



HAL
open science

Conception et optimisation robuste des réseaux de télécommunications

Mohamed Zied Ben Hamouda

► **To cite this version:**

Mohamed Zied Ben Hamouda. Conception et optimisation robuste des réseaux de télécommunications. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2010. Français. NNT: . tel-00512347

HAL Id: tel-00512347

<https://theses.hal.science/tel-00512347>

Submitted on 30 Aug 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.

UNIVERSITE de TOULOUSE - PAUL SABATIER
UFR Mathématiques, Informatique, Gestion

THESE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITE DE TOULOUSE

Délivré par l'Université Toulouse - Paul Sabatier
Discipline ou spécialité : Systèmes Informatiques

Présentée et soutenue par Zied Ben Hamouda
Le 25 juin 2010

Titre :

**Conception et Optimisation Robuste
des Réseaux de Télécommunications**

JURY

Christian BES (professeur)
Bernard FORTZ (chargé de cours)
Philippe MAHEY (professeur)
Prosper CHEMAUIL (directeur de recherche)
Olivier KLOPFENSTEIN (ingénieur de recherche)
Michel DIAZ (directeur de recherche)
Jean-Marie GARCIA (directeur de recherche)
Olivier BRUN (chargé de recherche)

Ecole doctorale : EDSYS
Unité de recherche : LAAS-CNRS groupe MRS
Directeur de Thèse : Jean-Marie GARCIA

Remerciements

Ce mémoire est l'aboutissement de trois années d'études effectuées au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS (LAAS-CNRS) au sein du groupe de Modélisation et contrôle des Réseaux et des Signaux (MRS) en partenariat avec la société QoS Design.

Je tiens à exprimer toute ma reconnaissance à Raja CHATILA, directeur du LAAS, pour m'avoir permis de réaliser l'ensemble de ces travaux dans d'excellentes conditions. Je voudrais également signaler ici l'exceptionnelle qualité des moyens informatiques et logistiques mis à disposition des chercheurs dans ce laboratoire.

J'adresse tous mes remerciements à Monsieur Jean-Marie GARCIA, directeur et fondateur de la société QoS Design, pour avoir dirigé cette thèse. Je le remercie plus particulièrement pour sa patience, la qualité de son encadrement et ses conseils, aussi bien scientifiques que personnels, qu'il a pu me prodiguer durant les années que j'ai passées au LAAS. Je tiens ici à lui témoigner mon amitié et mon respect.

Je remercie Monsieur Christian BES, Professeur à l'Université de Toulouse, pour avoir accepté de présider le jury de cette thèse.

Je suis très particulièrement reconnaissant envers Monsieur Bernard FORTZ, chargé de cours à l'Université Libre de Bruxelles, et Monsieur Philippe MAHEY, professeur à l'ISIMA, qui m'ont fait l'honneur d'être les rapporteurs de cette thèse.

Je remercie, Monsieur Michel DIAZ, Monsieur Prosper CHEMOUIL et Monsieur Olivier KLOPFENSTEIN en leurs qualités de membre du Jury.

Les remerciements exprimés ici ne seront jamais à la hauteur de son implication dans ce travail de thèse. J'exprime toute ma reconnaissance pour son aide, ses nombreux conseils, sa disponibilité, son dynamisme et sa bonne humeur à Monsieur Olivier BRUN. Qu'il sois assuré, de tout mon respect et de

ma profonde gratitude.

Je remercie toute l'équipe de QoS Design, pour leur rigueur et leur sérieux ainsi que pour les agréables moments passés ensemble. Une attention spéciale pour Laurent, Cédric, Charly, Anouar et David.

Je remercie également tous les membres du LAAS, pour leur accueil, et particulièrement les nombreux amis que j'y ai rencontré.

Je remercie, Messieurs Hakan Umit et Fausto Pascali de l'Université Libre de Bruxelles pour leurs aides sur certaines parties de ce travail.

Enfin, j'exprime ma dernière pensée à mes parents pour l'aide, la confiance et le soutien dont ils ont fait preuve tout au long de ces dernières années. Mais aussi à mes deux soeurs et mon frère.

Mes remerciements vont aussi à tous mes amis en dehors du LAAS, auprès desquels j'ai passé d'agréables moments. J'ai beaucoup appris de la vie auprès des plus proches.

Les derniers remerciements vont, à tout le personnel technique du LAAS dont le concours a grandement facilité la réalisation de ce mémoire.

Zied BEN HAMOUDA. Avril 2010.

Résumé

Les réseaux de communication devenant de plus en plus présents dans nos activités quotidiennes, l'interruption des services fournis par le réseau, ou même une dégradation significative de leurs qualités, deviennent de moins en moins tolérables. Une conception robuste des réseaux de communication, anticipant les pannes éventuelles d'équipements ou les variations sur la demande en trafic, devient donc de plus en plus nécessaire. Cette thèse traite de plusieurs problèmes de conception et de planification robustes apparaissant dans le domaine des télécommunications.

Nous étudions tout d'abord le problème de la conception et du dimensionnement d'une topologie de communication résilience. Nous proposons un nouveau modèle de conception de réseaux intégrant les coûts et les contraintes des différents équipements (liens, routeurs, cartes). Ce modèle prend également en compte de nombreuses contraintes opérationnelles telles que la présence de nœud potentiel et de capacités modulaires, mais aussi des contraintes sur les délais de communication. Un algorithme exact et deux approximations sont proposés pour résoudre ce problème. Les résultats numériques montrent que des économies substantielles peuvent être effectuées en intégrant les coûts d'équipements dans la phase amont du processus de conception.

Avec la croissance exponentielle de l'Internet et l'incroyable développement des applications réseaux, les variations sur les volumes de trafic sont devenus un des problèmes majeurs auxquels sont confrontés les opérateurs. Aujourd'hui, concevoir un réseau sur la base d'une unique matrice de trafic "en heure de pointe" est de moins en moins crédible du fait de la forte volatilité des trafics. Il devient ainsi nécessaire d'intégrer explicitement l'incertitude sur la demande en trafic dans les problèmes d'optimisation du routage. Nous étudions deux problèmes d'optimisation robuste du routage : (1) le problème de conception des VPN dans le cadre du modèle hose d'incertitude sur la demande et (2) le problème d'optimisation des métriques de routage IGP avec incertitude sur la demande. Nous formulons des modèles mathématiques de chacun de ces problèmes et proposons des heuristiques efficaces basées sur des techniques de recherche locale pour les résoudre.

L'efficacité pratique des techniques et méthodes développées dans le cadre de cette thèse est démontrée et validée par une batterie de tests sur des données réalistes. Ce travail a d'autre part donné lieu à plusieurs développements logiciels dans le logiciel de planification de réseaux NEST de QoS Design.

Abstract

With communication networks getting more and more present in our daily activities, network outages or even significant degradations of the quality of service become less and less tolerable. This calls for a robust design of communication networks anticipating possible failures or shifts in the expected traffic demands. This thesis addresses several robust design and planning problems arising in the telecommunication area.

We first address the problem of designing and dimensioning a survivable network topology. We propose a novel network design model integrating all equipment costs and constraints (links, routers and line cards). This model also include operational constraints such as the presence of potential nodes and modular capacities as well as delay constraints. One exact and two heuristic algorithms are proposed to solve this problem. Numerical results show that significant cost-savings can be achieved when equipment costs are taken into account in the early stages of the design process.

With the explosive growth of the Internet and the incredible development of network applications, the variation in traffic volumes has become one of the most important problems faced by network operators. Designing a network using a single “busy hour” traffic matrix more and more strains credibility due to the high volatility of traffic patterns. Thus, there is a need to incorporate demand uncertainty into the network routing problems explicitly. We study two robust route optimization problems : (1) the problem of VPN design under the hose model of demand uncertainty and (2) the problem of link weight optimization under demand uncertainty. We establish mathematical models for both problems and propose efficient approximation algorithms based on local-search techniques to solve them.

The practical effectiveness of the techniques and methods developed in this thesis is illustrated and validated by a set of numerical tests on realistic data. This work resulted in several software developments in the network planning tool NEST of QoS Design.

Préface

Cette thèse a été réalisée dans le cadre d'un partenariat entre la société QoS Design située à Toulouse et le groupe de recherche MRS du laboratoire LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) du CNRS.

Le LAAS en bref

Créé en 1967 sous le nom de "Laboratoire d'Automatique et de ses Applications Spatiales", le LAAS s'est installé dans ses locaux du complexe scientifique de Ranguel en 1968 avec un effectif de 140 personnes. Actuellement, le laboratoire compte près de 500 chercheurs dans son effectif.

Le LAAS est une unité propre du Centre National de la Recherche Scientifique (CNRS), rattachée à l'institut des sciences de l'ingénierie et des systèmes (INSIS).

Le laboratoire s'intéresse principalement à quatre classes de systèmes, à travers quatre pôles thématiques qui organisent ses 17 groupes de recherche :

- le pôle Modélisation, Optimisation et Conduite des Systèmes (MOCOSY),
- le pôle Robotique et Intelligence Artificiel (RIA),
- le pôle Systèmes Informatiques Critiques (SINC),
- le pôle Micro et Nano Systèmes (MINAS).

Le groupe MRS (Modélisation et Contrôle des Réseaux et Signaux) fait partie du pôle MOCOSY. L'activité de recherche du groupe concerne le développement de nouvelles approches méthodologiques et algorithmiques pour l'analyse, la conception et l'évaluation de performances de réseaux et systèmes de télécommunications, ainsi que le traitement avancé des signaux et systèmes.

Le groupe a de nombreuses collaborations académiques, nationales (INRIA et I3S à Sophia Antipolis, ENST à Paris et ISIMA à Clermont-Ferrand) et internationales (CWI à Amsterdam, HUT à

Helsinki et le California Institute of Technology à Pasadena). Le groupe est également engagé dans plusieurs partenariats avec des industriels : SFR, Index Multimedia, AIRBUS, FT R&D, Alcatel, Thales, DCN, NavoCap, QoS Design.

Le société QoS Design en bref

La société

Créée en 2003, QoS DESIGN est une société spécialisée dans les domaines de l'évaluation de performances, la simulation, la conception et la planification des réseaux de télécommunications. QoS DESIGN développe et commercialise la suite logicielle NEST (Network Engineering & Simulation Tool).

NEST propose un environnement intégré permettant de résoudre de manière efficace et ergonomique la planification, l'optimisation et la simulation des grands réseaux de télécommunications. La suite logicielle NEST se compose de plusieurs produits :

- **NEST IP/MPLS** : pour la simulation et la planification des réseaux IP/MPLS,
- **NEST Mobile** : pour la planification des réseaux mobiles,
- **NEST VPN** : pour la planification des VPN,
- **NEST Supervisor** : pour l'inventaire et la supervision des réseaux.

Secteurs d'activité

Les clients de QoS design sont principalement les opérateurs de réseaux et les entreprises grand comptes qui possèdent des infrastructures réseaux. QoS Design participe également à mener des études de R&D et des projets de recherche avec ses partenaires académiques et industriels. Les domaines de recherche sont variés :

- **Télécommunications** : évaluation de performances de réseaux, simulation du trafic, design et optimisation des réseaux, estimation du trafic ...
- **Aéronautique et Spatial** : maintenance avion, stabilité en avionique, traitement des essais en vol, calcul optimal de trajectoires avion, calcul de résistance de structures, détection pannes avion, modélisation fiabilité et maintenance moteurs avion, trajectoires spatiales et rendez-vous orbitaux, constellations de satellites...
- **Informatique et Calcul** : grilles de calcul, ASP, environnements de parallélisme, applications client/serveur, grands codes de calcul scientifique ...
- **Défense** : systèmes de navigation (GPS, Galiléo, LORAN-C), systèmes de détection (RADAR, SONAR, ...), problèmes de contrôle optimal...

Dates Importantes

- **Juillet 2003** : QoS DESIGN primé au 5^{ème} concours national de l'innovation.

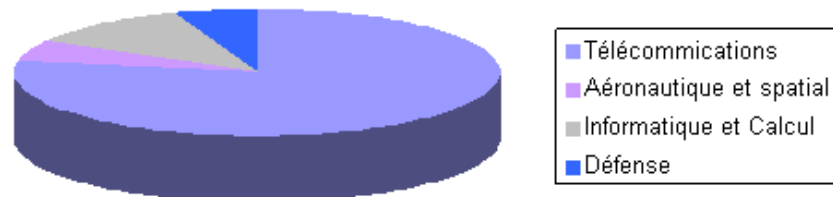


FIG. 1 – Secteurs d'activité en R&D de QoS Design

- **Décembre 2003** : QoS Design primé au 23^{ème} concours régional de l'innovation Midi-Pyrénées
- **Avril 2005** : QoS Design lauréat de Capital IT 2005
- **Mars 2006** : QoS Design lauréat des Trophées de l'Economie Numérique 2006

TABLE DES MATIÈRES

Remerciements	3
Résumé	5
Abstract	7
Préface	9
1 Réseaux de télécommunications : évolutions et problématiques	27
1.1 Introduction	27
1.2 Succès d'Internet et de son service best effort	27
1.2.1 Le protocole OSPF	29
1.2.2 Le protocole BGP	29
1.3 Evolutions majeures d'Internet depuis ses débuts	30
1.3.1 La technologie ATM	31
1.3.2 La technologie MPLS	32
1.3.3 Les réseaux privés virtuels VPN	34
1.4 Problèmes de planification des réseaux	35
1.4.1 Conception de réseaux	36
1.4.2 Planification du routage	36
1.4.3 Dimensionnement de réseaux	37
1.4.4 Contraintes opérationnelles	38
1.5 Conclusion	40
2 Panorama sur les techniques d'optimisation	41
2.1 Introduction	41
2.2 Programmation linéaire et en nombres entiers	42
2.2.1 Méthodes de recherches arborescentes	43

2.2.2	La Programmation dynamique	44
2.3	Méthodes heuristiques	46
2.3.1	La Recherche locale	47
2.3.2	Algorithmes gloutons	48
2.3.3	Recherche à écarts limités LDS	48
2.4	Métaheuristiques - méthode tabou	49
2.4.1	Intensification	50
2.4.2	Diversification	50
2.5	Problèmes de flots	50
2.5.1	Notion de flot	51
2.5.2	Le problème de flot maximum	51
2.5.3	Le problème de flot compatible	52
2.5.4	Le problème de flot maximal de coût minimum	52
2.5.5	Le problème de transport	53
2.6	Conclusion	54
3	Conception de réseaux de télécommunications	55
3.1	Introduction	55
3.1.1	Motivations	55
3.1.2	Le problème de conception de réseaux	56
3.1.3	Etat de l'art	57
3.1.4	Contributions	58
3.2	Une nouvelle approche de conception de réseaux	59
3.2.1	Description du modèle	59
3.2.2	Formulation mathématique du modèle	61
3.2.3	Problème à résoudre	66
3.2.4	Extension du modèle	66
3.3	Résolution exacte par des techniques de "Branch-and-Bound"	66
3.3.1	Branchement	67
3.3.2	Elagage	67
3.3.3	Génération des bornes supérieures et inférieures	68
3.3.4	Ordre de Branchement	69
3.3.5	Optimisation du calcul des charges des liens	69
3.3.6	Temps de Calcul	69
3.4	Algorithme glouton	70
3.5	Heuristique basée sur la recherche tabou	70
3.5.1	Initialisation	71
3.5.2	Espace de recherche et structure du voisinage	71
3.5.3	Diversification	72
3.5.4	Liste Tabou	72
3.6	Heuristique basée sur la méthode LDS	72
3.7	Tests et résultats	73
3.8	Conclusion	76

4	Conception robuste des VPN avec incertitude sur la demande	79
4.1	Introduction	79
4.1.1	Rappel sur les VPN	79
4.1.2	Spécification de la bande-passante d'un VPN	80
4.1.3	Réservation de la bande-passante d'un VPN	81
4.1.4	Contribution	82
4.1.5	Organisation de ce chapitre	82
4.2	Etat de l'art	83
4.3	Minimisation de la bande-passante totale réservée	84
4.3.1	Notations	84
4.3.2	Formulation en programmation linéaire mixte	85
4.3.3	Formulation nœud-chemin	87
4.3.4	Méthode heuristique	89
4.4	Minimisation de l'utilisation maximale des interfaces	96
4.4.1	Modèle et formulation	97
4.4.2	Méthode de résolution	98
4.5	Tests et résultats	100
4.5.1	Minimisation de la bande-passante totale réservée	101
4.5.2	Minimisation de l'utilisation maximale des interfaces	102
4.6	Conclusion	104
5	Optimisation robuste du routage IP avec incertitude sur la demande	107
5.1	Introduction	107
5.1.1	Optimisation des métriques de routage IP	107
5.1.2	Incertainitude sur la demande	108
5.1.3	Contribution	109
5.1.4	Organisation de ce chapitre	109
5.2	Etat de l'art	110
5.2.1	Optimisation des métriques de routage avec des demandes en trafic connues	110
5.2.2	Estimation de matrice de trafic	110
5.2.3	Optimisation robuste du routage	111
5.3	Formulation du problème	112
5.3.1	Structure de l'ensemble d'incertitude	112
5.3.2	Solutions admissibles	112
5.3.3	Fonction objectif	113
5.4	Un algorithme de recherche locale pour l'optimisation robuste du routage	114
5.4.1	Définition du voisinage	114
5.4.2	Génération du voisinage	115
5.4.3	Mise à jour dynamique des plus courts chemins	117
5.4.4	Charges maximales des liens	117
5.4.5	Test de convergence	119
5.4.6	Modes incrémental et multi-start	119
5.5	Résultats	119
5.5.1	Modèle d'incertitude sur la demande de type hose	119

5.5.2	Incertitude sur la demande de type Box+Hose	121
5.6	Conclusion	123
Conclusion et perspectives		125
5.7	Contributions	125
5.8	Perspectives	126
Glossaire		135

TABLE DES FIGURES

1	Secteurs d'activité en R&D de QoS Design	11
2	Convergence technologique au travers de l'Internet	22
3	Cycle d'ingénierie d'un réseau	23
4	Structure du document	25
1.1	Routage IP hiérarchique	28
1.2	Routeurs BGP	30
1.3	Réseau ATM	32
1.4	Réseau MPLS	33
1.5	Architecture d'un réseau privé virtuel.	34
1.6	Processus de conception de réseaux	36
1.7	Exemples d'instances du problème d'optimisation de routage	37
1.8	Réseau en deux couches	39
2.1	Premier chemin trouvé avec une recherche en profondeur.	45
2.2	Essai d'un autre nœud fils.	45
2.3	Exemples de flots	51
2.4	Exemple d'instance pour laquelle il n'existe pas de solution réalisable	52
2.5	Exemples d'instances du problème de transport	54
3.1	Processus de conception de réseaux	57
3.2	Les types de nœuds dans un réseau	59
3.3	Branchement des liens sur les routeurs à l'aide de cartes	65
3.4	Temps de calcul requis par l'algorithme de BB	70
3.5	Ecarts relatif (en %) entre le coût optimal et les coûts des heuristiques étudiées	75
3.6	Temps de calcul en (sec)	75
4.1	Représentation d'un lien virtuel	80
4.2	Modèles Pipe et Hose de spécification de la bande-passante d'un VPN.	81
4.3	Exemples d'instances de notre problème	82

4.4	Exemple d'instances générant des temps de calcul considérables.	87
4.5	Exemple de déviation de flux entre s et t	91
4.6	Construction du réseau de transport relatif à l'arc $e \in E$ (seuls les arcs avec un coût non nul sont représentés).	93
4.7	Tests sur la qualité de la borne supérieure.	96
4.8	Mauvaise utilisation des ressources	97
5.1	Représentation des plus court chemins dans un graphe	113
5.2	Exemple où $d_{\min}^l = 0$	116
5.3	Exemple où $0 < d_{\min}^l < \infty$	116
5.4	Problème de flot à coût minimal.	118
5.5	Taux d'utilisation maximal (%)	123

LISTE DES TABLEAUX

3.1	Exemples de modèles de liens.	60
3.2	Modèles de routeurs.	74
3.3	Modèles de cartes.	74
3.4	Modèles de liens.	74
3.5	TS v.s. algorithme glouton	76
4.1	Résumé des Résultats Connus	84
4.2	Ecarts relatifs en % entre le coût de l'heuristique et la borne inférieure et entre le coût optimal et la borne inférieure	101
4.3	Temps calcul en secondes.	102
4.4	Description des Topologies	102
4.5	Taux de réservation maximal (%).	103
4.6	Bande-passante totale réservée(%)	104
4.7	Temps de calcul (sec).	104
5.1	Description des Topologies	120
5.2	Coûts obtenus avec les algorithmes A_{INCR} , A_{MS} , CM et LM	121
5.3	Temps de calcul (sec) des algorithmes A_{INCR} , A_{MS} , CM et LM	121

Introduction générale

Avec aujourd’hui plus de 800 millions d’utilisateurs dans le monde, l’Internet a connu un succès fulgurant. Ce succès provient principalement de deux facteurs : la simplicité d’accès au réseau et l’offre de services adaptés aux besoins des utilisateurs. A l’origine, le protocole IP a été conçu pour l’interconnexion de réseaux hétérogènes. Cela avait permis aux développeurs de services de faire abstraction de la technologie de transport (ATM, SONET/SDH ...) et de développer, sur des API, plusieurs générations de services. Ceci a commencé par le développement des services World Wide Web, suivi par l’adoption par le monde professionnel des applications Internet telles que le courrier électronique, le transfert de fichiers (FTP), l’accès distant (telnet) et, plus récemment, par le déploiement des logiciels de partage de fichier pair à pair.

Aujourd’hui l’Internet s’est transformé en une architecture de communication globale supportant des services “temps-réel”, tels que la voix sur IP ou la vidéo. Ces services ont des contraintes de qualité de service beaucoup plus strictes que les services traditionnels. Le développement attendu de ces services est susceptible de bouleverser radicalement les réseaux actuels, en engendrant de nouveaux usages, en attirant beaucoup plus d’utilisateurs et ainsi en accroissant de plusieurs ordres de grandeur les volumes de trafic qu’ils doivent supporter.

En parallèle au développement des services, un autre fait marquant l’évolution de l’Internet au cours de ces dernières années est la multiplication des technologies d’accès radios telles que les WAN avec le WiFi, les PAN avec le Bluetooth, les MBWAN avec le WiMAX et les réseaux de 3^{ème} génération (UMTS, ...). De part leur diversité en termes de coût de déploiement, de couverture et de capacité, ces différentes technologies s’associent pour offrir aux usagers un Internet ambiant disponible dans tous les environnements qu’ils soient urbain ou domestique, professionnel ou ludique. Un certain nombre de recherches se sont focalisées sur le développement d’une architecture unifiée permettant aux terminaux d’utiliser séquentiellement ou simultanément ces réseaux d’accès hétérogènes pour accéder à l’Internet qui devient de fait omniprésent. L’internet de demain sera ainsi une interconnexion de réseaux d’accès hétérogènes permettant d’accéder, via des coeurs de réseaux IP/MPLS, à un grand nombre de services “universels” (cf. figure 2).

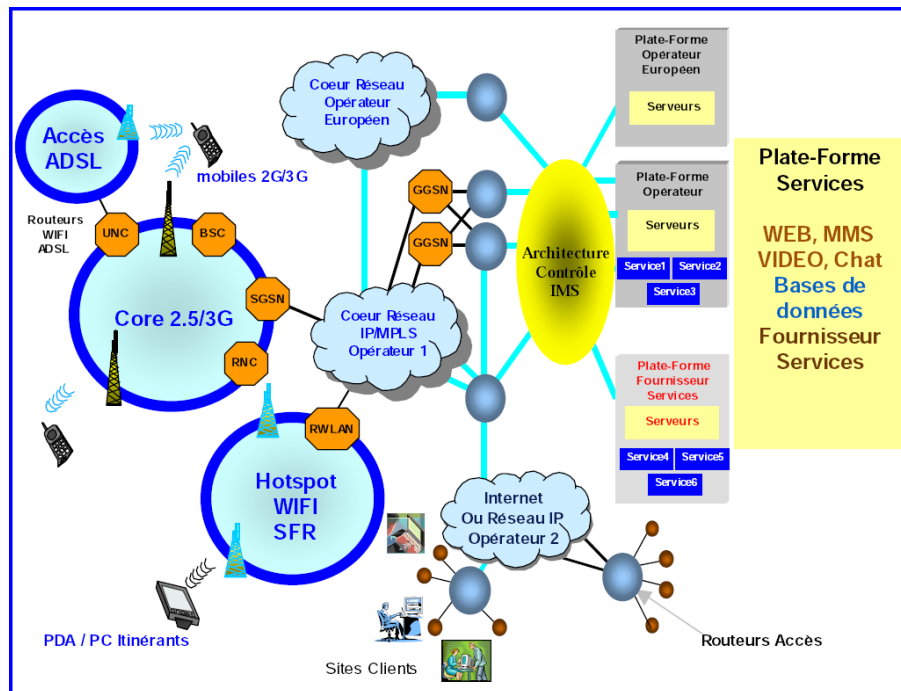


FIG. 2 – Convergence technologique au travers de l’Internet

Toutes ces évolutions conduisent à une multiplication des services offerts par le réseau et à une croissance sans précédent du nombre d’utilisateurs et des volumes de trafics qu’il génèrent. Dans une société où l’information et la communication ont pris une telle importance, l’interruption des services offerts par le réseau, ou même une dégradation significative de la qualité de service, sont de moins en moins acceptables. Le respect des nouvelles exigences de qualité de service est ainsi un enjeu majeur.

Ingénierie des Réseaux IP

Ces nouvelles exigences de qualité de service, l’augmentation continue du trafic Internet et les impératifs de plus en plus forts de sécurisation imposent aux fournisseurs d’accès Internet (FAI) et aux opérateurs une adaptation régulière et une mise à jour de leurs infrastructures pour faire face à cet accroissement.

L’objectif d’un opérateur consiste à trouver un compromis entre (a) une gestion optimale de ses ressources, (b) une minimisation des coûts associés à l’exploitation et à la gestion du réseau, et (c) la qualité du service fournis aux clients.

Dans une conjoncture économique difficile et très compétitive, les opérateurs ont du trouver une alternative à la solution de sur-dimensionnement “aveugle” des réseaux. Celle ci consiste à avoir un suivi

plus régulier et plus fin du réseau pour mettre en oeuvre des techniques d'ingénierie de trafic. Ces techniques ont pour objectif principal d'éviter les phénomènes de congestion du trafic et les dégradations du service qui en résultent. A notre avis, une telle approche de l'ingénierie de réseaux doit alors reposer sur deux axes essentiels :

- **Supervision du réseau** : il est illusoire de vouloir mettre en oeuvre une ingénierie de trafic efficace, sans une connaissance approfondie du réseau, des trafics qu'il achemine et de l'état des ressources utilisées par le réseau (routeurs, interfaces, liens physiques, ...). Les techniques de métrologie de réseaux peuvent apporter une réponse à ces problèmes. Ces techniques ont pour objectif principal de mesurer les volumes de trafic sur les liens du réseau, mesurer les paramètres de qualité de service, connaître l'état des équipements, afin de pouvoir, non seulement intervenir dans l'urgence en cas de problème, mais aussi anticiper l'évolution du réseau, planifier l'introduction de nouvelles applications et améliorer les performances pour les utilisateurs.
- **Planification du réseau** : les objectifs essentiels sont de concevoir et de dimensionner à moindre coût les équipements du réseau et d'optimiser le routage des flux. Pour être utilisé en situation opérationnelle, un outil de planification doit permettre de résoudre ces problèmes en tenant compte de l'ensemble des contraintes matérielles ou techniques auxquelles sont confrontés les gestionnaires d'infrastructures, mais aussi de la qualité de service cible et du niveau de robustesse recherché.

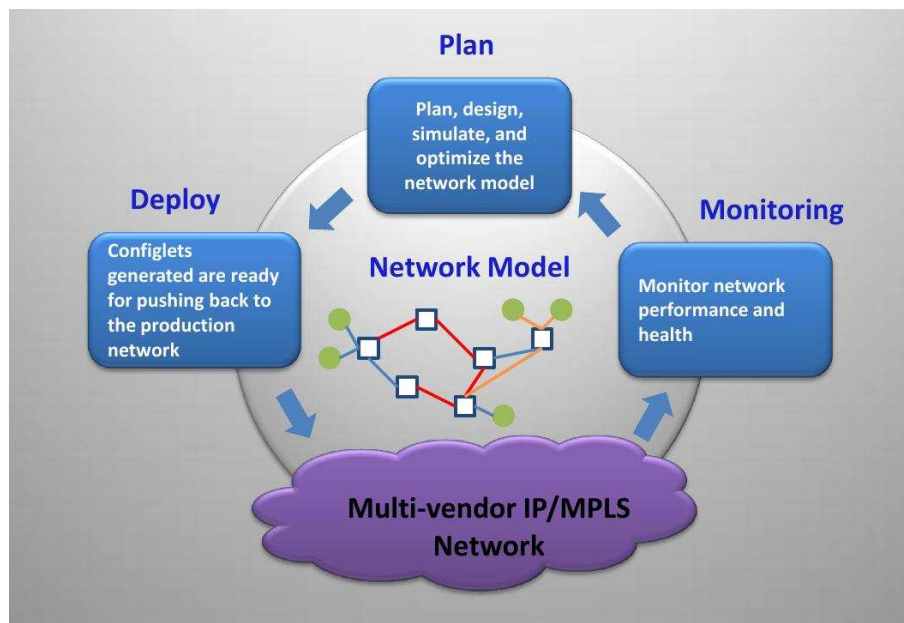


FIG. 3 – Cycle d'ingénierie d'un réseau

Contributions

Nos contributions portent sur la planification des réseaux et ont pour caractéristiques communes de tenir compte des contraintes :

- **technologiques** telles que les contraintes de routage dans les réseaux IP/MPLS, les caractéristiques des VPN, les modèles d'équipements, ...,
- **opérationnelles** puisque nous tenons compte des contraintes de résilience aux pannes et de l'incertitude sur la demande,
- **financières** par l'intégration de modèles économiques réalistes et utilisés par les opérateurs. De plus, elles sont intégrées dans la solution NEST dont l'ambition est d'être l'outil de référence de cette nouvelle approche d'ingénierie dans les réseaux IP.

L'organisation du présent mémoire est la suivante :

- La chapitre 1 approfondit le fonctionnement, les évolutions et les limitations des réseaux IP depuis leur apparition. Un panorama des principaux problèmes auxquels les opérateurs de ces réseaux sont souvent confrontés est ensuite proposé.
- Avant d'aborder certains de ces problèmes, nous faisons, dans le chapitre 2, le tour des techniques de résolution des problèmes d'optimisation. Nous mettons, plus particulièrement, l'accent sur les outils que nous utiliserons pour résoudre les problèmes de planifications étudiés dans ce mémoire.
- Le chapitre 3 traite le problème de conception des réseaux. Ce problème a longtemps été traité sans tenir compte des équipements autres que les liens. Nous proposons, dans ce chapitre, une nouvelle approche de conception des réseaux tenant compte des contraintes matérielles et utilisant des coûts d'équipements réalistes. Une méthode exacte et des heuristiques efficaces aux temps de calcul sont proposées. Comparées aux méthodes proposées dans la littérature, nos solutions amènent un gain économique significatif.
- Nous traitons le problème de planification des VPN avec incertitude sur la demande dans le chapitre 4. Nous considérons le modèle Hose d'incertitude sur la demande. Deux critères d'optimisation sont étudiés : (1) la bande passante totale réservée et (2) l'utilisation maximale des liens. Pour chacun de ces deux critères, une heuristique à base de recherche locale est présentée.
- Nous considérons dans le chapitre 5 le problème d'optimisation des métriques de routage IP avec incertitude sur la demande. En utilisant un voisinage de taille réduite et les techniques de type min-cost flow, nous proposons une nouvelle heuristique à temps de calcul faible qui permet d'obtenir des solutions incrémentales de bonne qualité.

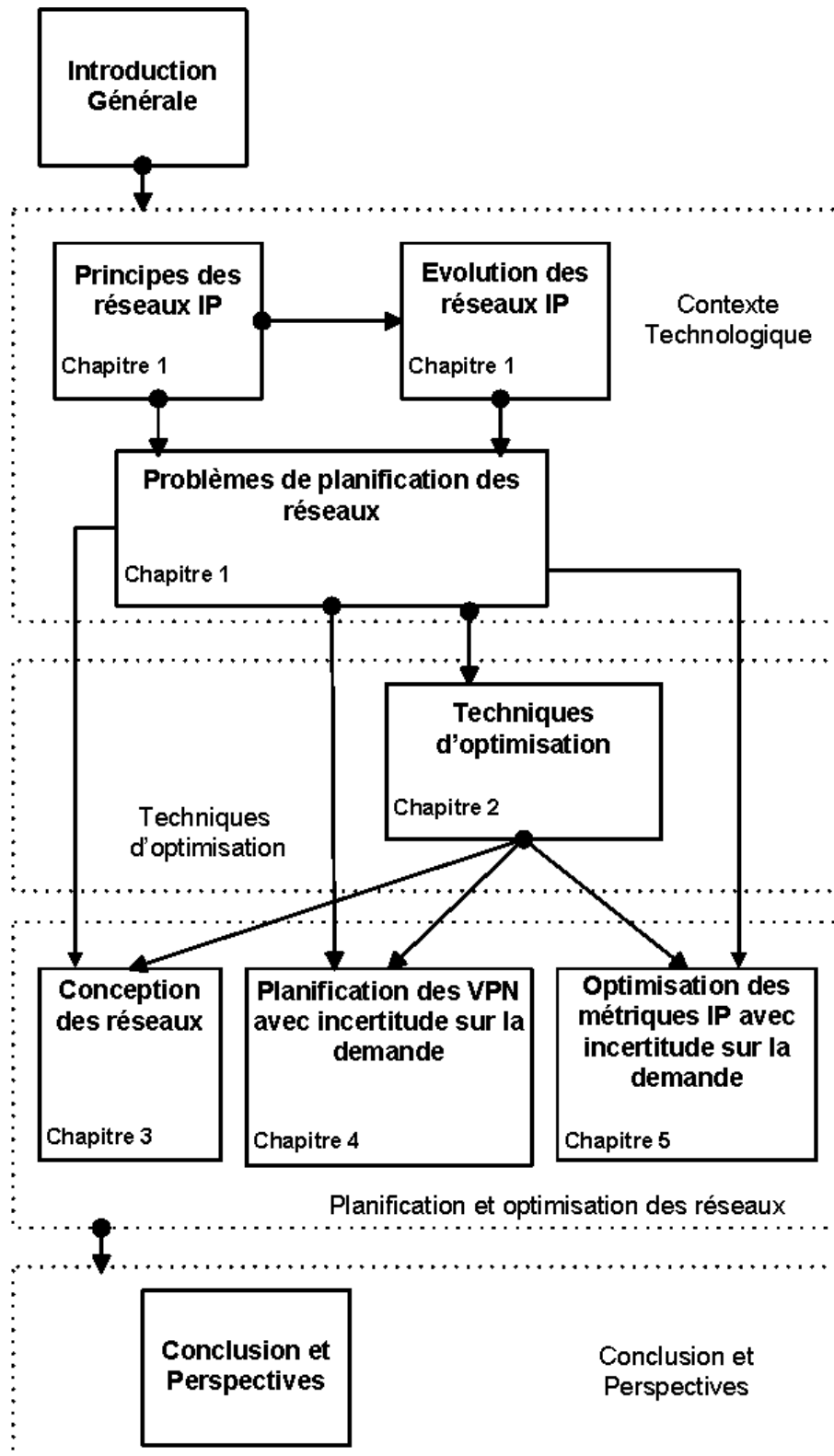


FIG. 4 – Structure du document

CHAPITRE 1

Réseaux de télécommunications : évolutions et problématiques

1.1 Introduction

Depuis les années 90, Internet a connu un succès fulgurant. D'un coté plusieurs évolutions technologiques importantes ont fait d'Internet un outil majeur dans le domaine de l'informatique et des télécommunications et de l'autre un nombre d'utilisateurs et des volumes de trafic en perpétuelle augmentation.

Avec ces évolutions, les fournisseurs de services Internet (FAI) doivent constamment relever le défi de faire évoluer leurs réseaux afin de soutenir des taux de croissance extrêmement rapide tout en maintenant une infrastructure fiable pour les applications nécessitant de la Qualité de Service (QoS).

Nous proposons, dans ce chapitre, d'exposer les évolutions intervenues dans le fonctionnement et l'utilisation d'Internet. Nous mettrons ensuite en évidence les principaux problèmes auxquels les opérateurs et les FAI doivent faire face pour adapter leurs réseaux à ces évolutions et pour maintenir une infrastructure fiable qui répond aux nouvelles exigences de QoS.

1.2 Succès d'Internet et de son service best effort

Internet est basé sur le protocole IP (Internet Protocol)[15] permettant la propagation des flots de données. Cette technologie simple associée aux réseaux IP est parfaitement adaptée aux services de type "Best Effort (BE)" initialement fournis par Internet. Ce type de service ne fournit aucune garantie (délai, pertes) aux flots transmis par le réseau. Si le protocole de transport TCP [16] est utilisé, les flots ont tout de même la garantie d'être transmis de manière intègre au bout d'un certain temps.

Le routage consiste en une mise en place de routes permettant la communication entre les différents éléments composant les réseaux IP : les routeurs. Ces derniers sont des point d'aiguillage formant l'architecture du réseau. Ils possèdent tous une table de routage résumant l'information de routage sous forme de prochain saut ("Next Hop"), c'est à dire qu'une entrée de cette table associe une destination au prochain routeur vers lequel le flot doit se diriger. De manière plus précise, il existe sur chaque routeur un ensemble d'interfaces permettant la communication avec les routeurs connectés à ce routeur. C'est en fait l'interface à emprunter qui est renseignée dans les tables de routage.

Les réseaux IP sont hiérarchiques car composés de systèmes autonomes (AS) gérés indépendamment les uns des autres comme illustré sur la figure 1.1. Chaque système autonome implémente un protocole de routage intra domaine ("Interior Gateway Protocol (IGP)") tel qu'OSPF (Open Shortest Path First)[42], IS-IS (Intermediate System to Intermediate System)[24], RIP (Routing Information Protocol)[21], ou EIGRP (Enhanced Interior Gateway Routing Protocol)[19]. Un protocole de routage met en place les routes entre les routeurs d'un même AS à l'aide d'un algorithme calculant les plus courts chemins (exemples : Dijkstra [5], Bellman Ford [4]) au sens des métriques associées à chaque interface.

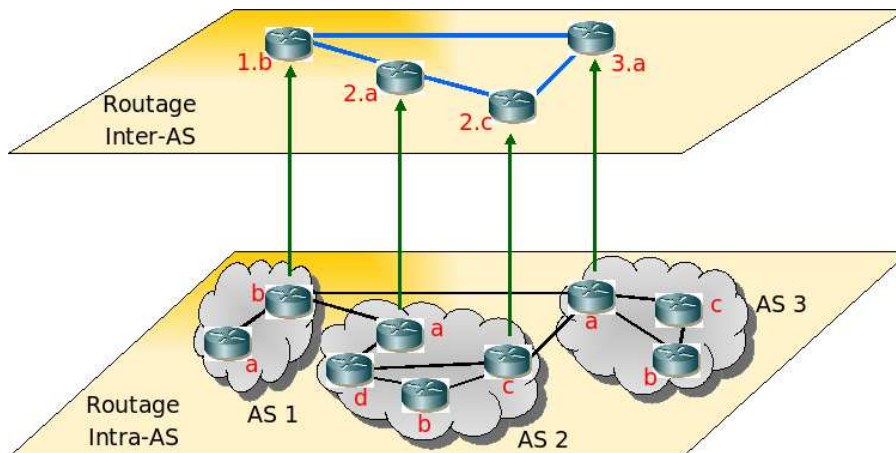


FIG. 1.1 – Routage IP hiérarchique

Le routage entre systèmes autonomes (ou routage interdomaine) est assuré par un "Exterior Gateway Protocol" (EGP) dont le plus communément utilisé est BGP (Border Gateway Protocol) [27].

Les principaux objectifs à atteindre par un protocole de routage sont :

- Optimisation (sélectionner le meilleur chemin selon les critères choisis),
- Simplicité (le plus simple possible),
- Robustesse et souplesse (faire face à toutes sortes d'imprévus),
- Convergence rapide (recalculer les routes rapidement pour éviter les boucles, les pannes réseau,...).

Dans la suite, nous allons nous intéresser aux deux protocoles de routage les plus utilisés dans l'Internet : OSPF et BGP

1.2.1 Le protocole OSPF

Le protocole OSPF est un protocole de routage à état de lien (Link-State) normalisé par l'IETF. Il est actuellement l'IGP (Interior Gateway Protocol) le plus répandu.

Les messages sur l'état des liens permettent au routeur d'avoir une vision globale du réseau et de sa topologie et l'émission des mises à jour n'est déclenchée qu'en cas de modifications topologiques.

Pour générer la table de routage, le protocole OSPF utilise un arbre de plus courts chemins (SPF Tree : Short Path First Tree) et un algorithme de plus court chemin (SPF Algorithm) correspondant à l'algorithme de Dijkstra, qui détermine le meilleur chemin en terme de coût.

Le coût du chemin dépend des métriques fixées par l'opérateur sur chacune des interfaces du réseau. La métrique est un coût sans dimension et aucune valeur n'est imposée. Libre à l'opérateur de choisir la valeur qu'il juge pertinente (CISCO préconise la valeur $10^8/\text{débit}$).

Lorsque plusieurs routes ont le même coût, un partage de charge équitable (Load-Balancing) entre ces routes peut être employé. Le nombre de routes sur lesquelles on peut faire du partage de charge est en général limité (à quatre sur certains routeurs).

1.2.2 Le protocole BGP

Le protocole BGP (Border Gateway Protocol) est un protocole de routage entre AS. BGP est utilisé pour l'échange d'informations de routage entre AS de l'Internet. Les informations de routage se présentent sous la forme de la suite des numéros d'AS à traverser pour atteindre la destination. C'est BGP qui assure la propagation des routes à l'échelle mondiale.

Quand BGP est utilisé entre AS, le protocole est connu sous le nom de BGP Externe (e-BGP). Si BGP est utilisé à l'intérieur d'un AS pour échanger des routes, alors le protocole est connu sous le nom de BGP interne (i-BGP). Les routeurs de frontière communiquant par i-BGP doivent être complètement maillés (mais pas forcément directement connectés).

BGP transmet la meilleure route vers chaque système autonome susceptible d'être adopté pour atteindre une destination donnée. Lorsqu'un routeur BGP reçoit des mises à jour en provenance de plusieurs systèmes autonomes décrivant différents chemins vers une même destination, il choisit alors le meilleur itinéraire pour l'atteindre et le propage vers ses voisins. Une décision de routage est fondée sur plusieurs attributs comme :

- **AS-path** : Cet attribut liste les numéros de tous les AS qu’une mise à jour a traversé pour atteindre une destination.
- **Origin** : Cet attribut donne des informations sur l’origine de la route. Ces informations peuvent être de type IGP (la route annoncée provient du même système autonome que l’annonceur), de type EGP (la route est apprise et ne provient pas du même système autonome) ou incomplètes (la route est apprise d’une autre manière).
- **Next hop** : Cet attribut contient l’adresse IP du routeur vers lequel l’information doit être émise pour atteindre le réseau.
- **Weight** : Cet attribut est utilisé dans le processus de sélection de chemin lorsqu’il existe plus d’une route vers une même destination. Cet attribut de poids est local et n’est pas propagé dans les mises à jour de routage.
- **Local preference** : Cet attribut a un rôle similaire à l’attribut de poids, si ce n’est qu’il fait partie des informations de mise à jour de routage.
- **Multi-exit discriminator** : Cet attribut indique aux routeurs voisins externes le chemin à privilégier vers un AS lorsque celui-ci possède plusieurs points d’entrée.
- **Community** : Cet attribut est utilisé pour grouper des destinations auxquelles des décisions de routage peuvent être appliquées.

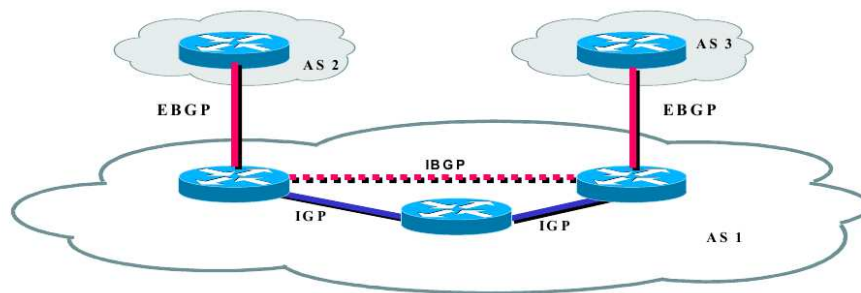


FIG. 1.2 – Routeurs BGP

1.3 Evolutions majeures d’Internet depuis ses débuts

Depuis ses débuts où de simples fichiers étaient transmis d’un ordinateur vers un autre, l’utilisation d’Internet a fait un long chemin.

Internet est devenu une architecture de communication globale supportant maintenant des applications temps réel tels que la voix sur IP ou la vidéo à la demande.

Ces nouvelles applications n’ont pas les mêmes besoins en termes de Qualité de Services (QoS) que les premiers services proposés par Internet. La QoS implique des limitations fortes sur :

- la latence ou le délai de bout en bout : le temps entre le début de l'émission d'un paquet de données et la fin de sa réception,
- les probabilités de perte des paquets, et
- la gigue : moyenne des écarts des latences des différents paquets appartenant à un même flot de communication.

En même temps, on assiste depuis des dizaines d'années à la convergence des services proposés par Internet. Etant défini comme une interconnexion de réseaux hétérogènes, il n'est pas surprenant de voir de plus en plus des réseaux d'accès de types différents permettre aux utilisateurs un accès à un nombre toujours croissant de services sur Internet. Un utilisateur peut donc accéder à un ensemble de services "universels".

Ces évolutions majeures conduisent donc à une augmentation sans précédent du nombre d'utilisateurs et des volumes de trafic qu'il génèrent. L'architecture best-effort établi par les réseaux IP dans les années 70 ne permet pas de garantir une quelconque qualité de service et il a été nécessaire de définir de nouvelles architectures de réseaux pour répondre à ces nouveaux besoins.

1.3.1 La technologie ATM

L'architecture ATM (Asynchronous Transfer Mode) a été conçue durant les années 80 pour les réseaux de télécommunications dont l'objectif est de transporter tout type de communication (voix, vidéo, données) tout en garantissant la qualité de service de chaque client. L'ATM est une technique de transfert asynchrone fondée sur la commutation de paquets de taille fixe (les cellules). Elle repose sur un multiplexage statistique des trafics pour bénéficier des fluctuations des sources de trafics et utiliser au mieux la bande passante de ses liens haut débit.

A chaque demande de connexion d'un utilisateur, on lui associe un circuit virtuel (PVC), qui est établi à l'intérieur du réseau ATM par des "étiquettes" (ou labels) laissées dans chaque noeud traversé. Les informations émises par l'utilisateur sont alors transmises par cellules, composées de 48 octets pour les données et de 5 octets pour l'entête. Cet entête permet d'identifier le chemin virtuel que devront emprunter les cellules.

Un réseau ATM est constitué de commutateurs reliés par des liens physiques (cf. figure 1.3). Les commutateurs sont chargés :

- d'acheminer les cellules reçues sur les ports d'entrée vers le bon port de sortie selon les informations de routage contenues dans l'entête de la cellule,
- d'effectuer le multiplexage des cellules sur la sortie requise,
- de contrôler l'établissement et la libération des connexions.

En proposant des solutions permettant le transport d'IP sur ATM, il est possible de réutiliser le savoir-faire et la technologie (par exemple, les matrices de commutation) développés dans un contexte purement ATM. Ainsi, la superposition en couches IP sur ATM sur SDH devint courante, et un certain nombre d'architectures et de protocoles virent le jour : Classical IP and ARP over ATM (CLIP), Next

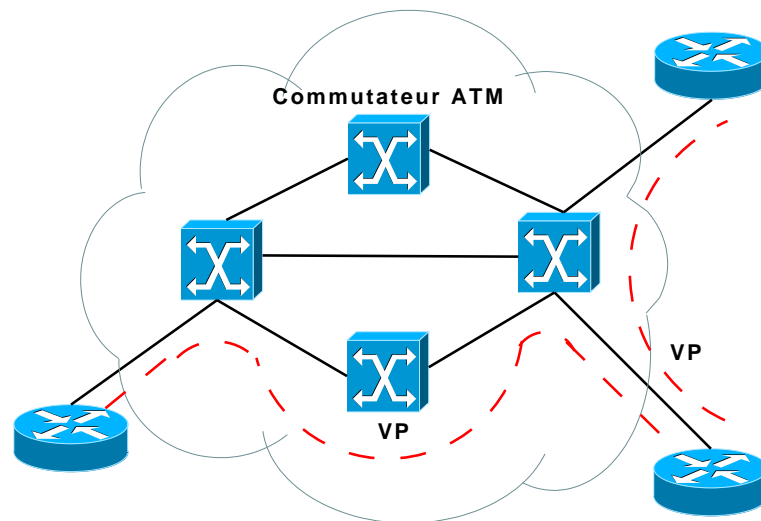


FIG. 1.3 – Réseau ATM

Hop Resolution Protocol (NHRP), MultiProtocol over ATM (MPOA) entre autres (voir par exemple [45] ou [46]).

1.3.2 La technologie MPLS

Le protocole MPLS, basé sur le paradigme de commutation d'étiquettes, dérive directement de l'expérience acquise avec l'ATM. L'idée de MPLS est de rajouter un label de couche 2 aux paquets IP dans les routeurs frontières d'entrée du réseau. Le réseau MPLS va pouvoir créer des chemins virtuels ou LSP (Label Switching Path), similaires aux PVC d'ATM.

Ces LSP définissent une route de bout en bout par une concaténation de labels. A l'entrée du réseau MPLS, le label est rajouté à l'entête IP par le routeur frontière LER (Label Edge Routers) puis à la sortie du réseau MPLS le label est retiré par un autre routeur frontière, ce qui permet de retrouver le paquet IP original. Les routeurs du cœur de réseau, appelés LSR (Label Switching Routers), routent les paquets de proche en proche par commutation de labels, "oubliant" ainsi les adresses IP.

L'un des objectifs initiaux de MPLS était d'accroître la vitesse de traitement des datagrammes dans l'ensemble des équipements intermédiaires. Cette volonté, avec l'introduction des gigarouteurs, est désormais passée au second plan. Depuis, l'aspect "fonctionnalité" a largement pris le dessus sur l'aspect "performance". L'avantage majeur du protocole MPLS est de permettre un routage particulier pour chacun des LSP et donc d'associer un chemin (éventuellement différent du plus court chemin utilisé par les IGP) à chaque groupe de flots considéré. La granularité du choix des routes est donc améliorée permettant ainsi une meilleure gestion de la QoS et surtout une ingénierie de trafic plus aisée.

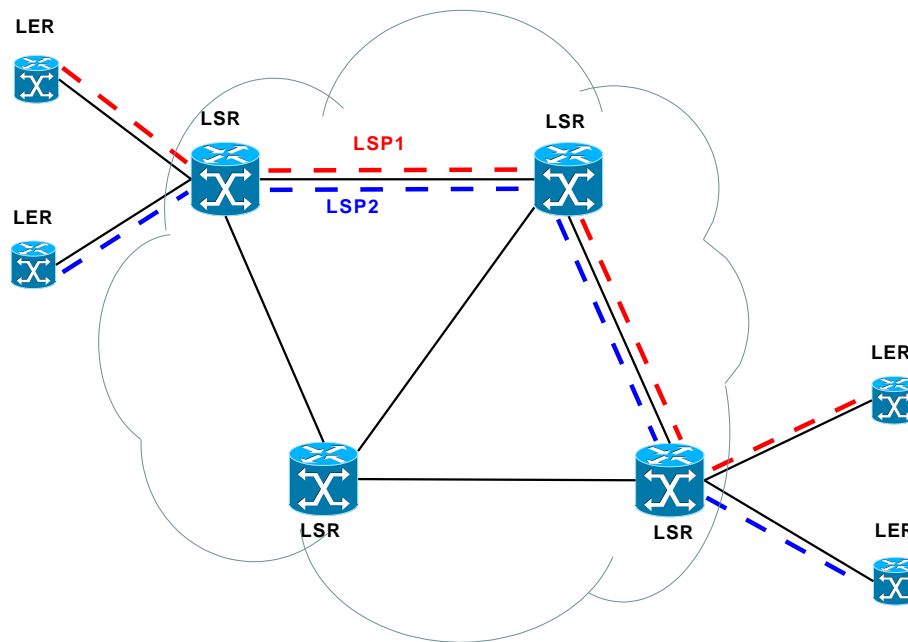


FIG. 1.4 – Réseau MPLS

Mais les avantages de MPLS ne s'arrêtent pas à ces améliorations. Il permet aussi un grand nombre de fonctionnalités très poussées :

- **Une intégration IP/ATM** : en effet, si la couche 2 est gérée par le protocole ATM (ou Frame Relay), MPLS permet l'utilisation de l'en-tête de couche 2. Cela évite l'ajout d'un en-tête supplémentaire et permet donc de diminuer la taille des données de signalisation.
- **La flexibilité** : MPLS permet l'utilisation de n'importe quelle couche 2 car ce protocole vient se placer entre la couche réseau et la couche de transmission de données. L'ajout d'une couche supplémentaire permet de coexister avec l'ensemble des protocoles existants (surtout pour la couche 2). De plus, il est possible de déployer un réseau IP/MPLS sur des infrastructures sous-jacentes hétérogènes (Ethernet, ATM, SDH, WDM, etc...)
- **La création de VPN** : la notion de réseau privé virtuel (VPN) s'appuie complètement sur le paradigme de routage intermédiaire proposé par MPLS. Il devient ainsi facile d'intégrer des VPN sur le cœur du réseau IP/MPLS de part l'affectation des labels directement associés aux VPN.
- **Approche DiffServ** : l'architecture d'un cœur de réseau IP/MPLS formé de LER aux frontières et de LSR dans le cœur est identique à celle retenue dans DiffServ nécessitant des fonctionnalités différentes aux extrémités et dans le cœur. De plus, l'association des trafics aux FEC permet et facilite l'affectation des différentes classes de trafic aux files d'attente dédiés. On trouve ainsi

dans MPLS la notion de “Per Hop Behaviour (PHB)” adaptée à l’architecture DiffServ.

- **L’ingénierie de trafic** : MPLS est souvent associé à son ingénierie de trafic (Traffic Engineering : TE) car la notion de routage explicite des LSP permet une simplification de la mise en place d’un routage calculé au préalable.

1.3.3 Les réseaux privés virtuels VPN

Les opérateurs de télécommunications proposent, depuis quelques années, des services VPN (Virtual Private Network). Les entreprises souscrivant à ces offres peuvent disposer d’une infrastructure privée et virtuelle de communication longue distance. Le client VPN peut ainsi connecter différents sites répartis géographiquement sans avoir à investir dans des liaisons louées et sans avoir à supporter les coûts liés à la gestion et à la maintenance d’une telle architecture.

L’architecture d’un VPN est décrite sur la figure 1.5. Pour l’entreprise souscrivant l’offre VPN, les noeuds du VPN apparaissent comme étant interconnectés suivant une topologie complètement maillée grâce à des liens virtuels. En fait, les noeuds VPN sont directement reliés à des routeurs IP du réseau opérateur, et les liens virtuels correspondent à des chemins entre ces routeurs dans le réseau IP.

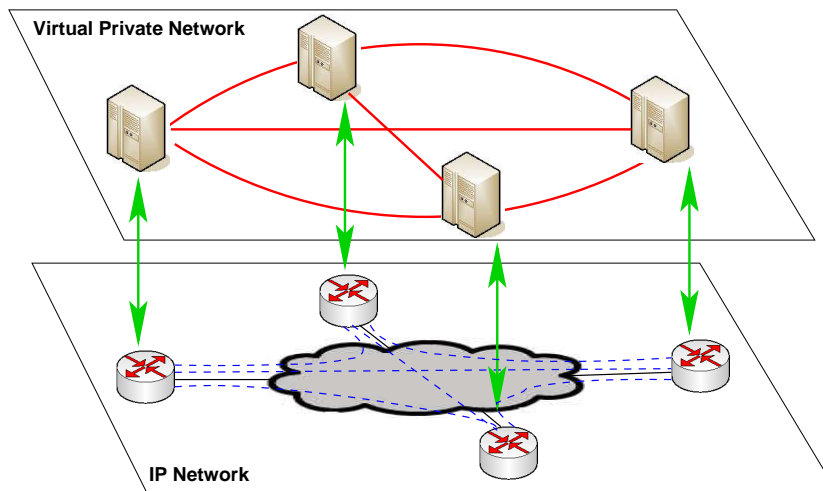


FIG. 1.5 – Architecture d’un réseau privé virtuel.

Il existe deux technologies largement déployées par les opérateurs pour la mise en place de services VPN : la technologie des VPN MPLS [48] et celle basée sur le protocole IP (IPSec, L2TP, GRE, UDP [50]). En termes de conception réseau, il y a une différence significative entre un VPN MPLS et un VPN IP. Tandis qu’avec un VPN IP les liens virtuels du VPN peuvent être routés sur plusieurs chemins en fonction des métriques de routage du réseau sous-jacent, chaque lien virtuel d’un VPN MPLS est routé sur un seul chemin en utilisant un LSP (Label Switched Path) à routage explicite. Le fournisseur

de services VPN peut ainsi mieux contrôler le placement des liens virtuels dans son réseau. De plus, la technologie MPLS lui permet de réserver de la bande-passante pour les LSP associés aux liens virtuels dans le réseau sous-jacent en utilisant le protocole RSVP [41].

Du point de vue de l'entreprise cliente, le VPN doit apparaître comme un réseau privé, offrant le même niveau de qualité de service qu'un réseau constitué de lignes louées, mais à un prix très inférieur. Pour cela, il est essentiel que l'opérateur garantisse la bande-passante allouée aux liens virtuels du VPN. Ceci suppose tout d'abord de spécifier la bande-passante nécessaire entre les différents noeuds du VPN, pour ensuite la réserver dans le réseau opérateur le long des chemins correspondants aux liens virtuels grâce au protocole RSVP[41].

1.4 Problèmes de planification des réseaux

Les évolutions de l'Internet, décrites dans les sections précédentes, conduisent à une multiplication des services offerts par les réseaux et à une croissance du nombre d'utilisateurs et des volumes de trafics qu'ils génèrent. Dans une société où l'information et la communication ont pris une telle importance, l'interruption des services offerts par le réseau, ou même une dégradation significative de la QoS sont de moins en moins acceptables. Ceci pose aux opérateurs ou FAI de nouveaux problèmes.

Initialement, pour faire face à ces problèmes, les opérateurs ou FAI se sont tournés vers un sur-dimensionnement des équipements. Les principales sources de dégradation de QoS étant les zones de congestion du réseau, limiter les risques de congestion par une augmentation conséquente des ressources permet d'écouler les volumes de trafic tout en garantissant la QoS requise.

Cependant, cette démarche n'est plus viable économiquement. Le contexte concurrentiel qui induit des marges bénéficiaires réduites ne permet plus d'améliorer les performances d'un réseau IP par un sur-dimensionnement excessif des équipements. D'un point de vue technique, cette démarche doit être modifiée si l'on veut vraiment maîtriser l'évolution du réseau.

La garantie de performances, ne peut s'obtenir sans une nouvelle approche de planification de réseaux. Les techniques de planification de réseaux, consistent à adapter le réseau (existant ou non encore installé) aux volumes de trafic et exigences de QoS qu'il doit supporter. Elles intègrent également les notions de résilience pour garantir non seulement une utilisation adéquate des ressources dans le réseau nominal, mais aussi des performances "acceptables" si le réseau est dans un état de panne. Planifier son réseau revient également à anticiper l'évolution globale du trafic, par exemple l'évolution des services dans le réseau et le nombre de clients par service.

Gourdin et al [75] classifie en différents types les problèmes de planification des réseaux :

- les problèmes de conception de réseaux,
- les problèmes de dimensionnement de réseaux (capacity planning),

- les problèmes d’allocation de ressources. Ce dernier type de problèmes est également connu sous le nom de problème de planification du routage ; il suppose que le réseau soit construit et dimensionné.

1.4.1 Conception de réseaux

Le problème de conception de réseaux, connu dans la littérature sous le nom de “Network Design”, se pose généralement aux opérateurs souhaitant la mise en place de nouvelles infrastructures de communication ou le remplacement de celles existantes afin de pouvoir répondre aux nouvelles exigences des utilisateurs (communications intranet, extranet, internet,...).

Ce problème consiste, étant donné un ensemble de sites prédéfini pour l’installation d’équipements ainsi que des caractéristiques (le plus souvent, estimées) des flux inter-sites, à sélectionner un sous-ensemble des sites à activer et à trouver un plan pour les connecter à moindre coût. Ce coût est constitué de coûts d’installation des équipements et de connexion des sites. En général, le réseau résultant doit satisfaire à certains standards de fiabilité et de performance (cf. figure 1.6).

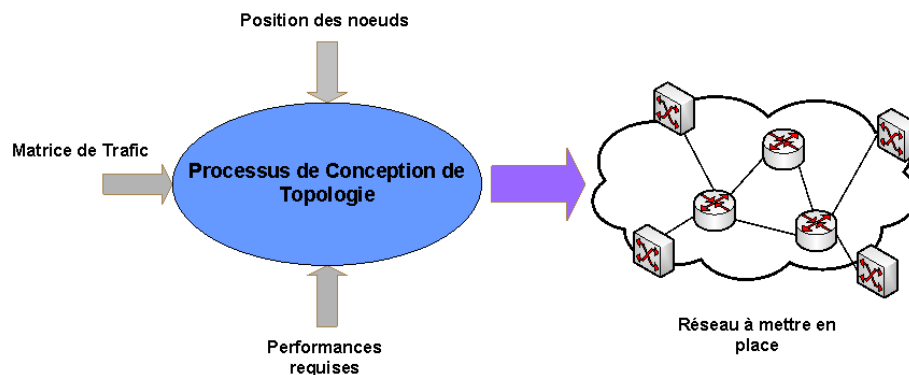


FIG. 1.6 – Processus de conception de réseaux

Une variante des problèmes de conception de réseaux concerne l’extension de la topologie d’un réseau. On suppose ici que le réseau est connu, mais on cherche à modifier sa topologie. Par exemple, on peut ajouter de nouveaux nœuds pour desservir de nouveaux points de présence. On peut également vouloir supprimer certains liens et en ajouter d’autres pour faire face à l’évolution de la demande.

1.4.2 Planification du routage

Comme nous l’avons précisé précédemment, le terme routage désigne l’ensemble des mécanismes mis en œuvre dans un réseau pour déterminer les routes qui vont acheminer les paquets d’un terminal émetteur à un terminal récepteur. Les algorithmes de routage sont utiles bien évidemment pour l’acheminement des données mais aussi pour l’allocation des ressources le long des chemins. L’introduction

de la qualité de service de bout en bout rend ces problèmes de routage plus complexes.

L'approche la plus simple pour effectuer le routage d'un flot donné entre une source et une destination consiste à choisir le plus court chemin. C'est l'idée sous-jacente du routage best-effort des réseaux IP (c'est aussi le routage par défaut dans MPLS), qui utilise des algorithmes distribués de plus court chemin.

Le problème d'optimisation du routage consiste à établir un schéma d'acheminement des flux permettant d'éviter la congestion des équipements et de répartir au mieux les flux dans le réseau. Une instance de ce problème est illustré dans la figure 1.7.

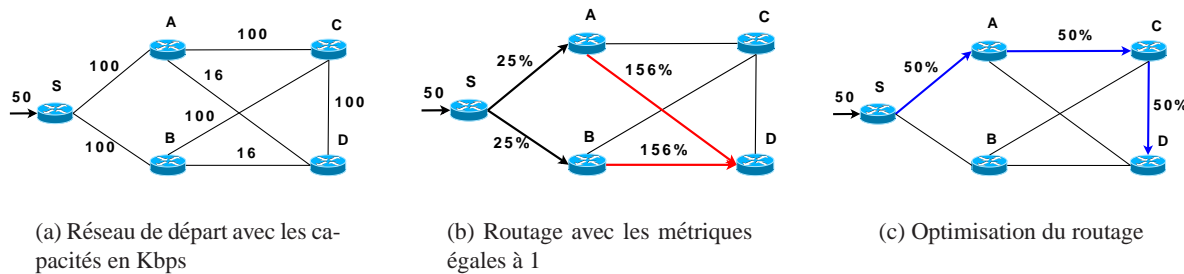


FIG. 1.7 – Exemples d'instances du problème d'optimisation de routage

Cette figure illustre l'intérêt des modifications du plan de routage dans les réseaux OSPF. La figure 1.7(a) illustre la topologie de départ où toutes les métriques sont unitaires. Les valeurs des capacités des interfaces sont données en Kbps sur la figure et un seul trafic, de demande 50 Kbps, est émis entre S et D.

Dès lors, un premier routage nous donne deux routes de S vers D (voir figure 1.7(b)). Le partage de flot équitable répartit donc 25 Kbps sur chaque route. Mais ces routes empruntent les interfaces A/D et B/D qui ne peuvent donc pas supporter tout le trafic (leurs capacité est de 16 Kbps). Leurs utilisations valent donc 156%. L'opérateur est donc appelé à optimiser le routage en changeant les métriques de certains liens (ou interfaces). Si l'opérateur garde, par exemple, les métriques des interface S/A et A/C à 1 et augmente les métriques des autres interfaces, il obtient le plan de routage décrit dans la figure 1.7(c). Ainsi, l'utilisation maximale des interfaces diminue en passant de 156% à 50%.

1.4.3 Dimensionnement de réseaux

Les technique d'optimisation du routage ont comme particularité principale de permettre une amélioration des ressources existantes sans surcoût supplémentaire. En effet, l'attrait des techniques d'optimisation du routage est de permettre une amélioration de QoS sur le réseau sans avoir à modifier les équipements même de ce réseau.

Cependant, du fait d'une constante évolution de la demande en trafic, il est souvent nécessaire de faire évoluer le réseau lorsque l'optimisation du routage ne permet plus d'atteindre le niveau de perfor-

mance cible. Si les ressources ne sont pas suffisantes, aucune ingénierie ne pourra permettre de garantir une QoS acceptable.

Le dimensionnement de réseaux connu sous le nom de “capacity planning” est un des problèmes auxquels les opérateurs ont à faire face dans ce cas là. L’objectif est d’affecter à chaque équipement du réseau une capacité pour que le réseau puisse atteindre le niveau de performance cible à moindre coût.

1.4.4 Contraintes opérationnelles

Généralement, les problèmes de planification de réseaux doivent tenir compte de contraintes opérationnelles souvent exigées par les opérateurs. Nous détaillons dans ce paragraphe certaines contraintes qui sont classiquement utilisées.

Gestion des pannes

Certains services proposés par les opérateurs de télécom nécessitent une disponibilité permanente du réseau. Les interruptions non prévues de ces services peuvent contribuer à la diminution de la fidélité des clients. Il est donc important d’intégrer la notion de panne dans le processus de planification du réseau pour garantir non seulement une utilisation adéquate des ressources dans le réseau nominal, mais aussi des performances acceptables si le réseau est dans un état de panne. Planifier son réseau revient donc à anticiper par la mise en place de scénarii de secours pour chaque panne possible.

Planification multi-couches des réseaux

Outre sa décomposition en niveaux hiérarchiques (réseaux d’accès et backbone), une architecture de télécommunications peut être décomposée en couches technologiques empilées les unes au dessus des autres. Par exemple, une architecture est IP sur ATM, le réseau de transport ATM étant lui même supporté par un réseau de transmission SDH ou WDM. Les couches technologiques opèrent suivant une relation client/serveur : les liens d’une couche cliente correspondent à des chemins dans la topologie de réseau de la couche inférieure. Ainsi, par exemple, les liens d’une architecture IP sont réalisés par des PVC ATM, qui eux mêmes peuvent correspondre à des chemins dans un réseau SDH ou WDM (cf figure 1.8).

Les couches d’un réseau développent chacune leur propre architecture, indépendamment les unes des autres. Rien ne garantit l’optimalité de l’ensemble des couches superposées [47, 57] . Par exemple, le choix du routage dans une couche cliente peut induire une consommation excessive de ressources au niveau de la couche transport. Cette approche couche par couche n’est donc pas adaptée pour concevoir et faire évoluer un réseau à moindre coût.

Le problème de planification multi-couche concerne la conception de méthodes de planification et d’optimisation multi-couche et multi-niveau de réseaux de télécommunications. Typiquement, il s’agit de déterminer la topologie optimale d’un réseau, de dimensionner à moindre coût ses équipements

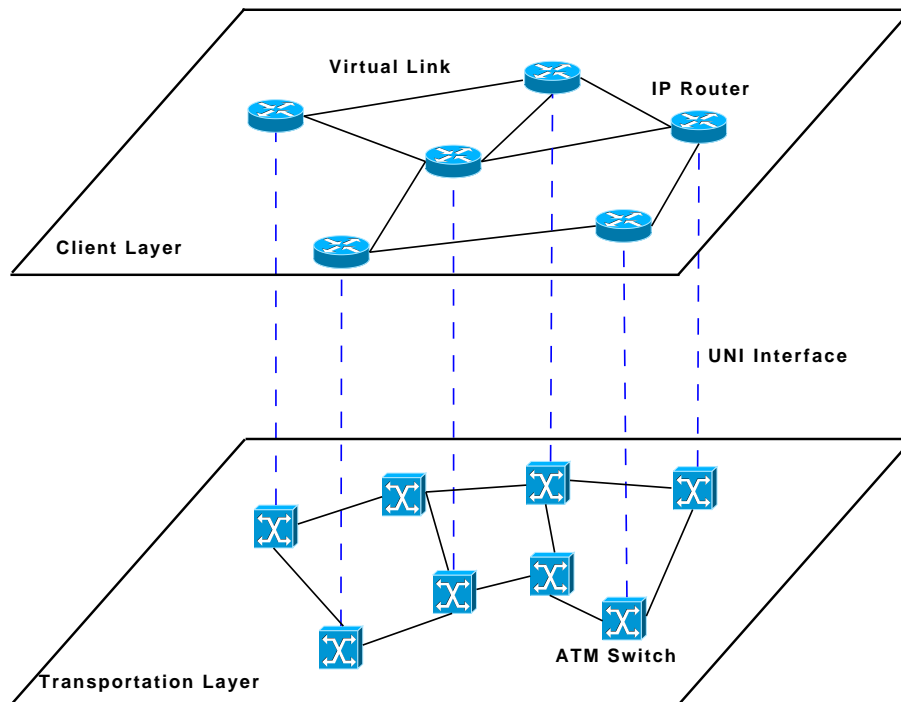


FIG. 1.8 – Réseau en deux couches

(routeurs, commutateurs, liens, ...) et de router les flots de communications, en prenant en compte les contraintes de résilience et de QoS. L'objectif est de faire une conception simultanée des différentes couches de communication d'une architecture de réseau hiérarchique. Comme expliquée précédemment, une telle approche intégrée des problèmes de planification revêt une importance grandissante pour les opérateurs de télécommunication du fait des réductions de coût considérables qu'elle doit permettre et aussi du fait du couplage dynamique existant dans les nouvelles couches protocolaires.

Incertitudes sur la demande

La plupart des techniques de planification de réseau (optimisation du routage, dimensionnement, etc.) suppose connaître la demande en trafic de manière précise. Cette hypothèse a l'avantage de simplifier grandement les problèmes d'optimisation auxquels le planificateur est confronté, et permet souvent de les résoudre de manière optimale.

Toutefois, en pratique, on ne connaît pas la volumétrie et la composition du trafic. De plus, la matrice de trafic change continuellement. Avec la popularité croissante de nombreuses applications gourmandes en bande-passante, on a d'ailleurs pu constater un comportement beaucoup plus volatil du trafic, avec des changements abrupts de la matrice de trafic. Ceci constitue un obstacle majeur à l'utilisation des techniques traditionnelles d'ingénierie du trafic.

La prise en compte des incertitudes sur la matrice de trafic et une planification robuste des réseaux permettant de faire face à l'incertitude sur la demande sont des enjeux majeurs pour les opérateurs.

1.5 Conclusion

Dans ce chapitre, nous avons décrit brièvement le fonctionnement d'un réseau IP à travers les deux protocoles de routages les plus répandus (OSPF, BGP). Nous avons ensuite présenté les évolutions majeures d'IP sur deux plans : (1) le plan technologique et (2) le plan de la convergence des services fournis aux utilisateurs.

La multiplication des services que doivent fournir les réseaux, l'augmentation des trafics et leur nature de plus en plus complexe, les exigences des services en terme de qualité de service demandent des architectures de réseaux de plus en plus élaborées. La gestion de tels réseaux devient complexe et nécessite des mécanismes "intelligents". La planification de son réseau permet à l'opérateur de mettre en place des mécanismes pour faire face aux divers problèmes de gestion de ressources dans les réseaux multi-services.

C'est dans ce contexte très général que se situe ces travaux de thèse. Nous nous intéressons plus précisément aux problèmes de conception et d'optimisation des réseaux. Nous proposons, dans les prochains chapitres, d'étudier en détails certains de ces problèmes.

Mais avant de s'intéresser à ces problèmes, nous présentons dans le chapitre qui suit un panorama des méthodes d'optimisation. Nous nous attardons sur les techniques qui seront utilisées dans la suite de ce rapport.

CHAPITRE 2

Panorama sur les techniques d'optimisation

2.1 Introduction

Comme nous l'avons précisé dans le chapitre précédent, un fournisseur d'accès ou un opérateur est souvent confronté au problème de concevoir et d'exploiter des réseaux de télécommunications de manière efficace, notamment sur le plan économique. Le but est d'offrir une haute qualité de service aux clients tout en maintenant les coûts d'équipement et d'opération les plus bas possible. La rentabilité des investissements est évidemment un facteur clé pour la réussite de l'entreprise.

Les problèmes d'optimisation qui apparaissent lors de la conception ou de l'exploitation de réseaux sont regroupés, dans ce document, sous le nom de problèmes d'optimisation de réseau. La formalisation et la résolution de ces problèmes fait appel à des modélisations et des méthodes qui sont issues de la théorie des graphes, de la recherche opérationnelle, de la théorie des files d'attente ...

Il existe deux familles de méthodes pour résoudre les problèmes d'optimisation de réseau (et les problèmes d'optimisation d'une manière générale) : les méthodes exactes et les méthodes approximatives aussi appelées heuristiques.

Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles telles que la programmation linéaire, la programmation linéaire en nombres entiers, l'optimisation par séparation-évaluation (branch-and-bound), ou les algorithmes avec retour arrière (backtracking). Les méthodes exactes permettent de trouver des solutions optimales pour des problèmes de taille raisonnable. Mais malgré les progrès réalisés (notamment en matière de programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés avec les instances de taille importante.

Si les méthodes de résolution exactes permettent d'obtenir une ou plusieurs solutions dont l'optimalité est garantie, dans certaines situations, on peut cependant se contenter de solutions de bonne

qualité, sans garantie d'optimalité, mais obtenues en un temps de calcul réduit. Typiquement ce type de méthodes, dites heuristiques est particulièrement utile pour les problèmes nécessitant une solution en temps réel (ou très court) ou pour résoudre des problèmes difficiles sur des instances numériques de grande taille.

Dans ce chapitre, nous présentons les principales techniques, issues de la théorie des graphes et de la recherche opérationnelle, de résolution des problèmes d'optimisation. L'objectif est d'introduire les notions et les outils qui seront utilisés dans la suite de ce mémoire. Ce chapitre ne présente pas un état de l'art exhaustif des techniques du domaine ; nous renvoyons donc à la lecture de [17], par exemple, pour un exposé plus complet.

Nous proposons en premier lieu d'introduire quelques techniques traditionnellement utilisées pour la résolution exacte des problèmes d'optimisation : la programmation linéaire et la programmation linéaire en nombres entiers. Nous présentons ensuite quelques méthodes heuristiques. Nous abordons, à la fin de ce chapitre, les problèmes de flots dans les réseaux. Nous introduisons dans ce contexte les problèmes classiques de flots ainsi que les techniques utilisées pour les résoudre.

2.2 Programmation linéaire et en nombres entiers

En général, un problème d'optimisation est caractérisé par un ensemble de solutions S , et par une fonction coût f (appelée aussi fonction objectif). La fonction f affecte à chaque solution $s \in S$ la valeur $f(s)$. L'ensemble S est généralement défini par un système d'équations ou d'inéquations appelées "contraintes". Ces contraintes traduisent l'autorisation ou l'interdiction d'une combinaison de valeurs.

Résoudre un tel problème consiste à chercher le minimum s^* de la fonction coût f sur l'ensemble S :

$$(P) \begin{cases} f(s^*) = \min f(s) \\ \text{s.t.} \\ s \in S \end{cases}$$

Une telle solution s^* s'appelle une solution optimale, ou un optimum global. La définition concerne la minimisation, mais il suffit de remarquer que maximiser f équivaut à minimiser $-f$.

Si l'ensemble S est formulé comme un ensemble de variables à valeurs dans l'ensemble des réels \mathbb{R} et contraintes à satisfaire des inégalités linéaires et si f est une fonction linéaire en ces variables, on parle alors d'un problème de programmation linéaire (PL).

La programmation linéaire est un outil très puissant de la recherche opérationnelle qui permet de résoudre un grand nombre de problèmes linéaires. On distingue dans la programmation linéaire, la programmation linéaire en nombres réels, pour laquelle les variables des équations sont dans \mathbb{R} et la programmation en nombres entiers (PLNE), pour laquelle les variables sont dans \mathbb{N} . Bien entendu, il est possible d'avoir les deux en même temps (programmation linéaire mixte). Cependant, la résolution

d'un problème avec des variables entières est nettement plus compliquée qu'un problème en nombres réels.

Des résultats d'algèbre linéaire ont montré qu'un programme linéaire a une interprétation géométrique simple. L'ensemble S des solutions réalisables est par définition un polyèdre (un polytope si S est borné) de l'espace euclidien \mathbb{R}^n . Si S est non vide, le problème (PL) possède une solution. On dit alors que PL est réalisable. Dans ce cas toute solution optimale de (PL) se trouve nécessairement à un sommet du polytope.

La méthode de résolution du Simplexe repose sur cette observation. L'algorithme du Simplexe construit tout d'abord une solution réalisable qui est un sommet du polytope des solutions admissibles puis se déplace selon les arêtes du polytope pour atteindre des sommets pour lesquels la valeur de l'objectif est de plus en plus petite, jusqu'à atteindre l'optimum. En théorie, la méthode du Simplexe a une complexité non polynomiale et est donc supposée peu efficace. Cependant, dans la pratique, les algorithmes de Simplexe sont à ce jour encore les plus utilisés. Ces derniers sont généralement implémentés dans les logiciels de résolution linéaire.

Cependant l'algorithme du Simplexe ne peut s'appliquer qu'aux problèmes en variables continues ou à des problèmes en variables entières ayant une matrice de contraintes T unimodulaire (car dans ce cas, tous les sommets de l'ensemble de recherche sont entiers) comme les problèmes de transport ou d'affectation. Pour les PLNE il existe plusieurs méthodes de résolution exacte telles que :

- les méthodes de recherches arborescentes,
- la programmation dynamique,
- les méthodes de plan de coupe (cutting plane) consistant à ajouter progressivement des contraintes supplémentaires.

2.2.1 Méthodes de recherches arborescentes

Le principe de ces méthodes consiste à représenter l'ensemble (fini) des solutions du programme linéaire en nombres entiers à résoudre par une arborescence et à parcourir cette arborescence pour chercher la/les meilleure(s) solution(s). L'algorithme de Séparation & Evaluation, plus connu sous son appellation anglaise Branch & Bound (BB, ou B&B), est la technique la plus connue et la plus utilisée de cette famille de méthodes de résolution. Cette méthode est une mise en application du principe latin "Diviser pour Régner"

Le principe de l'algorithme B&B consiste à ne parcourir que les sous-ensemble de solutions réalisables du problème dont on a réussi à montrer qu'ils contiennent (au moins) une solution optimale. Il s'agit de séparer (branch) les solutions du problèmes en plusieurs ensembles de solutions et d'utiliser une évaluation partielle pour éliminer des solutions grâce à des bornes sur les valeurs possibles des solutions (bound). Grâce à cette étape d'élimination on empêche l'algorithme d'énumérer explicitement toutes les solutions.

Il y a donc trois ingrédients principaux qui caractérisent une procédure de B&B : (1) l'ordre de sélection des nœuds, (2) la façon de séparer l'espace ou schéma de branchement et (3) la méthode d'évaluation des nœuds pour tenter d'identifier ceux ne contenant pas de solution optimale.

1. **Ordre de sélection** : il existe trois stratégies usuellement mises en œuvre :
 - Profondeur d'abord,
 - Largeur d'abord,
 - Meilleur d'abord : le parcours dépend de l'estimation des nœuds restant à visiter. Selon cette estimation, on choisira d'examiner en premier un nœud susceptible de contenir une solution optimale, ou menant rapidement à une meilleure solution réalisable.

2. **Schémas de branchement** : il s'agit de séparer l'espace de recherche en plusieurs parties. Dans la pratique, ces espaces sont le plus souvent disjoints.

3. **Evaluations** : Un nœud de l'arbre n'est exploré que s'il conduit à une solution strictement meilleure que la meilleure solution connue. A chaque nœud, on évalue la meilleure solution contenue dans le sous-espace E associé à ce nœud. Puisque ce PLNE est généralement aussi dur que le programme original, on se contente d'encadrer sa valeur optimale z_E par une borne inférieure LB_E . On pose UB la valeur de la meilleure solution réalisable connue. Si $LB_E \geq UB$ alors E ne contient pas de solution strictement améliorante et il n'est pas nécessaire d'explorer plus en avant le sous-arbre de solutions issues de ce nœud.

Illustrons le fonctionnement du B&B sur un exemple classique du monde des réseaux. Considérons un réseau reliant un certain nombre de nœuds, et supposons que l'on cherche à utiliser le plus court chemin pour router le trafic d'un nœud i vers un nœud j . Un nœud de l'arbre de recherche sera ici un chemin dans le réseau, dont une extrémité sera i . On saura que l'on a atteint une feuille de l'arbre (une solution potentielle) lorsque l'autre extrémité du chemin sera j . La racine de l'arbre est un chemin de longueur nulle, ne contenant que i . Les nœuds fils sont simplement les chemins obtenus en considérant les nœuds voisins du dernier nœud atteint, comme illustré sur la figure 2.1. On choisit comme fonction de coût la distance parcourue.

Supposons qu'en explorant l'arbre en profondeur, on ait trouvé le chemin 1, décrit en figure 2.1. Cette solution est mémorisée, puis on remonte dans l'arbre d'états en revenant au nœud précédent j dans ce chemin, avant d'essayer un autre nœud fils, comme illustré sur la figure 2.2. Le nouveau chemin testé, qui se termine par le nœud k , est plus long que la solution déjà trouvée. Ce n'est donc pas la peine de continuer à chercher à joindre j en considérant les nœuds voisins de k , et on arrête l'exploration de cette branche de l'arbre de solutions.

2.2.2 La Programmation dynamique

Quelquefois, en essayant de résoudre un problème de grande taille par l'approche arborescente, on s'aperçoit que la décomposition génère plusieurs sous problèmes identiques et qu'on est conduit à cal-

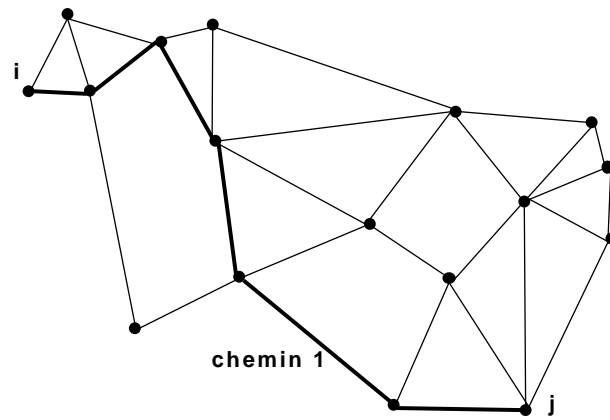


FIG. 2.1 – Premier chemin trouvé avec une recherche en profondeur.

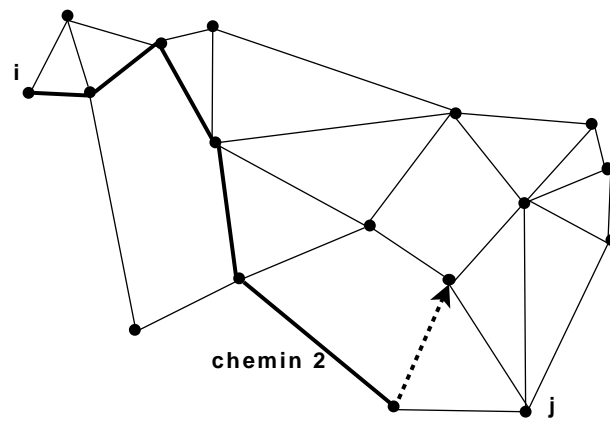


FIG. 2.2 – Essai d'un autre nœud fils.

culer de nombreuses fois la solution de mêmes sous problèmes.

Le principe de base de la programmation dynamique est d'éviter de calculer deux fois la même chose, normalement en utilisant une table de résultats déjà calculés, remplie au fur et à mesure qu'on résout les sous problèmes. L'efficacité de cette méthode d'optimisation repose sur le principe d'optimalité de Bellman : "Toute politique optimale est composée de sous-politiques optimales".

En effet, la programmation dynamique s'appuie sur une relation entre la solution optimale du problème et celles d'un nombre fini de sous-problèmes. Concrètement, cela signifie que l'on va pouvoir déduire la solution optimale d'un problème de taille n à partir de solutions optimales de sous-problèmes de taille inférieure à n .

Le problème du plus court chemin qu'on a introduit dans la section précédente peut se résoudre avec la technique de programmation dynamique. La longueur d'un chemin est la somme des poids des arcs de ce chemin. Considérons un plus court chemin entre i et j et $k \neq i, j$ un sommet intermédiaire sur ce chemin. Les sous-chemins de ce chemin, de i à k et de k à j sont aussi de coût minimal (sinon, en remplaçant un de ces sous-chemins par un chemin de moindre distance de mêmes extrémités, on diminuerait la distance du chemin de i à j , ce qui est impossible). Ce problème satisfait donc au principe d'optimalité. L'algorithme de résolution est une méthode ascendante qui calcule les plus courts chemins entre chaque pair de sommets en utilisant comme sommets intermédiaires, dans l'ordre et de façon successives, les sommets $1, 2, 3 \dots n$, avec n la taille du graphe. Notons d_{ij}^k la distance du chemin de i à j minimal parmi les k -chemins, c'est-à-dire ceux ayant comme sommets intermédiaires $1, 2, \dots, k$. A l'état initial aucun sommet intermédiaire n'est utilisé, on a donc $d_{ij}^0 = \infty$. A l'itération k on a la relation suivante :

$$d_{ij}^k = \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}).$$

2.3 Méthodes heuristiques

Les méthodes de recherche complètes (basées sur des parcours arborescents) sont souvent utilisées pour obtenir des solutions optimales et effectuer des preuves d'optimalité. Mais, du fait de leur comportement (souvent) exponentiel, ces méthodes peuvent s'avérer trop gourmandes en temps de calcul.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. Les méthodes approchées sont fondées principalement sur diverses heuristiques, souvent spécifiques à un type de problème. Dans cette section, nous passons en revue les principales heuristiques utilisées dans les prochains chapitres de ce document.

2.3.1 La Recherche locale

La recherche locale, appelée aussi l'amélioration itérative, représente une classe de méthodes heuristiques très anciennes. La recherche locale est une technique d'approximation très utilisée en pratique pour résoudre les problèmes d'optimisation combinatoire difficiles.

Une méthode de recherche locale est un processus itératif fondé sur deux éléments essentiels : un voisinage N et une procédure exploitant le voisinage. L'idée est d'améliorer itérativement la solution courante x^i en explorant un voisinage $N(x^i)$ de celle-ci pour y trouver une solution de moindre coût. Les solutions dites voisines sont généralement obtenues en appliquant une transformation élémentaire à la solution courante.

Cette procédure fait intervenir à chaque itération le choix d'un voisin qui améliore la configuration courante. Plusieurs possibilités peuvent être envisagées pour effectuer ce choix. Il est possible d'énumérer les voisins jusqu'à ce qu'on en découvre un qui améliore strictement (première amélioration). Cette technique est connue dans la littérature sous le nom de "first descent". On peut également rechercher le meilleur voisin. C'est la méthode de plus grande descente connue sous le nom de "steepest descent". Cette dernière solution peut sembler plus coûteuse, mais le voisin découvert sera en général de meilleure qualité. La version de recherche locale décrite par l'Algorithme 1 est celle de "steepest descent".

algorithm 1 Algorithme de Recherche Locale

```
1: procédure Local_Search
2: choose initial point ( $x$ )
3: repeat
4:    $z \leftarrow x$ 
5:   for  $y \in N(x)$  do
6:     if  $F(y) < F(z)$  then
7:        $z \leftarrow y$ 
8:     end if
9:   end for
10: until  $z = x$ 
```

L'algorithme s'arrête si la dernière configuration trouvée ne possède pas de voisin strictement meilleur qu'elle-même. La recherche locale est alors dans un minimum local.

L'avantage principal de cette méthode réside dans sa grande simplicité et sa rapidité. Mais les solutions produites peuvent être de qualité médiocre et de coût très supérieur au coût optimal. Pour remédier à ce problème, la solution la plus simple est la méthode de relance aléatoire (ou "restart", ou "multi-start") qui consiste à générer une nouvelle configuration de départ de façon aléatoire et à recommencer une descente. On remarque cependant que cette solution ne tire aucun profit des optima locaux déjà découverts. Une autre solution consiste à accepter des voisins de même performance que la configuration courante. Cette approche permet à la recherche de se déplacer sur les plateaux, mais n'est pas suffisante pour sortir de tous les optima locaux. D'autres techniques plus élaborées sont également

possibles, par exemple, : l'introduction de voisinages variables [11, 13] et les techniques de réduction [9, 13] ou d'élargissement [10].

2.3.2 Algorithmes gloutons

Les algorithmes gloutons (en anglais "greedy") appartiennent à la famille des approches de résolution par construction. Le principe d'une méthode gloutonne est très simple : on construit une solution de manière incrémentale en rajoutant à chaque pas un élément selon un critère glouton, i.e. celui qui nous paraît "localement" le meilleur. Ce choix local ne remet pas en cause les choix effectués précédemment.

Si cette vision à court terme nous donne toujours une solution optimale, on parlera d'algorithme glouton exact sinon d'heuristique gloutonne.

Pour assurer que l'algorithme glouton donne une solution optimale pour un problème, il faut montrer que ce problème a les propriétés suivantes :

1. **Propriété du choix glouton** : Une solution globalement optimale peut être trouvée en faisant des choix localement optimaux (choix gloutons).
2. **Propriété de sous-structure optimale** : Après avoir effectué une première étape gloutonne, le problème qui reste à résoudre est une forme réduite du problème initial. En d'autres termes, si on retire la première étape d'une solution optimale pour le problème initial, on obtient une solution optimale pour le problème réduit.

L'algorithme de Kruskal de recherche d'un arbre couvrant de poids minimum (ACM) [3] est un exemple d'algorithme glouton exact.

2.3.3 Recherche à écarts limités LDS

LDS (Limited Discrepancy Search) est une méthode de recherche partielle introduite par Harvey et Ginsberg [34]. Cette méthode est une méthode arborescente.

L'idée est de se baser sur une solution initiale, obtenue avec une méthode heuristique h , et d'autoriser des divergences par rapport à cette solution.

En considérant que h peut se tromper un petit nombre η de fois, on s'autorise donc η écarts (discrepancies) à l'heuristique h . Pour un nombre maximal η d'écarts autorisé, LDS explore l'arbre de recherche de manière itérative selon un nombre croissant d'écarts allant de $i = 0$ à $i = \eta$. A chaque itération, i est incrémenté de 1. Ceci revient à remettre en cause progressivement les choix de l'heuristique h , et ce, afin d'améliorer à chaque itération la qualité de la solution. Si la valeur de η est très grande, cela conduit au développement complet de l'arborescence.

L'inconvénient principal de LDS est sa redondance dans le parcours de l'arborescence. En effet, pour avoir les solutions à η divergences, les solutions de 0 à $\eta - 1$ divergences sont revisitées. Pour éviter ce problème, ILDS ("Improved Limited Discrepancy Search") a été proposée dans [36]. Cette méthode permet à la recherche d'éviter de se diriger vers les feuilles de profondeur η dont la divergence est inférieure à celle de l'itération courante. Ainsi, chaque feuille n'est plus visitée qu'une seule fois.

La méthode LDS peut être utilisée dans un contexte de recherche locale. L'idée est de prendre comme voisinage toutes les solutions ayant η écarts de la solution courante. L'exploration du voisinage se fait donc par un parcours arborescent.

2.4 Métaheuristiques - méthode tabou

Depuis des années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées métaheuristiques [28], [40]. Une métaheuristique est constituée d'un ensemble de règles et de mécanismes (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation.

Les métaheuristiques sont représentées essentiellement par les méthodes de voisinage comme le recuit simulé et la recherche tabou, et les algorithmes évolutifs comme les algorithmes génétiques et les stratégies d'évolution. Dans cette section nous détaillons les métaheuristiques de recherche tabou.

La méthode tabou a été développée par Fred Glover [23, 25]. Son objectif est de surmonter le problème des optima locaux de la recherche locale tout en évitant au maximum de cycliser.

Commençant à partir d'une solution initiale réalisable, le processus consiste, à chaque itération, à choisir la meilleure solution dans le voisinage de la solution courante, même si cette solution n'entraîne pas une amélioration. La recherche tabou ne s'arrête donc pas au premier optimum trouvé.

Cette stratégie peut entraîner des cycles. Pour empêcher ce type de cycle, on crée une liste T qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Par ailleurs, en s'interdisant ainsi certaines permutations, un mouvement qui aboutirait à une solution meilleure que toutes celles déjà visitées risquerait d'être écarté. Pour éviter cela, on incorpore dans l'algorithme des critères d'aspiration, qui autorisent certains mouvements, bien qu'interdits par la liste T , parce qu'on suppose qu'ils vont améliorer la recherche.

Deux composants importants de la recherche tabou sont les stratégies d'intensification et de diversification. Ces techniques font appel à des mémoires à long terme.

2.4.1 Intensification

L'idée à la base de l'intensification est qu'on devrait explorer de façon plus approfondie les régions qui semblent les plus prometteuses (où l'optimum global est susceptible de se trouver), évitant ainsi les optima locaux. De nombreuses techniques ont été proposées dans ce sens :

- Repartir de bonnes solutions (dites solutions d'élite) déjà rencontrées. Ces solutions sont conservées dans l'ordre pour que l'on puisse examiner leurs voisins immédiats. Cette technique a été proposée par Chakrapani et Skorin-Kapov [31].
- Reconstruire une solution de départ qui tente de combiner des attributs qui ont été présents souvent dans les meilleures configurations. Pour ce faire, on ne considère que les attributs souvent présents dans les configurations d'élite et on incite à les introduire dans la solution reconstruite,
- Geler certains attributs qui ont été souvent présents dans les configurations visitées. Dans la pratique, ceci consiste à fixer un seuil de fréquence et à ne prendre en compte que les mouvements au dessus de ce seuil.

2.4.2 Diversification

L'objectif de la diversification, opposé à celui de l'intensification, est d'orienter la recherche vers de nouvelles régions non encore explorées, et peut-être, très intéressantes.

Deux techniques de diversification sont couramment utilisées :

- **Diversification par relance** : selon cette technique, on interrompt périodiquement la recherche. On effectue alors une relance en construisant une nouvelle solution qui contient des attributs de fréquence faible. Une technique possible pour reconstruire la nouvelle solution est de partir d'un attribut qui n'a jamais été présent dans la solution (attribut de fréquence nulle). Puis, on complète la solution de manière gloutonne.
- **Diversification en continu** : Périodiquement et pendant une certaine durée (phase de diversification), on modifie l'évaluation des mouvements en ajoutant à l'objectif un terme relié à l'ancienneté ou à la fréquence des attributs.

2.5 Problèmes de flots

Les problèmes de flots interviennent très fréquemment dans de nombreux problèmes d'optimisation combinatoire, notamment les problèmes de routage dans les réseaux de télécommunication. Ces problèmes consistent à acheminer une certaine demande ou commodité (en anglais "commodity"), entre des sites de production et des sites de consommation, via un graphe appelé réseau de transport.

Il existe des algorithmes efficaces, c'est-à-dire en temps polynomial, capables de les résoudre de manière exacte. Il est donc important de savoir modéliser un problème comme un problème de flot lorsque cela est possible, et de connaître les principaux algorithmes employés pour résoudre ces

problèmes.

De nombreux problèmes d'optimisation se ramènent à la recherche de flots particuliers sur un graphe, les plus connus sont le problème de flot maximum, le problème de flot compatible, le problème de flot maximal de coût minimum ou encore le problème de transport. Dans la suite, nous faisons une description succincte de ces problèmes. Pour plus de détails, nous invitons le lecteur à se référer à [32].

2.5.1 Notion de flot

Un flot f dans un réseau de transport est une fonction qui associe à chaque arc e une quantité $f(e)$ qui représente la quantité transportée sur cet arc, en provenance de la source et en destination du puits. Un flot doit respecter la loi de conservation dite aussi loi de Kirchhoff : à chaque sommet i différent de la source et du puits, la totalité du flot entrant dans i est égale à la totalité du flot sortant de i .

2.5.2 Le problème de flot maximum

Connaissant les capacités des arcs d'un réseau de transport, le problème du flot maximum consiste à trouver quelle est la quantité maximum de flot qui peut circuler d'une source s à un puits t . La figure 2.3 montre trois solutions différentes d'une instance du problème de flot maximum. Sur chaque arc (i, j) figure en premier la valeur du flot, puis la capacité de l'arc.

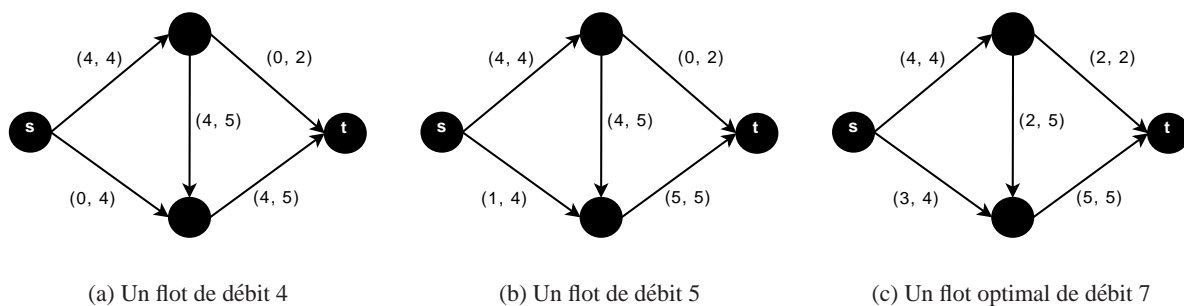


FIG. 2.3 – Exemples de flots

L'algorithme le plus connu pour résoudre ce problème est celui de Ford et Fulkerson [2]. Cet algorithme se base sur le théorème de Ford et Fulkerson qui annonce que, dans un réseau de transport avec des capacités bien déterminées pour chaque arc, le flot maximum est celui qui sature l'arc de capacité minimum, c'est à dire celui qui remplit le plus petit goulot d'étranglement du graphe. En langage formel, on appelle cela la coupe de capacité minimum. L'algorithme consiste alors à partir d'un flot F réalisable (le flot nul par exemple) et à l'améliorer itérativement. A chaque étape l'algorithme vérifie s'il existe un chemin augmentant pour le flot, c'est à dire un chemin (orienté) de s à t dans le réseau résiduel. Ceci revient à chercher une chaîne améliorante dans le graphe d'écart. Si un tel chemin existe,

il est saturé dans le réseau résiduel et le flot correspondant est ajouté à F .

2.5.3 Le problème de flot compatible

Ce problème est une extension du problème de flot maximum. Dans cette variante du problème on suppose que chaque arc e possède non seulement une borne supérieure b_e mais également une borne inférieure a_e sur la quantité de flot pouvant y circuler. Un flot compatible est un flot qui vérifie les conditions de limitation de capacité suivantes :

$$a_e \leq f(e) \leq b_e \quad \forall e$$

Le réseau de transport représenté sur la figure 2.4 n'admet pas de flot compatible. Sur cette figure, sur chaque arc (i, j) est indiqué en première (resp. seconde) position la quantité minimale (resp. maximale) de flot qui peut transiter par cet arc. Il ne peut exister de flot compatible pour ce réseau de transport car on doit avoir un flot au moins égal à 4 sur l'arc (b, t) et au plus égal à 3 sur l'arc (s, b) .

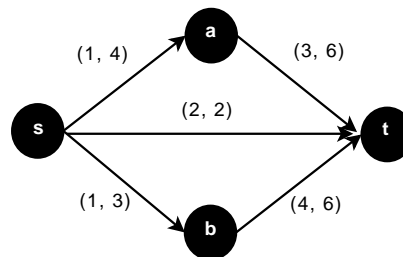


FIG. 2.4 – Exemple d'instance pour laquelle il n'existe pas de solution réalisable

Le théorème du flot compatible de Hoffman [14] annonce qu'une condition nécessaire et suffisante pour qu'il existe une solution réalisable dans un graphe est que, à tout sommet, les capacités des arcs sortants soit supérieure ou égale à celles des arcs entrants. On peut ramener tout problème vérifiant le théorème de Hoffman à un problème de flot maximum (cf [14] pour une démonstration détaillée).

2.5.4 Le problème de flot maximal de coût minimum

Dans cette variante du problème, on suppose qu'un coût c_e est associé à chaque unité de flot circulant sur l'arc e . Le coût d'un flot est alors le coût total des unités de flot circulant sur les arcs. Le

problème consiste donc à trouver un flot compatible maximal de coût minimum.

La recherche d'un flot maximal à coût minimal peut être résolue à l'aide de l'algorithme de Busacker-Gowen [6]. Cet algorithme détermine un flot de coût minimum de la même façon que l'algorithme de Ford et Fulkerson. On augmente à chaque phase le flot dans le réseau en utilisant une chaîne augmentante de coût minimum de s à t dans le graphe d'écart. Le coût d'une chaîne étant obtenu en additionnant les coûts des arcs qui sont dans le sens direct puis en soustrayant les coûts des arcs inverses. Le graphe d'écart est valué de la façon suivante :

- un arc (x, y) du graphe d'écart issu d'un arc (x, y) non saturé du réseau de transport est valué par le coût $c_{(x,y)}$ que possède cet arc,
- un arc (y, x) du graphe d'écart issu d'un arc (x, y) du réseau de transport portant un flux non nul est valué par l'opposé $-c_{(x,y)}$ du coût que possède cet arc.

Notons R le graphe valué ainsi obtenu. Rechercher dans le réseau d'origine une chaîne augmentante de coût minimum revient exactement à rechercher le plus court chemin de s à t dans R .

2.5.5 Le problème de transport

Ce problème consiste à transporter une certaine marchandise depuis des points (sources) d'un réseau à d'autres points (puits) d'un réseau. Le transport de marchandises à travers le réseau a un coût connu, nous cherchons donc à satisfaire la demande des puits en minimisant le coût total du transport.

Pour ce problème, le réseau de transport est représenté par un graphe bi-parti à deux étages : l'étage des sources et l'étage des puits. Il n'y a pas de point intermédiaire dans le réseau, c'est à dire chaque point est soit une source soit un puits. De plus, chaque source est reliée à chaque puits et il n'y a pas de chemin d'une source à une autre ni d'un puits à un autre. Ceci est illustré par la figure 2.5.

Dans la figure 2.5, nous représentons une instance du problème de transport. Dans cette instance, il y a deux sources (I et II) et trois puits (A, B et C). Les deux sources I et II ont respectivement 350 et 550 unités à envoyer. Les trois puits A, B et C demandent chacun 300 unités. Les coûts de transport des sources vers les puits sont représentés sur les arcs du graphe. Deux solutions différentes de ce problème sont représentées par les figures 2.5(b) et 2.5(c) ; sur chaque arc figure en premier le coût unitaire de transport, puis la valeur de flot.

L'algorithme le plus connu pour résoudre le problème de transport est celui de Stepping Stone [1]. Cet algorithme est un algorithme de programmation linéaire spécifiquement conçu pour les problèmes de transport. A partir d'une solution de base réalisable, l'algorithme interchange successivement les variables en base et hors base de façon à améliorer notre solution jusqu'à atteindre une solution optimale. Trois méthodes sont généralement conseillées dans la littérature pour générer les solutions de base réalisables [32] :

1. La méthode de la matrice minimale,
2. La méthode du coin nord-ouest,

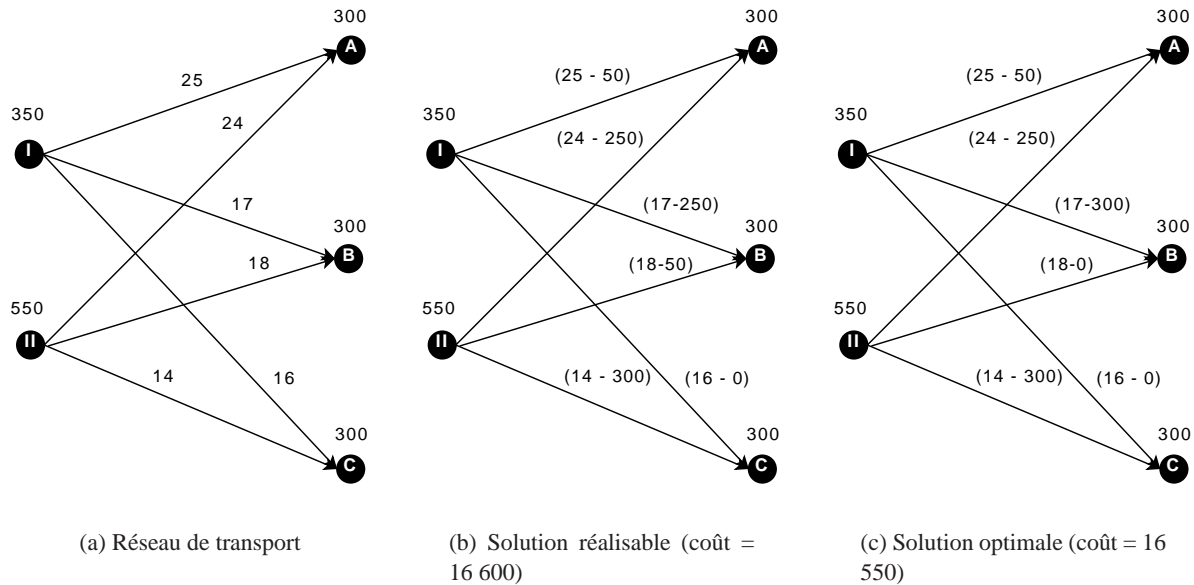


FIG. 2.5 – Exemples d’instances du problème de transport

3. La méthode de Vogel,

Nous invitons le lecteur à se référer à [1], pour un exposé plus complet sur les algorithmes de Stepping Stone.

2.6 Conclusion

Nous avons présenté dans ce chapitre un panorama sur les techniques d’optimisation. Nous avons particulièrement mis l’accent sur les techniques de PL et PLNE. Nous avons ensuite étudié quelques méthodes heuristiques et métaheuristiques classiquement utilisées pour la résolution des problèmes d’optimisation. A la fin de ce chapitre, nous avons fait le tour des problèmes de flots dans les graphes ainsi que les techniques de résolution de ces problèmes.

Nous proposons, dans la suite de ce document, d’étudier trois problèmes de planification de réseaux. Nous adapterons les techniques d’optimisation introduites dans ce chapitre pour résoudre ces problèmes.

CHAPITRE 3

Conception de réseaux de télécommunications

3.1 Introduction

Dans le chapitre 1, nous avons évoqué les problèmes de planification et de conception souvent rencontrés par les opérateurs. Dans ce chapitre, nous étudions en détails l'un des ces problèmes : le problème de conception de topologies de réseaux.

Nous proposons en premier lieu de définir le problème de conception de réseaux. Nous présentons ensuite les travaux de référence qui ont traité ce problème. Nous introduisons alors un modèle réaliste de conception de réseaux qui prend en considération les contraintes des équipements. Des méthodes de résolution sont présentées dans les sections 3.3, 3.4, 3.5 et 3.6. Nous évaluons enfin les résultats obtenus avec ces méthodes dans la partie 3.7 avant de conclure et de présenter quelques perspectives de ce travail dans la section 3.8.

3.1.1 Motivations

Ces dernières années ont été marquées par l'essor des nouvelles technologies de l'information et de la communication qui ont pénétré tous les secteurs d'activité : industriel, commercial, bancaire, éducatif. Phénomène de mode, ou réel besoin pour les entreprises, la conséquence immédiate a été la mise en place de nouvelles infrastructures de communication ou le remplacement de celles existantes afin de pouvoir répondre aux exigences des utilisateurs (communications intranet, extranet, internet,...). Partant de cet état de fait, l'élaboration ou la conception d'une topologie optimale nous est apparue un point important à étudier.

La conception des réseaux de télécommunications représente une tâche très complexe et, en règle générale, fort coûteuse étant donné l'importance des investissements nécessaires. En outre, l'équipe de

planification doit passer en revue les besoins déjà existants ou anticiper les besoins futurs, la multiplicité des contraintes topologiques, les coûts des différentes composantes des systèmes (routeurs, commutateurs, cartes réseau, ...), les contraintes imposées aux performances, la fiabilité, la capacité d'adaptation aux évolutions, le service de contrôle de la qualité, etc.

Par le passé, pour faire face aux nouvelles exigences de QoS et à l'augmentation soutenue du trafic, les opérateurs ou FAI se sont souvent tournés vers la mise en place de réseaux surdimensionnés. Les principales sources de dégradation de la QoS étant les zones de congestion du réseau, limiter les risques de congestion par une augmentation conséquente des ressources permettait d'écouler les volumes de trafic tout en garantissant la QoS requise. Cependant, cette démarche n'est pas viable économiquement. Etant donné l'importance des investissements nécessaires, des économies considérables peuvent être effectuées si la topologie du réseau est correctement conçue.

Notre objectif, dans ce chapitre, est donc d'étudier le problème de conception à moindre coût d'une infrastructure réseau respectant un certain nombre de contraintes opérationnelles définies par l'architecte réseau. Nous proposons des algorithmes performants permettant de générer des architectures de réseaux optimales répondant à ces différentes contraintes.

3.1.2 Le problème de conception de réseaux

Le problème de conception de réseaux, connu dans la littérature sous le nom de "Network Design", concerne la définition de la topologie d'un réseau permettant d'interconnecter un ensemble de sites/nœuds aux caractéristiques différentes. En général, cette topologie de réseau doit satisfaire à certains standards de fiabilité et de performance, et ce à moindre coûts. Le processus de conception comprend également les procédures de routage ainsi que le dimensionnement des lignes et nœuds.

En général, l'architecte réseau tient compte d'un nombre considérable de données qui lui permettent de proposer un modèle satisfaisant les conditions requises, notamment :

- des informations sur l'ensemble des positions des sites/nœuds à interconnecter (les entreprises clientes, les abonnés au service, les nœuds d'entrée au cœur du réseau, etc),
- des caractéristiques (le plus souvent estimées) des flux inter-sites/nœuds (volume, comportement dans le temps, etc).

Quant à l'aspect des composantes du réseau, le concepteur dispose d'une liste d'équipements possibles selon la nature, les caractéristiques et les coûts du réseau en question. Nous désignons par le terme équipements les dispositifs à installer dans le réseau pour acheminer le trafic, notamment les routeurs dans le cas d'un réseau basé sur un système datagramme (c'est le cas de la technologie IP), les commutateurs dans les réseaux X25, Frame Relay, ATM ..., les cartes réseau, les câbles entre les nœuds du réseau, etc.

La nature technique du réseau examiné conduit à des procédures spécifiques de routage qui devraient être prises en considération pour assurer des solutions efficaces, voire, dans la mesure du pos-

sible, optimales.

Il peut y avoir d'autres données complémentaires, notamment quels sont les sites les plus aptes, ou les moins adaptés pour installer des concentrateurs, quelles sont les caractéristiques particulières ou les restrictions de sécurité applicables à certains flux, etc.

Sur la base de cet ensemble de données, le processus de conception des réseaux doit spécifier la topologie du réseau ainsi que les caractéristiques des différents nœuds et connexions, le routage des trafics, etc.

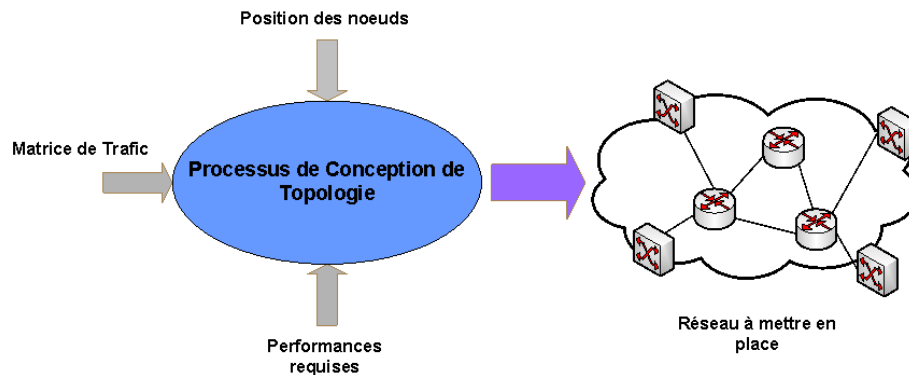


FIG. 3.1 – Processus de conception de réseaux

3.1.3 Etat de l'art

L'étude des problèmes de conception de topologies pour les réseaux de télécommunications remonte aux années 60 avec l'introduction des problèmes 1-median et p-median par Hakimi [7, 8]. Plusieurs modèles ont vu le jour depuis. Ces modèles se différencient par la façon dont ils évaluent le réseau obtenu.

Les premiers travaux dans ce domaine supposent un coût total du réseau dépendant uniquement des distances euclidiennes des liens reliant les nœuds du réseau [20, 22, 18]. Les approches qui ont été proposées dans ce sens ne tiennent pas compte du problème du dimensionnement et des coûts des équipements dans le processus de conception de topologie. De ce fait, la conception d'un réseau est subdivisée en deux sous-problèmes différents. Un bon exemple d'une démarche possible de décomposition du processus de conception est présenté ci après :

1. Conception de la topologie du réseau. On part de connaissances spécifiques sur les coûts de mise en place de lignes entre différents sites du réseau. Ces coûts sont souvent indépendants du type de ligne qui sera effectivement mis en place, puisqu'ils modélisent les coûts fixes (les coûts de

creuser des tranchées en cas d'installation de fibre optique, de mettre en service la fibre, etc). Une bonne partie du budget total de mise en place du réseau est affectée à cette phase.

2. Dimensionnement des lignes appelées à relier les différents nœuds du réseau, et des équipements à mettre en place dans ces nœuds.

Cette approche a longtemps été considérée comme acceptable dans la mesure où le coût des équipements était très inférieur au coût des liens. Néanmoins, de nos jours avec le déploiement massif des technologies optiques, le coût des liens devient de moins en moins cher voire comparable au coût des équipements, d'où la nécessité d'intégrer le problème de dimensionnement dans le processus de conception de topologie.

Pour faire face à ce problème, plusieurs travaux ont proposé des modèles qui prennent en compte la charge des liens [26, 44, 37, 38, 29, 30, 35]. Ces modèles supposent que les liens du réseau ont des capacités modulaires et que le coût de chaque lien sera d'autant plus fort que la distance (euclidienne) entre les extrémités sera grande et que la capacité de ce lien sera grande.

Récemment, Frangioni et Gendron ont proposé dans [96] une formulation 0-1 du problème de conception des réseaux avec des capacités modulaires des liens.

3.1.4 Contributions

La majorité des approches proposées jusqu'à présent ne tiennent pas compte des contraintes matérielles portant sur les équipements utilisés comme le modèle exact de routeur à installer dans un nœud ou le nombre de slots disponibles sur un routeur. Il n'existe donc aucune garantie quant à la mise en place effective d'une solution au problème de conception de topologie.

Partant de ce constat, notre contribution au problème de conception de réseaux est double :

- **Coût des équipements** : nous proposons une nouvelle définition du problème de conception de réseaux qui tient compte des coûts réels des équipements qui seront choisis. Ainsi, grâce à une définition fine des coûts, nous proposons une conception optimisée avec un coût réaliste très proche de ce que l'opérateur aura à investir pour mettre en place son réseau.
- **Contraintes matérielles** : comme notre approche tient compte des caractéristiques des routeurs à installer dans les nœuds et des cartes réseau permettant la connexion des liens sur les routeurs, elle permet d'intégrer complètement les contraintes des équipements : nombre de slots sur un routeur, nombre de ports sur une carte, capacité de traitement d'un routeur. De part l'intégration de ces contraintes, les solutions que nous proposons sont directement configurables et ceci est un aspect essentiel pour les opérateurs réseaux.

3.2 Une nouvelle approche de conception de réseaux

Nous proposons dans ce chapitre de poser le problème de conception de réseaux d'une manière nouvelle en y intégrant les contraintes des équipements. Ce nouveau problème sera défini dans cette partie.

3.2.1 Description du modèle

Un réseau de télécommunications est composé de nœuds et de liens d'interconnexion. Dans la pratique, les opérateurs distinguent deux types de nœuds (cf. figure 3.2) :

1. **Les nœuds terminaux** : ces nœuds se comportent comme des clients du réseau. Ce sont ces nœuds qui génèrent les demandes à acheminer à travers le réseau (exemple : les routeurs via lesquels les entreprises clientes accèdent au réseau de l'opérateur).
2. **Les nœuds de transit** : comme leur nom l'indique, ces nœuds sont utilisés comme nœuds de transit pour relier les nœuds terminaux. Un nœud de transit ne génère pas de trafic. Un tel nœud est optionnel et peut ne pas être utilisé dans le réseau.

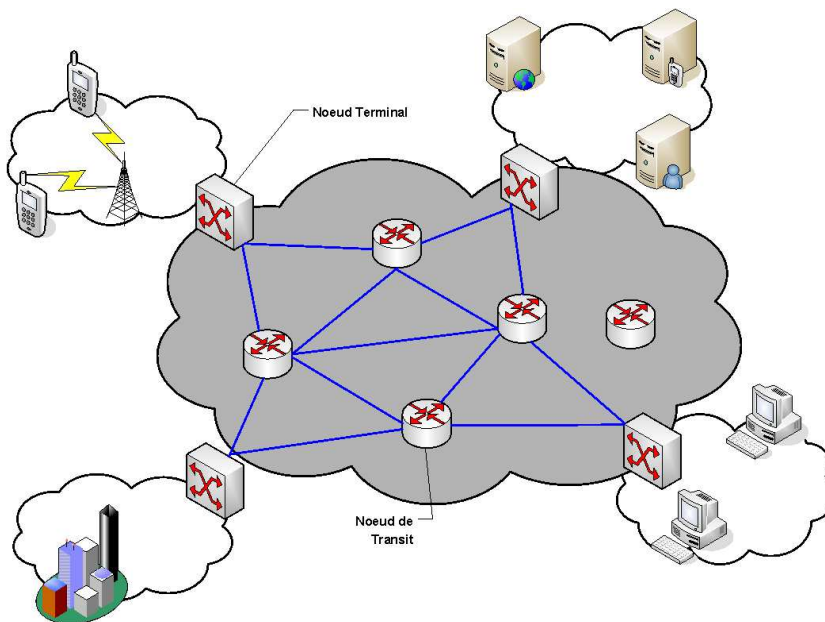


FIG. 3.2 – Les types de nœuds dans un réseau

Modèle	Capacité en Kbps
E1	1920
E2	7680
E3	30720
OC-1	50725
OC-3	152174
OC-12	608698
OC-48	2434793
OC-192	10^7

TAB. 3.1 – Exemples de modèles de liens.

Chaque lien du réseau a un modèle qui caractérise sa bande passante. Le tableau 3.1 décrit quelques modèles de liens disponibles sur le marché.

Chaque nœud du réseau contient des équipements permettant de concentrer le trafic venant des autres nœuds. Ces équipements correspondent à des routeurs et à des cartes que l'on place sur ces routeurs. Un modèle de routeur est caractérisé par :

- les cartes compatibles,
- le nombre de slots disponibles pour accueillir des cartes,
- le débit maximal que peut supporter le bus interne du routeur,
- le coût d'achat du châssis.

De même, un modèle de carte est caractérisé par :

- la technologie de la carte,
- le nombre de ports de communication disponibles,
- le débit crête de chaque port,
- le coût d'achat de la carte.

Nous étudions dans ce chapitre le problème réaliste de conception de réseaux suivant :
Etant donné :

1. les positions des nœuds terminaux,
2. les positions des nœuds de transit,
3. la liste des équipements possibles,
4. une estimation des demandes en trafic entre les nœuds terminaux,

L'objectif est de définir la topologie d'un réseau permettant d'interconnecter l'ensemble des nœuds terminaux à un coût minimal. Le coût global d'une solution est la somme des coûts d'acquisition et

d'installation de tous les liens et des équipements (routeurs et cartes) requis pour établir le réseau.

Le réseau résultant doit également satisfaire les critères de performances définis par l'opérateur. Nous considérons dans notre modèle les deux critères de performances suivant, qui sont classiquement utilisés par les opérateurs :

Résilience aux pannes

Le réseau construit doit être résilient, c'est à dire qu'il garantit, même en cas de panne d'un équipement, qu'il existe toujours un chemin entre deux nœuds terminaux et qu'aucune panne n'engendrera la congestion du réseau. Ceci implique des contraintes sur la connexité du graphe du réseau.

Limitation du délai de bout-en-bout

Le réseau doit garantir que le délai de transmission des paquets entre les nœuds terminaux est limité, même en cas de défaillance d'un nœud ou d'un lien. Dans la pratique, le concepteur du réseau doit assurer que les longueurs des chemins entre les nœuds terminaux sont délimitées. Il doit également garantir que la défaillance d'un nœud ou d'un lien n'augmente pas la distance entre deux nœuds terminaux au delà d'un seuil donné [51].

3.2.2 Formulation mathématique du modèle

Contraintes topologiques

Le réseau à concevoir est représenté par un graphe non orienté $G = (V, E)$. Les nœuds du graphe correspondent aux routeurs et les arcs correspondent à tous les liens possibles entre les routeurs. Soit $A \subset V$ l'ensemble des nœuds terminaux.

Pour chaque lien potentiel $e \in E$, nous définissons la variable de décision suivante :

$$x_e = \begin{cases} 1 & \text{si le lien } e \text{ est utilisé dans la solution finale} \\ 0 & \text{sinon} \end{cases}$$

Une solution de notre problème peut alors être représentée par le vecteur $\mathbf{x} = (x_e)_{e \in E}$.

On peut représenter une solution par un sous-graphe $G_{\mathbf{x}} = (V_{\mathbf{x}}, E_{\mathbf{x}})$ du graphe G , tel que $E_{\mathbf{x}} = \{e \in E : x_e = 1\}$, et $V_{\mathbf{x}}$ est l'ensemble des nœuds extrémités des arrêtes $e \in E_{\mathbf{x}}$.

Contraintes de connexité : Une solution admissible doit satisfaire les contraintes de résiliences aux pannes. Pour ce faire, une solution admissible doit vérifier les deux contraintes suivantes de connexité dans le graphe du réseau [51] :

$$\sum_{u \in W} \sum_{v \in V-W} x_{uv} \geq 2 \quad W \subset V, W \cap A \neq \emptyset, A \quad (3.1)$$

$$\sum_{u \in W} \sum_{v \in V-z-W} x_{uv} \geq 1 \quad z \in V, W \subset V-z, W \cap A \neq \emptyset, A \quad (3.2)$$

Il est facile de démontrer, en utilisant le théorème de Menger [99], que pour toute solution vérifiant les inégalités (3.1) la suppression d'un lien conserve la connexité du graphe du réseau. Si cette même solution satisfait, en plus, les inégalités (3.2), alors tous les nœuds de l'ensemble A sont situés dans la même composante biconnexe. Cette dernière propriété garantit que même en cas de défaillance d'un nœud ou d'un lien, il existe toujours un chemin entre chaque paire de nœuds terminaux.

Contraintes de délai : Une solution admissible doit également satisfaire la contrainte de limitation des délais de bout-en-bout. Plus précisément, les longueurs des chemins entre les nœuds terminaux doivent être bornées par une constante K_n dans le réseau nominal et par la constante $K_f \geq K_n$ en cas de pannes.

Sur la base de l'idée introduite dans [51], la contrainte de délai peut être exprimée par une formulation linéaire, et ce comme suit :

Notons par Π_{uv} (respectivement Π_{uv}^z) l'ensemble des chemins entre les nœuds u et v dans G (respectivement $G-z$) dont la longueur est inférieure ou égale à K_f (respectivement K_n). Notons aussi par y_{uv}^π la variable de décision tel que :

$$y_{uv}^\pi = \begin{cases} 1 & \text{si tous les liens du chemin } \pi \text{ appartiennent à } E_x \\ 0 & \text{sinon} \end{cases}$$

pour chaque chemin π entre u et v . La contrainte de délai peut alors être formulée de la manière suivante :

$$y_{uv}^\pi \leq x_e \quad e \in \pi, \pi \in \Pi_{uv}, u, v \in A \quad (3.3)$$

$$\sum_{\pi \in \Pi_{uv}} y_{uv}^\pi \geq 1 \quad u, v \in A \quad (3.4)$$

$$\sum_{\pi \in \Pi_{uv}^z} y_{uv}^\pi \geq 1 \quad z \in V, z \neq u, v, u, v \in A \quad (3.5)$$

Les inégalités (3.3) assurent qu'un chemin n'est présent dans la solution finale que si tous ses liens y sont présents. Les inégalités (3.4) indiquent que la solution doit inclure au moins un chemin de Π_{uv} pour n'importe quel couple d'origine-destination $u, v \in A$. Enfin, les inégalités (3.5) indiquent que la solution doit contenir au moins un chemin de Π_{uv}^z pour n'importe quel couple d'origine-destination

$u, v \in A$, et quel que soit le nœud z en panne.

En résumé, une solution admissible est un vecteur $\mathbf{x} = (x_e)_{e \in E}$ de variables binaires qui satisfont les contraintes de connexité (3.1) et (3.2). Cette solution doit également vérifier que pour chaque couple d'origine-destination $u, v \in A$, il existe une variable binaire y_{uv}^π qui vérifie les contraintes (3.3)-(3.5). Dans la suite, nous noterons par X l'ensemble des solutions réalisables. Nous noterons aussi par $\Pi_{uv}(\mathbf{x})$ l'ensemble des chemins π entre les nœuds u et v tels que $y_{uv}^\pi = 1$.

Coût d'une solution

Comme nous avons pu l'introduire rapidement dans le début de ce chapitre, une de nos contributions principales porte sur l'aspect économique du modèle que nous proposons.

Le coût d'une solution $\mathbf{x} \in X$ est composé des coûts suivants :

- le coût des liens du réseaux,
- le coût des routeurs installés dans les nœuds du réseau,
- le coût des cartes installées sur les routeurs.

Charge des liens : Les opérateurs se tournent aujourd'hui, pour la plupart, vers des fournisseurs de bande passante qui leur permettent de mettre en place des liens de capacité donnée. Nous proposons une formulation du coût d'un lien qui respecte pour beaucoup la réalité (aux ristournes et offres promotionnelles près).

Soit d_{uv} la demande entre les nœuds $u, v \in A$. Nous supposons dans notre modèle que les demandes seront routées le long des plus courts chemins (PCC). Si plusieurs chemins sont équivalents d'un point de vue distance, le partage de charge entre ces chemins est employé. Nous supposons que la longueur d'un chemin est le nombre de liens qui forment ce chemin.

Soit $\bar{\Pi}_{uv}(\mathbf{x})$ l'ensemble des PCC entre u et v dans le graphe G . Considérons pour chaque arc $(i, j) \in E$ et pour chaque nœud $v \in A$ la variable de décision suivante :

$$\delta_{ij}^v = \begin{cases} 1 & \text{s'il existe } \pi \in \bar{\Pi}_{iv}(\mathbf{x}) \text{ tel que } (i, j) \in \pi \\ 0 & \text{sinon} \end{cases}$$

Les variables δ_{ij}^v indiquent si l'arc (i, j) est sur l'un des PCC entre les nœuds i et v .

On notera :

- $n_i^v = \sum_j \delta_{ij}^v$: le nombre de PCC entre i et v ,

– $\gamma_i^v(\mathbf{x})$: le trafic reçu au nœud i , direct et en transit, pour la destination v . On a alors :

$$\gamma_i^v(\mathbf{x}) = d_{iv} + \sum_{j \neq v} \frac{\delta_{ji}^v}{n_j^v} \gamma_j^v$$

La charge du lien (i, j) s'écrit alors :

$$y_{i,j}(\mathbf{x}) = \sum_{v \in A} \frac{\delta_{ij}^v}{n_i^v} \gamma_i^v \quad (3.6)$$

Coût des liens : On suppose, dans notre modèle, que le coût du lien (i, j) est une fonction croissante F_{ij} de sa capacité. Cette capacité est choisie parmi un ensemble de capacités modulaires. Soit T le nombre de modèles de liens et r_t la capacité du modèle $t = 1, \dots, T$ (triés dans l'ordre croissant des capacités).

Considérons la variable binaire suivante :

$$\beta_{ij}^t(\mathbf{x}) = \begin{cases} 1 & \text{si } t \text{ est la valeur minimale tel que } r_t \geq \max(y_{i,j}(\mathbf{x}), y_{j,i}(\mathbf{x})) \\ 0 & \text{sinon} \end{cases}$$

Le coût du lien (i, j) s'écrit alors :

$$F_{ij} \left(\sum_{t=1}^T \beta_{ij}^t(\mathbf{x}) r_t \right) \quad (3.7)$$

Coût des équipements : Si le nœud v est utilisé dans la solution $\mathbf{x} \in X$, alors il faut y installer un routeur permettant de supporter le trafic en émission et réception. Chaque lien du réseau est connecté à ses deux routeurs extrémités par une carte offrant le même débit par port que le modèle choisi pour le liens. La figure 3.3 illustre la manière dont les liens sont branchés sur les routeurs à l'aide des cartes adaptées.

Soit T le nombre de modèles de carte utilisables. On suppose que pour chaque modèle de lien on dispose d'un et un seul modèle de carte qui le supporte. Chaque modèle de carte $t = 1, \dots, T$ est caractérisé par :

- son nombre de ports de communication disponibles p_t ,
- son coût d'achat ϕ_t .

La configuration de cartes sur un routeur v , c'est à dire le nombre de cartes de chaque modèle, est une variable qui dépend du nombre de liens reliés au routeur et de leurs capacités. On définit cette variable comme un vecteur $\mathbf{s} = [s^1, s^2, \dots, s^T]$ de dimension T où s^t représente le nombre de cartes

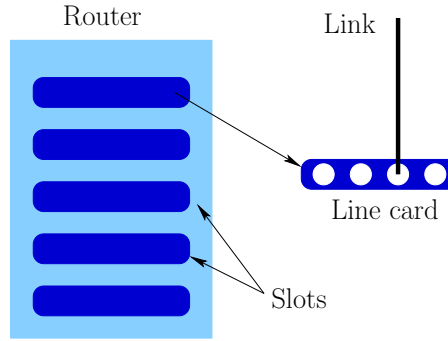


FIG. 3.3 – Branchement des liens sur les routeurs à l'aide de cartes

du modèle t à placer. Etant donné une solution \mathbf{x} et un nœud v utilisé par cette solution, il est facile de vérifier que les valeurs des s^t relatifs au nœud v se calculent de la manière suivante :

$$s^t(\mathbf{x}, v) = \left\lceil \frac{\sum_{i \in V} \beta_{iv}^t(\mathbf{x})}{p_t} \right\rceil$$

où, pour toute valeur réelle z , $\lceil z \rceil$ représente le plus petit entier n tel que $z \leq n$.

Un routeur sur lequel une configuration de cartes sera branchée doit être choisi parmi R modèles de routeurs. Rappelons que chaque modèle de routeur $r = 1, \dots, R$ est caractérisé par son nombre de slots \bar{s}_r , son coût d'achat et la liste des modèles de cartes qu'il supporte.

Soit \mathbf{x} une solution de notre problème et v un nœud utilisé par cette solution. Soit $\mathbf{s}(\mathbf{x}, v)$ la configuration de cartes du nœud v . Calculer le coût des équipements à installer sur le nœud v revient donc à calculer le coût de la configuration $\mathbf{s}(\mathbf{x}, v)$. Ce coût est décomposé en deux parties.

1. le coût des cartes qui composent la configuration \mathbf{s}_v , c'est à dire $\sum_{t=1}^T s_v^t(\mathbf{x}) \cdot \phi_t$
2. le coût minimal du routeur pouvant accueillir ces cartes,

Soit $\bar{h} = \max_{r=1, \dots, R} \bar{s}_r$. L'ensemble Λ de toutes les configurations de cartes possibles est donné par :

$$\Lambda = \left\{ \mathbf{s} = (s_1, \dots, s_T) : \sum_t s_t \leq \bar{h} \right\}$$

Considérons la fonction $\mu^* : \Lambda \rightarrow \mathbb{R}$ qui à chaque configuration $\mathbf{s} \in \Lambda$ associe son coût. Le coût des équipements d'une solutions \mathbf{x} est alors la somme, sur tous les nœuds v , des valeurs retournées par la

fonction $\mu^*(\mathbf{s}(\mathbf{x}, v))$. Pour construire la fonction μ^* , il est possible d'utiliser la technique de programmation dynamique. L'idée de l'algorithme est la suivante :

On définit la relation d'ordre " $<$ " tels que : pour chaque paire de configuration \mathbf{s} et $\mathbf{s}' \in \Lambda$, $\mathbf{s} < \mathbf{s}'$ si et seulement si $s^t < s'^t, t = 1, \dots, T$. Pour construire μ^* , il faut d'abord trier selon un ordre croissant toutes les configurations $\mathbf{s} \in \Lambda$. On obtient donc les configurations $\mathbf{s}^0 < \mathbf{s}^1 < \dots < \mathbf{s}^{k-1} < \mathbf{s}^k < \dots < \mathbf{s}^N$, où \mathbf{s}^k est la configuration d'ordre k . Le calcul du coût de la configuration \mathbf{s}^k , nécessite le calcul du coût du routeur pouvant accueillir cette configuration. Pour déterminer le modèle de routeur r_k qui va accueillir la configuration \mathbf{s}^k , deux cas peuvent se présenter :

1. Si le modèle de routeur r_{k-1} , utilisé pour la configuration \mathbf{s}^{k-1} , peut accueillir les cartes de \mathbf{s}^k alors $r_k = r_{k-1}$.
2. Sinon on choisit le modèle de routeur de coût minimal qui supporte \mathbf{s}^k .

On définit ainsi une relation entre deux configurations successives \mathbf{s}^{k-1} et $\mathbf{s}^k \in \Lambda$.

3.2.3 Problème à résoudre

Le coût total d'une solution $\Gamma(\mathbf{x})$ peut donc être calculé à partir de la somme des coûts des routeurs, des cartes et des liens. Le problème à résoudre peut maintenant être énoncé comme suit,

$$\min_{\mathbf{x} \in X} \Gamma(\mathbf{x}) = \sum_{e \in E} x_e F_e \left(\sum_{t=1}^T \beta_e^t(\mathbf{x}) r_t \right) + \sum_{v \in V_{\mathbf{x}}} \mu^*(\mathbf{s}(\mathbf{x}, v)) \quad (3.8)$$

3.2.4 Extension du modèle

Pour faciliter la présentation, il a été supposé que les charges des liens sont calculées seulement pour le réseau nominal. Il arrive souvent que l'architecte réseau prenne en compte, en plus de l'état nominal, les états où une défaillance d'un lien ou d'un nœud se produit. Cette contrainte peut être facilement intégrée dans notre modèle en l'incorporant dans la définition de la charge des liens. Nous considérons donc que la charge $y_{i,j}(\mathbf{x})$ associée à un lien (i, j) est le maximale sur tous les états possibles du réseau. En divisant cette valeur par un facteur U (par exemple 70%), nous pourrions garantir que chaque lien est dimensionné de telle manière que son utilisation ne dépasse jamais U quel que soit l'état du réseau.

3.3 Résolution exacte par des techniques de "Branch-and-Bound"

Pour résoudre notre problème de conception de réseaux, nous proposons d'utiliser une approche à base de la technique Branch-and-Bound (BB). Dans cette section, nous décrivons comment nous avons adapté cette technique pour résoudre notre problème. Le pseudo-code de la méthode proposée est décrit

algorithm 2 Algorithme exact à base de BB

```

1: procédure BB( $\mathbf{x}, l, ub$ )
2: if  $\mathbf{x} \notin X$  or  $l = |E| + 1$  then
3:   return  $\infty$ 
4: end if
5: if  $lb(\mathbf{x}) \geq ub$  then
6:   return  $\Gamma(\mathbf{x})$ 
7: end if
8: if  $\Gamma(\mathbf{x}) < ub$  then
9:    $ub = \Gamma(\mathbf{x})$ 
10: end if
11:  $\mathbf{x}^{right} = \mathbf{x}; x_l^{right} = 0$ 
12:  $rightCost = BB(\mathbf{x}^{right}, l + 1, ub)$ 
13:  $\mathbf{x}^{left} = \mathbf{x}$ 
14:  $leftCost = BB(\mathbf{x}^{left}, l + 1, ub)$ 
15: return  $\min(\Gamma(\mathbf{x}), leftCost, rightCost)$ 

```

dans l’algorithme 2.

Notre algorithme 2 est récursif et travaille sur un arbre binaire de recherche où les nœuds représentent des solutions partielles. Le nœud racine de l’arbre est la solution $\mathbf{x} = (1)_{e \in E}$ (le réseau entièrement maillé). L’algorithme prend comme entrée :

- la solution partielle courante \mathbf{x} ,
- le prochain lien potentiel l qui sera examiné,
- une borne supérieure ub sur le coût optimal.

Il explore alors le sous-arbre binaire de recherche de racine \mathbf{x} . La valeur retournée par cet algorithme est le coût de la solution optimale.

3.3.1 Branchement

L’opération de branchement consiste à diviser l’arbre de recherche associé au nœud parent \mathbf{x} en deux descendants ou sous-arbres : un sous-arbre droit et un sous-arbre gauche. Le sous-arbre droit est enraciné au nœud \mathbf{x}^{right} où \mathbf{x}^{right} est la solution obtenue en supprimant le lien l de $E_{\mathbf{x}}$. Le sous-arbre gauche est enraciné au nœud \mathbf{x}^{left} où \mathbf{x}^{left} est juste \mathbf{x} , c’est à dire tous les liens de \mathbf{x} sont conservés dans \mathbf{x}^{left} .

3.3.2 Elagage

Un sous-arbre de recherche est élagué lorsqu’il ne peut pas conduire à une solution qui sera meilleure que la meilleure solution trouvée jusqu’à présent. Un arbre de recherche enraciné au nœud \mathbf{x} est élagué

si l'une des affirmations suivantes est vraie :

- la solution \mathbf{x} n'est pas admissible. (Il est clair que si une solution \mathbf{x} n'est pas admissible, toutes les solutions dans le sous-arbre ayant comme racine le nœud \mathbf{x} ne sont pas non plus admissibles).
- la borne inférieure sur le coût de \mathbf{x} est supérieure à l'actuelle borne supérieure.

3.3.3 Génération des bornes supérieures et inférieures

La borne supérieure est initialisée avec le coût retourné par un algorithme glouton, détaillé plus loin dans ce chapitre. Cette borne supérieure est mise à jour chaque fois qu'une meilleure solution est découverte.

En règle générale, la qualité des bornes inférieures et le temps nécessaire à leur calcul sont décisifs pour l'efficacité d'un algorithme de Branch & Bound. Dans notre algorithme, une borne inférieure sur le coût optimal est générée à chaque nœud de l'arbre de recherche comme décrit ci-dessous :

Considérons une solution admissible \mathbf{x} de l'arbre de recherche pour laquelle les valeurs de $x_e \forall e < l$ sont déjà décidées. Soit :

- $Subtree(\mathbf{x})$: l'ensemble des solutions réalisables dans la sous-arborescence enracinée au nœud \mathbf{x} ,
- $F = \{e < l : x_e = 1\}$: l'ensemble des liens déjà sélectionnés dans la solution \mathbf{x} ,
- $F' = \{e > l\}$: les liens non encore examinés,
- $G[F]$: le sous-graphe de $G_{\mathbf{x}}$ induit par les liens de F .

Il est clair que, pour toute solution $\mathbf{x}' \in Subtree(\mathbf{x})$, on a $F \subset E_{\mathbf{x}'}$. Il est évident aussi que si $G[F]$ ne satisfait pas les contraintes de connexité (3.1)-(3.2), alors, pour que \mathbf{x}' soit admissible certains liens de F' devront être sélectionnés, c'est à dire $F' \cap E_{\mathbf{x}'} \neq \emptyset$.

Il en résulte que la borne inférieure sur le coût d'une solution appartenant à $Subtree(\mathbf{x})$ peut être calculée à partir de la somme des éléments suivants :

1. une borne inférieure sur le coût des liens de l'ensemble F . Comme la solution \mathbf{x}' est obtenue en supprimant certains liens dans G , il s'ensuit que la charge d'un lien appartenant à F ne peut qu'augmenter par rapport à sa charge dans la solution \mathbf{x} . Une borne inférieure sur le coût des liens de F est ainsi égale au coût des liens de F en considérant leurs charges dans \mathbf{x} .
2. une borne inférieure sur le coût des liens de F' sélectionnés si $G[F]$ ne satisfait pas les contraintes de connexité (3.1)-(3.2). Le nombre minimum d'arêtes de F' à sélectionner peut facilement être calculé en utilisant l'algorithme proposé dans [87]. Une borne inférieure sur le coût de ces liens peut ainsi être obtenue en prenant les n liens les moins chers de F' tel que a) les nœuds extrémités de ces liens n'appartiennent pas à la même composante biconnexe dans le graphe $G_{\mathbf{x}}$, et b) aucun de ces nœuds n'est un point d'articulation dans $G_{\mathbf{x}}$. Le coût de ces liens est calculé en utilisant leurs charges dans \mathbf{x} .

3. une borne inférieure sur le coût des cartes et des routeurs à installer dans la solution \mathbf{x}' . Une telle borne inférieure est obtenue en considérant les liens de F , avec les capacités qu'ils ont dans la solution \mathbf{x} .

3.3.4 Ordre de Branchement

La rapidité de notre algorithme dépend grandement de l'ordre dans lequel les sous-arbres sont considérés. Un ordre de branchement efficace permet d'améliorer rapidement la borne supérieure et d'élaguer le maximum de sous-arbres de recherche.

Comme le coût des solution partielles \mathbf{x} et \mathbf{x}^{right} sont différents, un ordre naturel de branchement est de considérer le sous-arbre dont la racine est \mathbf{x}^{right} avant celui enraciné au nœud \mathbf{x}^{left} .

3.3.5 Optimisation du calcul des charges des liens

Etant donné une solution \mathbf{x} dans l'arbre de recherche, le fils droit \mathbf{x}^{right} correspond à la solution obtenue en supprimant un lien l de $E_{\mathbf{x}}$. Evaluer le coût de \mathbf{x}^{right} suppose de déterminer la charge de chaque lien de $E_{\mathbf{x}^{right}}$. Pour ce faire, on calcule les PCC entre les nœuds terminaux, puis on propage le trafic le long de ces chemins.

La suppression d'un lien l de $E_{\mathbf{x}}$ n'impacte souvent qu'un petit nombre de PCC. Des algorithmes, dits de plus courts chemins dynamiques, permettent de traiter ce type de problèmes plus efficacement que les algorithmes de Dijkstra ou de Bellman-Ford. Nous avons utilisé une amélioration de l'algorithme de Ramalingam et Reps [39], décrite en détail dans [68].

Pour accélérer la procédure de propagation de trafic, nous avons utilisé la même technique que Fortz et Thorup dans [56] connue sous le nom de propagation dynamique des trafics.

3.3.6 Temps de Calcul

Nous avons testé cet algorithme dans notre outil de simulation NEST [108] dans le but principal d'en estimer le temps de calcul. Nous savons que les temps de calcul de ces techniques ne supportent pas la mise à l'échelle. Les résultats sont représentés dans la figure 3.4. On peut voir que les temps de calcul, requis pour trouver les solutions optimales, augmentent exponentiellement avec la taille des instances. Nous prévoyons que les temps de calcul se situeront au-delà des valeurs tolérables pour des réseaux de grandes tailles.

Nous gardons donc cet algorithme comme un outil nous donnant la solution exacte. Mais nous sommes obligés de proposer des heuristiques adaptées pour permettre des temps de calcul raisonnables.

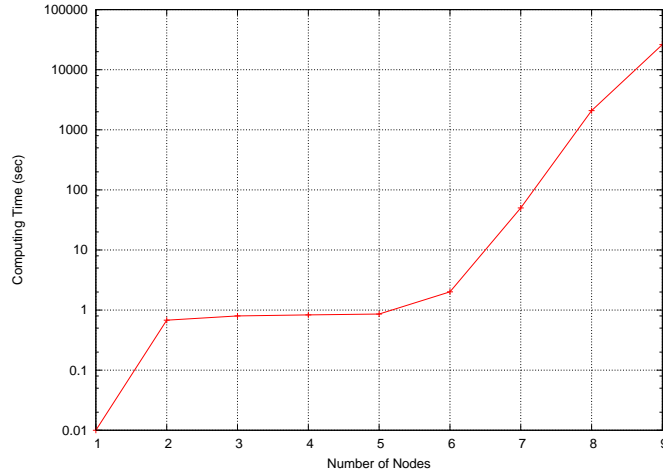


FIG. 3.4 – Temps de calcul requis par l’algorithme de BB

3.4 Algorithme glouton

Cet algorithme a été initialement introduit par Fortz dans [51]. Il appartient à la classe des approches qui n’intègrent pas la charge des liens dans le processus de conception de réseaux.

Partant d’une solution initiale (par exemple un réseau complètement maillé), à chaque itération, l’algorithme supprime le lien le plus long, tout en préservant la faisabilité de la solution. Le pseudo-code est décrit dans l’Algorithme 3.

algorithm 3 Algorithme glouton

Require: an initial solution \mathbf{x}_0

- 1: $\mathbf{x}_i \leftarrow \mathbf{x}_0$
 - 2: sort edges of $E_{\mathbf{x}_0}$ in increasing order of length
such that $E_{\mathbf{x}_0} = \{e(1), e(2), \dots, e(m)\}$, $length(i) \leq length(i) \forall i < k$
 - 3: **for** $k \leftarrow m$ to 1 **do**
 - 4: **if** $\mathbf{x}_i - e(k)$ is feasible **then**
 - 5: $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i - e(k)$
 - 6: **end if**
 - 7: **end for**
-

3.5 Heuristique basée sur la recherche tabou

Comme nous l’avons signalé dans le chapitre 2, la technique de recherche tabou (TS) est une technique bien connue pour résoudre les problèmes d’optimisation combinatoire difficiles. TS a prouvé son

efficacité dans de nombreux problèmes d'optimisation combinatoire, et en particulier pour la conception des réseaux résilient aux pannes [51]. Dans la suite, nous décrivons comment nous avons adapté les idées de TS à notre problème de conception des réseaux.

Une description complète de notre heuristique de TS est donnée dans l'algorithme 4. Nous décrivons ci-dessous les détails de cet algorithme.

algorithm 4 Algorithme de TS

Require: an initial feasible solution \mathbf{x}_0

```

1:  $\mathbf{x}_i = \mathbf{x}_0$ ;  $\mathbf{x}^* = \mathbf{x}_0$ ;  $tabuList = \emptyset$ 
2: while the stopping criterion is not satisfied do
3:   select the best solution  $x \in \mathcal{N}(\mathbf{x}_i)$  such that  $x$  is not tabu or  $x$  satisfies the aspiration criterion
4:   if  $x$  does not improve  $\mathbf{x}_i$  then
5:     select the best solution  $x' \in \mathcal{N}'(\mathbf{x}_i)$  such that  $x'$  is not tabu or  $x'$  satisfies the aspiration
       criterion
6:   end if
7:    $\mathbf{x}_{i+1} = \min(x, x')$ 
8:   update  $tabuList$ 
9:   if  $\Gamma(\mathbf{x}_{i+1}) < \Gamma(\mathbf{x}^*)$  then
10:     $\mathbf{x}^* = \mathbf{x}_{i+1}$ 
11:   end if
12: end while

```

3.5.1 Initialisation

Notre algorithme TS part d'une solution initiale admissible \mathbf{x}_0 et génère une séquence $\mathbf{x}_0, \mathbf{x}_1, \dots$ de solutions potentielles. La solution initiale \mathbf{x}_0 peut être obtenue, par exemple, avec l'algorithme glouton ci-dessus.

3.5.2 Espace de recherche et structure du voisinage

Les deux aspects importants à définir pour un algorithme de TS sont le voisinage d'une solution et la prise de décision suite à l'exploration de celui-ci.

Considérons, à l'itération i , une solution admissible et "minimale" \mathbf{x}_i . Nous entendons par le terme "minimale" qu'en supprimant n'importe quel lien, la solution \mathbf{x}_i devient non admissible ou on augmente son coût. Si une meilleure solution minimale \mathbf{x}_{i+1} existe, il est facile de voir que \mathbf{x}_{i+1} contient au moins un lien qui n'est pas dans \mathbf{x}_i .

Sur la base de cette observation, nous définissons le voisinage $\mathcal{N}(\mathbf{x}_i)$ de la solution \mathbf{x}_i de la manière suivante : $\mathcal{N}(\mathbf{x}_i)$ contient toutes les solutions qui peuvent être obtenues à partir de \mathbf{x}_i en appliquant les

transformations suivantes sur le graphe $G_{\mathbf{x}_i}$:

1. ajouter un arc reliant deux nœuds non isolés,
2. puis lancer l'algorithme glouton sur le graphe résultant.

3.5.3 Diversification

Comme nous l'avons indiqué dans le chapitre 2, une stratégie de diversification est conçue pour conduire la recherche dans de nouvelles régions et générer des solutions qui diffèrent de celles observées auparavant. La stratégie de diversification est particulièrement utile lorsque la recherche est piégée dans un minimum local.

La notion de diversification peut facilement être adaptée à notre problème. Soit \mathbf{x}_i la meilleure solution obtenue à l'itération i . Puisque un nœud de transit n'est généralement utilisé que s'il réduit le coût total du réseau, nous définissons un autre voisinage $\mathcal{N}'(\mathbf{x}_i)$ de la solution \mathbf{x}_i . $\mathcal{N}'(\mathbf{x}_i)$ contient toutes les solutions qui peuvent être obtenues à partir de \mathbf{x}_i en appliquant les transformations suivantes sur le graphe $G_{\mathbf{x}_i}$:

1. connecter un nœud de transit, non utilisé dans le graphe $G_{\mathbf{x}_i}$, aux autres nœuds non isolés,
2. puis lancer l'algorithme glouton sur le graphe résultant.

Nous adoptons, ainsi, la politique de recherche suivante : si, à l'itération i , aucune solution améliorante de \mathbf{x}_i n'est trouvée dans $\mathcal{N}(\mathbf{x}_i)$, l'algorithme diversifie la recherche en explorant les solutions de $\mathcal{N}'(\mathbf{x}_i)$.

3.5.4 Liste Tabou

Pour éviter les cycles pendant la recherche, nous avons utilisé une mémoire (une liste tabou T) pour stocker les mouvements qui pourraient conduire à des solutions récemment visitées. Nous stockons dans cette structure les arcs ajoutés à chaque itération. Nous donnons, ainsi, un statut "Tabou" pour ces arcs. Lors de l'exploration d'un voisinage, on interdit la suppression des arcs qui sont déjà déclarés tabou pendant un certain nombre d'itérations. Un arc déclaré tabou ne peut être supprimé que si la solution résultante améliore le coût du réseau. Ce dernier mécanisme est connu dans la littérature sous le nom d'"aspiration".

3.6 Heuristique basée sur la méthode LDS

Les méthodes de recherche complètes (basées sur des parcours arborescents) sont souvent utilisées pour obtenir des solutions optimales. Mais, du fait de leur comportement (souvent) exponentiel, ces méthodes peuvent s'avérer trop gourmandes en temps de calcul.

Grâce à leur façon opportuniste d'explorer l'espace de recherche, les méthodes approchées, basées sur les recherches locales (comme la recherche tabou), sont supposées produire de bonnes solutions dans des temps de calcul raisonnables. Malheureusement, de telles méthodes ne sont pas toujours capables de se sortir facilement d'optima locaux et peuvent perdre beaucoup de temps dans l'exploration de voisinages inintéressants.

Les algorithmes hybrides [58] offrent un réel compromis entre ces deux types d'approches. Plus précisément, ils sont capables de combiner efficacement les avantages des méthodes de recherche complète, avec l'exploration opportuniste des méthodes de recherche locale.

Nous avons adapté ce principe pour résoudre notre problème en utilisant un algorithme à base d'une hybridation TS/LDS. Le principe est d'explorer les solutions pouvant être à une distance limitée d'une solution générée par l'algorithme TS. Le choix de LDS garantit une exploration plus diversifiée et rapide de l'espace de recherche, améliorant ainsi le comportement de l'algorithme TS. LDS part de la solution du TS, puis explore successivement les autres solutions, en augmentant le nombre d'écarts (discrepancies), jusqu'à atteindre le nombre maximum d'écarts autorisés.

Le nombre d'écarts est défini de la manière suivante : soit $\bar{x} \in X$ la solution du TS. Soit $\mathbf{x} \neq \bar{x}$ une autre solution admissible. La distance entre \mathbf{x} et \bar{x} est définie par :

$$d(\mathbf{x}, \bar{x}) = \sum_{e \in E} |x_e - \bar{x}_e| \quad (3.9)$$

Etant donné un paramètre entier β (qui sera fixé par l'utilisateur), nous notons par $X_\beta(\bar{x})$ le sous-ensemble de solutions admissibles $\mathbf{x} \in X$ tel que $d(\mathbf{x}, \bar{x}) \leq \beta$. L'algorithme à base de LDS est une légère modification de l'algorithme 2. Cet algorithme effectue une recherche arborescente, mais au lieu d'explorer tout l'espace de recherche, il explore uniquement la région $X_\beta(\bar{x})$.

3.7 Tests et résultats

Nous avons intégré les algorithmes définis dans les parties précédentes dans l'outil NEST. Nous avons également développé la résolution exacte du problème à l'aide de l'algorithme 2. Nous proposons donc dans cette partie de décrire le protocole de test utilisé ainsi que les résultats obtenus. Nous donnerons également un certain nombre de commentaires sur ces résultats.

Pour tous les tests que nous avons effectués, nous nous sommes limités à un ensemble de modèles de lien, de carte et de routeur restreint. Cette restriction est visible dans les tableaux 3.2, 3.3 et 3.4.

Tous les tests ont été effectués sous un processeur Pentium Centrino 2.8 Ghz, fonctionnant sous un système d'exploitation Linux avec 2 Go de mémoire disponible.

Modèle de routeur	# Nombre de Slots	Débit (Gbps)	Coût (K €)
A	12	3	5
B	10	5	10
C	8	10	15

TAB. 3.2 – Modèles de routeurs.

Modèle de carte	# Nombre de ports	Débit (Mbps)	Coût (K €)
1	12	12 × 20	2
2	8	8 × 50	3
3	4	4 × 100	5
4	2	2 × 1000	8

TAB. 3.3 – Modèles de cartes.

Les résultats ci-dessus ont été produits pour une période d'un an pour des instances aléatoires avec des sommets uniformément générés dans un carré de taille $200 \text{ Km} \times 200 \text{ Km}$. Les demandes de trafic entre les paires de nœuds terminaux varient de 1 à 20 *Mbps*.

Dans un premier temps, nous avons testé les algorithmes sur des instances de petites tailles. La Figure 3.5 montre les écarts relatifs en termes de coût entre la solution optimale d'une part, et, d'autre part, les solutions générées par les algorithmes TS, TS+LDS et l'algorithme glouton. La Figure 3.6 illustre les temps de calcul obtenus pour chaque algorithme. Notez que dans cette dernière figure, nous avons utilisé une échelle logarithmique.

A partir de ces résultats on peut remarquer que les approches exactes sont applicables uniquement aux petites instances de notre problème. A partir de dix nœuds, les méthodes exactes deviennent trop gourmandes en temps de calcul.

L'heuristique TS permet de conserver des temps de calcul raisonnables, tout en fournissant des solutions proches de l'optimum. Ces solutions sont toujours améliorées par l'heuristique LDS.

Les résultats ci dessus soulignent donc les avantages associés aux mécanismes de TS et LDS élaborés dans le présent document, à savoir leur capacité à concevoir des réseaux sous des contraintes réelles.

Capacité (Mbps)	20	50	100	1000
Coût annuel/Km (K €)	0.3	0.4	0.5	0.6

TAB. 3.4 – Modèles de liens.

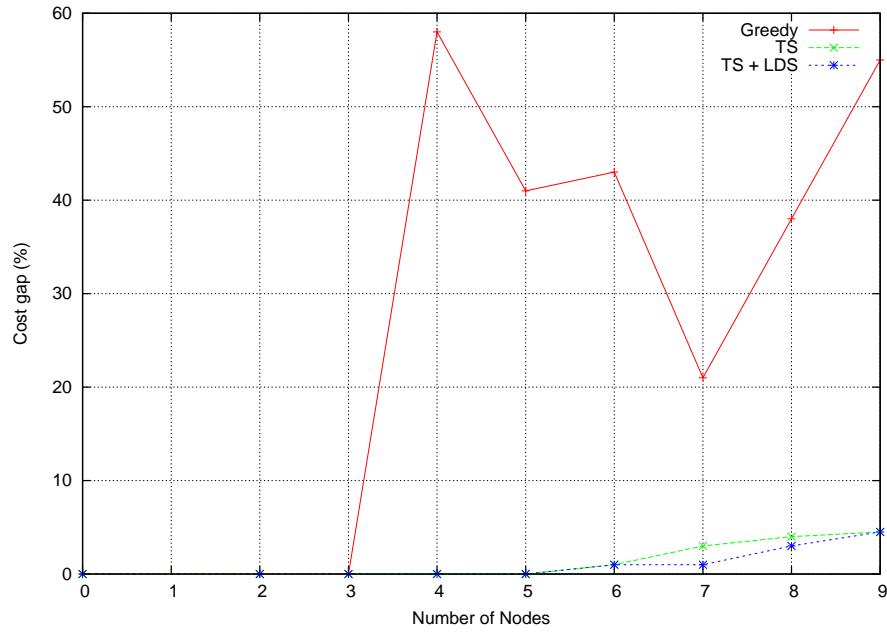


FIG. 3.5 – Ecarts relatif (en %) entre le coût optimal et les coûts des heuristiques étudiées

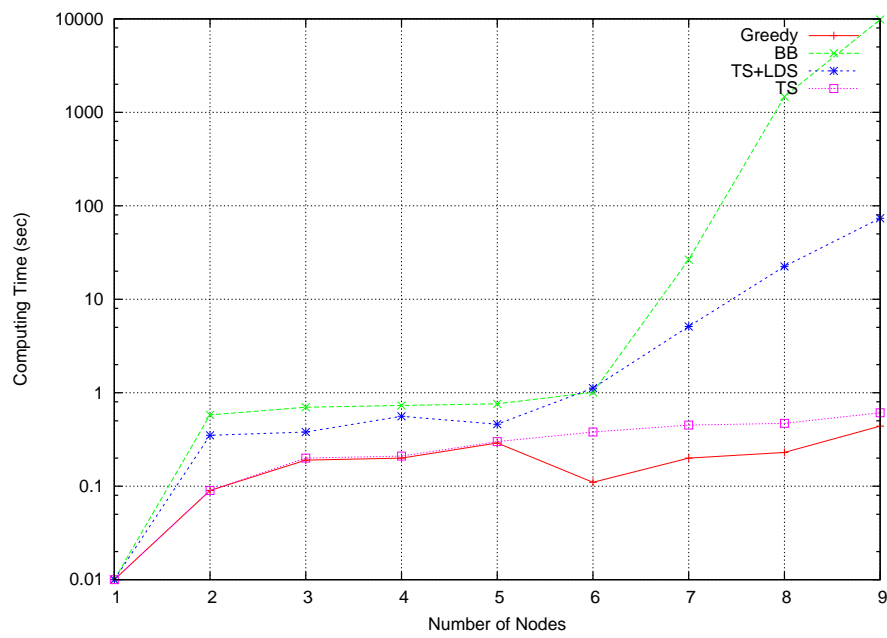


FIG. 3.6 – Temps de calcul en (sec)

# Noeuds	algorithme glouton		TS	
	Coût(K€)	Temps de calcul(sec)	Coût(K€)	Temps de calcul (sec)
10	968	1.05	716	2.1
20	1058	15.5	895	54.2
30	1352	65.5	925	203.1
40	1438	175.1	1265	731.2
50	1574	289.5	1293	1524.6

TAB. 3.5 – TS v.s. algorithme glouton

Le tableau 3.5 propose d'autres résultats permettant une comparaison des performances de l'algorithme TS avec celles de l'algorithme glouton sur des topologies réalistes. Nous avons testé plusieurs instances de chaque taille.

Les résultats montrent que l'algorithme TS surpasse la méthode gloutonne. TS trouve de meilleures solutions que l'heuristique gloutonne, même si cela prend plus de temps de calcul. Ces temps de calcul restent raisonnables vis à vis des tailles des instances étudiées.

Rappelons que l'algorithme glouton ne prend pas en compte le problème de dimensionnement et suppose que le coût du réseau ne dépend que des longueurs des liens. De ce fait, les résultats précédents démontrent que des économies conséquentes peuvent être faites si on l'on tient compte des coûts des équipements lors du processus de design de réseaux ; d'où l'utilité du modèle introduit dans ce chapitre.

3.8 Conclusion

La conception des réseaux de télécommunications représente une tâche très complexe et, en règle générale, fort coûteuse étant donné l'importance des investissements nécessaires. Partant de cet état de fait, l'élaboration d'une topologie optimale a toujours été pour les opérateurs un point important à étudier.

De nombreux travaux portant sur la conception de topologie ont été menés. La plupart de ces travaux ne prennent pas en compte les coûts des équipements. Cette approche a longtemps été considérée comme acceptable dans la mesure où le coût des équipements était très inférieur au coût des liens. Néanmoins, de nos jours, le coût des équipements, et principalement celui des routeurs et des cartes, permettant le branchement des liens sur les routeurs, ne peut plus être négligé.

Nous avons donc proposé dans ce chapitre une nouvelle approche de conception de réseaux basée sur :

- une définition réaliste des coûts menant à la minimisation du coût réel du réseau,

- un ensemble de contraintes matérielles permettant de garantir la faisabilité matérielle de la solution proposée,
- des contraintes de QoS (limitation du délai de bout-en-bout) permettant de proposer une solution respectant les SLA,
- une prise en compte des performances en cas de panne.

Le problème ainsi défini peut être résolu par des techniques de Branch-and-Bound permettant d'explorer la combinatoire. Cependant de tels algorithmes ne passant pas à l'échelle, nous avons proposé deux heuristiques efficaces pour résoudre ce problème.

Chez un opérateur réseau, l'étape postérieure au processus de design de réseau concerne la mise en place du nouveau réseau puis l'exploitation et la gestion opérationnelle de ce réseau. Lorsqu'un réseau existe et doit être exploité, il faut prendre en compte la variabilité du trafic des clients connectés et que les demandes peuvent changer continuellement. On parle dans ce cas d'ingénierie de trafic, dont les prises de décision concernent des échelles de temps relativement courtes (typiquement quelques mois). Dans ce contexte, nous proposons dans les chapitres suivants, d'étudier les problèmes d'ingénierie de réseaux avec des demandes incertaines.

CHAPITRE 4

Conception robuste des VPN avec incertitude sur la demande

4.1 Introduction

Dans le chapitre précédent, nous avons étudié le problème de conception de réseaux. Ce problème se pose généralement aux opérateurs qui souhaitent mettre en place une nouvelle infrastructure de télécommunications. Le processus de conception se base sur une estimation des demandes en trafic dans le futur réseau concernant des échelles de temps relativement longues (typiquement plusieurs années). Ces estimations sont généralement faites grâce à des études de marché (la part de marché du nouvel opérateur, le comportement des utilisateurs, ...).

Cependant, dans la pratique, ces estimations ne sont pas assez précises et ne reflètent pas de manière exacte les ressources dont l'opérateur a besoin durant la phase d'exploitation de son réseau. Il faut anticiper l'évolution globale du trafic, par exemple l'évolution des services dans le réseau et le nombre de clients par service.

Dans ce chapitre nous étudions cette problématique. Nous prenons comme exemple de réseaux les Réseaux Privés Virtuels (VPN).

4.1.1 Rappel sur les VPN

Comme nous l'avons précisé dans le chapitre 1, un réseau privé virtuel (VPN), consiste comme son nom l'indique à créer des liaisons privées en s'appuyant sur un réseau public comme Internet. Cette infrastructure est conçue typiquement pour permettre à des entreprises de connecter différents sites répartis géographiquement sans avoir à investir dans des liaisons louées et sans avoir à supporter les coûts liés à la gestion et à la maintenance d'une telle architecture.

Pour l'entreprise souscrivant l'offre VPN, les nœuds du VPN apparaissent comme étant interconnectés par des liens directs (cf. figure 4.1). Pour le fournisseur du service VPN, ces liens sont des liens virtuels. En fait, les nœuds VPN sont directement reliés à des routeurs IP/MPLS du réseau opérateur, et les liens virtuels correspondent à des chemins entre ces routeurs dans le réseau IP/MPLS.

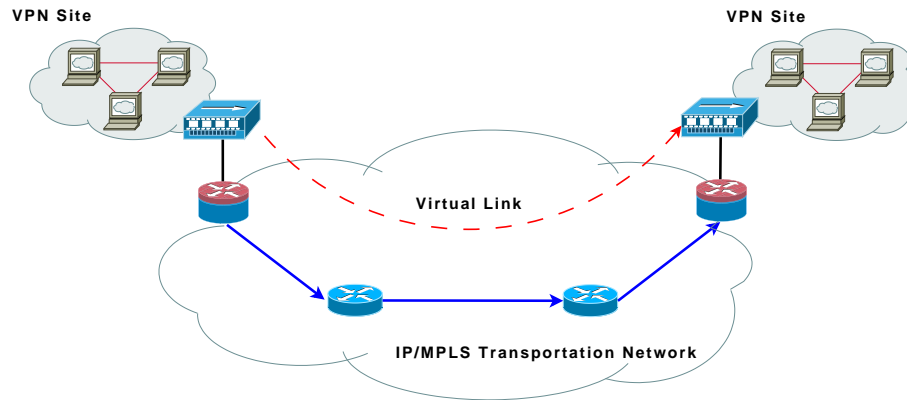


FIG. 4.1 – Représentation d'un lien virtuel

Aujourd'hui, les offres VPN les plus répandues utilisent MPLS comme support pour établir les tunnels. La majorité des opérateurs utilisent un plan de routage en mono-chemin pour relier les différents sites du VPN.

Dans la suite de ce chapitre, nous considérons les VPN MPLS avec un plan de routage en mono-chemin.

4.1.2 Spécification de la bande-passante d'un VPN

Deux approches sont généralement utilisées pour permettre à un client VPN de spécifier la capacité de communication dont il a besoin. La première, connue sous le nom de modèle "Pipe" [69] est illustrée sur la figure 4.2(a). Elle suppose que le client VPN décrit la demande maximale entre chaque paire de nœuds du VPN. S'il veut interconnecter N nœuds, cela suppose donc qu'il fournisse $N(N-1)$ valeurs de capacité, ce qui peut être fastidieux quand le VPN relie plusieurs dizaines de nœuds. Surtout, le modèle "Pipe" a l'inconvénient majeur d'exiger du client VPN une connaissance précise de la demande entre ses différents sites, ce qui n'est généralement pas le cas.

Le modèle "Hose" (cf. figure 4.2(b)), introduit par Duffield et al. en 1999 [49], est beaucoup plus simple et plus flexible pour le client VPN. Celui-ci n'a qu'à spécifier une bande-passante en émission et une autre en réception pour chaque nœud. Ainsi, avec N nœuds dans le VPN, il suffit de spécifier $2N$ valeurs. La bande-passante en émission d'un nœud VPN représente le débit maximal auquel il pourra émettre vers les autres, tandis que sa bande-passante en réception représente le débit maximal auquel il pourra recevoir des données en provenance des autres nœuds. Le modèle "Hose" est de loin le plus

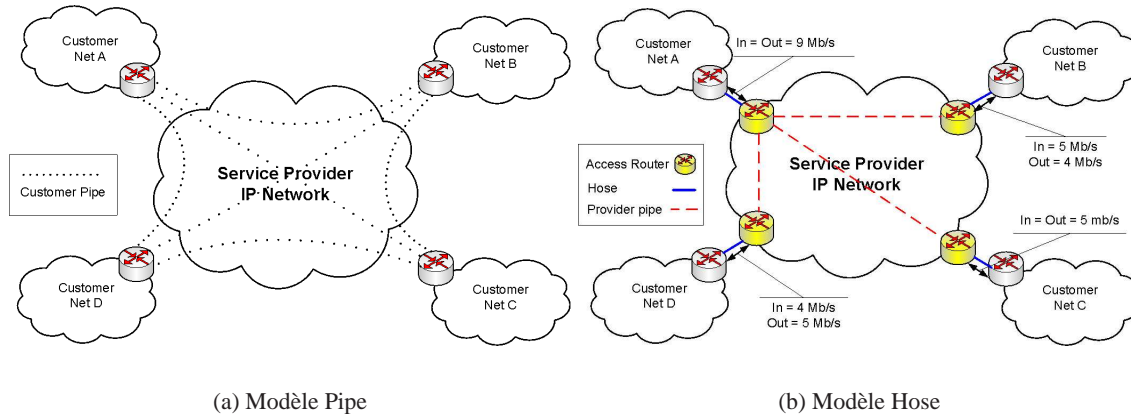


FIG. 4.2 – Modèles Pipe et Hose de spécification de la bande-passante d'un VPN.

utilisé actuellement, et dans la suite, nous supposons que la demande en bande-passante des VPN est décrite suivant ce modèle.

4.1.3 Réserve de la bande-passante d'un VPN

Si le modèle hose simplifie grandement la spécification de la demande en bande-passante du VPN pour le client, la réserve des ressources devient plus complexe pour l'opérateur dans la mesure où le modèle hose ne spécifie pas une seule matrice de trafic mais un ensemble de matrices de trafic. En fait, le modèle hose définit un polytope de matrices de trafic : toute matrice de trafic telle que la somme sur chaque ligne est inférieure à la bande-passante en émission du nœud source correspondant, et telle que la somme sur chaque colonne est inférieure à la bande-passante en réception du nœud destination correspondant, est compatible avec le modèle hose. Ainsi, l'opérateur est confronté à un problème d'incertitude sur la matrice de trafic.

La bande-passante réservée par l'opérateur sur un lien du réseau doit être supérieure au volume maximal de trafic pouvant s'écouler sur ce lien dans le cadre du modèle hose, c'est à dire quelle que soit la matrice de trafic pourvu qu'elle appartienne à la région d'incertitude définie par ce modèle. Le problème de conception optimale d'un VPN consiste à déterminer le routage des liens virtuels dans le réseau IP/MPLS sous-jacent de manière à optimiser une certaine fonction des bandes-passantes réservées.

Dans la majorité des travaux qui ont été consacrés à ce problème, la fonction coût à minimiser correspond à la bande-passante totale réservée, c'est à dire à la somme des bandes-passantes réservées sur les liens. Ce problème est illustré sur la figure 4.3. Dans cette figure, nous présentons dans 4.3(b) et 4.3(c) les capacités réservées pour deux plans de routage différents du problème représenté en 4.3(a). Chaque nœud est étiqueté par une paire (in, out) indiquant les bandes passantes en réception et émission sur ce nœud. La capacité de chaque lien physique du réseau est supposée infinie. Les valeurs indiquées

sur les liens dans les figures 4.3(b) et 4.3(c) correspondent à la bande passante à réserver sur chacun de ces liens. Nous n'avons pas représenté les liens pour lesquels la réservation est nulle. Dans la solution représentée sur la figure 4.3(b), la capacité totale reversée est de 54 ; les trafics sont routés selon les chemins de l'arbre représenté sur cette figure. Dans la solution représentée sur la figure 4.3(c), la capacité totale réservée est de 46 ; le plan de routage est défini de la manière suivante : les trafics sont routés selon les plus courts chemins ; s'il existe deux plus courts chemins entre deux nœuds, le trafic est routé suivant le sens des aiguilles d'une montre.

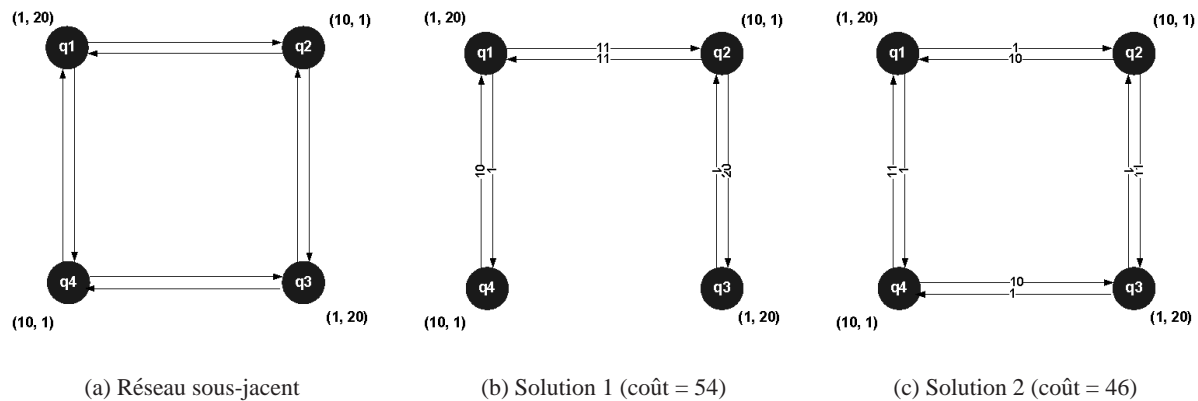


FIG. 4.3 – Exemples d'instances de notre problème

4.1.4 Contribution

Dans ce chapitre, nous étudions le problème de la conception des VPN Hose avec un plan de routage en mono-chemin. C'est le cas des VPN MPLS qui sont de nos jours largement répandus.

Nous considérons deux critères d'optimisation qui sont classiquement utilisés par les opérateurs :

1. la bande-passante totale réservée,
2. l'utilisation maximale des liens.

Pour optimiser ces deux critères, nous proposons des algorithmes, basés sur les techniques de recherche locale, très simples et qui donnent souvent de bons résultats comme on le verra. Ces algorithmes sont utilisés en mode multi-start à partir d'un grand nombre de solutions initiales générées aléatoirement.

4.1.5 Organisation de ce chapitre

Le reste de ce chapitre est organisé de la façon suivante. La section 4.2 dresse un état de l'art des travaux antérieurs sur le sujet. La section 4.3 est consacrée à l'étude du problème de minimisation du

critère correspondant à la bande-passante totale réservée. Dans la section 4.4 nous traitons le problème de minimisation du critère correspondant à l'utilisation maximale des liens. Les résultats expérimentaux sont présentés au paragraphe 4.5, tandis que les conclusions et perspectives de ce travail sont analysées au paragraphe 4.6.

4.2 Etat de l'art

L'incertitude sur le trafic est un aspect crucial du problème de conception des VPN. L'intégration de l'incertitude sur la demande est une tendance récente de la communauté scientifique travaillant sur les problèmes d'optimisation de réseaux. La plupart des travaux traitant le problème de conception de réseaux avec incertitude sur la demande ont d'ailleurs été consacrés au problème de conception des VPN, bien que certains articles récents aient abordés d'autres problèmes de routage, comme celui du routage optimal multi-chemin [70, 97, 81], et celui de l'optimisation des métriques de routage IP [100, 82, 102].

Dans la majorité des travaux consacrés au problème de conception des VPN Hose, la fonction coût à minimiser correspond à la bande-passante totale réservée.

Ce problème a été considéré dans [76] en supposant qu'un routage multi-chemin des liens virtuels est possible. En utilisant une formulation de type programmation linéaire (LP) avec un nombre exponentiel de contraintes, les auteurs ont proposé une approche par plans de coupe et ont montré que la réservation de bande-passante peut être significativement réduite par rapport à un routage arborescent des liens virtuels. Pour le même problème, une formulation LP compacte avec un nombre polynomial de contraintes a été proposée dans [91]. Malgré ses avantages, le routage multi-chemin des liens virtuels n'est pas possible en pratique. Pour les réseaux MPLS, ce problème doit être considéré avec la contrainte supplémentaire d'un routage monochemin des liens virtuels.

Certains auteurs ont traité le problème de routage monochemin en se restreignant à un routage arborescent des liens virtuels. Dans ce cas, le réseau de réservation (union des arcs avec une bande-passante réservée strictement positive) doit être un arbre, et il existe une seule route pour chaque lien virtuel. Pour des bandes-passantes en émission et en réception symétriques et des capacités de liens infinies, un VPN avec une réservation minimale de bande-passante peut être construit en temps polynomial grâce à des calculs de plus courts chemins [61, 59, 62]. Cependant, le problème devient fortement NP-difficile si on suppose des capacités de liens finies ou des bandes-passantes en émission et en réception asymétriques [59, 71]. Le meilleur algorithme d'approximation connue garantit une réservation de bande-passante qui est au plus 9 fois supérieure à l'optimum [63].

Le problème de conception des VPN a aussi été considéré pour un routage monochemin général (sans la restriction d'un routage arborescent). Pour des bandes-passantes en émission et en réception symétriques, on sait que tout routage arborescent optimal fournit une 2-approximation au problème de routage monochemin. Il est même conjecturé dans [98] que dans ce cas le problème admet un routage arborescent en tant que solution optimale (voir [92] pour une preuve dans le cas de topologies en anneau). Pour des bandes-passantes en émission et en réception asymétriques, Gupta et al. ont proposé un

Routage	Asymétrique	Symétrique
Arbre	\mathcal{NP} -Difficile	Polynomial
mono-chemin	\mathcal{NP} -Difficile	\mathcal{NP} -Difficile
Multi-chemin	Polynomial	Polynomial

TAB. 4.1 – Résumé des Résultats Connus

algorithme de 5.55-approximation [71]. Dans [91], Altin et al. présentent une formulation linéaire en nombres entiers (MIP) qui peut être utilisée pour résoudre des problèmes de taille moyenne avec des solveurs MIP du commerce. Ils proposent également un algorithme combinant branch-and-price et plan de coupe pour traiter des instances de plus grandes dimensions.

Un résumé des résultats connus dans la littérature est donné dans le tableau 4.1.

4.3 Minimisation de la bande-passante totale réservée

Classiquement, le critère d'optimisation considéré dans la littérature correspond à la bande-passante totale réservée dans le réseau. Il s'agit de minimiser la somme des bandes-passantes réservées sur les liens du réseau. Dans cette partie, nous proposons une formulation de ce problème d'optimisation. Nous présentons également une approche de résolution basée sur les techniques de recherche locale.

4.3.1 Notations

Le réseau sous-jacent sur lequel on doit réserver des ressources pour le VPN est modélisé par un graphe dirigé $G = (V, E)$, où V est l'ensemble des nœuds du réseau et E est l'ensemble des liens de communication entre nœuds adjacents. Chaque arc $e \in E$ a une capacité limitée C_e . On notera c_e le coût induit par la réservation d'une unité de bande-passante sur ce lien. Pour chaque nœud $v \in V$, on note $\delta^-(v)$ et $\delta^+(v)$ l'ensemble des arcs sortants et entrants, respectivement.

Soit $Q \subset V$ le sous ensemble des nœuds extrémités du VPN. La demande en bande-passante du VPN est décrite à l'aide du modèle hose. On note $b^+(v)$ and $b^-(v)$ les bandes-passantes en émission (ingress) et réception (egress) du nœud $v \in Q$, respectivement. Dans la suite, nous supposons que $\sum_{v \in Q} b_v^+ = \sum_{v \in Q} b_v^-$. Une matrice de trafic $\mathbf{d} = (d_{s,t})_{s,t \in Q}$ est compatible avec le modèle hose si elle satisfait les contraintes linéaires suivantes :

$$\sum_{t \in Q} d_{s,t} \leq b^+(s) , s \in Q \quad (4.1)$$

$$\sum_{s \in Q} d_{s,t} \leq b^-(t) , t \in Q \quad (4.2)$$

$$d_{s,s} = 0 , s \in Q \quad (4.3)$$

Nous noterons \mathcal{D} le polytope de matrices de trafic défini par ces contraintes. Ce polytope représente l'ensemble d'incertitudes sur la demande du VPN.

Le fournisseur de service VPN doit définir un plan de routage mono-chemin pour les liens virtuels du VPN. En d'autres termes, pour chaque couple (s, t) de nœuds extrémités du VPN, la demande d_{st} doit être routée sur un unique chemin entre les nœuds s et t . Soit $f_{s,t}^e$ la proportion de trafic entre les nœuds s et t routée à travers le lien e . Nous définissons un plan de routage comme un vecteur $\mathbf{f} = [f_{s,t}^e]_{s,t \in Q, e \in E}$ tels que :

$$\sum_{e \in \delta^+(v)} f_{s,t}^e - \sum_{e \in \delta^-(v)} f_{s,t}^e = 0, \quad s, t \in Q, v \neq s, t, \quad (4.4)$$

$$\sum_{e \in \delta^+(v)} f_{s,t}^e - \sum_{e \in \delta^-(v)} f_{s,t}^e = 1, \quad s, t \in Q, v = s, \quad (4.5)$$

$$\sum_{e \in \delta^+(v)} f_{s,t}^e - \sum_{e \in \delta^-(v)} f_{s,t}^e = -1, \quad s, t \in Q, v = t \quad (4.6)$$

$$f_{s,t}^e \in \{0, 1\}, \quad e \in E, s, t \in Q \quad (4.7)$$

Les contraintes (4.4)-(4.6) représentent les contraintes de conservation de flots pour des nœuds intermédiaires, le nœud source et le nœud destination, respectivement. Les contraintes (4.7) spécifient que le routage des liens virtuels doit être un routage monochemin.

Dans la suite, nous noterons \mathcal{F} l'ensemble des plans de routage, c'est à dire l'ensemble des vecteurs \mathbf{f} satisfaisant les contraintes ci-dessus. Etant donné un plan de routage $\mathbf{f} \in \mathcal{F}$, le vecteur $\mathbf{f}_{s,t} = [f_{s,t}^e]_{e \in E}$ représente la stratégie de routage du lien virtuel (s, t) . Ce lien virtuel est routé sur le chemin $\pi_{s,t} = \{e \in E : f_{s,t}^e = 1\}$.

4.3.2 Formulation en programmation linéaire mixte

Notons x_e la quantité de bande-passante réservée par l'opérateur sur le lien $e \in E$ et $\mathbf{x} = [x_e]_{e \in E} \in \mathbb{R}^{|E|}$ le vecteur de réservation de bande-passante. L'objectif de l'opérateur est de déterminer un plan de routage minimisant le coût total $\sum_{e \in E} c_e x_e$ de la bande-passante réservée, tout en garantissant que la bande-passante réservée sur chaque lien sera suffisante quelle que soit la matrice de trafic $D \in \mathcal{D}$. Le problème peut être formulé sous la forme d'un programme linéaire mixte :

$$\min_{\mathbf{x}, f} \sum_{e \in E} c_e x_e \quad (4.8)$$

s.t.

$$\sum_{u, v \in Q} f_{u, v}^e d_{u, v} \leq x_e, \quad D \in \mathcal{D}, e \in E \quad (4.9)$$

$$0 \leq x_e \leq C_e, \quad e \in E \quad (4.10)$$

$$\mathbf{x} \in \mathbb{R}^{|E|}, \quad (4.11)$$

$$f \in \mathcal{F}, \quad (4.12)$$

Dans la formulation précédente, les contraintes (4.9) garantissent que le trafic sur chaque lien $e \in E$ ne dépasse pas la capacité x_e réservée sur ce lien, et ce quelle que soit la matrice de trafic $D \in \mathcal{D}$. Les contraintes (4.10) sont des contraintes de capacité. La formulation ci-dessus correspond à un problème d'optimisation qui comporte un nombre infini de contraintes puisque qu'il y a un nombre infini de matrices \mathbf{d} dans \mathcal{D} . De manière classique, on peut contourner cette difficulté en dualisant la contrainte (4.9), comme il a été fait dans [91, 92, 70]. La formulation du problème devient ainsi :

$$\min_{\mathbf{x}, f} \sum_{e \in E} c_e x_e \quad (4.13)$$

s.t.

$$\sum_{v \in Q} b^+(v) \cdot \pi_e(v) + b^-(v) \cdot \lambda_e(v) \leq x_e, \quad e \in E \quad (4.14)$$

$$f_{s, t}^e \leq \pi_e(s) + \lambda_e(t), \quad s, t \in Q, e \in E \quad (4.15)$$

$$0 \leq x_e \leq C_e, \quad e \in E \quad (4.16)$$

$$\pi_e(u), \lambda_e(u) \geq 0, \quad u \in Q, e \in E \quad (4.17)$$

$$\mathbf{x} \in \mathbb{R}^{|E|} \quad (4.18)$$

$$f \in \mathcal{F}, \quad (4.19)$$

Le problème ci-dessus peut être résolu en temps polynomial si l'on relaxe les contraintes (4.7) en $0 \leq f_{s, t}^e \leq 1$ [76], autrement dit si on renonce au mono-routage pour accepter un partage de charge sur plusieurs chemins. Avec la contrainte de mono-routage, ce problème est connu pour être NP-difficile [59].

Non seulement les temps calcul augmentent exponentiellement avec la taille du problème, mais ils peuvent déjà être prohibitifs avec des solveurs MIP du commerce pour de petites instances. Ceci est illustré dans la figure 4.4.

Dans cette figure, nous considérons deux réseaux de petite taille. Le réseau représenté dans la figure 4.4(a) contient 7 nœuds terminaux (ou clients), alors que celui de la figure 4.4(b) contient 8 nœuds terminaux. Nous supposons dans les deux exemples que chaque terminal envoie et reçoit au maximum 8 Mb/s. On indique sur chaque lien physique sa capacité en Mb/s. La fonction coût considérée est l'utilisation maximale des liens. Pour résoudre le problème (3), nous avons utilisé le solveur CPLEX [107]. Nous indiquons pour chaque exemple le temps de calcul mis par CPLEX pour trouver la solution optimale.

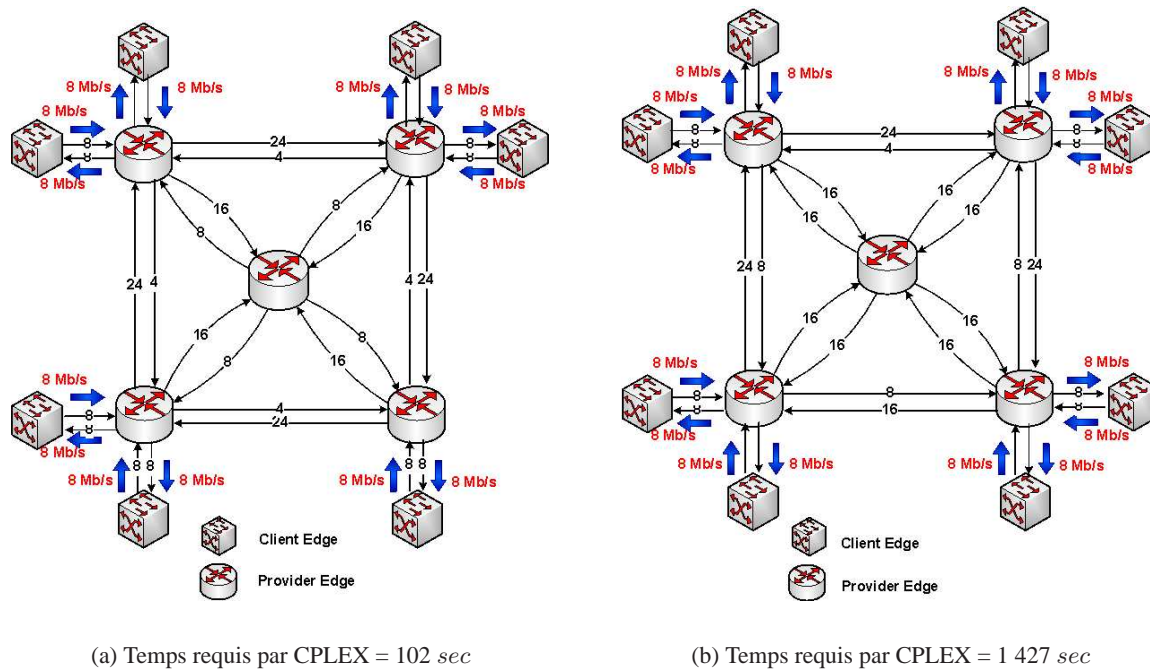


FIG. 4.4 – Exemple d’instances générant des temps de calcul considérables.

4.3.3 Formulation nœud-chemin

Nous proposons dans cette section de réécrire le problème ci dessus suivant une formulation nœud-chemin.

Notons $\Pi_{s,t}$ l’ensemble de tous les chemins entre les nœuds $s, t \in Q$ dans G et $\Pi = \cup_{s,t \in Q} \Pi_{s,t}$. Pour chaque chemin $\pi \in \Pi$, on notera $s(\pi)$ le nœud source et $t(\pi)$ le nœud destination. Introduisons les variables de décision suivantes :

$$f_{\pi} = \begin{cases} 1 & \text{si le trafic entre } s(\pi) \text{ et } t(\pi) \text{ est routé sur } \pi, \\ 0 & \text{sinon.} \end{cases}$$

Un plan de routage peut alors être défini de manière équivalente comme un vecteur $\mathbf{f} \in \mathcal{F}_\pi$, où,

$$\mathcal{F}_\pi = \left\{ \mathbf{f} = \{0, 1\}^{|\mathcal{Q}| \cdot |\mathcal{Q}-1|} : \sum_{\pi \in \Pi_{s,t}} f_\pi = 1 \quad \forall s, t \in \mathcal{Q} \right\}$$

On remarquera que l'on obtient facilement la définition précédente d'un plan de routage avec la formule suivante :

$$f_{u,v}^e = \sum_{\pi \in \Pi_{u,v}} \gamma_\pi^e f_\pi \quad e \in E, u, v \in \mathcal{Q}$$

où γ_π^e est égal à 1 si le chemin π passe par l'arc e , et 0 sinon.

Nous définissons une solution du problème comme un vecteur $\mathbf{f} \in \mathcal{F}_\pi$. Une solution est dite admissible si elle est telle que $x_e(\mathbf{f}) \leq C_e$ pour tout $e \in E$, où $x_e(\mathbf{f})$ représente le trafic maximal qui peut passer sur le lien e . $x_e(\mathbf{f})$ est donc la solution du programme linéaire suivant :

$$x_e(\mathbf{f}) = \max_{\mathbf{D}} \sum_{\pi \in \Pi} \gamma_\pi^e f_\pi \cdot d_{s(\pi),t(\pi)} \quad (4.20)$$

s.t.

$$\sum_{t \in \mathcal{Q}} d_{s,t} \leq b^+(s), \quad s \in \mathcal{Q} \quad (4.21)$$

$$\sum_{t \in \mathcal{Q}} d_{t,s} \leq b^-(s), \quad s \in \mathcal{Q} \quad (4.22)$$

$$d_{s,s} = 0, \quad s \in \mathcal{Q} \quad (4.23)$$

$$d_{s,t} \geq 0, \quad s, t \in \mathcal{Q}, s \neq t \quad (4.24)$$

Le problème (4.8)-(4.12) est alors équivalent au problème d'optimisation bi-niveau suivant :

$$\min_{\mathbf{f} \in \mathcal{F}_\pi} \sum_{e \in E} c_e x_e(\mathbf{f}) \quad (4.25)$$

s.t.

$$x_e(\mathbf{f}) \leq C_e, \quad e \in E \quad (4.26)$$

4.3.4 Méthode heuristique

Les techniques de résolutions exactes ont le défaut principal d'avoir des temps de calcul trop importants dès que la taille du problème augmente. Pour résoudre ce problème de passage à l'échelle, nous proposons dans la suite une méthode heuristique.

L'Algorithme 5 décrit l'heuristique proposée. Il s'agit d'un algorithme multi-start qui de manière répétitive utilise une procédure de recherche locale à partir de points initiaux générés aléatoirement. Partant d'une solution initiale, la procédure de recherche locale réduit son coût itérativement jusqu'à ce qu'un minimum local soit atteint. Nous décrivons les détails de cet algorithme ci-dessous.

algorithm 5 Heuristique de routage mono-chemin

procédure Single_Path_Routing

Let $\hat{\mathbf{f}}$ be the optimal multi-path routing scheme.

$\Phi(\mathbf{f}^*) = +\infty$.

§Initialisation de la solution de coût min .

for $k = 1 \dots N_{\max}$ **do**

$\mathbf{f}^k = \text{random_generation}(\hat{\mathbf{f}})$.

 Local_Search (\mathbf{f}^k).

if $\Phi(\mathbf{f}^k) < \Phi(\mathbf{f}^*)$ **then**

$\mathbf{f}^* = \mathbf{f}^k$.

end if

end for

Génération aléatoire des points initiaux

L'algorithme 5 se déroule en N_{\max} itérations. A chacune d'elles, il génère aléatoirement un nouveau point initial (un nouveau plan de routage monochemin) pour la procédure de recherche locale.

Pour générer ces points initiaux, nous utilisons la version multi-chemin du problème (4.13)-(4.19), c'est à dire que nous relaxons les contraintes de monoroutage (4.7) en $0 \leq f_{s,t}^e \leq 1$. Ce problème se résout très facilement avec un solveur linéaire.

Le routage optimal multi-chemin $\hat{\mathbf{f}}$ utilise en général plusieurs chemins pour router chaque lien virtuel (s, t) . Soit $\Pi_{s,t}$ l'ensemble des chemins entre s et t utilisés par ce routage. Définissons le poids d'un chemin $\pi \in \Pi_{s,t}$ de la façon suivante : $w_\pi = \prod_{e \in \pi} f_{s,t}^e$. Pour générer un routage monochemin admissible, nous routons itérativement chacun des chemins virtuels (s, t) sur un chemin de $\Pi_{s,t}$. Le chemin $\pi \in \Pi_{s,t}$ est sélectionné avec la probabilité w_π . En appliquant cette procédure pour chacun des chemins virtuels, on obtient une solution initiale du problème de monoroutage générée aléatoirement.

Nous avons déjà signalé que la détermination d'une solution admissible pour le problème (4.8)-(4.12) est NP-difficile [59]. Cela constitue un obstacle majeur pour la mise en oeuvre d'une méthode de recherche locale, puisque la solution initiale doit satisfaire les contraintes (4.10). Pour résoudre cette difficulté, nous proposons l'utilisation des méthodes de pénalités [12]. L'idée est de s'affranchir des

contraintes (4.10) pour pouvoir démarrer d'une solution arbitraire dans \mathcal{F}_π , pour ensuite guider la recherche vers une zone de l'espace des solutions contenant des solutions admissibles. Le problème que nous allons chercher à résoudre dans la suite, s'écrit de la façon suivante :

$$\min_{\mathbf{f} \in \mathcal{F}_\pi} \Phi(\mathbf{f}) = \sum_{e \in E} c_e x_e(\mathbf{f}) + A \cdot \sum_{e \in E} p_e(\mathbf{f}) \quad (4.27)$$

où $A > 0$ est un paramètre ajustable, et $p_e(\mathbf{f})$ est définie de la façon suivante :

$$p_e(\mathbf{f}) = \begin{cases} 0 & \text{si } x_e(\mathbf{f}) \leq C_e, \\ [C_e - x_e(\mathbf{f})]^2 & \text{sinon.} \end{cases}$$

Algorithme de recherche locale

La méthode que nous proposons est basée sur la recherche locale. Elle est décrite dans l'Algorithme 6.

Son originalité principale est d'utiliser une borne supérieure $\Psi(\mathbf{f})$ sur le coût d'une solution \mathbf{f} . En effet, l'évaluation du coût exact $\Phi(\mathbf{f})$ de \mathbf{f} nécessite le calcul de $x_e(\mathbf{f})$ pour chaque lien $e \in E$, ce qui veut dire résoudre le problème (4.20)-(4.24) pour chaque lien. Pour explorer le voisinage d'une solution, c'est à dire évaluer le coût de tous ses voisins, il faut alors résoudre le problème (4.20)-(4.24) un très grand nombre de fois, ce qui est pénalisant en temps calcul. L'idée est alors d'utiliser une borne supérieure $\Psi(\mathbf{f})$ sur le coût $\Phi(\mathbf{f})$, à la fois très simple à calculer et proche du coût réel, pour déterminer si le voisinage contient une solution améliorante. Si c'est le cas, ce voisin sera choisi comme solution courante de la prochaine itération. Sinon, on refait une exploration du voisinage en utilisant cette fois le coût exact pour pouvoir garantir que le voisinage ne contient pas de solution améliorante. Les détails de l'algorithme sont décrits ci-dessous.

algorithm 6 Algorithme de recherche locale

```

 $\mathbf{f}_0$  : solution initiale
 $k = 0$  and  $\mathbf{f}^* = \mathbf{f}_0$ 
while a local minimum is not reached do
   $\mathbf{f}_{k+1} = \operatorname{argmin}_{\mathbf{f} \in N(\mathbf{f}_k)} \Psi(\mathbf{f})$ 
  if  $\Phi(\mathbf{f}_{k+1}) > \Phi(\mathbf{f}_k)$  then
     $\mathbf{f}_{k+1} = \operatorname{argmin}_{\mathbf{f} \in N(\mathbf{f}_k)} \Phi(\mathbf{f})$ 
  end if
  if  $\Phi(\mathbf{f}_{k+1}) < \Phi(\mathbf{f}^*)$  then
     $\mathbf{f}^* = \mathbf{f}_{k+1}$ 
  end if
end while

```

Structure du voisinage

Une solution admissible du problème d'optimisation (4.27) peut être construite en choisissant, pour chaque paire origine-destination (s, t) , un chemin de l'ensemble $\pi_{s,t}$.

Etant donné une solution f de notre problème, l'idée de base de notre algorithme est très simple : pour réduire le coût la solution f , il faut transformer le plan de routage en déviant au moins un flux (lien virtuel) sur un chemin différent de celui emprunté par ce flux dans f (cf figure 4.5).

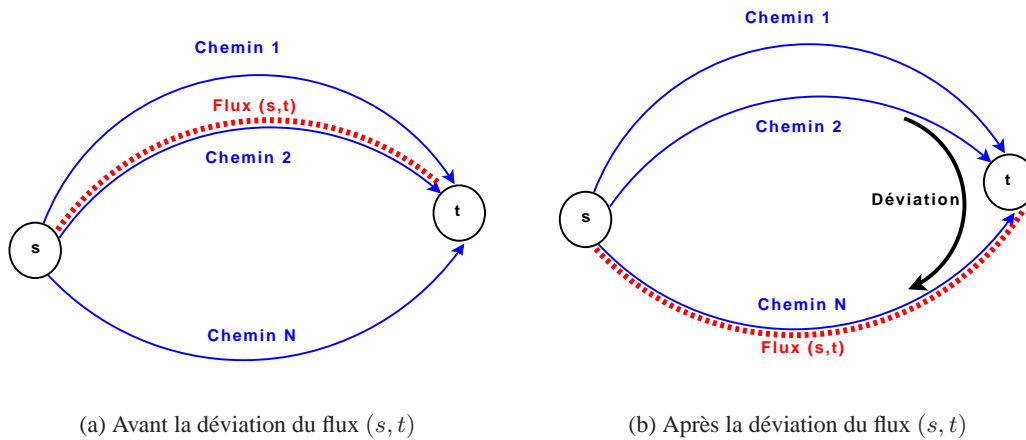


FIG. 4.5 – Exemple de déviation de flux entre s et t

Soit f une solution. On définit le voisinage $N(f)$ de f par :

$$N(f) = \{f^{s,t}\}_{s,t \in Q} \quad (4.28)$$

où

$$f^{s,t} = \operatorname{argmin}_{f' \in D_{s,t}(f)} \Phi(f') \quad (4.29)$$

avec

$$D_{s,t}(f) = \{f' \in \mathcal{F} : (\forall \pi \in \Pi \setminus \Pi_{s,t}, f'_\pi = f_\pi) \quad \text{et} \quad (\exists! \pi \in \Pi_{s,t}, f'_\pi = 1 \quad \text{et} \quad f_\pi = 0)\} \quad (4.30)$$

$D_{st}(f)$ est l'ensemble des solutions obtenues en déviant seul le flux (s, t) sur un chemin différent de celui emprunté par ce flux dans f . Le voisin $f^{s,t}$ correspond donc à la meilleure solution qu'on peut obtenir à partir de f en déviant le flux (s, t) sur un chemin différent de celui emprunté par ce flux dans f .

Ainsi, le voisinage d'une solution \mathbf{f} contient exactement $|Q| \times (|Q| - 1)$ solutions. Chacune de ces solutions est associée à un flux (s, t) , $s, t \in Q$.

La génération du voisinage d'une solution \mathbf{f} consiste à déterminer, pour chaque flux (s, t) , le chemin $\pi^* \in \Pi_{s,t}$ tel que pour chaque chemin $\pi \in \Pi_{s,t}$ le coût de la solution obtenue en déviant le flux (s, t) sur π est supérieur ou égal au coût de la solution obtenue en déviant (s, t) sur π^* . Pour trouver le chemin alternatif π^* , nous définissons tout d'abord l'ensemble de liens :

$$U_{s,t} = \{e \in \pi : \pi \in \Pi_{s,t} \text{ et } f_\pi = 0\}, \quad (4.31)$$

L'ensemble ci-dessus regroupe tous les liens e situés sur des chemins entre s et t qui sont non utilisés par le routage. On définit ensuite pour chaque lien de l'ensemble $U_{s,t}$ un poids w_e qui correspond à la variation de la contribution de ce lien au coût global si l'on route le flux (s, t) sur ce lien :

$$w_e = x_e(\mathbf{f} + \mathbf{e}_\pi) - x_e(\mathbf{f}), \quad e \in U. \quad (4.32)$$

où $\mathbf{e}_{s,t}^l$ est le vecteur $(0, \dots, 1, \dots, 0)$ avec un 1 en position (s, t, e) et des 0 partout ailleurs.

Déterminer le chemin π^* revient alors à résoudre un problème de plus court chemin entre s et t dans le graphe réduit $(V, U_{s,t})$ en fonction des métriques w_e . Le choix est arbitraire s'il y a plusieurs plus courts chemins.

A chaque itération, une fois qu'un voisin de la solution courante \mathbf{f} est sélectionné (on suppose que ce voisin correspond au flot (s, t)), on met à jour le plan de routage de la façon suivante :

$$f_{s,t}^e = \begin{cases} 1 & e \in \pi^* \\ 0 & \text{sinon} \end{cases} \quad (4.33)$$

L'algorithme s'arrête quand un minimum local est atteint.

Evaluation exacte du coût d'une solution

Le coût d'une solution $\mathbf{f} \in \mathcal{F}_\pi$ est donné par :

$$\Phi(\mathbf{f}) = \sum_{e \in E} c_e x_e(\mathbf{f}) + A \cdot \sum_{e \in E} p_e(\mathbf{f})$$

Le calcul de ce coût suppose de déterminer pour chaque lien e la valeur $x_e(\mathbf{f})$, qui représente le trafic maximal qui peut passer sur le lien e pour l'ensemble des matrices de trafic valides du polyèdre \mathcal{D} .

On a vu que, pour un lien e donné, $x_e(\mathbf{f})$ est en fait le coût optimal du problème (4.20)-(4.24). Remarquons que la solution de ce problème ne nous intéresse pas, puisque nous voulons seulement connaître le coût optimal. Le programme linéaire (4.20)-(4.24) peut être efficacement résolu en utilisant

une technique générique tel que l'algorithme du Simplexe. Néanmoins, il faut bien observer que pour calculer le coût d'une solution il faudra résoudre cela pour chaque lien $e \in E$. Par conséquent, l'exploration du voisinage d'une solution va nécessiter un nombre important de résolutions de ce problème.

Pour accélérer la procédure de calcul de $x_e(\mathbf{f})$, nous remarquons que le problème (4.20)-(4.24) est en fait un problème de transport. Comme illustré sur la figure 4.6, on construit un graphe biparti en plaçant à gauche les nœuds qui envoient du trafic sur un chemin passant par le lien e et à droite les nœuds qui reçoivent du trafic sur un de ces chemins. En d'autres termes, pour chaque chemin p passant par le lien e , on place à gauche le nœud $s(p)$ et à droite le nœud $t(p)$. Il peut bien sûr arriver qu'un même terminal apparaisse à gauche et à droite. Chaque nœud u situé à gauche a une offre de $b^+(u)$, et chaque nœud v à droite une demande égale à $b^-(v)$. Il existe dans le graphe biparti un arc entre les sommets u et v si on peut trouver un chemin p passant par e tel que $u = s(p)$ et $v = t(p)$. Le poids de cet arc est alors fixé à -1 (pour se ramener à un problème de minimisation). La résolution du problème de transport peut alors se faire en utilisant l'algorithme de Stepping Stone [1].

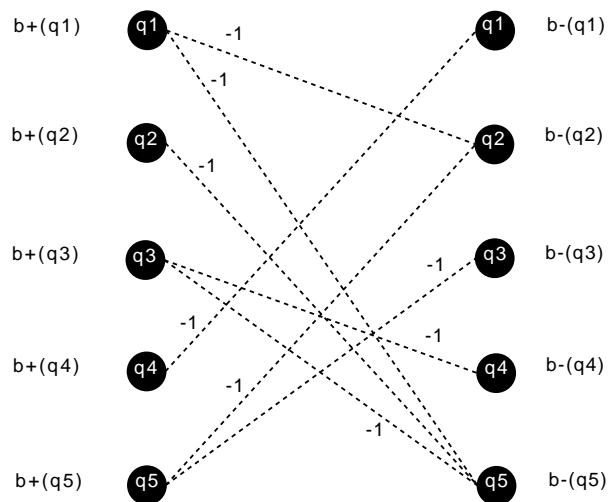


FIG. 4.6 – Construction du réseau de transport relatif à l'arc $e \in E$ (seuls les arcs avec un coût non nul sont représentés).

Borne Supérieure sur le coût d'une solution

Considérons les deux problèmes linéaires suivants qui sont des relaxations du problème (4.20)-(4.24) obtenues en remplaçant respectivement les contraintes (4.22) et (4.21) par (4.36) dans le problème (4.34)-(4.38) et par (4.40) dans le problème (4.39)-(4.43).

$$\alpha_e = \max_{\mathbf{D}} \sum_{\pi \in \Pi} \gamma_{\pi}^e f_{\pi} \cdot d_{s(\pi),t(\pi)} \quad (4.34)$$

s.t.

$$\sum_{t \in Q} d_{s,t} \leq b^+(s), \quad s \in Q \quad (4.35)$$

$$d_{t,s} \leq b^-(s), \quad t, s \in Q \quad (4.36)$$

$$d_{s,s} = 0, \quad s \in Q \quad (4.37)$$

$$d_{s,t} \geq 0, \quad s, t \in Q, s \neq t \quad (4.38)$$

$$\beta_e = \max_{\mathbf{D}} \sum_{\pi \in \Pi} \gamma_{\pi}^e f_{\pi} \cdot d_{s(\pi),t(\pi)} \quad (4.39)$$

s.t.

$$d_{s,t} \leq b^+(s), \quad s, t \in Q \quad (4.40)$$

$$\sum_{t \in Q} d_{t,s} \leq b^-(s), \quad s \in Q \quad (4.41)$$

$$d_{s,s} = 0, \quad s \in Q \quad (4.42)$$

$$d_{s,t} \geq 0, \quad s, t \in Q, s \neq t \quad (4.43)$$

Il est clair que toute matrice $\mathbf{d} \in \mathcal{D}$ solution de (4.20)-(4.24) est aussi une solution de (4.34)-(4.38) et de (4.39)-(4.43). Il est aussi évident que les coûts optimaux α_e de (4.34)-(4.38) et β_e de (4.39)-(4.43) sont des bornes supérieures sur le coût optimal $x_e(\mathbf{f})$ de (4.20)-(4.24), c'est à dire sur la capacité à réserver sur le lien e pour la stratégie de routage \mathbf{f} . Dans ce qui suit, nous proposons un algorithme efficace pour calculer les valeurs de α_e et β_e .

Concentrons nous d'abord sur le problème (4.34)-(4.38). Vu que le routage est en mono-chemin, ce problème peut être écrit sous la forme suivante :

$$\alpha_e = \max \sum_{s \in Q} \sum_{t \in \Gamma_e^+(s)} d_{s,t} \quad (4.44)$$

s.t.

$$\sum_{t \in \Gamma_e^+(s)} d_{s,t} \leq b^+(s), \quad s \in Q \quad (4.45)$$

$$d_{s,t} \leq b^-(t), \quad s \in Q, t \in \Gamma_e^+(s) \quad (4.46)$$

$$d_{s,t} \geq 0, \quad s \in Q, t \in \Gamma_e^+(s) \quad (4.47)$$

$$d_{s,t} = 0, \quad s \in Q, t \notin \Gamma_e^+(s) \quad (4.48)$$

avec $\Gamma_e^+(s)_{s \in Q} = \{t(\pi) : \pi \in \Pi, s(\pi) = s \text{ et } \gamma_\pi^e = 1\}$. Intuitivement, $\Gamma_e^+(s)$ est l'ensemble des nœuds situés à droite du graphe biparti qui sont directement reliés au nœud source s . Il est facile de vérifier qu'une solution du problème (4.34)-(4.38) est telle que chaque nœud source envoie le maximum possible sans pour autant dépasser les capacités des nœuds récepteurs. Formellement, ceci peut être écrit de la façon suivante :

$$\alpha_e = \sum_{u \in Q} \alpha_e(u) \quad \text{où} \quad \alpha_e(u) = \min \left(b^+(u), \sum_{v \in \Gamma_e^+(u)} b^-(v) \right). \quad (4.49)$$

De la même façon, le coût optimal β_e du problème (4.39)-(4.43) est donnée par la formule suivante :

$$\beta_e = \sum_{u \in Q} \beta_e(u) \quad \text{où} \quad \beta_e(u) = \min \left(b^-(u), \sum_{v \in \Gamma_e^-(u)} b^+(v) \right) \quad (4.50)$$

Finalement une borne supérieure $\bar{x}_e(\mathbf{f})$ sur $x_e(\mathbf{f})$ est donnée par $\bar{x}_e(\mathbf{f}) = \min(\alpha_e, \beta_e)$. On obtient ainsi très facilement une borne supérieure sur la bande-passante à réserver sur un lien, connaissant la stratégie de routage.

Nous avons évalué la qualité de cette borne supérieure expérimentalement. Nous avons effectué 10000 tests sur des réseaux de transport générés aléatoirement (figure 4.7). Nous avons comparé l'écart entre le coût exact obtenu avec la méthode du Stepping Stone et les valeurs de la borne supérieure. Comme on peut le constater sur la figure 4.7(a), l'écart moyen ne dépasse pas 0.013%, valeur maximale obtenue pour 50 nœuds (c'est à dire 50 nœuds à droite et 50 nœuds à gauche dans le graphe biparti de transport). La comparaison entre les temps de calcul des deux méthodes est présentée sur la figure 4.7(b). Dans cette dernière figure, "ub Algorithm" représente l'algorithme de calcul de la borne supérieure. On peut voir que notre borne supérieure est très proche de la valeur exacte de la réservation. On peut voir aussi que le temps de calcul nécessaire pour calculer cette borne supérieure est négligeable par rapport à celui nécessaire au calcul du coût exact avec l'algorithme du Stepping Stone.

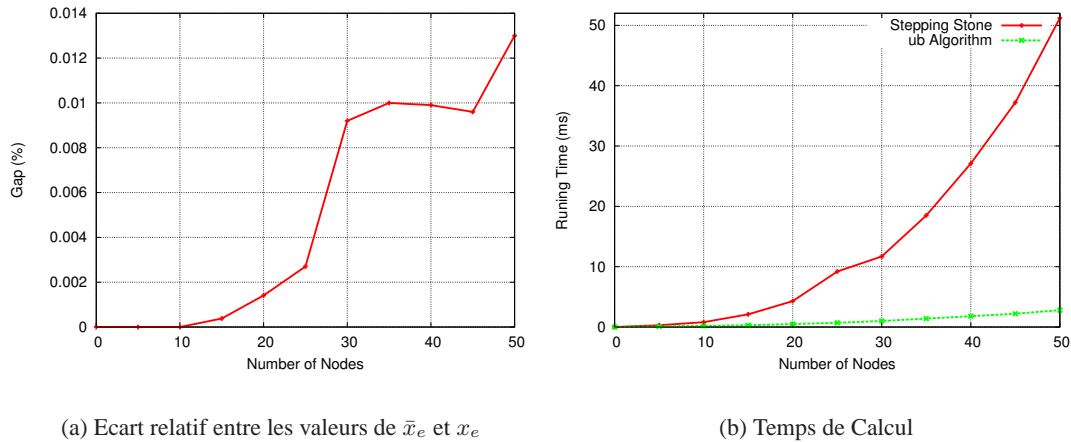


FIG. 4.7 – Tests sur la qualité de la borne supérieure.

La borne supérieure $\Psi(\mathbf{f}) = \sum_{e \in E} c_e \cdot \bar{x}_e(\mathbf{f})$ sur le coût peut ainsi être calculée très rapidement par rapport au coût exact $\Phi(\mathbf{f})$. De plus, $\Psi(\mathbf{f})$ est en général une très bonne approximation de $\Phi(\mathbf{f})$, tout en étant toujours supérieure. Ainsi, cette borne supérieure permet de trouver rapidement une solution améliorante dans le voisinage de la solution courante \mathbf{f}_k . En effet, si l'on détermine une solution $\mathbf{f} \in \mathcal{N}(\mathbf{f}_k)$ telle que $\Psi(\mathbf{f}) < \Phi(\mathbf{f}_k)$, alors on peut affirmer que $\Phi(\mathbf{f}) < \Phi(\mathbf{f}_k)$, et donc que la transition de \mathbf{f}_k vers $\mathbf{f}_{k+1} = \mathbf{f}$ va permettre de faire décroître le coût.

4.4 Minimisation de l'utilisation maximale des interfaces

Si le critère qui correspond à la bande-passante totale réservée est évidemment important pour l'opérateur, cette fonction coût a un inconvénient majeur. Elle conduit très souvent à une mauvaise utilisation des ressources existantes : suivant les demandes en trafic, certains liens du réseau sont utilisés à 100 %, tandis que d'autres liens sont sous-utilisés.

Pour l'expliquer, il est intéressant de faire le parallèle avec le même problème lorsque l'on connaît la matrice de trafic. Clairement, si l'opérateur veut minimiser la bande-passante totale réservée, il lui suffit de router les liens virtuels au plus court chemin (en nombre de sauts). Cependant, on sait bien qu'une telle approche conduit très souvent à une mauvaise utilisation des ressources existantes : suivant les demandes en trafic, certains liens du réseau sont utilisés à 100 %, tandis que d'autres liens sont sous-utilisés. Le même phénomène se produit lorsqu'il y a incertitude sur la demande en trafic si l'on utilise la bande-passante totale réservée comme critère d'optimisation.

Une utilisation efficace des ressources existantes est capitale pour fournir un service stable à un prix raisonnable. La figure 4.8 illustre cette nécessité. Les résultats représentés sur cette figure sont fournis par le logiciel NEST. On remarque que les ressources du réseau sont largement suffisantes étant donné que l'utilisation moyenne des interfaces est très faible (moins de 6%). Avec une interface chargée à

plus de 70%, la seconde plus chargée étant à moins de 40%, le routage proposé engendre une mauvaise utilisation de ces ressources et donc une QoS médiocre.

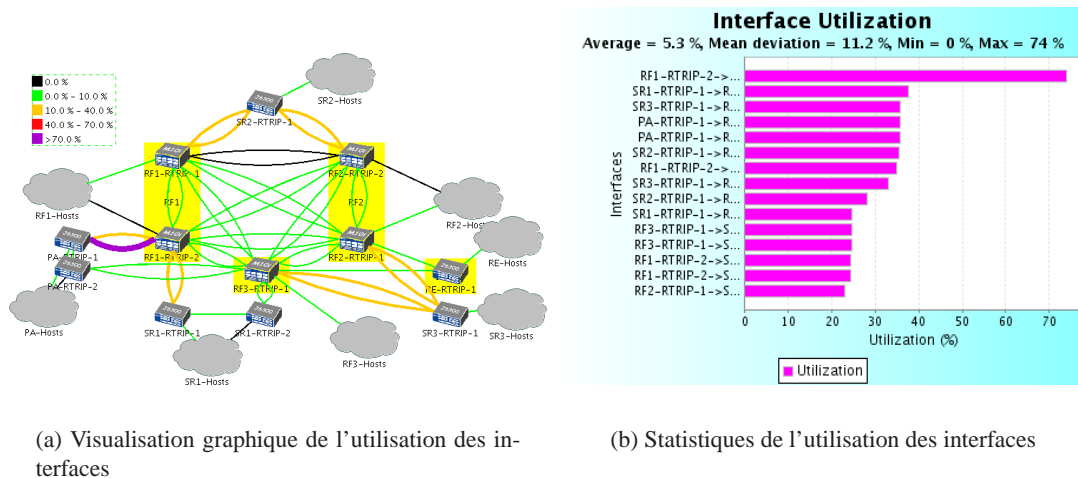


FIG. 4.8 – Mauvaise utilisation des ressources

Les fournisseurs de service VPN cherchent souvent à avoir une répartition équilibrée des capacités réservées entre les liens en proportion de leurs capacités. Dans ce contexte, une bonne mesure de l'équité d'un routage est le taux maximal de réservation, c'est à dire le maximum sur tous les liens du ratio entre bande-passante réservée et capacité. Dans la suite, nous utiliserons le taux maximal de réservation comme critère d'optimisation. Pour résoudre ce problème, nous proposons d'adapter la méthode heuristique proposée dans les sections précédentes.

4.4.1 Modèle et formulation

Nous considérons le problème de la conception optimale des VPN Hose avec un plan de routage en mono-chemin, en prenant le taux maximal de réservation comme critère d'optimisation. Comme le critère qui correspond à la bande-passante totale réservée est important pour les opérateurs, qui cherchent souvent à ne pas gaspiller de la bande passante, nous proposons d'intégrer ce critère dans notre modèle.

Soit $\rho_e = x_e/C_e$ le taux de réservation du lien e et $\rho = \max_{e \in E} \rho_e$ le taux maximal de réservation sur les liens du réseau. L'objectif est de déterminer un plan de routage monochemin minimisant le coût total ρ , tout en garantissant que la bande-passante réservée sur chaque lien $e \in E$ ne dépasse pas un seuil M_e fixé par l'opérateur, et ce quelle que soit la matrice de trafic $D \in \mathcal{D}$. Le problème peut être formulé sous la forme d'un programme linéaire mixte :

$$\min_{\mathbf{f}} \rho \quad (4.51)$$

s.t.

$$\frac{x_e}{C_e} \leq \rho \quad e \in E \quad (4.52)$$

$$x_e \leq M_e \quad e \in E \quad (4.53)$$

$$\sum_{u,v \in Q} f_{u,v}^e d_{u,v} \leq x_e \quad \mathbf{D} \in \mathcal{D}, e \in E \quad (4.54)$$

$$\mathbf{f} \in \mathcal{F} \quad (4.55)$$

$$\mathbf{x} \in \mathbb{R}_+^{|E|}, \rho \in \mathbb{R}_+ \quad (4.56)$$

Les contraintes (4.52) expriment que ρ est le taux maximal de réservation. Les contraintes (4.53) garantissent que la bande-passante réservée sur chaque lien $e \in E$ n'excédera pas le seuil M_e . Les contraintes (4.54) assurent que pour toute matrice de trafic compatible avec les spécifications de bande-passante, le volume total de trafic sur le lien e n'excédera pas la bande-passante réservée x_e . La contrainte (4.55) garantit que le vecteur \mathbf{f} est un plan de routage en monochemin admissible. La contrainte (4.56) précise que les variables $x_e, e \in E$, et ρ sont des variables réelles positives.

Comme, nous l'avons précisé dans la section 4.3.2, le problème précédent est semi-infini. Cette difficulté peut être contournée en utilisant la théorie de la dualité.

Le paramètre M_e est fixé par l'opérateur du réseau. Ce paramètre permet à l'opérateur de contrôler la bande-passante réservée dans le réseau. A travers ce paramètre l'opérateur peut garantir que la bande-passante totale réservée ne dépasse pas un certain seuil. Nous proposons, à titre d'exemple, de prendre $M_e = A \cdot \hat{x}_e$, où \hat{x}_e est la réservation obtenue pour la version multi-chemin du problème (4.8)-(4.12), c'est à dire que nous relaxons la contrainte de monoroutage (4.7) en $0 \leq f_{s,t}^e \leq 1$. Ce problème se résout très facilement avec un solveur linéaire.

4.4.2 Méthode de résolution

Le problème précédent est connu pour être NP-difficile. Comme pour le problème de minimisation de la bande passante totale réservée, nous proposons d'utiliser un algorithme multi-start qui de manière répétitive utilise une méthode de recherche locale à partir de points initiaux générés aléatoirement. L'algorithme de recherche locale mis en œuvre est l'Algorithme 7. Les détails de l'algorithme sont décrits ci-dessous.

Structure du voisinage

Pour minimiser le critère correspondant à la bande passante totale réservée, nous avons défini les voisins d'une solution \mathbf{f} comme toutes les solutions possibles qu'on peut obtenir en déviant les flux

algorithm 7 Local Search procedure

Let \mathbf{f} be the initial solution and $\rho(\mathbf{f})$ its cost.
 $\mathbf{f}^* = \mathbf{f}$ Initialisation of minimum cost solution

while Convergence() = false **do**
 for $l \in E_{\max}(\mathbf{f})$ **do**
 for (s, t) such that $l \in \pi_{s,t}(\mathbf{f})$ **do**
 $U = \{e \in E : x_e(\mathbf{f} + \mathbf{e}_{s,t}^l)/C_e < \rho(\mathbf{f})\}$.
 $w_e = (x_e(\mathbf{f} + \mathbf{e}_{s,t}^l) - x_e(\mathbf{f}))/C_e, e \in U$.
 $\pi : (s, t)$ -shortest path in (V, U, \mathbf{w}) .
 if $\pi \neq \emptyset$ **then**
 Update solution \mathbf{f} and cost $\rho(\mathbf{f})$.
 end if
 end for
 end for
end while

entre les nœuds terminaux (cf. section 4.3.4). La taille du voisinage d'une solution est alors égale à $|Q| \cdot (|Q| - 1)$.

Pour le critère correspondant au taux maximal de réservation, on peut faire nettement mieux et définir un voisinage plus restreint. Pour ce faire, nous définissons l'ensemble de liens :

$$E_{\max}(\mathbf{f}) = \{e \in E : \rho_e(\mathbf{f}) = \rho(\mathbf{f})\},$$

comme l'ensemble des liens ayant un taux de réservation maximal.

Puisque $\rho(\mathbf{f}) = \rho_e(\mathbf{f})$ pour tout lien $e \in E_{\max}(\mathbf{f})$, nous ne pouvons espérer réduire le taux maximal de réservation du réseau sans modifier le routage des flots passant par les liens de $E_{\max}(\mathbf{f})$. C'est l'idée de base de l'algorithme. Pour réduire le taux de réservation de la solution \mathbf{f} , il faut se focaliser sur les liens $l \in E_{\max}(\mathbf{f})$. Etant donné un de ces liens, disons l , pour réduire son taux de réservation il faut dévier au moins un lien virtuel de ce lien. Supposons que (s, t) est un tel lien virtuel, c'est à dire que $l \in \pi_{s,t}(\mathbf{f})$. Si on peut trouver un autre chemin pour ce lien virtuel tel que, quand le lien virtuel sera routé dessus, tous les liens auront un taux de réservation strictement inférieur à $\rho(\mathbf{f})$, alors il est possible que le taux de réservation $\rho_l(\mathbf{f})$ du lien l soit réduit.

Sur la base de cette idée, le voisinage de \mathbf{f} sera restreint aux solutions associées aux flots passant par les liens de $E_{\max}(\mathbf{f})$. Ainsi, le taille du voisinage de la solution \mathbf{f} est égale au nombre de source-destination ayant des chemins passant par les liens de $E_{\max}(\mathbf{f})$.

Pour trouver un chemin alternatif pour un lien virtuel (s, t) , il faut vérifier que quand le lien virtuel sera routé dessus, tous les liens auront un taux de réservation strictement inférieur à $\rho(\mathbf{f})$. L'ensemble $U_{s,t}$ des liens qui seront considérés lors de la résolution du problème de plus court chemin entre s et t

est alors restreint à l'ensemble :

$$U_{s,t} = \left\{ e \in \Pi_{s,t} : \frac{x_e(\mathbf{f} + \mathbf{e}_{s,t}^l)}{C_e} < \rho(\mathbf{f}) \right\}, \quad (4.57)$$

où $\mathbf{e}_{s,t}^l$ est le vecteur $(0, \dots, 1, \dots, 0)$ avec un 1 en position (s, t, e) et des 0 partout ailleurs. L'ensemble $U_{s,t}$ regroupe donc tous les liens e tels que si le lien virtuel (s, t) est routé sur e (en gardant le routage des autres liens virtuels inchangés), alors le taux de réservation résultant sera strictement inférieur aux taux maximal de réservation du réseau.

Finalement, les poids à affecter aux liens de $U_{s,t}$, pour faire le calcul des plus courts chemins dans le graphe réduit $(V, U_{s,t})$, sont définis de la manière suivante :

$$w_e = \frac{x_e(\mathbf{f} + \mathbf{e}_{s,t}^l) - x_e(\mathbf{f})}{C_e}, \quad e \in U_{s,t} \quad (4.58)$$

Choix du voisin

L'algorithme utilise la technique "first descent" pour le choix du voisin de la solution courante. A chaque itération, on sélectionne le premier voisin qui améliore la solution courante.

Test de convergence

L'algorithme n'assure pas forcément une décroissance monotone du coût pour pouvoir "sortir" des optima locaux. La technique mise en place n'est pas de type tabou mais simplement une exploration limitée de solutions. Le test de convergence permettant de terminer l'algorithme est basé sur trois critères :

- Nombre maximum d'itérations (Q_1),
- Nombre maximum d'augmentations successives du coût (Q_2),
- Voisinage de la solution courante vide.

4.5 Tests et résultats

Cette partie est dédiée à l'étude de l'efficacité et de la pertinence de nos approches et des algorithmes associés. Nous proposons donc de débiter cette partie avec l'expérimentation des méthodes que nous avons développées pour optimiser le critère bande-passante totale réservée. Puis, nous donnerons des résultats obtenus pour le critère utilisation maximale des interfaces.

Scenario	$ V $	$ Q $	Ecart MS_{bp} /BorneInf (%)	Ecart Exact/Borne-Inf (%)
1	10	7	13.76	12.68
2	15	10	24.67	22.61
3	20	12	24.31	-
4	25	16	5.4	-
5	30	20	1.87	-
5	40	24	10.08	-

TAB. 4.2 – Ecarts relatifs en % entre le coût de l’heuristique et la borne inférieure et entre le coût optimal et la borne inférieure

Tous les tests ont été effectués sur un processeur Pentium Centrino 2.8 Ghz, fonctionnant sous un système d’exploitation Linux avec 2 Go de mémoire disponible. Nos méthodes heuristiques d’optimisation ont été implémentées en C++ et nous avons utilisé le solveur CPLEX 8.11 pour résoudre les problèmes linéaires ainsi que les problème liniaire mixtes. Nous avons fixé un temps limite de calcul de 10h pour CPLEX 8.11 ; au delà de cette limite, CPLEX 8.11 est interrompu.

4.5.1 Minimisation de la bande-passante totale réservée

Dans ce paragraphe, nous noterons par MS_{bp} notre méthode heuristique pour l’optimisation de la bande-passante totale réservée. Afin d’évaluer notre heuristique, nous avons effectué des tests sur des réseaux ayant des caractéristiques fréquemment observées dans la pratique. Pour ces topologies, la solution du problème du routage en multi-chemin n’est pas une solution en mono-chemin. Le nombre de nœuds de ces réseaux varie entre 10 et 40, et pour chaque instance, cinq topologies de test sont générées. Les valeurs des trafics en entrée et en sortie varient entre $1Mb/s$ et $50Mb/s$. Nous avons supposé que les coûts d’utilisation des liens sont unitaires ($c_e = 1, e \in E$). D’autre part, pour chaque instance, le nombre de points initiaux générés aléatoirement (le paramètre N_{max}) est égal à 50.

Les résultats de calcul sont résumés dans les tableaux 4.2 et 4.3, qui incluent les informations suivantes :

- La taille des instances,
- Ecart MS_{bp} /BorneInf : L’écart relatif en % entre le coût obtenu avec notre heuristique et la borne inférieure sur le coût optimal obtenue en relaxant la contrainte de monoroutage. Rappelons que cette borne inférieure peut être obtenue en temps polynomial,
- Ecart Exact/BorneInf : L’écart relatif en % entre le coût optimal et la borne inférieure sur ce coût,
- Les temps calcul en secondes pour calculer les valeurs de la borne inférieure, le coût de l’heuristique et le coût optimal (on met le symbole ∞ quand le seuil de temps est dépassé).

On peut voir de ces tableaux que pour des réseaux de taille entre 10 et 15 nœuds, notre algorithme MS_{bp} génère des solutions situées à 1 ou 2 % de l’optimum. Pour ces réseaux les temps de calcul des solutions optimales sont considérables.

Scenario	$ V $	$ Q $	MS_{bp}	Exact
1	10	7	8.2	1124
2	15	10	38	2140
3	20	12	87	∞
4	25	16	179	∞
5	30	20	487	∞
5	35	22	954	∞
6	40	24	1378	∞

TAB. 4.3 – Temps calcul en secondes.

Instance	$ V $	$ E $	$ Q $
arpanet	24	100	10
abovenet	19	69	12
bhvac	19	46	11
eon	19	74	15
example	10	32	10
metro	11	84	9
nsf	8	21	8
pacbell	15	42	10

TAB. 4.4 – Description des Topologies

On peut voir aussi que les temps de calcul requis pour trouver les solutions optimales, augmentent exponentiellement avec la taille du problème. Au delà de 15 nœuds, le calcul du coût optimal par une méthode exacte prend un temps non raisonnable. L'utilisation de la méthode MS_{bp} proposée permet alors d'obtenir des solutions proches des valeurs exactes avec des temps de calcul réduits.

4.5.2 Minimisation de l'utilisation maximale des interfaces

Dans la suite, nous noterons par MS_{um} l'algorithme multi-start que nous avons proposé pour minimiser l'utilisation maximale des liens. Pour justifier les performances de l'algorithme MS_{um} , nous donnons ici les résultats obtenus sur des topologies réelles de réseau de tailles moyennes à grandes. Les informations sur les topologies ont été collectées soit à partir de la littérature IEEE (topologies *bhvac*, *pacbell*, *eon*, *metro*, *arpanet*), soit à partir du projet Rocketfuel [77] pour les topologies *abovenet* et *vnsf*. Pour ces réseaux, nous avons défini manuellement les nœud terminaux, les capacités des liens et les bornes sur les trafics ingress et egress des routeurs d'extrémités. Les topologies sont décrites dans le tableau 4.4.

Instance	Opt	pCALC	Min_Band	MS_um	MS_um
				$N_{\max} = 1$	$N_{\max} = 50$
arpanet	70	70	100	70	70
abovenet	40	102	100	74	59
bhvac	54	80	100	73	73
eon	62	73	100	64	64
example	56	89	100	74	74
metro	71	91	100	68	64
nsf	61	74	100	63	63
pacbell	41	75	100	52	50

TAB. 4.5 – Taux de réservation maximal (%).

Pour chaque instance, nous avons utilisé le solveur CPLEX 8.11 pour calculer le coût de la solution optimale. Nous avons également calculé les coûts des solutions générées par les algorithmes suivants :

- notre algorithme MS_{um} d’optimisation de l’utilisation maximale des liens,
- une méthode exacte Min_Band de résolution du problème de minimisation de la bande passante totale réservée. A ce propos, nous avons utilisé CPLEX 8.11 comme méthode exacte de résolution de ce problème,
- l’algorithme PCALC de Cisco. Cet algorithme propose une implantation du protocole OSPF-TE pour la distribution des labels et l’établissement automatique des LSP dans le réseau. Ce protocole, comme OSPF, est basé sur la notion de plus courts chemins au sens des métriques associées au réseau. L’algorithme PCALC se base sur la notion de bande passante résiduelle associée à un chemin qui correspond au minimum des bandes passantes résiduelles des liens qui le composent.

Le tableau 4.5 présente les valeurs du taux maximal d’utilisation des liens obtenus pour les 8 topologies de réseau. On peut constater que, dans tous les cas, notre algorithme surpasse nettement les algorithmes PCALC et Min_Band. Les solutions générées par notre algorithme sont les plus proches des solutions optimales. En moyenne, l’écart entre le taux de réservation maximal généré par notre heuristique avec $N_{max} = 50$ et celui des solutions optimales est de 11,2%. Nous remarquons aussi qu’il n’y a souvent aucun avantage à utiliser de nombreux points de départ au lieu d’un seul.

Dans le tableau 4.6, nous présentons les écarts relatifs entre la bande passante totale réservée dans les solutions de chacun des algorithmes, et celles générées par l’algorithme Min_Band (la réservation optimale). Pour notre algorithme, nous avons fixé un seuil de réservation M_e sur chaque lien e égal à $150\% \cdot \hat{x}_e$, où \hat{x}_e est la réservation obtenue pour la version multi-chemin du problème (4.13)-(4.19). On peut constater que l’optimisation du taux maximal de réservation pénalise le critère bande-passante totale réservée. Notre algorithme permet de faire un bon compromis entre les deux critères.

Les temps de calcul sont reportés dans le tableau 4.7. Les valeurs sont exprimées en secondes. Les résultats montrent que pour la plupart des topologies il faut plusieurs heures de temps de calcul pour déterminer la solution exacte, ce qui n’est manifestement pas acceptable pour un fournisseur de

Instance	Opt	pCALC	MS_um
			$N_{\max} = 1$
arpanet	98	65	24
abovenet	87	57	18
bhvac	102	82	21
eon	89	71	31
example	106	69	16
metro	91	41	27
nsf	101	54	23
pacbell	88	45	22

TAB. 4.6 – Bande-passante totale réservée(%)

Instance	CPLEX	MS_um	MS_um
		$N_{\max} = 1$	$N_{\max} = 50$
arpanet	34600	0.82	254
abovenet	∞	0.51	70
bhvac	∞	0.54	39
eon	∞	0.43	31
example	∞	0.1	6
metro	∞	0.31	46
nsf	5691	0.12	3
pacbell	∞	0.11	21

TAB. 4.7 – Temps de calcul (sec).

services VPN. Puisque ces temps de calcul augmentent exponentiellement avec la taille du problème, l'utilisation d'une méthode exacte est inappropriée pour des réseaux de grandes tailles. En revanche, notre algorithme est très rapide et propose toujours des solutions proches des valeurs exactes avec des temps de calcul raisonnables. Nous pouvons voir que le temps de calcul maximal qu'on obtient est de 254 sec. Cette durée est manifestement négligeable devant celle requise par la méthode exacte (34600 sec).

4.6 Conclusion

L'optimisation des VPN Hose est un sujet qui a été traité par de nombreux chercheurs mais qui reste tout de même un problème relativement récent car les premiers travaux de Kumar [61] datent de 2002. De nombreuses approches ont été proposées et nous avons présenté au début de ce chapitre un résumé rapide des techniques les plus connues.

Nous avons développé dans ce cadre une méthode basée sur la technique de recherche locale pour l'optimisation des VPN Hose avec contraintes de mono-routage. Nous avons considéré deux critères d'optimisation : (1) la bande-passante totale réservée et (2) l'utilisation maximale des interfaces.

La méthode proposée fournit des résultats intéressants. Une comparaison des résultats obtenus avec ceux souvent utilisées par les opérateurs, notamment PCALC, a montré une amélioration nette de la qualité de la solution pour des temps de calcul très courts.

Notre approche s'applique dans le cadre de réseaux avec un plan de routage explicite. C'est le cas des réseaux MPLS qui sont largement utilisés de nos jours. Cependant, certains opérateurs gardent jusqu'à aujourd'hui leurs infrastructures IP (par exemple OSPF). L'optimisation du routage dans ces réseaux sous la contrainte d'incertitude de la demande est une perspective intéressante à développer. C'est l'objectif du prochain chapitre.

CHAPITRE 5

Optimisation robuste du routage IP avec incertitude sur la demande

5.1 Introduction

Dans le chapitre précédent, nous avons étudié le problème de planification des réseaux (VPN) sous l'hypothèse que chaque flux est routé sur un seul chemin et que le routage est explicite. C'est le cas des réseaux basés sur la technologie MPLS qui sont de nos jours largement répandus. Les opérateurs qui utilisent cette technologie peuvent spécifier explicitement le placement des flux et réserver de la bande-passante pour les chemins (ou LSP) associés aux flux en utilisant le protocole RSVP.

Cependant, jusqu'à aujourd'hui, certains opérateurs n'ont pas totalement abandonné leurs infrastructures IP pour migrer vers la technologie MPLS. Une problématique clé pour ces opérateurs est celle de l'optimisation du routage dans les réseaux IP, qui permet une utilisation plus efficace des ressources existantes en adaptant le routage aux trafics supportés par le réseau. Il est cependant bien connu que, à cause du principe même du routage dans ces réseaux, le problème d'optimisation du routage, connu sous le nom d'optimisation des métriques IP, est un problème complexe à résoudre.

5.1.1 Optimisation des métriques de routage IP

Le succès phénoménal de l'Internet au cours de la dernière décennie – bien au delà des espérances initiales des pionniers de l'Internet – en fait aujourd'hui un élément incontournable de l'infrastructure mondiale de communication. Ce succès va encore s'amplifier dans les années à venir avec le développement très rapide de nouvelles applications, comme par exemple les applications de grid computing et de SaaS (Software as a Service), ou encore celles liées aux réseaux pair-à-pair ou aux réseaux sociaux.

Les opérateurs réseaux doivent adapter régulièrement leur infrastructure de communication de manière à faire face à l’augmentation du trafic tout en satisfaisant des exigences de qualité de service de plus en plus forte. Dans un contexte fortement concurrentiel, une alternative efficace à un surdimensionnement excessif est d’optimiser le routage du réseau de manière à éviter les phénomènes de congestion et les dégradations de service résultantes. L’optimisation du routage permet une meilleure utilisation des ressources existantes en adaptant les routes utilisées aux conditions de trafic.

Open Shortest Path First (OSPF) et Intermediate System to Intermediate System (IS-IS) sont les deux protocoles de routage intra-domaine Internet les plus utilisés [42, 24]. Les flux de trafic sont routés le long de plus courts chemins, en partageant équitablement le trafic quand un noeud a plusieurs liens qui sont sur des plus courts chemins vers la destination. Les poids des liens, et donc les plus courts chemins, peuvent être changés par l’opérateur. Ces poids peuvent être fixés à 1 ou bien proportionnels aux distances physiques, mais la pratique usuelle recommandée par Cisco consiste à prendre des poids inversement proportionnels aux capacités des liens. Bien que parfois efficaces, ces heuristiques conduisent souvent à une utilisation assez pauvre des ressources.

Etant donné un ensemble de demandes entre des couples origine/destination (OD), le problème d’optimisation du routage IP consiste à déterminer un ensemble de métriques à affecter aux liens qui optimisent une certaine mesure de performance. Ce problème est donc assez différent du problème de routage traditionnel dans lequel il n’y a aucune restriction sur la structure des routes utilisées. Depuis son introduction par Fortz et Thorup [56, 78], de nombreuses études ont été dédiées à ce problème d’optimisation du routage IP.

5.1.2 Incertitude sur la demande

Dans la définition ci-dessus du problème d’optimisation du routage IP, nous avons supposé que le trafic offert au réseau est connu ou peut être mesuré. Toutefois, en pratique, les demandes entre les noeuds du réseau ne peuvent être mesurées que de manière imprécise, et de plus elles changent continuellement. Concevoir un réseau sur la base d’une unique matrice de trafic en “heure de pointe” est de moins en moins crédible du fait du caractère de plus en plus volatil du trafic Internet. Une telle approche peut conduire à une mauvaise utilisation des ressources du réseau si à un moment donné la véritable matrice de trafic du réseau diverge significativement de celle utilisée pour l’optimisation du routage. Une approche bien connue permettant de faire face aux variations temporelles de la matrice de trafic consiste à s’appuyer sur des mesures de trafic en temps réel et à adapter dynamiquement le routage quand des changements sont observés. Néanmoins, ces mises à jour distribuées du routage peuvent conduire à une certaine complexité et même engendrer des instabilités du réseau.

Une approche hors ligne alternative qui a été récemment proposée consiste à optimiser sur un ensemble de matrices de trafic possibles [82, 102, 100, 101]. Cet ensemble, qui est souvent un polyèdre, doit contenir toutes les réalisations possibles des demandes aléatoires en trafic, ou à tout le moins les plus vraisemblables. L’objectif est alors de déterminer un routage garantissant les meilleures performances possibles quelle que soit la réalisation de la matrice de trafic dans cet ensemble. Un tel routage est qualifié de *oublieux* dans la mesure où il n’est pas déterminé par rapport à une matrice de trafic spécifique. Les méthodes permettant d’obtenir ce routage oublieux relèvent de l’optimisation robuste

dans la mesure où elles intègrent l'incertitude sur des demandes en trafic inconnues et variables dans le temps. La plupart des travaux intégrant l'incertitude sur le trafic ont été consacrés au problème de conception des VPN avec le très célèbre *hose model* d'incertitude sur la demande introduit dans le chapitre précédent, ou au problème de routage traditionnel dans le cadre d'un ensemble d'incertitude polyédral. Le problème d'optimisation robuste du routage IP n'a été considéré que récemment.

5.1.3 Contribution

Les travaux précédents sur l'optimisation robuste du routage IP supposent une incertitude sur la demande polyédrale, mais ne font aucune autre hypothèse sur la structure de l'ensemble d'incertitude. Ceci a l'avantage de permettre de prendre en compte toute définition polyédrale de l'incertitude sur la demande. Toutefois, cette approche a l'inconvénient de laisser ouverte la question de ce que doit être réellement un ensemble d'incertitude ayant du sens d'un point de vue pratique. De plus, l'hypothèse d'un ensemble polyédral d'incertitude est très générale, et ne permet pas de s'appuyer sur une structure spécifique pour réduire les temps de calcul. Comme nous l'expliquons ci-dessous, les techniques d'estimation de matrice de trafic peuvent être utilisées pour obtenir des bornes supérieures et inférieures sur le volume de trafic de chaque flot OD, de même que des bornes supérieures sur le trafic total entrant et sortant de chaque noeud aux extrémités du réseau. Le modèle d'incertitude résultant est une combinaison de deux modèles précédemment proposés : le *box model* de Belotti et al. [97] et le *hose model* de Duffield et al. [49]. Il fournit une structure concrète d'incertitude sur la demande qui a du sens pour les opérateurs réseaux. De plus, en adaptant nos algorithmes à cette structure spécifique d'incertitude, nous sommes capables d'obtenir une réduction considérable des temps de calcul.

L'algorithme d'optimisation du routage IP proposé dans ce chapitre adapte les techniques de recherche locale proposées dans [83] à cette structure spécifique d'incertitude sur la demande. Il y a plusieurs avantages par rapport aux travaux mentionnés ci-dessus. Le premier concerne la versatilité de l'algorithme obtenu, qui peut être utilisé aussi bien en mode incrémental, permettant ainsi d'améliorer le routage existant avec quelques modifications, qu'en mode multi-start pour une optimisation globale des métriques de routage qui permet à l'algorithme de s'extraire des minima locaux. De nombreux résultats expérimentaux montrent que cet algorithme fournit des solutions qui sont souvent assez proches de la borne inférieure obtenue avec un routage multi-chemin robuste, et ce dans des temps de calcul très inférieurs à ceux des approches concurrentes proposées dans la littérature.

5.1.4 Organisation de ce chapitre

Dans le prochain paragraphe, nous présentons un état de l'art détaillé sur ce problème. Le paragraphe 5.3 est consacré à la formulation mathématique du problème d'optimisation robuste des métriques. L'algorithme proposé est décrit au paragraphe 5.4, tandis que les résultats expérimentaux sont présentés au paragraphe 5.5.

5.2 Etat de l'art

Dans ce paragraphe, nous présentons une synthèse des travaux antérieurs sur le problème d'optimisation des métriques de routage IP, sur l'estimation de matrice de trafic, et sur l'optimisation robuste du routage.

5.2.1 Optimisation des métriques de routage avec des demandes en trafic connues

Le problème traditionnel d'optimisation des métriques de routage IP suppose que la demande en trafic est connue pour chaque flot OD et vise à déterminer les métriques des liens, et donc les routes utilisées, de manière à optimiser un certain critère de performance (cf. [103] pour un état de l'art détaillé et certaines extensions au problème de base). Ce problème est connu pour être NP-difficile [56]. Bien qu'une formulation sous forme de programme linéaire en nombres entiers soit possible [88], elle ne permet pas une résolution exacte du problème en des temps de calcul raisonnables. En conséquence, la plupart des travaux effectués sur ce problème se sont focalisés sur des algorithmes d'approximation. On peut distinguer deux grandes catégories d'algorithmes : ceux qui tentent de résoudre un problème de plus court chemin inverse [52, 54, 55, 33, 64, 72], et ceux qui, partant d'une solution initiale, c'est à dire d'un ensemble de métriques initiales, essayent de l'améliorer itérativement [56, 65, 73, 93, 60, 83, 84]. Comme déjà mentionné, dans ce chapitre nous allons nous baser sur l'algorithme de recherche locale proposé dans [83] et l'étendre à notre structure spécifique d'incertitude sur la demande.

5.2.2 Estimation de matrice de trafic

Les méthodes traditionnelles d'ingénierie du trafic supposent que la matrice de trafic du réseau est connue. Malheureusement, en pratique, elle s'avère très difficile à mesurer directement le trafic car les outils de métrologie actuels (par exemple Netflow [104]) engendrent une charge de calcul importante sur les routeurs et génèrent un trafic de métrologie significatif. Certaines tentatives ont été menées pour réduire ces surcoûts en utilisant des techniques d'échantillonnage [105]. Une approche alternative consiste à utiliser les mesures de trafic agrégé fournies pour chaque lien par le protocole SNMP pour retrouver les véritables demandes en trafic. Cependant, comme il y a en général beaucoup plus de flots OD dans un réseau que de liens, cela conduit à un problème inverse mal posé qui ne peut être résolu sans information supplémentaire. Après le travail précurseur de Vardi [43], une première génération de méthodes a tenté d'utiliser la covariance sur la charge des liens en tant qu'information supplémentaire [66, 53, 89, 90]. Ces méthodes sont basées sur des hypothèses statistiques sur le trafic dont la validité est parfois contestable. Une seconde génération de méthodes utilisent une information spatiale ou temporelle *à priori* [74, 85, 79, 94]. La plus connue et la plus simple de ces méthodes est celle de tomogravité [74] qui suppose que le trafic entre deux noeuds i et j est proportionnel au trafic total entrant au noeud i et au trafic total sortant au noeud j . Comme pour la première génération de méthodes, l'inconvénient de ces approches est lié à l'information *à priori* qui est utilisée et qui ne permet pas de garantir la validité de la matrice de trafic obtenue après estimation. Pour dépasser cette difficulté, les méthodes les plus récentes utilisent une information spatiale ou temporelle *a posteriori*. Elles nécessitent une phase de calibration pendant laquelle la matrice de trafic du réseau est régulièrement mesurée en utilisant un outil comme Netflow et utilisent ensuite des techniques de filtrage pour prédire l'évolution de la matrice de trafic [86, 80]. Bien que la précision de ces méthodes soit significativement supérieures à celles des

méthodes précédentes, leur inconvénient est le surcoût induit sur le réseau pendant la phase de calibration.

En utilisant n'importe lequel des algorithmes d'estimation de matrice de trafic ci-dessus, on peut facilement obtenir des bornes sur les fluctuations au cours du temps de la demande de chaque flot OD autour de sa valeur moyenne. Il suffit pour cela de prendre les valeurs minimales et maximales du volume de trafic de ce flot estimées durant la phase de métrologie. Bien sûr, ces bornes inférieures et supérieures se doivent d'intégrer les imprécisions liées à la méthode d'estimation. De plus, l'opérateur est libre d'adopter une approche plus prudente en diminuant légèrement la borne inférieure et en augmentant légèrement la borne supérieure de manière à se protéger contre des variations plus fortes de la matrice de trafic.

Enfin, des bornes supérieures sur le volume de trafic reçu et émis par chaque routeur d'extrémité peuvent être facilement obtenues en utilisant les mesures SNMP. Là encore, il suffit de prendre les valeurs maximales des trafics ingress et egress des routeurs d'extrémités obtenues avec SNMP pendant la phase de métrologie.

Notre modèle d'incertitude va donc combiner une borne inférieure et une borne supérieure sur la demande de chaque flot OD, avec des bornes supérieures sur les trafics ingress et egress des noeuds d'extrémités. On peut voir ce modèle comme une combinaison du modèle d'incertitude de type intervalle introduit dans [97] avec le célèbre hose model d'incertitude sur la demande de [49]. Notre modèle généralise ces deux modèles d'incertitude.

5.2.3 Optimisation robuste du routage

Comme nous l'avons précisé dans le chapitre précédent, un des modèles d'incertitude sur la demande les plus connus est le hose model [49] introduit par Duffield et al. pour la conception de VPN [91, 59, 69, 61, 95, 76]. Ce modèle donne le trafic maximal entrant et sortant en chaque noeud terminal du VPN. L'ensemble d'incertitude est alors constitué de toutes les matrices de trafic dont la somme par ligne est inférieure à la bande passante entrante du noeud source correspondant, et dont la somme par colonne est inférieure à la bande passante sortante du noeud destination correspondant.

Certains auteurs ont également travaillé sur le problème de routage traditionnel en intégrant l'incertitude sur la demande. Applegate et Cohen [70] traitent ce problème en faisant très peu d'hypothèses sur les demandes en trafic. Belotti et Pinar [97] incorporent au problème une incertitude sur la demande de chaque flot OD de type intervalle (*box model*), ainsi qu'une incertitude statistique. Ben-Ameur et Kerivin [81] étudient le problème de routage multi-chemin à moindre coût en supposant un polytope d'incertitude sur la demande et proposent un algorithme basé sur la génération itérative de chemins et de contraintes pour le résoudre.

Le problème d'optimisation robuste des métriques de routage IP n'a été considéré que récemment. Chu et Lea traitent ce problème pour des réseaux IP supportant des VPN conçus suivant le hose model [100]. Mulyana et Killat [82] considèrent le problème d'optimisation des métriques avec des contraintes sur le trafic sortant. Altin et al. [102] étudient le routage OSPF avec incertitude polyédrale sur la de-

mande et proposent une formulation en programmation linéaire mixte ainsi qu'un algorithme de branch-and-price pour résoudre exactement de petites instances. Dans [101], Altin et al. étendent l'algorithme tabou de [67] pour prendre en compte une incertitude polyédrale sur la demande.

Comme mentionné dans l'introduction, une des principales contributions de notre approche est liée à notre modèle d'incertitude. En exploitant ses spécificités, nous sommes capables de réduire significativement les temps de calcul nécessaires pour évaluer la charge des liens en réduisant ce problème à un problème classique de flot à coût minimal que l'on sait résoudre très efficacement.

5.3 Formulation du problème

On considère un réseau de N noeuds et M liens de communication représenté par un graphe dirigé $G = (V, E)$. On note $Q \subset V$ l'ensemble des routeurs d'extrémités. La capacité du lien $l \in E$ est C_l . Le réseau supporte un ensemble de $K = |Q|(|Q| - 1)/2$ flots OD, et chaque flot OD $k = 1, \dots, K$ est défini par son noeud source $s(k) \in Q$, son noeud destination $t(k) \in Q$ et sa demande (incertaine) en bande passante $d_{s(k),t(k)}$. Dans la suite, on notera \mathbf{d} le vecteur des demandes en trafic.

5.3.1 Structure de l'ensemble d'incertitude

Comme expliqué au paragraphe 5.2.2, notre modèle d'incertitude correspond à une combinaison d'un modèle d'incertitude de type intervalle avec un modèle de type hose. Soit $a_{s,t}$ et $b_{s,t}$ les bornes inférieure et supérieure sur la demande en trafic du flot OD $(s, t) \in Q$, respectivement. Soit b_s^{out} et b_s^{in} les volumes maximaux de trafic que peut émettre et recevoir le noeud d'extrémité $s \in Q$, respectivement. Notre modèle d'incertitude correspond au polytope \mathcal{D} des matrices de trafic \mathbf{d} satisfaisant les contraintes linéaires suivantes :

$$a_{s,t} \leq d_{s,t} \leq b_{s,t}, \quad s, t \in Q, \quad (5.1)$$

$$\sum_{t \neq s} d_{s,t} \leq b_s^{out}, \quad s \in Q, \quad (5.2)$$

$$\sum_{t \neq s} d_{t,s} \leq b_s^{in}, \quad s \in Q. \quad (5.3)$$

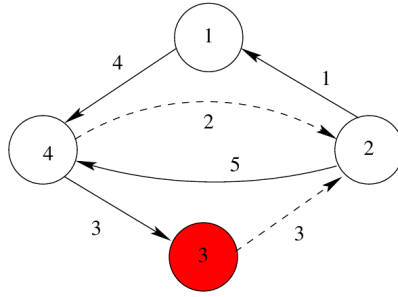
Dans la suite, nous supposons que $\sum_{s \in Q} b_s^{out} = \sum_{t \in Q} b_t^{in}$. On supposera également que $\sum_{t \neq s} b_{s,t} > b_s^{out}$ et $\sum_{s \neq t} b_{s,t} > b_t^{in}$.

5.3.2 Solutions admissibles

Une solution admissible du problème d'optimisation des métriques de routage IP est définie comme un vecteur $\mathbf{w} = (w_1, \dots, w_M)$, où $w_l \in \Omega$ est le poids affecté à l'arc $l \in E$ et $\Omega = \{1, \dots, w_{max}\}$ est l'ensemble des valeurs entières que peuvent prendre les métriques. Dans le cas du protocole OSPF on a $w_{max} \leq 2^{16} - 1$. Etant donné une solution admissible $\mathbf{w} = (w_1, \dots, w_M)$, un algorithme de plus court chemin peut être utilisé pour calculer les paramètres suivants :

- $D_u^t(\mathbf{w})$ est la distance du noeud u à la destination t ,
- $\delta_{u,v}^t(\mathbf{w}) = 1$ si le lien $l = (u, v)$ est sur un plus court chemin vers la destination t , i.e. $D_u^t(\mathbf{w}) = w_l + D_v^t(\mathbf{w})$, et 0 sinon.
- $n_u^t(\mathbf{w}) = \sum_v \delta_{u,v}^t(\mathbf{w})$ est le nombre d'arcs sortant du noeud u qui sont sur des plus courts chemins vers la destination t .

Ces paramètres résument toute l'information sur les plus courts chemins entre le noeud u et la destination t pour le vecteur de métriques \mathbf{w} . Ils sont illustrés sur la figure 5.1 dans laquelle la destination t est le noeud 3, et où les métriques sont indiquées sur les arcs du graphe.



$$\begin{aligned}
 D_1^3 &= 7, D_2^3 = 8, D_3^3 = 0, D_4^3 = 3 \\
 \delta_{1,4}^3 &= 1, \delta_{2,1}^3 = 1, \delta_{2,4}^3 = 1, \delta_{3,2}^3 = 0, \delta_{4,2}^3 = 0, \delta_{4,3}^3 = 1 \\
 n_1^3 &= 1, n_2^3 = 2, n_3^3 = 0, n_4^3 = 1
 \end{aligned}$$

FIG. 5.1 – Représentation des plus court chemins dans un graphe

En supposant que le trafic est partagé équitablement entre tous les arcs sortants qui sont sur des plus courts chemins vers la destination, la proportion du flot OD (s, t) passant sur le lien $l = (u, v)$ est alors

$$f_{s,t}^l(\mathbf{w}) = \frac{\delta_{u,v}^t(\mathbf{w})}{n_u^t(\mathbf{w})} \sum_{(i,u) \in E} f_{s,t}^{i,u}. \quad (5.4)$$

Remarquons que la relation précédente est valide quelle que soit la matrice de trafic $\mathbf{d} \in \mathcal{D}$. Elle permet le calcul récursif des proportions $f_{s,t}^{u,v}(\mathbf{w})$ pour chaque flot OD (s, t) et chaque lien $(u, v) \in E$ un fois que les paramètres $\delta_{u,v}^t(\mathbf{w})$ et $n_u^t(\mathbf{w})$ ont été calculés.

5.3.3 Fonction objectif

Considérons un vecteur de métriques $\mathbf{w} \in \Omega^M$ fixé. Le volume maximal de trafic sur le lien $l = (u, v)$ est la solution du problème suivant :

$$y_l(\mathbf{w}) = \max_{\mathbf{d} \in \mathcal{D}} \sum_{s,t \in Q} f_{s,t}^l d_{s,t}. \quad (5.5)$$

Notre objectif est de minimiser le taux de congestion du réseau, c'est à dire le taux d'utilisation du lien le plus chargé. C'est une fonction coût classique qui est largement acceptée par les opérateurs réseaux. Le coût d'une solution admissible $\mathbf{w} \in \Omega^M$ est alors le suivant :

$$\Phi(\mathbf{w}) = \max_{l \in E} \frac{y_l(\mathbf{w})}{C_l}. \quad (5.6)$$

On peut donc formuler le problème à résoudre de la manière suivante :

$$\min_{\mathbf{w} \in \Omega^M} \Phi(\mathbf{w}). \quad (5.7)$$

On notera que notre algorithme n'est toutefois pas limité à cette fonction objectif. Il peut notamment être utilisé avec des fonctions coûts additives, par exemple pour minimiser le délai moyen de transmission d'un paquet dans le réseau.

5.4 Un algorithme de recherche locale pour l'optimisation robuste du routage

Pour résoudre le problème d'optimisation robuste des métriques de routage, nous proposons d'utiliser une heuristique de recherche locale. Le pseudo-code 8 décrit l'algorithme proposé en détail.

L'originalité principale de cet algorithme est liée à la structure de voisinage utilisée et à la méthode utilisée pour évaluer les charges maximales des liens.

5.4.1 Définition du voisinage

Notons \mathbf{e}_l le M -vecteur $(0, \dots, 1, \dots, 0)$ avec un 1 en position l et des 0 partout ailleurs. Le voisinage d'une solution $\mathbf{w} = (w_1, \dots, w_M)$ est défini de la façon suivante :

$$\mathcal{N}(\mathbf{w}) = \{\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^M\}, \quad (5.8)$$

où $\mathbf{w}^l = \mathbf{w} + \Delta_l \mathbf{e}_l$, pour $l \in E$, avec

$$\Delta_l = \operatorname{argmin}_{\Delta \geq 1} \left[\sum_{s,t \in Q} f_{s,t}^l(\mathbf{w} + \Delta \mathbf{e}_l) < \sum_{s,t \in Q} f_{s,t}^l(\mathbf{w}) \right]. \quad (5.9)$$

Ainsi, le voisinage d'une solution \mathbf{w} contient au plus M solutions. Chacune est associée à un lien l et est obtenue en augmentant la métrique de ce lien de la quantité minimale permettant de dévier au moins un flot OD de ce lien (complètement ou en partie). On remarquera que cette structure de voisinage permet également de générer automatiquement les situations de partage de charge.

algorithm 8 Algorithme d'optimisation robuste des métriques OSPF.

```

1: Metric Optimisation()
2: Let  $\mathbf{w} = (w_1, \dots, w_M)$  be the initial solution, and  $\Phi(\mathbf{w})$  the cost of  $\mathbf{w}$ .
3:  $\mathbf{w}^* = \mathbf{w}$  § Initialisation de la solution de coût min .

4: while Convergence() = false do
5:    $\Phi_{min} = \infty$ 
6:   for  $l = 1 \dots M$  do
7:     Calculate  $\Delta_l$  et  $\mathbf{w}^l = \mathbf{w} + \Delta_l \mathbf{e}_l$ 
8:     Calculate the shortest paths :  $\delta_{u,v}^t(\mathbf{w}^l), n_u^t(\mathbf{w}^l)$ 
9:     Calculate the maximum load  $y_l(\mathbf{w}), l \in E$ 
10:    Calculate the cost  $\Phi(\mathbf{w}^l)$ 
11:    if  $\Phi(\mathbf{w}^l) \leq \Phi_{min}$  then
12:       $\mathbf{w}_{next} = \mathbf{w}^l$  et  $\Phi_{min} = \Phi(\mathbf{w}^l)$ 
13:    end if
14:  end for
15:   $\mathbf{w} = \mathbf{w}_{next}$ 
16:  if  $\Phi(\mathbf{w}) < \Phi(\mathbf{w}^*)$  then
17:     $\mathbf{w}^* = \mathbf{w}$  § Mise à jour de la solution de coût min.
18:  end if
19: end while

```

5.4.2 Génération du voisinage

Notons \mathcal{F}_l l'ensemble des flots OD transmis sur le lien l ($f_{s,t}^l > 0$). Si $\mathcal{F}_l = \emptyset$, alors Δ_l ne peut être défini puisqu'il est impossible de dévier du trafic de ce lien. Sinon, nous procédons de la façon suivante pour calculer Δ_l :

- (a) Fixer $\overline{\mathbf{w}}^l = (w_1, \dots, w_{i-1}, \infty, w_{i+1}, \dots, w_M)$.
- (b) Pour tout $u, t \in V$, mettre à jour les distances $D_u^t(\overline{\mathbf{w}}^l)$ en utilisant un algorithme de plus court chemin.
- (c) Soit

$$d_{\min}^l = \min_{f \in \mathcal{F}_l} \left[D_{s(f)}^{t(f)}(\overline{\mathbf{w}}^l) - D_{s(f)}^{t(f)}(\mathbf{w}) \right], \quad (5.10)$$

l'augmentation minimale de la longueur des plus courts chemins sur l'ensemble des flots $f \in \mathcal{F}_l$.

- (d) Δ_l est donné par

$$\Delta_l = \begin{cases} 1 & \text{si } d_{\min}^l = 0 \\ d_{\min}^l & \text{si } 0 < d_{\min}^l < \infty \\ \infty & \text{si } d_{\min}^l = \infty \end{cases}$$

La solution $\mathbf{w}^l = \mathbf{w} + \Delta_l \mathbf{e}_l$ dévie du trafic du lien l :

- Si $d_{\min}^l = 0$, il y avait plusieurs plus courts chemins pour au moins un des flots OD de \mathcal{F}_l (partage de charge). En augmentant la métrique w_l de $\Delta_l = 1$, ce flot sera complètement dévié du lien l (cf. figure 5.2),
- Si $0 < d_{\min}^l < \infty$, la solution \mathbf{w}^l introduit du partage de charge pour au moins un flot OD appartenant à \mathcal{F}_l (cf. figure 5.3),
- Si $d_{\min}^l = \infty$, le lien l est obligatoire pour tous les flots OD $f \in \mathcal{F}_l$ et on peut donc le supprimer de l'ensemble des liens à optimiser sans changer le problème.

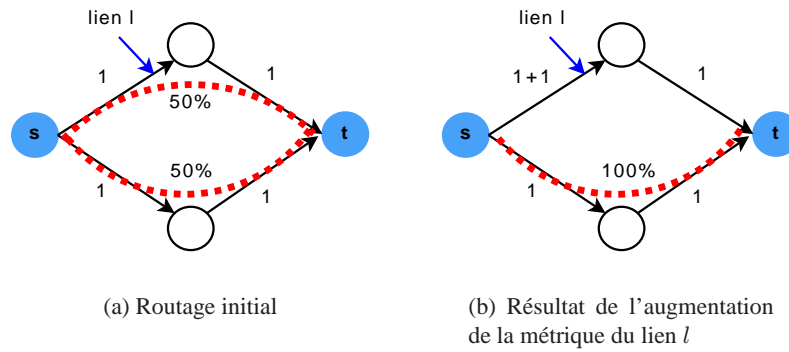


FIG. 5.2 – Exemple où $d_{\min}^l = 0$

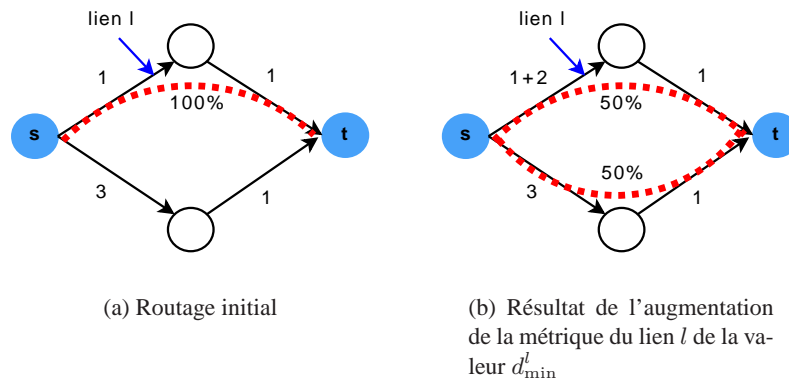


FIG. 5.3 – Exemple où $0 < d_{\min}^l < \infty$

5.4.3 Mise à jour dynamique des plus courts chemins

Deux opérations clés sont très souvent réalisées au cours de l'exploration du voisinage. La première concerne la mise à jour des plus courts chemins suite à l'augmentation de la métrique d'un seul lien. Ceci se produit quand on fixe la métrique d'un lien l à $+\infty$ pour calculer Δ_l , mais aussi quand on fixe cette métrique à $w_l + \Delta_l$ pour évaluer le coût de la solution voisine \mathbf{w}^l .

Si l'on considère un seul changement de métrique (une augmentation dans notre cas), on aura très souvent qu'une petite partie du graphe qui sera affectée. Il est donc pertinent d'éviter de recalculer l'ensemble des plus courts chemins à partir de zéro, comme c'est le cas avec l'algorithme de Dijkstra, mais au contraire de ne mettre à jour les distances que pour les nœuds du graphe affectés par le changement de métrique. Ce problème est connu sous le nom de problème des plus courts chemins dynamiques. Plusieurs algorithmes ont été proposés pour le résoudre, l'un des plus connus étant celui de Ramalingam et Reps [39]. Comme suggéré dans [56], nous avons utilisé une version améliorée de cet algorithme [68].

5.4.4 Charges maximales des liens

L'autre opération qui est très souvent réalisée concerne le calcul de la charge maximale des liens. Cette opération est réalisée à chaque fois que l'on doit évaluer le coût d'une solution voisine \mathbf{w}^m , $m \in E$. Si l'on considère un voisin \mathbf{w}^m , nous observons tout d'abord qu'il n'est pas nécessaire de recalculer la charge maximale de tous les liens, mais seulement des liens dont les paramètres $f_{s,t}^l$ ont été mis à jour. Ces derniers sont facilement identifiés durant la mise à jour dynamique des plus courts chemins. On n'a donc en général, pour évaluer un voisin, qu'à recalculer la charge maximale d'un petit nombre de liens.

Soit l un de ces liens. La charge maximale $\eta_l(\mathbf{w})$ de ce lien est la valeur optimale du problème de maximisation (5.5) avec les valeurs des coefficients de routage $f_{s,t}^l$ mises à jour. D'après nos hypothèses $\sum_{t \neq s} b_{s,t} > b_s^{out}$ et $\sum_{s \neq t} b_{s,t} > b_t^{in}$, il est clair qu'il existe au moins une solution optimale de ce problème telle que $\sum_{t \neq s} d_{s,t} = b_s^{out}$ et $\sum_{s \neq t} d_{s,t} = b_t^{in}$. Observons que ceci implique que

$$\sum_{s,t \in Q} d_{s,t} = \sum_{s \in Q} b_s^{out} = \sum_{t \in Q} b_t^{in}. \quad (5.11)$$

En ce qui concerne la valeur optimale du problème de maximisation (5.5), nous pouvons donc remplacer les contraintes d'inégalité (5.2)-(5.3) par les contraintes d'égalité suivantes :

$$\sum_{t \neq s} d_{s,t} = b_s^{out}, \quad s \in Q, \quad (5.12)$$

$$\sum_{s \neq t} d_{s,t} = b_t^{in}, \quad t \in Q. \quad (5.13)$$

Soit \mathcal{D}^* le polytope de matrices de trafic défini par les contraintes (5.1) et (5.12)-(5.13). La charge maximale du lien l peut maintenant être écrite de la façon suivante :

$$\begin{aligned}
y_l(\mathbf{w}) &= \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} f_{s,t}^l d_{s,t}, \\
&= \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} f_{s,t}^l d_{s,t} + \sum_{s \in Q} b_s^{out} - \sum_{s \in Q} b_s^{out}, \\
&= \sum_{s \in Q} b_s^{out} + \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} f_{s,t}^l d_{s,t} - \sum_{s,t \in Q} d_{s,t}, \\
&= \sum_{s \in Q} b_s^{out} + \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} (f_{s,t}^l - 1) d_{s,t}, \\
&= \sum_{s \in Q} b_s^{out} - \min_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} (1 - f_{s,t}^l) d_{s,t}.
\end{aligned}$$

Le calcul de $y_l(\mathbf{w})$ revient ainsi à résoudre le problème de minimisation suivant :

$$\left\{ \begin{array}{l} \min_{\mathbf{d} \in \mathbb{R}^K} \sum_{s,t \in Q} (1 - f_{s,t}^l) d_{s,t} \\ \text{s.t.} \\ \quad a_{s,t} \leq d_{s,t} \leq b_{s,t} \quad , \quad s, t \in V \\ \quad \sum_{t \neq s} d_{s,t} = b_s^{out} \quad , \quad s \in V \\ \quad \sum_{t \neq s} d_{t,s} = b_s^{in} \quad , \quad s \in V \end{array} \right.$$

Comme illustré sur la Figure 5.4, la structure de ce problème est celle d'un problème classique de flot à coût minimal dans un graphe biparti. Les noeuds à gauche représentent les sources de trafic tandis que les noeuds à droite représentent les destinations. Chaque noeud source s a une fourniture correspondant à b_s^{out} unités et chaque noeud destination t a une demande de b_t^{in} unités. Le coût de l'arc (s, t) est $1 - f_{s,t}^l$, sa capacité est de $b_{s,t}$ unités et la borne inférieure sur son flot est de $a_{s,t}$ unités. Ce problème peut être résolu très efficacement en utilisant une des techniques de flot à coût minimal décrites dans les chapitres 9, 10 et 11 de [32].

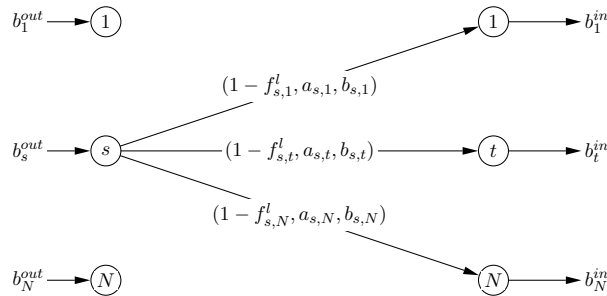


FIG. 5.4 – Problème de flot à coût minimal.

5.4.5 Test de convergence

Le test de convergence est basé sur trois critères : (a) un nombre maximal d'itérations Q_1 , (b) un nombre maximal Q_2 d'itérations sans amélioration de la meilleure solution trouvée et (c) un voisinage vide pour la solution courante ($\Delta_l = \infty, \forall l \in E$). Si $Q_2 = 0$, alors l'algorithme converge vers un minimum local, mais sinon il peut s'en échapper en sélectionnant le voisin correspondant à l'augmentation minimale du coût. L'algorithme s'arrête alors si on ne trouve pas une solution meilleure que \mathbf{w}^* en Q_2 itérations.

5.4.6 Modes incrémental et multi-start

Dans la suite, nous allons considérer plusieurs variantes de cet algorithme. Tout d'abord, l'algorithme peut être utilisé en mode "single-start" en partant des métriques existantes pour une optimisation incrémentale du routage. L'algorithme correspondant sera noté A_{INCR} ci-dessous. Ces paramètres sont $Q_1 = 100$ et $Q_2 = 5$.

L'algorithme 8 peut aussi être utilisé en mode "multi-start". Dans ce cas, l'algorithme 8 est exécuté plusieurs fois à partir de solutions initiales différentes. Les tests expérimentaux ont montré que l'on pouvait se contenter des trois solutions initiales suivantes : (a) les métriques existantes, (b) les métriques toutes égales à 1 et (c) des métriques inversement proportionnelles aux capacités des liens. On se référera à cet algorithme sous le nom de A_{MS} .

5.5 Résultats

Nous avons testé nos algorithmes d'optimisation A_{INCR} et A_{MS} sur des topologies réelles de réseau de tailles moyennes à grandes. Les informations sur les topologies ont été collectées soit à partir de la littérature IEEE (topologies *bhvac*, *pacbell*, *eon*, *metro*, *arpanet*), soit à partir du projet Rocketfuel [77] pour les topologies *abovenet* et *vnsf*. Pour ces dernières, les capacités des liens et les bornes sur les trafics ingress et egress des routeurs d'extrémités ont été obtenues en utilisant la méthode décrite dans [101]. Les topologies sont décrites dans le tableau 5.1.

Tous les tests ont été effectués sur un processeur Pentium Centrino 3 *Ghz* fonctionnant sous Linux avec 2 *GB* de mémoire. Les algorithmes A_{INCR} et A_{MS} ont été implémentés en *C++*. Nous avons utilisé la bibliothèque *LEMON* [106] pour résoudre le problème de flot à coût minimal. Dans les résultats présentés ci-dessous, la solution initiale de A_{INCR} a été obtenue en fixant toutes les métriques à 1.

5.5.1 Modèle d'incertitude sur la demande de type hose

Dans ce paragraphe, nous comparons nos algorithmes aux méthodes proposées par Altin et al. [103]. A des fins de comparaison, nous supposons comme eux que le modèle d'incertitude est celui du modèle Hose et adoptons leur fonction objectif : $\Phi(\mathbf{w}) = \sum_{l \in E} \Phi_l(\mathbf{w})$, où

Instance	$ V $	$ E $	$ Q $
abovenet	19	68	5
arpanet	24	100	10
bhvac	19	46	11
eon	19	74	15
example	10	31	4
metro	11	84	5
nsf	8	20	5
pacbell	15	7	42
telstra	44	88	7
vns1	9	21	3

TAB. 5.1 – Description des Topologies

$$\Phi_l(\mathbf{w}) = \begin{cases} 0 & \frac{y_l(\mathbf{w})}{C_l} = 0 \\ y_l(\mathbf{w}) & 0 \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{1}{3} \\ 3y_l(\mathbf{w}) - \frac{2C_l}{3} & \frac{1}{3} \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{2}{3} \\ 10y_l(\mathbf{w}) - \frac{16C_l}{3} & \frac{2}{3} \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{9}{10} \\ 70y_l(\mathbf{w}) - \frac{178C_l}{3} & \frac{9}{10} \leq \frac{y_l(\mathbf{w})}{C_l} < 1 \\ 500y_l(\mathbf{w}) - \frac{1468C_l}{3} & 1 \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{11}{10} \\ 5000y_l(\mathbf{w}) - \frac{16318C_l}{3} & \frac{11}{10} \leq \frac{y_l(\mathbf{w})}{C_l} < \infty \end{cases} .$$

La fonction coût ci-dessus a été initialement introduite par [56] et a ensuite été utilisée par de nombreux travaux sur l'optimisation des métriques IP. Pour que la comparaison soit équitable, nous utilisons exactement les mêmes données réseaux que dans [103]. Les algorithmes présentés dans [103] utilisent la méthode d'optimisation de [67] en tant que sous-routine. Cette dernière permet l'optimisation des métriques IP pour un ensemble discret de matrices de trafic. De manière à prendre en compte l'incertitude polyédrale sur la demande, Altin et al. proposent d'ajouter dynamiquement des matrices de trafic à cet ensemble discret. Ils proposent ainsi deux algorithmes différents. Le premier, appelé *CM*, génère une matrice de demandes qui, étant donné une solution courante \mathbf{w} , met le réseau dans la situation la pire du point de vue du coût global $\Phi(\mathbf{w})$. Au contraire, la seconde méthode, appelée *LM*, génère une matrice de demandes correspondant au pire cas pour un seul lien en termes de taux d'utilisation.

Les tableaux 5.2 et 5.3 comparent les coûts et les temps de calcul des différents algorithmes sur chacune des huit topologies. Les tests concernant les algorithmes *CM* et *LM* ont été faits par des membres de l'unité de recherche Graphes et Optimisation Mathématique (G.O.M.) de l'université libre de Bruxelles, et ce dans le cadre d'une collaboration avec cette équipe.

On voit qu'à l'exception de la topologie *abovenet*, nos algorithmes ont de bien meilleures performances que les méthodes *CM* et *LM*. On notera également que nos algorithmes sont beaucoup plus

Instance	A_{INCR}	A_{MS}	CM	LM
abovenet	34.9	34.9	9.4	9.4
arpanet	533.4	533.4	$5.51 \cdot 10^6$	$1.01 \cdot 10^7$
bhvac	$2.45 \cdot 10^7$	$2.45 \cdot 10^7$	$3.70 \cdot 10^7$	$3.90 \cdot 10^7$
eon	$3.63 \cdot 10^7$	$3.63 \cdot 10^7$	$5.15 \cdot 10^7$	$5.11 \cdot 10^7$
example	9010.3	9010.3	158094.0	171720.0
metro	19858.0	843.3	1582.9	1492.0
nsf	40213.0	40200.0	$2.55 \cdot 10^6$	$2.56 \cdot 10^6$
pacbell	4735.6	4735.6	$1.49 \cdot 10^6$	541683.0
telstra	0.27	0.27	0.28	0.28
vns1	128357.0	128357.0	128357.0	128357.0

TAB. 5.2 – Coûts obtenus avec les algorithmes A_{INCR} , A_{MS} , CM et LM

Instance	A_{INCR}	A_{MS}	CM	LM
abovenet	0.57	1.02	71	440
arpanet	52	101	124074	18331
bhvac	4	19	67	3084
eon	59	127	8135	10500
example	0.47	0.94	8	23
metro	1.87	2.35	78	1029
nsf	0.28	0.87	9	3
pacbell	2	4	111	485
telstra	0.8	1.14	200	96
vns1	0.3	0.88	2	1

TAB. 5.3 – Temps de calcul (sec) des algorithmes A_{INCR} , A_{MS} , CM et LM

rapides que ces méthodes.

5.5.2 Incertitude sur la demande de type Box+Hose

Dans ce paragraphe, nous revenons à notre modèle d'incertitude original (modèle Box + Hose). Nous comparons ci-dessous les solutions obtenues avec notre algorithme A_{MS} avec celles produites par les algorithmes suivants :

- Les deux heuristiques classiques, c'est à dire les métriques fixées à 1 (UNIT) et les métriques inversement proportionnelles aux capacités des liens (INV_CAPA),
- A_{MS} dans le cadre d'un modèle d'incertitude de type hose.

Pour obtenir un encadrement des fluctuations de chaque demande en trafic autour de sa valeur moyenne, nous avons utilisé la méthode de tomogravité [74] pour calculer la demande en trafic moyenne de chaque flot OD et avons supposé une variation de $\pm 50\%$ autour de cette valeur moyenne. De plus, nous avons calculé une borne inférieure sur le coût global en résolvant le problème de routage optimal multi-chemin :

$$\min \quad z \quad (5.14)$$

s.t.

$$\sum_{s,t \in Q} f_{s,t}^l d_{s,t} \leq C_l z \quad , \quad l \in E, \mathbf{d} \in \mathcal{D} \quad (5.15)$$

$$\sum_{l \in \delta^+(v)} f_{s,t}^l - \sum_{l \in \delta^-(v)} f_{s,t}^l = h_{s,t}^v \quad , \quad s, t \in Q, v \in V, \quad (5.16)$$

$$0 \leq f_{s,t}^l \leq 1 \quad , \quad l \in E, s, t \in Q \quad (5.17)$$

où $\delta^+(v)$ (resp. $\delta^-(v)$) représente l'ensemble des arcs entrants (resp. sortants) au noeud v , tandis que $h_{s,t}^v$ est 1 si $v = s$, -1 si $v = t$ et 0 sinon. Notons que les contraintes (5.17) garantissent qu'il n'y a aucune restriction sur le plan de routage. Les contraintes (5.15) doivent être satisfaites pour toute matrice de trafic \mathbf{d} dans la région d'incertitude \mathcal{D} . On est donc confronté à un problème d'optimisation semi-infini. Comme suggéré dans [91], cette difficulté peut être contournée en dualisant les contraintes (5.1)-(5.3).

Les résultats obtenus sont représentés sur la Figure 5.5. On observe tout d'abord que A_{MS} avec une incertitude sur la demande de type Box+Hose fournit systématiquement de meilleures solutions que les heuristiques classiques UNIT et INV_CAPA (améliorations relatives de 39.9% et 77.8% en moyenne, respectivement). Ceci montre que A_{MS} améliore très significativement ses solutions initiales. On observe également que pour la plupart des instances A_{MS} est assez proche de la borne inférieure (+20% en moyenne). L'écart absolu le plus grand est de 45% (*metro*), et le plus petit est de 4% (*arpanet*). Pour ces deux instances, les écarts absolus de INV_CAPA sont respectivement de 227% et 30%.

Un autre résultat intéressant émerge de ces expérimentations : le taux d'utilisation maximal obtenu dans le cadre du modèle Box+Hose est toujours très inférieur à celui obtenu avec le modèle Hose. Pour certaines instances comme *bhvac* et *metro*, le taux maximal d'utilisation avec le modèle Hose peut être trois fois celui obtenu avec le modèle Box+Hose. Ces résultats confirment que le modèle Hose est vraiment trop pessimiste pour la planification des réseaux. On voit ainsi qu'en rajoutant de l'information supplémentaire, on peut, tout en gardant un ensemble d'incertitude assez vaste, ne pas prendre en compte certaines matrices de trafic qui bien que cohérentes avec le modèle Hose, apparaissent très peu vraisemblables.

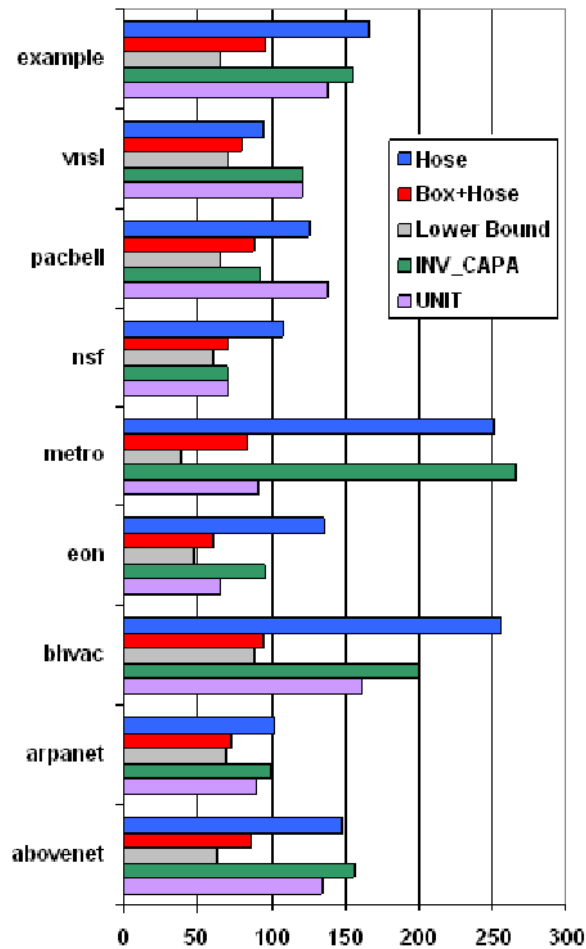


FIG. 5.5 – Taux d'utilisation maximal (%)

5.6 Conclusion

Nous avons étudié dans ce chapitre le problème d'optimisation des métriques avec incertitude sur la demande. Nous avons utilisé un modèle d'incertitude réaliste qui combine le modèle Hose et le modèle Box. L'originalité de ce modèle est qu'il fournit une structure concrète d'incertitude sur la demande qui a du sens pour les opérateurs réseaux.

Nous avons développé de nouveaux algorithmes d'optimisation des métriques permettant d'obtenir de bon résultats dans des temps de calcul raisonnables. Les contributions apportées portent sur plusieurs points clés de l'ingénierie des trafics :

- **Approche incrémentale** : nous proposons un algorithme qui fournit une solution de bonne qualité basée sur la configuration initiale du réseau étudié. La suite de modifications proposées

peuvent être appliquées en totalité ou de manière partielle tout en conservant la garantie que le critère sera amélioré.

- **Approche multi-start** : nous proposons également un algorithme multi-start qui fournit une solution proche de l'optimum. L'opérateur peut mettre en place cette solution dans son réseau avant de le rendre opérationnel.

L'ensemble des algorithmes proposés ont des résultats assez proches de l'optimum dans la majorité des cas testés. Leurs temps de calcul sont très courts comparés aux algorithmes existant. Cela provient essentiellement de la structure de voisinage utilisée, qui est bien adapté au problème.

Conclusion et perspectives

Internet et les réseaux IP sont victimes d'un paradoxe. La simplicité du service "Best Effort" initialement proposé a entraîné un succès planétaire des réseaux IP, caractérisé par une utilisation de plus en plus massive par un nombre toujours croissant d'utilisateurs. Mais ce succès a à son tour entraîné l'apparition de nouveaux services "universels" qui nécessitent plus de bande passante et une meilleure qualité de service et qui ne peuvent fonctionner sur l'architecture IP initiale.

Les réseaux IP ont donc subi un ensemble d'évolutions technologiques majeures pour leur permettre d'écouler des volumes de trafics en perpétuelle croissance et aux besoins variés en qualité de service. Cependant comme nous l'avons montré dans le chapitre 1, une nouvelle approche d'ingénierie des réseaux doit être mise en place pour permettre une planification et une gestion efficace des ressources dans le but de garantir, de manière continue, la qualité de service tout en minimisant les coûts résultants.

Cette approche doit, à notre avis, être basée sur deux axes importants : (1) la supervision du réseau et (2) la planification du réseau. C'est dans cette dernière optique que l'ensemble de nos travaux ont été menés.

5.7 Contributions

Nous avons traité le problème de conception de réseaux. Ce problème se pose généralement aux opérateurs qui souhaitent mettre en place une nouvelle infrastructure de télécommunications. La résolution de ce problème représente une tâche très complexe pour un opérateur et, en règle générale, fort coûteuse étant donné l'importance des investissements nécessaires. Dans le chapitre 3, nous avons donc proposé une nouvelle approche du problème de conception de réseaux. La formulation de ce nouveau problème est notre contribution principale et se distingue des travaux antérieurs par la prise en compte des contraintes matérielles et par l'utilisation de coûts réalistes. Nous avons proposé un algorithme exact pour résoudre ce problème combinatoire. Cependant, pour contourner le problème du passage à l'échelle, nous avons aussi proposé deux méthodes heuristiques aux temps de calcul faibles. Lors de nos

différents tests, nous avons montré l'intérêt d'intégrer les contraintes matérielles pour pouvoir fournir une solution configurable par la suite. Nous avons aussi obtenu des gains économiques conséquents grâce à la prise en compte du matériel existant.

Généralement, lors de la conception de son réseau, l'opérateur se base sur une estimation des demandes en trafic dans le futur réseau concernant des échelles de temps relativement longues (typiquement plusieurs années). Ces estimations sont généralement faites grâce à des études de marché. Dans la pratique, ces estimations ne sont pas assez précises. Durant la phase d'exploitation du réseau, l'opérateur doit anticiper l'évolution globale du trafic, par exemple l'évolution des services dans le réseau et le nombre de clients par service. On est donc dans le contexte de planification des réseaux avec incertitude sur la demande. C'est dans ce contexte que se situe le chapitre 4. Dans ce chapitre nous avons étudié le problème de la conception et l'optimisation des VPN avec un plan de routage en mono-chemin. Nous avons considéré le modèle d'incertitude Hose. Deux critères d'optimisation ont été étudiés dans de ce chapitre : (1) la bande-passante totale réservée et (2) l'utilisation maximale des liens. Pour chacun de ces critères nous avons proposé une méthode heuristique de résolution basée sur les techniques de recherche locale.

Les techniques proposées dans le chapitre 4 de ce mémoire s'appliquent généralement sur des réseaux avec un routage explicite. C'est le cas des réseaux utilisant la technologie MPLS qui sont de nos jours largement répandus. Les opérateurs qui utilisent cette technologie peuvent spécifier explicitement le placement des flux et réserver de la bande-passante pour les chemins (ou LSP) associés aux flux en utilisant le protocole RSVP. Cependant, jusqu'à aujourd'hui, certains opérateurs n'ont pas totalement abandonné leurs infrastructures IP pour migrer vers la technologie MPLS. Un concept clé pour ces opérateurs est celui de l'optimisation du routage dans les réseaux IP, qui permet une utilisation plus efficace des ressources existantes en adaptant le routage au trafics supportés par le réseau. C'est dans ce contexte que se situe le chapitre 5. La recherche des métriques IP permettant d'obtenir des performances optimales sous l'hypothèse que la demande est incertaine est un problème combinatoire connu pour être NP-difficile. Nous avons proposé un nouvel algorithme donnant des solutions de bonne qualité dans des temps de calcul plus faibles que ceux des algorithmes existants. Dans notre algorithme, nous nous sommes basé sur des techniques de flots dans les graphes pour calculer le coût d'une solution.

Toutes les contributions traitées dans ce mémoire sont en cours d'intégration dans le logiciel NEST de la société QoS Design.

5.8 Perspectives

Ces résultats ouvrent de nombreuses pistes intéressantes que nous souhaitons explorer dans l'avenir :

- **Etude du problème d'optimisation dynamique du routage :** La configuration des réseaux changent souvent pour des raisons diverses : surcharge, incidents, changement de matériel, etc. Il faut donc maintenir un réseau robuste à la fois aux changements du trafic et de la topologie pour

assurer la continuité du service. Chaque routeur doit donc adapter dynamiquement et en fonction de l'état actuel du réseau sa table de routage à la nouvelle configuration.

Cela n'est possible qu'à travers un processus automatique. Ce processus consiste à exécuter, périodiquement ou suite à un événement particulier (panne, ajout/remplacement d'un équipement ...), les tâches suivantes :

1. Estimer, au moyen de mesures SNMP par exemple, l'ensemble d'incertitude sur le trafic.
2. En fonction des résultats obtenus, optimiser le routage.

En faisant ainsi, le plan de routage devient robuste puisqu'il s'adapte automatiquement aux changements de la topologie et du trafic.

- **Etude de la robustesse dans les réseaux Overlay :** De nombreuses limitations de l'architecture actuelle de l'Internet sont devenues évidentes ces dernières années : manque inhérent de sécurité et de garanties de qualité de service, performances du routage assez pauvres, etc. Bien que de nombreuses propositions de changement aient été faites, l'ossification de l'Internet a empêché même les changements les plus indispensables d'avoir lieu. Dans ce contexte, les réseaux overlays apparaissent comme une solution viable fournissant aux fournisseurs de service et aux utilisateurs un moyen de résoudre les problèmes précédents à une échelle réduite et sans nécessiter un consensus universel.

Un réseau overlay est un réseau de communication logique établi au-dessus du réseau physique. La complexité du réseau sous-jacent n'est pas visible par le réseau overlay. De nombreux travaux de recherche ont utilisé avec succès les réseaux overlays pour résoudre différents problèmes. Par exemple, ils ont été employés pour implémenter du multicast au niveau applicatif, pour contourner les limitations du protocole BGP ou encore pour fournir des contre-mesures aux attaques de type DoS. Les résultats obtenus ont fait apparaître des améliorations significatives et ont clairement montré l'intérêt de déplacer une partie du contrôle sur le réseau dans les mains des systèmes terminaux.

La mise en oeuvre de réseaux overlays soulèvent toutefois des problématiques de recherche importantes, notamment en ce qui concerne l'adaptation dynamique de la topologie et du routage de ces réseaux aux conditions de trafic dans le réseau sous-jacent. Dans la mesure où il apparaît difficile d'avoir une connaissance précise de ces conditions de trafic, les approches d'optimisation robuste développées au cours de cette thèse pourraient être adaptées pour aborder ces problèmes.

BIBLIOGRAPHIE

- [1] A. Charnes and W. W. Cooper (1954) , "The Stepping Stone Method of Explaining Linear Programming Calculations in Transportation Problems", *Mark. Sci.* 1, 49-69.
- [2] Ford, L. R., Jr. ; Fulkerson, D. R. (1956), "Maximal flow through a network", *Canadian Journal of Mathematics* 8 : 399-404
- [3] J. B. Kruskal (1956) "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem", *Proceedings of the American Mathematical Society*, Vol 7, No. 1 (Feb, 1956), pp. 48-50.
- [4] R. Bellman (1958), "On a routing problem, in quarterly of applied mathematics".
- [5] E.W. Dijkstra (1959), "A note on two problems in connexion with graphs".
- [6] R.G. Busacker, P.J. Gowen (1961) "A procedure for determining minimal-cost network flow patterns", ORO Technical Rep. 15, Operational Research Office, John Hopkins University, Baltimore.
- [7] S. L. Hakimi, (1964), "Optimum Locations of Switching Centers And The Absolute Centers and Medians of a Graph", *Operations Research*, (12), 450-459.
- [8] S. L. Hakimi, (1965), "Optimum Distribution of Switching Centers in a communication Network and Some Related Graph Theoretic Problems ", *Operations Research*, (13), 462-475.
- [9] S. LIN (1965), "Computer solutions of the travelling salesman problem", *Bell System Technology Journal* 44(10) : 2245-2269.
- [10] K. STERGLITZ, P. WEINER (1968) "Some improved algorithms for computer solution of the travelling salesman problem", *Proc. of 6th Allerton Conf. on Circuit and System Theory*, Urbana, Illinois, p. 814-921.
- [11] B.W. KERNIGHAN, S. LIN (1970), "An efficient heuristic for partitioning graphs". *Bell System Technology Journal* 49(2) : 291-307.
- [12] L. Lasdon (1970), "Optimization Theory for Large Systems", Mac Millan Company Ed.
- [13] S. LIN, B.W. KERNIGHAN (1973), "An efficient heuristic for the traveling-salesman problem", *Operations Research* 21 : 498-516.
- [14] A. J. Hoffman (1974), "A Generalization of Max Flow-Min Cut". *Math. Prog.*, 6, 352-359.

- [15] IETF. RFC 791 (1981), "ip-internet protocol".
- [16] IETF. RFC 793 (1981), "tcp-transmission control protocol".
- [17] C. H. Papadimitriou and K. Steiglitz (1982), "Combinatorial optimization : algorithms and complexity", Englewoods Cliffs, Prentice Hall.
- [18] T.L. Magnanti and R.T. Wong (1984), "Network Design and Transportation Planning : Models and Algorithms", Transportation Science.
- [19] Cisco. (1986), "EIGRP-enhanced interior gateway routing protocol".
- [20] A. Balakrishnan, T. L. Magnanti, A. Shulman and R. T. Wong (1989), "A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design", Operation Research.
- [21] IETF. RFC 1058 (1988), "rip routing information protocol".
- [22] M. Minoux (1989), "Network Synthesis and Optimum Network Design Problems : Models, Solution Methods and Applications", Networks.
- [23] F. Glover (1989), "Tabu Search–Part I", ORSA J. on Computing.
- [24] IETF. RFC 1142 (1990), "is-is intra-domain routing protocol".
- [25] F. Glover (1990), "Tabu Search–Part II", ORSA J. on Computing.
- [26] B. Gavish and A. Altinkemer (1990), "Backbone Network Design Tools with Economic Tradeoffs", ORSA J. on Computing.
- [27] IETF. RFC 1267 (1991), "bgp-border gateway protocol".
- [28] C.R. REEVES (1993), "Modern heuristic techniques for combinatorial problems", Blackwell Scientific Publications, Oxford.
- [29] T. L. Magnanti and P. Mirchandani (1993), "Shortest Paths, Single Origin-Destination Network Design, and Associated Polyhedra", Networks.
- [30] T.L. Magnanti, P. Mirchandani and R. Vachani (1993), "The Convex Hull of Two Core Capacitated Network Design Problems", Mathematical Programming.
- [31] J. Chakrapani, J. Skorin-Kapov (1993), "Massively Parallel Tabu Search for the Quadratic Assignment problem", Annals of Operations Research Vol.41, 327-341
- [32] R. K. Ahuja and T. L. Magnanti and J. B. Orlin (1993), "Network Flows", Prentice Hall.
- [33] D. Burton (1993), "On the inverse shortest path problem", MIT PhD thesis.
- [34] W. D. Harvey, M. L. Ginsberg (1995), "Limited Discrepancy Search". The 14th Int. Conf. on Artificial Intelligence.
- [35] T.L. Magnanti, P. Mirchandani and R. Vachani (1995), "Modeling and Solving the Two-Facility Capacitated Network Loading Problem", Operations Research.
- [36] R. E. Korf (1996, "Improved limited discrepancy search", Proceedings AAAI-96, 286-291.
- [37] D. Bienstock and O. Gunluk (1996), "Capacitated Network Design-Polyhedral Structure and Computation", INFORMS J. on Computing.
- [38] F. Barahona (1996), "Network Design Using Cut Inequalities", SIAM J. on Optimization.
- [39] G. Ramalingam, Thomas W. Reps (1996) "An Incremental Algorithm for a Generalization of the Shortest-Path Problem", J. Algorithms 21(2) : 267-305.

- [40] E.H.L. AARTS, J.K. LENSTRA (1997), "Local search in combinatorial optimization", John Wiley & Sons.
- [41] R. Braden and L. Zhang and S. Berson and S. Herzog and S. Jamin (1997) "Resource ReSerVation Protocol (RSVP)", RFC 2205, IETF Network Working Group,
- [42] IETF. RFC 1247 (1998), "ospf open shortest path first".
- [43] Y. Vardi (1998), "Bayesian Inference on Network Traffic Using Link Count Data", Journal of the American Statistical Association, volume 93.
- [44] B. Gendrom, T.G. Crainic and A. Frangioni (1999), "Multicommodity Capacited Network Design", in : P. Soriano, B. Sanso (Eds.), Telecommunications Network Planning, Kluwer Academics Publisher.
- [45] J.L. Rougier (1999), "Routage Dynamique Unicast et Multicast", Th de doctorat, ENST.
- [46] D. Kofman and M. Gagnaire (1999), "Raux Haut Dt : Tome I Raux ATM et raux locaux", 2ed., Dunod.
- [47] M. Pi M. Szczesniak (1999), "Design of Multi-Layer Robust Telecommunication Networks : Cost Analysis of Protection Mechanisms using Lagrangean Relaxation", Teletraffic Engineering in a Competitive World (eds. P.Key and D.Smith), Teletraffic Science and Engineering , Elsevier Science B.V., vol.3b, pp.989-998.
- [48] E. Rosen and Y. Rekhter (1999), " BGP/MPLS VPNs", RFC 2547, IETF Network Working Group.
- [49] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan and J. E. van der Merive (1999), "A flexible model for resource management in virtual private networks", ACM SIGCOMM, San Diego, CA, USA.
- [50] B. Gleeson, A. Lin, J. Heinanen, G. Armitage and A. Malis (2000), "IP Based Virtual Private Networks", RFC 2764, IETF Network Working Group.
- [51] B. Fortz and M. Labbé and F. Maffioli (2000), "Solving the Two-Connected Network with Bounded Meshes Problem", Operations Research.
- [52] W. Ben-Ameur and E. Gourdin and B. Liau (2000), "Dimensioning of internet networks", Proceedings of the DRCS2000, Munich.
- [53] J. Cao and D. Davis and S. Wiel and B. Yu (2000), "Time-Varying Network Tomography : Router Link Data", Bell Labs Tech. Memo.
- [54] W. Ben-Ameur and B. Liau (2001), "Calcul des métriques de routage pour Internet", Ann. Telecommun 56, pages 3-4.
- [55] W. Ben-Ameur, N. Michel and B. Liau, "Routing Strategies for IP Networks", Teletronikk Magazine 2, 3 (2001).
- [56] B. Fortz and M. Thorup (2001), "Internet Traffic Engineering by optimizing OSPF weights", Proc. 19th IEEE Conf. on Computer Communications (INFOCOM).
- [57] M. Pi T. Szymanski (2001), "Basic Reconfiguration Options in Multi-Layer Robust Telecommunication Networks - Design and Performance Issues", Teletraffic Engineering in the Internet Era (eds. J.Moreira de Souza, N.L.S. Fonseca and E.A. de Souza e Silva), Teletraffic Science and Engineering , Elsevier Science B.V., vol.4. pp. 271-284.

- [58] F. Focacci , F. Laburthe and A. Lodi (2001), "Local Search and Constraint Programming", in F. Glover and G. Kochenber (Eds.), Handbook on Metaheuristics, Kluwer Academic Publishers.
- [59] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi and B. Yener (2001), "Provisioning a Virtual Private Network : A Network Design Problem for Multicommodity Flow", in Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC), pp. 389-398.
- [60] Z. Wang and Y. Wang and L. Zhang (2001), "Internet traffic engineering without full mesh overlaying", Proceedings of INFOCOM 2001, Anchorage, Alaska.
- [61] A. Kumar and R. Rastogi and A. Silberschatz and B. Yener (2002), "Algorithms for provisioning virtual private networks in the hose model", IEEE/ACM Transactions on Networking, volume 10, number 4, pages 565-578.
- [62] G. Italiano and S. Leonardi and G. Oriolo (2002), "Design of networks in the hose model", 3rd Workshop Approx. Random. Algorithms Commun. Networks (ARACNE).
- [63] C. Swamy and A. Kumar (2002), "Primal-dual algorithms for connected facility location problems", 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX02).
- [64] M. Pioro and A. Szentesi and J. Harmatos and A. Juttner and P. Gajowniczek and S.Kozdrowski (2002), "On open shortest path first related network optimization problems", Performance evaluation journal, New York, pages = 201-223.
- [65] M. Ericsson and M.G.C. Resende and P.M. Pardalos (2002), "A genetic algorithm for the weight setting problem in OSPF routing", Journal of Combinatorial Optimization, volume 6, pages 299-333.
- [66] A. Medina, N. Taft, S. Battacharya, C. Diot and K. Salamatian (2002), "Traffic matrix estimation : Existing techniques compared and new directions", In SIGCOMM'2002, Pittsburgh.
- [67] B. Fortz and M. Thorup (2002), "Optimizing OSPF/IS-IS weights in a changing world", IEEE Journal on Selected Areas in Communications, volume 20, pages 756-767.
- [68] L. S. Buriol , M. G. C. Resende , M. Thorup (2003), "Speeding Up Dynamic Shortest Path Algorithms", INFORMS Journal on Computing.
- [69] A. Jttner and I. Szabo and A. Szentesi (2003), "On bandwidth efficiency of the hose resource management model in virtual private networks", INFOCOM2003, San Francisco, USA.
- [70] D. Applegate and E. Cohen (2003), "Making intra-domain routing robust to changing and uncertain traffic demands : understanding fundamental tradeoffs", in Proc. ACM SIGCOMM Karlsruhe, Germany, pp. 313 - 324
- [71] A. Gupta, A. Kumar, and T. Roughgarden (2003), "Simpler and better approximation algorithms for network design", in Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), pp. 365-372.
- [72] A. Sridharan and R. Guérin and C. Diot (2003), "Achieving Near Optimal Traffic Engineering Solutions for Current OSPF/ISIS Networks", Proceedings of INFOCOM 2003, San Fransisco, USA.
- [73] L.S. Buriol and M.G.C. Resende and C.C. Ribeiro and M. Thorup (2003), "A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing", journal of Networks, volume 46, pages 36-56.

- [74] Y. Zhang and M. Roughan and N. Duffield and A. Greenberg (2003), "Fast accurate computation of large-scale IP traffic matrices from link loads", In ACM SIGMETRICS, San Diego, USA.
- [75] E. Gourdin, B. Liao, A. Ouorou, and D. Nace, (2004), "Optimisation des taux de communications", Bulletin de la Soci Franse de Recherche Optionnelle et dAide Dsion, (12) :812.
- [76] T. Erlebach, and M. Ruegg (2004), " Optimal bandwidth reservation in hose-model VPNs with multipath routing", in proceeding of INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies Vol. 4, pp.2275-2282.
- [77] N. Springs, R. Mahajan, D. Wetherall and T. Anderson (2004), "Measuring ISP topologies with Rocketfuel", IEEE/ACM Transactions on Networking, volume 12, pages 2-16.
- [78] B. Fortz and M. Thorup (2004), "Increasing internet capacity using local search", Computational Optimization and Applications journal, volume 29, pages 13-48.
- [79] A. Nucci and R. Cruz and N. Taft and C. Diot (2004), "Design of IGP Link Weight Changes for Estimation of Traffic Matrices", IEEE INFOCOM'2004, Hong Kong.
- [80] K. Papagiannaki and N. Taft and A.Lakhina (2004), "A distributed approach to measure IP traffic matrices", Internet Measurement Conference.
- [81] W. Ben-Ameur and H. Kerivin (2005), " Routing of uncertain demands", Optimization and Engineering journal, volume 3, pages 283-313.
- [82] E. Mulyana and U. Killat (2005), "Optimizing IP networks for uncertain demands using outbound traffic constraints", INOC2005.
- [83] C. Fortuny and O. Brun and J. M. Garcia (2005), "Metric Optimization in IP Networks, 19th International Teletraffic Congress", Beijing, China.
- [84] H. it (2005), "A column generation approach for IGP weight setting problem", CoNEXT, Toulouse, France.
- [85] A. Soule and A. Lakhina and N. Taft and K. Papagiannaki and K. Salamatian and A. Nucci and M. Crovella and C. Diot (2005), "Traffic matrices : balancing measurements, inference and modeling", SIGMETRICS Perform. Eval. Rev.
- [86] A. Soule and K. Salamatian and A. Nucci and N. Taft (2005), "Traffic matrix tracking using Kalman filters", SIGMETRICS Perform. Eval. Rev.
- [87] T. Mashima et al. (2006), "Bi-Connectivity Augmentation for Specified Vertices of a Graph with Upper Bounds on Vertex-Degree Increase", IEICE-Transactions on Information and Systems.
- [88] A. Parmar, S. Ahmed and J. Sokol (2006), "An integer programming approach to the OSPF weight setting problem", School of Industrial & Systems Engineering, Georgia Tech.
- [89] I. Juva and S. Vaton and J. Virtamo (2006), "Quick Traffic Matrix Estimation Based on Link Count Covariances", IEEE International Conference on Communications (ICC 2006), Istanbul.
- [90] P. Bermolen and S. Vaton and I. Juva (2006), "Search for Optimality in Traffic Matrix Estimation : a Rational Approach by Cramer-Rao Lower Bounds", 2nd EuroNGI Conference on Next Generation Internet Design and Engineering, Valencia, Spain.
- [91] A. Altin et al. (2007), "Provisioning Virtual Private Networks under traffic uncertainty", Networks, Vol. 20, No. 1, pp. 100-115.

- [92] C. A. J. Hurkens, J. C. M. Keijsper and L. Stougie (2007), "Virtual Private Network Design : A Proof of the Tree Routing Conjecture on Ring Networks", *SIAM Journal on Discrete Mathematics*, Volume 21 , Issue 2, pp. 482-503.
- [93] B. Fortz and H. Umit (2007), "Efficient techniques and tools for intra-domain traffic engineering", research report, ULB Computer Science Departement.
- [94] C. Fortuny and O. Brun and J. M. Garcia, "Fanout Inference from Link Counts", 4th European Conference on Universal Multiservice Networks (ECUMN 2007).
- [95] G. S. Poo and H. Wang (2007), "Multi-path routing versus tree routing for VPN bandwidth provisioning in the hose model", *journal of Computer Networks*, volume 51, pages 1725-1743.
- [96] A. Frangioni, B. Gendron (2008) "0-1 Reformulations of the Multicommodity Capacitated Network Design Problem", *Discrete Applied Mathematics*.
- [97] P. Belotti and M.C. Pinar (2008), Optimal oblivious routing under statistical uncertainty, *Optimization and Engineering journal* volume 9, pages 257-271.
- [98] N. Goyal, N. Olver, F. B. Shepherd (2008), "The VPN conjecture is true", *Proceedings of the 40th annual ACM symposium on Theory of computing*.
- [99] Aharoni, Ron and Berger, Eli (2009), "Mengers Theorem for infinite graphs", *Inventiones Mathematicae* 176 : 162
- [100] J. Chu and C. T. Lea (2009), "Optimal link weights for IP-based networks supporting hose-model VPNs", *IEEE/ACM Transactions on Networking*, volume 17, number 3, pages 778-786.
- [101] A. Altin, B. Fortz and H. Umit (2009), "Oblivious OSPF routing with weight optimization under polyhedral demand uncertainty", *International Network Optimization Conference (INOC2009)*, Pisa, Italy.
- [102] A. Altin, P. Belotti and M.C. Pinar (2009), "OSPF routing with optimal oblivious performance ratio under polyhedral demand uncertainty", *journal of Optimization and Engineering*.
- [103] A. Altin and B. Fortz and M. Thorup and H. Umit (2009), "Intra-domain traffic engineering with shortest path routing protocols", *4OR : A Quarterly Journal of Operations Research*.
- [104] Cisco ios Netflow, http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html
- [105] Cisco Systems : Sampled NetFlow, http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.html
- [106] COIN-OR :LEMON, Lemon Graph Library, <https://lemon.cs.elte.hu/trac/lemon>
- [107] <http://www-01.ibm.com/software/integration/optimization/cplex/>
- [108] www.qosdesign.com

Glossaire

API	Application Programming Interface
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
FAI	Fournisseurs Accès Internet
FTP	File Transfert Protocol
GRE	Generic Route Encapsulation
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
L2TP	Layer 2 Tunneling Protocol
LDS	Limited Discrepancy Search
LER	Label Edge Router
LSP	Label Switched Path
MBWAN	Mobile Broadband Wireless Access Networks
MPLS	MultiProtocol Label Switching
MPOA	Multi Protocol Over ATM
NHRP	Next Hop Resolution Protocol
OSPF	Open Shortest Path First
PAN	Personal Area Network
PCC	Plus Court Chemin
PHB	Per Hop Behaviour
PL	Programmation Linéaire
PLNE	Programmation Linéaire en Nombres Entiers
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RSVP	Resource ReSerVation Protocol
SaaS	Software as a Service

SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
TCP	Transmission Control Protocol
TE	Traffic Engineering
TS	Tabu Search
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VPN	Virtual Private Network
WAN	Wide Area Network
WDM	Wave-length Division Multiplexing