq \frac{2n(n-1)}{\max \left\{ 2, \frac{2(n-1)}{(2(n-1)-1)^\alpha} \right\}}$$

On a donc, si $n \geq 2^{\left(\frac{\alpha}{1-\alpha}\right)}$,

$$\mathcal{K}_\alpha(I_n) \geq 2^\alpha n^{1+\alpha} + o(n^{1+\alpha})$$

□

REMARQUE. La borne inférieure de la proposition 7.3.4 peut être généralisée aux cas où chaque sommet est la source de d requêtes ainsi que la destination de d requêtes : La capacité $c(a)$ est majorée par $2d - 1$ et donc l'efficacité par $\frac{2d}{(2d+1)^\alpha}$, en utilisant les mêmes arguments on a un coût supérieur à $\frac{nd}{\max\left\{2, \frac{2d}{(2d+1)^\alpha}\right\}} \leq 2^\alpha d^\alpha n + o(d^\alpha n)$.

Nous conjecturons que les constructions données pour le calcul des bornes supérieures sont les constructions optimales.

Si les requêtes ne sont pas unitaires mais que la somme de leur débit associés et la même que celles de l'instance d'échange total unitaire alors nous conjecturons que le coût optimal est inférieur à celui de l'échange total unitaire :

Conjecture 7.3.5 *Soit I une instance de communications telles que $\sum_{R_i \in I} r_i = n(n-1)$, alors $K_\alpha(I) \leq K_\alpha(I_n)$.*

Ayant une bonne approximation du coût optimal d'une construction avec des capacités réelles, il serait intéressant d'estimer le coût optimale d'une construction n'ayant que des capacités discrètes de la forme $\{1, 4, 16, 64\}$, c'est à dire en puissance croissante, vu que celà correspond aux capacités utilisées réellement dans les réseaux SDH.

Conclusion et perspectives

Nous avons étudié dans cette thèse des problèmes d'optimisation modélisant l'établissement de réseaux virtuels de télécommunications et en particulier le problème dit de «positionnement de chemin virtuel» (VPL) issu de la modélisation des communications dans les réseaux ATM, récemment étudié par la communauté scientifique.

Ce problème est un cas particulier d'empilement de réseaux où l'on cherche à réaliser les communications sur un réseau virtuel avec une qualité de service et à plonger ce réseau virtuel dans le réseau physique avec des contraintes de charge. Nous avons pris une approche «simple» de la qualité de service représentée par le nombre de sauts (distance entre deux nœuds dans le réseau virtuel) et la charge maximale (nombre de lien du réseau virtuel utilisant un même lien physique). Avec ces paramètres «simples», les problèmes à résoudre, tant pour des instances de communication particulières que pour des réseaux particuliers, sont «difficiles».

Ainsi dans le cas où le réseau physique est une chaîne ou un cycle avec l'instance de communication d'échange total, nous avons obtenus des résultats quasi-optimaux pour la chaîne quel que soit le nombre maximal de sauts et, pour le cycle, si le nombre de sauts est 2 (il resterait encore à déterminer la valeur exacte dans deux cas de congruence sur le nombre de sommets du cycle, voir la conjecture 3.2.3). Pour le cycle, si le nombre h de sauts maximum (pour réaliser l'instance de communication d'échange total) est supérieur à 2, nous savons que la valeur optimale de la charge est de la forme $c_h n^{2/h} + o(n^{2/h})$ mais nous n'avons qu'un encadrement de la valeur de c_h (il peut être intéressant de déterminer exactement la valeur de c_h).

Dans le quatrième chapitre, les réseaux physiques considérés sont les arbres. Nous donnons des algorithmes polynomiaux permettant de trouver les solutions optimales pour les instances relativement quelconques (*Many – To – Many* avec un ensemble d'initiateurs de taille bornée) après avoir présenté la méthode utilisée pour des cas plus simples : la diffusion en 2 sauts, la diffusion en h sauts et la diffusion depuis deux sources en h sauts.

L'étude de la complexité de ce problème dans le cas général (réseau quelconque, instance d'échange total) est un problème ouvert. Des travaux pour trouver un algorithme polynomial dans les arbres pour l'instance d'échange total et un nombre de sauts maximal de 3 sont en cours. Dans le but de converger vers le cas général, il serait intéressant de savoir où est la limite en terme d'instance de requêtes et de nombre maximum de sauts de la technique utilisée dans les algorithmes ainsi que de trouver d'autres moyens de construire des solutions optimales.

Plus généralement, il serait intéressant de prendre en compte d'autres critères de qualité de service.

Dans ce modèle, la capacité des liaisons physiques est prise en compte mais pas la capacité des liaisons virtuelles ; ainsi la quantité de trafic envoyée dans ces liaisons virtuelles peut être illimitée or ce n'est pas le cas dans la réalité. Il faudrait donc établir un modèle permettant d'intégrer cette composante, où la charge dans le réseau physique dépendrait de la charge des liens virtuels, ceci comprend entre autres des problèmes de «groupage» de trafic (qui sont étudiés par ailleurs dans le projet).

La tolérance aux pannes n'est pas non plus prise en compte dans ce modèle, si un lien (physique ou virtuel) vient à être défaillant, la connectivité n'est plus assurée. On peut donc étudier le problème consistant à établir des réseaux virtuels dans lesquels un lien défaillant peut être compensé par d'autres liens en minimisant le nombre de connexions perturbées ou le nombre de modifications à apporter pour rétablir toutes les connexions.

Une extension du modèle classique est donnée dans le cinquième chapitre, elle consiste à considérer des chemins virtuels optiques pour faire un lien avec un autre problème classique qui est l'allocation de fréquences dans les réseaux de communications optiques de type WDM. On remarque déjà que certains résultats existant dans le contexte des réseaux WDM sans conversion ne peuvent s'étendre à ce nouveau modèle.

En marge du problème du VPL, nous avons étudié deux problèmes d'optimisation dans les réseaux hiérarchiques. Les résultats obtenus dans le premier (chapitre 6) sont une estimation fine du nombre de sommets dans un réseau en anneaux en fonction du diamètre ainsi que les constructions optimales pour une structure quasi-identique, améliorant ainsi un résultat connu. Le second problème (chapitre 7) a été posé par France Telecom R&D dans le cadre de la CTI «Dimensionnement pour les services SDH aux entreprises». Nous donnons une modélisation d'un problème de minimisation du coût d'une boucle SDH étant donné que le coût d'un lien suit une fonction concave (et croissante) de sa capacité. Nous donnons des bornes inférieures dans le cas où la fonction de coût est continue ainsi que des solutions (mais pas toujours optimales) permettant de satisfaire les requêtes. Les travaux futurs sur ce problème consisteront à adapter

les résultats au cas où la fonction de coût n'est pas continue mais en escalier, ce qui correspond à la réalité des contraintes de la SDH et à faire converger le coût des solutions proposées et les bornes inférieures dans le cas continu.

Bibliographie

- [ABC⁺99] W. Aiello, S. N. Bhatt, F. R. K. Chung, A. L. Rosenberg, and R. K. Sitaraman. Augmented ring networks. In *The 6th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 1–16. Carleton Scientific, 1999.
- [BBGG00] L. Becchetti, P. Bertolazzi, Carlo Gaibisso, and Giorgio Gambosi. On the design of efficient ATM routing schemes. Une version courte est déjà publiée dans [BG97b], 2000.
- [BCCT01] J-C. Bermond, L. Chacon, D. Coudert, and F. Tillerot. Cycle Covering. In *International Colloquium on Structural Information and Communication Complexity – SIROCCO*, pages 21–34, 27-29 June 2001.
- [BCG97] M. Burlet, P. Chanas, and O. Goldschmidt. Minimum diameter vp network design. In *INFORM'S FALL*, 1997.
- [BCG99] M. Burlet, P. Chanas, and O. Goldschmidt. Virtual path layout design in atm networks. In *INFORMS FALL '99*, 1999.
- [BCML02] J-C. Bermond, D. Coudert, and M-L.Yu. On DRC-covering of K_n by cycles. *Journal of Combinatorial Designs to appear*, 2002.
- [BCP02] J-C. Bermond, S. Choplin, and S. Pérennes. Hierarchical ring network design. In *9th International Colloquium on Structural Information and Communication Complexity (SIROCCO'02)*, pages 1–16, 2002.
- [Bea00] B. Beauquier. *Communications dans les réseaux optiques par multiplexage en longueur d'onde*. PhD thesis, Université de Nice - Sophia Antipolis, janvier 2000.
- [Ber83] C. Berge. *Graphes*. Gauthiers-Villars, 1983.
- [BG97a] Becchetti and Gaibisso. Lower bounds for the virtual path layout problem in ATM networks. In *Theory and Practice of Informatics, Seminar on Current Trends in Theory and Practice of Informatics, LNCS*, volume 24. Springer-Verlag, 1997.
- [BG97b] L. Becchetti and C. Gaibisso. Lower bounds for the virtual path layout problem in ATM networks. *Lecture Notes in Computer Science*, 1338 :375 – 382, 1997.

- [BGP⁺00] J-C. Bermond, L. Gargano, S. Pérennes, A.A. Rescigno, and U. Vaccaro. Efficient collective communications in optical networks. *Theoretical computer science*, 233 :165–189, 2000.
- [BHP98] B. Beauquier, P. Hell, and S. Pérennes. Optimal wavelentgh-routed multicasting. *Discrete Applied Mathematics*, 84 :15–20, 1998.
- [BM76] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. MacMillan Press, 1976.
- [BMPP98] J-C. Bermond, N. Marlin, D. Peleg, and S. Pérennes. Directed Virtual Path layout in ATM networks. In *Proc. of the 12th International Conference on Distributed Computing, Andros Greece, 1998*. LNCS 1499, pages 75-88.
- [BMPP99] J-C. Bermond, N. Marlin, D. Peleg, and S. Pérennes. Virtual Paths Layout with Low Congestion or Low Diameter in ATM Networks. In *1^{ère} rencontre francophone... ALGOTEL '99, Mai 1999*.
- [CCRS87] F. R. K. Chung, E. G. Coffman, M. I. Reiman, and B. E. Simon. The forwarding index of communication networks. *IEEE Transactions on Information Theory*, 33 :224–232, 1987.
- [CGZ94] I. Cidon, O. Gerstel, and S. Zaks. A scalable approach to routing in ATM networks. In *Proc. of the 8th International Workshop on Distributed Algorithms, WDAG '94*, volume 857 of *Lecture Notes in Computer Science*, pages 209–222. Springer Verlag, 1994. La version longue est [CGZ96].
- [CGZ96] I. Cidon, O. Gerstel, and S. Zaks. The layout of virtual paths in ATM networks. In *ACM/IEEE Transaction on networking*, volume 4 – 6, pages 873 – 884, 1996. La version courte est [CGZ94].
- [Cha98] P. Chanas. *Réseaux ATM : conception et optimisation*. PhD thesis, Université J. Fourier, 1998.
- [Cho01] S. Choplin. Virtual path layout in ATM path with given hop count. *Lecture Notes in Computer Science*, 2094 :527–537, 2001.
- [DFZ97a] Y. Dinitz, M. Feighelstein, and S. Zaks. On optimal graphs embedded into paths and rings, with analysis using l1-Spheres. *Lecture Notes in Computer Science*, 1335 :171–183, 1997.
- [DFZ97b] Yefim Dinitz, Marcelo Feighlstein, and Shmuel Zaks. On optimal graphs embedded into paths and rings, with analysis using l1-spheres. In *Proc. of the 23rd Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 1335 of *Lecture Notes in Computer Science*, pages 171–183. Springer Verlag, 1997.
- [dR94] Jean de Rumeur. *Communications dans les Réseaux de Processeurs*. Collection Études et Recherche en Informatique. Masson, Paris, 1994.

- [EFZ98] T. Eilam, M. Flammini, and S. Zaks. Complete characterization of the path layout construction problem for ATM networks with given hop count and load. *Parallel Processing Letters*, 8(2) :207–220, 1998.
- [FNP97] M. Flammini, E. Nardelli, and G. Proietti. ATM layouts with bounded hop count and congestion. In *Proc. of the 11th International Workshop on Distributed Algorithms (WDAG)*, volume 1320 of *Lecture Notes in Computer Science*, pages 52–65. Springer Verlag, 1997.
- [FZ97] M. Feighlstein and S. Zaks. Duality in chain ATM virtual path layouts. In *Proc. of the 4th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Monte Verita, Ascona, Switzerland, July 1997.
- [GCZ96] O. Gerstel, I. Cidon, and S. Zaks. Optimal virtual path layout in ATM networks with shared routing table switches. *Chicago Journal of Theoretical Computer Science*, October 1996. Published in *Selected Papers from PODC 1994*, David Peleg editor.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1979.
- [GWZ95a] O. Gerstel, A. Wool, and S. Zaks. Optimal layouts on a chain ATM network. In *3rd Annual European Symposium on Algorithms (ESA), (LNCS 979), Corfu, Greece*, pages 508–522, 1995. To appear in *Discrete Applied Mathematics*.
- [GWZ95b] O. Gerstel, A. Wool, and S. Zaks. Optimal layouts on a chain ATM network (extended abstract). In *Proc. of the 3rd Annual European Symposium on Algorithms (ESA'95)*, volume 979 of *Lecture Notes in Computer Science*, pages 508–522. Springer Verlag, 1995. La version longue est [GWZ98].
- [GWZ98] Gerstel, Wool, and Zaks. Optimal layouts on a chain ATM network. *DAMATH : Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 83, 1998. La version courte est [GWZ95b].
- [GZ94a] O. Gerstel and S. Zaks. The virtual path layout problem in fast networks (extended abstract). In *Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 235–243, Los Angeles, California, 14–17 August 1994.
- [GZ94b] O. Gerstel and S. Zaks. The virtual path layout problem in fast networks (extended abstract). In *Proc. of 13th Annual ACM Symposium on Principles of Distributed Computing (PODC'94)*, pages 235–243, USA, August 1994. ACM Press.
- [HMS89] M-C. Heydemann, J-C. Meyer, and D. Sotteau. On forwarding indices of networks. *Discrete Applied Mathematics*, 23 :103–123, 1989.

- [JF56] L. R. Ford Jr. and D. R. Fulkerson. Maximal flow through a network. *Canadian J. Math.*, 8 :399–404, 1956.
- [KG98] D. Kofman and M. Gagnaire. *Réseaux Haut Débit, réseaux ATM, réseaux locaux et réseaux tout-optiques*. InterEditions-Masson, 1998. 2eme édition.
- [Mar00] N. Marlin. *Communications Structurées dans les Réseaux*. PhD thesis, Université de Nice - Sophia Antipolis, <http://www.inria.fr/RRRT/TU-0631.html>, juin 2000.
- [Min89] M. Minoux. Network synthesis and optimum network design problems : Models, solution methods and applications. *Networks*, 19 :313–360, 1989.
- [SB99] T. Stern and K. Bala. *Multiwavelength Optical Networks, A Layered Approach*. Addison-Wesley, 1999.
- [Sta] L. Stacho. Virtual path layouts in atm networks.
- [SV96] L. Stacho and I. Vr̂to. Virtual path layouts for some bounded degree networks. In *Proc. of the 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 269–278. Carleton University Press, 1996.
- [Tan95] A. S. Tanenbaum. *Computer Networks*. Eastern Economy Edition. Prentice-Hall(India) Pvt Ltd, New Delhi, 2 edition, 1995. Purchase Date Approximate.
- [Zak97] S. Zaks. Path layout in ATM networks. *Lecture Notes in Computer Science*, 1338 :144–??, 1997.

ANNEXES

Annexe A

Notions élémentaires de théorie de graphes

Nous donnons seulement les définitions nécessaires.

A.1 Définitions

- Un **graphe orienté** (en anglais, *digraph*) $G = (V, A)$ est constitué d'un ensemble fini $V = \{x_1, x_2, \dots, x_N\}$ d'éléments, appelés **sommets**, et d'une famille finie $A = \{a_1, a_2, \dots, a_m\}$ de représentants de couples de sommets, appelés **arcs**.
- Un graphe orienté est dit **symétrique** si l'existence d'un arc $a = (x, y)$ implique l'existence de l'arc $a' = (y, x)$.
- Notons qu'un couple de sommets $(x, y) \in V \times V$ peut être représenté plusieurs fois dans la famille A ; on parle alors d'**arc multiple** (x, y) et de **multigraphe** G .
- Un **graphe non orienté** $G = (V, E)$ est constitué d'un ensemble fini V de sommets et d'une famille finie $E = \{e_1, e_2, \dots, e_m\}$ de représentants de paires de sommets, appelées **arêtes**.
- Étant donné un graphe non orienté G , on note G^* le graphe orienté symétrique ayant les mêmes sommets que G et les arcs (x, y) et (y, x) si et seulement si $[x, y]$ est une arête de G .
- Si $e = [x, y]$ est une arête, les sommets x et y sont dits **adjacents** l'un à l'autre et l'arête e est dite **incidente** à x et y .

- Un arc de la forme (x, x) ou une arête de la forme $[x, x]$ est appelé(e) **boucle**.
- Un (multi)graphe ne comportant ni boucle, ni arc ou arête multiple, est appelé **graphe simple**.
- Nous appellerons par la suite **graphes** (sans autre adjectif) les graphes simples non orientés.

Degré

- On appelle **degré sortant** (resp. **entrant**) d'un sommet x dans un graphe orienté, noté $d^+(x)$ (resp. $d^-(x)$) le nombre d'arcs d'extrémité initiale (resp. finale) x .
- On appelle **degré** d'un sommet x dans un graphe, noté $d(x)$, le nombre d'arêtes incidentes à x .
- On appelle **degré maximum** (resp. **minimum**) d'un graphe, noté Δ (resp. δ), le maximum (resp. minimum) des degrés des sommets.

Chemins, chaînes

- On appelle **chemin** (*dipath*) dans un graphe orienté une suite $P = (a_1, a_2, \dots, a_q)$ d'arcs, telle que l'extrémité finale de a_i est l'extrémité initiale de a_{i+1} , pour $1 \leq i < q$. La **longueur** du chemin P est alors le nombre d'arcs qui le composent.
- On peut également définir un chemin $P = (a_1, a_2, \dots, a_q)$ dans un graphe orienté simple par la suite des sommets (x_0, x_1, \dots, x_q) , telle que $a_i = (x_{i-1}, x_i)$. On dit alors que P est un chemin de x_1 vers x_q .
- On appelle **chaîne** (*path*) dans un graphe non orienté une suite $P = (e_1, e_2, \dots, e_q)$ d'arêtes, telle que deux arêtes consécutives sont incidentes à un même sommet. La **longueur** de la chaîne P est alors le nombre d'arêtes qui la composent.
- On s'autorise par la suite à appeler une chaîne un chemin, la distinction dépendant de la nature du graphe considéré.
- Un chemin qui ne comporte pas deux fois le même lien est dit **simple**. Un chemin qui ne comporte pas deux fois le même sommet est dit **élémentaire**.

- On appelle **circuit** dans un graphe orienté un chemin d'un sommet vers lui-même.
- On appelle **cycle** dans un graphe une chaîne d'un sommet vers lui-même.

Distances

- On appelle **distance** entre deux sommets x et y dans un graphe (orienté ou non), notée $d(x, y)$, la longueur d'un plus court chemin de x vers y (cette longueur étant prise égale à $+\infty$ s'il n'existe pas de tels chemins). Notons que dans le cas orienté ce n'est pas une distance au sens topologique du terme car l'égalité $d(x, y) = d(y, x)$ n'est pas forcément vérifiée.
- On appelle **excentricité** d'un sommet x la plus grande distance entre x et les autres sommets du graphe.
- On appelle **diamètre** d'un graphe (orienté ou non), noté D , le maximum des distances entre les sommets.

Propriétés

- Un graphe non orienté est dit **connexe** s'il existe une chaîne entre toute paire de sommets.
- Un graphe orienté est dit **fortement connexe** s'il existe un chemin de tout sommet vers tout autre sommet.
- Deux chaînes (resp. chemins) sont dits arêtes (resp. arcs) disjoints d'ils n'ont pas d'arête (resp. arc) en commun.

A.2 Graphes classiques

La chaîne. On note P_N la **chaîne** (élémentaire) d'ordre N , de longueur $N - 1$ (voir figure A.1).



FIG. A.1 – Chaîne à 6 sommets P_6

Le cycle. On note C_N le **cycle** (ou **anneau**) (élémentaire) d'ordre N , de longueur N (voir figure A.2).

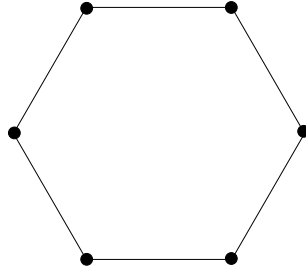


FIG. A.2 – Cycle à 6 sommets C_6

Le graphe complet. On note K_N le **graphe complet** d'ordre N , ayant N sommets deux à deux adjacents (voir figure A.3). On note K_N^* la version orientée.

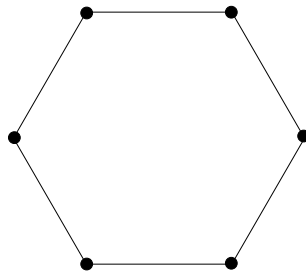


FIG. A.3 – Graphe complet à 6 sommets K_6

L'arbre. Un **arbre** est un graphe connexe sans cycle. Souvent, les arbres sont dits **enracinés** ; cela signifie qu'un sommet appelé **racine** est distingué. Soit $[x, y]$ une arête d'un arbre enraciné en r avec $d(r, y) = d(r, x) + 1$, on dit que y est un **fil** de x et que x est le **père** de y (voir figure A.4).

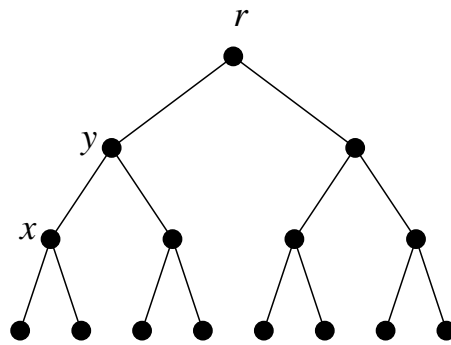


FIG. A.4 – Arbre enraciné

Les arbres de cycle. Un **arbre de cycles** est un graphe formé par une union de cycles qui s'intersectent deux à deux en au plus un sommet, et tel que deux sommets quelconques peuvent être reliés par exactement deux chaînes élémentaires arête-disjointes (voir figure A.5).

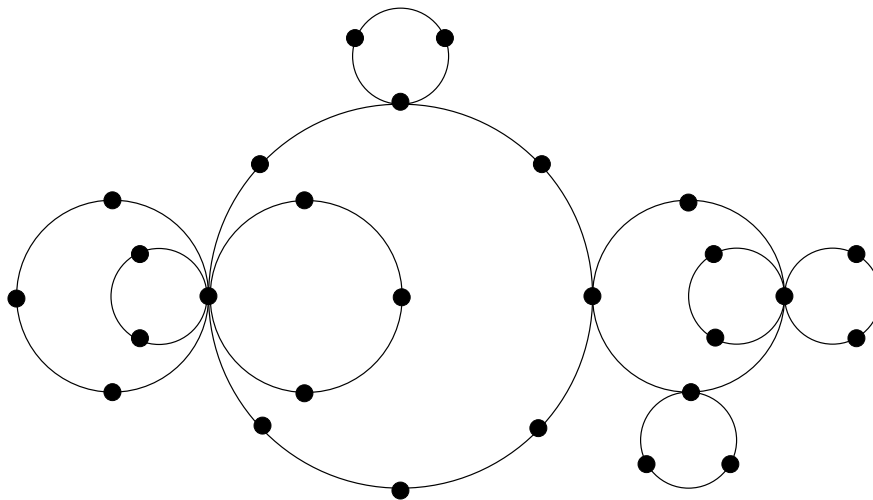


FIG. A.5 – Arbre de cycles.

Annexe B

Algorithmes pour les problèmes du VPL

B.1 Instance *One – to – All* en h sauts

Algorithme 5 Calcule les traces des feuilles pour le diffusion dans l'arbre (avec construction de la solution)

Calcule_*Traces_feuilles*(Arbre T enraciné en r, v sommet de T, c charge maximale)

```
si  $v$  est une feuille alors
   $Traces(v) \leftarrow \emptyset$ 
pour tout  $i$  de 1 à  $h$  faire
   $Q_i \leftarrow \emptyset$ 
   $P_{i-1} \leftarrow \emptyset$ 
fin pour
si  $v$  est la source de la diffusion alors
  trace  $t \leftarrow (\{v\} (c \text{ fois}), \emptyset_{1 \leq i \leq h-1}, \emptyset_{1 \leq i \leq h}, \emptyset)$ 
   $Traces(v) \leftarrow Traces(v) \cup Traces\_Possibles(t, c)$ 
sinon
  pour tout  $d$  de 1 à  $h$  /*choix de la distance du sommet  $v$  à la source*/
  faire
  trace  $t \leftarrow (\emptyset_{0 \leq i \leq d-1}, \{v\} (c \text{ fois}), \emptyset_{d+1 \leq i \leq h-1}, \emptyset_{1 \leq i \leq d-1}, \{v\}, \emptyset_{d+1 \leq i \leq h}, \emptyset)$ 
   $Traces(v) \leftarrow Traces(v) \cup Traces\_Possibles(t, c)$ 
  fin pour
fin si
fin si
```

Algorithme 6 Renvoie toutes les traces possibles respectant la charge à partir d'une trace temporaire

Traces_Possibles(trace t , charge maximale c)

$Traces \leftarrow \emptyset$

pour tout p_0, p_1, \dots, p_h tels que $\forall i, 0 \leq p_i \leq |P_i|$ et $\sum_{0 \leq i \leq h} p_i + \sum_{1 \leq i \leq h} |t.Q_i| \leq c$ **faire**

$t_{tmp} \leftarrow t$

$cut(t_{tmp}.P_i, p_i)$ /* on ne garde que les p_i premiers sommets */

$Traces \leftarrow Traces \cup \{t_{tmp}\}$

fin pour

Renvoie($Traces$)

Algorithme 7 Combine deux tables de traces pour h sauts, avec la source n'importe où dans l'arbre et la construction du graphe virtuel

Combine_Tables(Table T_1 , Table T_2) renvoie une table.

$T_Resultat \leftarrow \emptyset$

pour tout combinaisons (t_1, t_2) de traces de T_1 et T_2 **faire**

$E \leftarrow t_1.E \cup t_2.E$

pour tout d de 1 à h **faire**

$P_{d-1} \leftarrow t_1.P_{d-1} \cup t_2.P_{d-1}$

$Q_d \leftarrow t_1.Q_d \cup t_2.Q_d$

tant que $P_{d-1} \neq \emptyset$ et $Q_d \neq \emptyset$ **faire**

$sp = pop(P_{d-1})$

$sq = pop(Q_d)$

$E \leftarrow E \cup [sp, sq]$

fin tant que

fin pour

$T_Resultat \leftarrow T_Resultat \cup (\{P_i\}_{0 \leq i \leq h-1}, \{Q_i\}_{1 \leq i \leq h}, E)$

fin pour

Renvoie $T_Resultat$

Algorithme 8 Calcule les traces avec construction de la solution pour le diffusion pour les sommets qui ne sont pas des feuilles de l'arbre

Calcule_Traces_feuilles(Arbre T enracine en r, v sommet de T, c charge maximale)

```

Traces( $v$ )  $\leftarrow \emptyset$ 
Traces_temp  $\leftarrow \emptyset$ 
pour tout fils  $v'$  de  $v$  faire
    Calcule_Traces( $T, v', c$ )
    Traces_temp  $\leftarrow$  Combine_Tables(Traces_temp, Traces( $v'$ ))
fin pour
pour tout Traces  $t$  de Traces_temp faire
    /* On connecte à  $v$  tous les sommets du sous-arbre à distance  $d - 1$  de la
    source et on genere toutes les traces possibles respectant la charge*/
    si  $v$  est la source de la diffusion alors
        pour tout  $x \in Q_1$  faire  $t.E \leftarrow t.E \cup [v, x]$  fin pour
         $Q_1 \leftarrow \emptyset$ 
        /*  $v$  pourra connecter au plus  $c$  sommets à distance 1 en dehors du sous-
        arbre */
         $t.P_0 \leftarrow t.P_0 \cup \{\{v\}\}$  ( $c$  fois)
        Traces( $v$ )  $\leftarrow$  Traces( $v$ )  $\cup$  Traces_Possibles( $t, c$ )
    sinon
        pour tout  $d$  de 1 à  $h$  /*choix de la distance du sommet  $v$  à la source*/
        faire
             $t_{tmp} \leftarrow t$ 
            si  $|t_{tmp}.P_{d-1}| \geq 1$  /*on peut connecter  $v$  dans le sous-arbre*/ alors
                 $x \leftarrow pop(P_{d-1})$ 
                 $t_{tmp}.E \leftarrow t_{tmp}.E \cup [x, v]$ 
            sinon
                 $t_{tmp}.Q_d \leftarrow t_{tmp}.Q_d \cup \{v\}$ 
            fin si
            /* connexions à  $v$  dans le sous-arbre */
            si  $d \leq h - 1$  alors
                pour tout  $x \in Q_{d+1}$  faire  $t_{tmp}.E \leftarrow t_{tmp}.E \cup [v, x]$  fin pour
            fin si
            /*  $v$  pourra connecter au plus  $c$  sommets à distance  $d+1$  */
             $t_{tmp}.P_d \leftarrow t_{tmp}.P_d \cup \{\{v\}\}$  ( $c$  fois)
             $Q_{d+1} \leftarrow \emptyset$ 
            Traces( $v$ )  $\leftarrow$  Traces( $v$ )  $\cup$  Traces_Possibles( $t_{tmp}, c$ )
        fin pour
    fin si
fin pour

```

B.2 Programme linéaire pour une instance quelconque en h sauts

Programme Linéaire 2 Résolution du problème de minimisation de la charge dans l'arbre pour une instance quelconque I et au plus h sauts

Description des variables :

$$\begin{aligned}
 X[a, b] &= \begin{cases} \text{indicateur de présence d'une arête virtuelle} \\ 1 \text{ si } [a, b] \in E' \\ 0 \text{ sinon} \end{cases} \\
 X_s^i(a, b) &= \begin{cases} \text{propagation du flot issu de la } s \\ > 0 \text{ si } [a, b] \in E', \text{ } dist(s, b) = i \text{ et } dist(s, a) = i - 1 \\ = 0 \text{ sinon} \end{cases} \\
 L_{[u, v]} &= \text{charge sur l'arête physique } [a, b] \\
 L_{max} &= \text{charge maximale}
 \end{aligned}$$

Minimiser : L_{max}

Avec les contraintes :

$$X_s^i(a, b) \leq \sum_{x \in V \setminus \{b\}} X_s^{i-1}(x, a), \forall (a, b) \in V \times V, \forall s \in S, \forall 2 \leq i \leq h \quad (\text{B.1})$$

$$1 \leq \sum_{a \in V \setminus \{b\}, 1 \leq i \leq h} X_s^i(a, b), \forall s \in S, \forall b \in V \text{ tel que } [s, b] \in I \quad (\text{B.2})$$

$$X[a, b] \geq \sum_{1 \leq i \leq h} (X_s^i(a, b) + X_s^i(b, a)), \forall s \in S, \forall [a, b] \in V \times V. \quad (\text{B.3})$$

$$L_{[u, v]} \geq \sum_{[a, b] \in V \times V, [u, v] \in \mathcal{P}([a, b])} X[a, b], \forall [a, b] \in E \quad (\text{B.4})$$

$$L_{max} \geq L_{[u, v]}, \forall [u, v] \in E \quad (\text{B.5})$$

$$X_s^1(a, b) = 0, \forall s \in S, \forall a \neq s, \forall b \in V \quad (\text{B.6})$$

$$X[a, b] \in \{0, 1\} \quad (\text{B.7})$$

Nombre de variables :

de type $X[a, b]$: $\frac{|V|(|V|-1)}{2}$

de type $X_s^i(a, b)$: $h|S||V|(|V|-1)$

de type $L[u, v]$: $|E|$

Soit $O(h|S||V|^2)$ variables fractionnaires et $O(|V|^2)$ variables binaires.

Nombre de contraintes :

de type B.3 : $\frac{|V|(|V|-1)}{2}$

de type B.1 : $(h-1)|S||V|(|V|-1)$

de type B.2 : $|I|$

de type B.4 : $\frac{|V|(|V|-1)}{2}$

de type B.5 : $|E|$

Soit $O(h|S||V|^2)$ contraintes, avec $|S|$ la valeur d'une 2-approximation du «vertex cover» sur le graphe des requêtes.

Résumé

Les résultats obtenus dans cette thèse portent sur le **dimensionnement de réseaux virtuels de télécommunications**.

Dans le **chapitre 1**, nous présentons brièvement la technologie des réseaux étudiés. Le **chapitre 2** est consacré à la modélisation des réseaux de télécommunications à l'aide de la théorie des graphes.

Les **chapitres 3 et 4** traitent du problème du positionnement de chemins virtuels qui consiste à trouver un graphe ayant certaines propriétés tel que son plongement dans un graphe donné soit de congestion minimum. Pour les arbres, nous donnons des algorithmes polynomiaux permettant de trouver une solution optimale lorsque le nombre de sauts est fixé. Dans le **chapitre 5** est introduit une extension optique de ce modèle.

Le **chapitre 6** est consacré aux réseaux hiérarchiques en anneaux. Le problème de maximisation du nombre de sommets d'une telle structure ayant un diamètre donné est résolu. Dans le **chapitre 7**, nous étudions un problème d'optimisation lié à la tarification d'une boucle SDH.

Mots clés : Positionnement de chemins virtuels, réseaux virtuels, réseaux hiérarchiques, ATM, boucle SDH, routage, graphe, plongement, diamètre, diffusion, échange total.

Abstract

This thesis deals with **virtual telecommunication networks design**. In **chapter 1**, we briefly present the technology of the studied networks. **Chapter 2** is devoted to the modelization of the telecommunication networks using the graph theory.

Chapters 3 and 4 deal with the Virtual Path Layout problem which consists in finding a graph having given properties such as its embedding in a given graph has a minimum congestion. For trees, we give polynomial algorithms allowing to find an optimal solution when the maximum hop count is fixed. In **chapter 5** we introduce an optical extension of this model.

Chapter 6 is devoted to hierarchical ring networks. The problem which consists in the maximization of the number of nodes of such a structure having a given diameter is solved. In **chapter 7**, we study an optimization problem involved in the pricing of a SDH loop.

Keywords : Virtual Path Layout, virtual Network, hierarchical network, ATM, SDH loop, routing, graph, embedding, diameter, broadcasting, gossiping.