

Analyse et optimisation de performance des réseaux de communication

Olivier Brun

▶ **To cite this version:**

Olivier Brun. Analyse et optimisation de performance des réseaux de communication. Réseaux et télécommunications [cs.NI]. Université Paul Sabatier - Toulouse III, 2012. tel-00728757

HAL Id: tel-00728757
https://theses.hal.science/tel-00728757

Submitted on 6 Sep 2012

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.

Habilitation à Diriger des Recherches

delivrée par l'Université Toulouse III

Spécialité : Réseaux et Communications

par

Olivier BRUN

Analyse et Optimisation de Performance des Réseaux de Communication

Soutenue publiquement le 6 février 2012 devant le jury composé de :

M.	**Thierry**	**Gayraud**	**Président**
M.	**Jean-Marie**	**Garcia**	**Directeur de recherche**
M.	**Konstantin**	**Avrachenkov**	**Rapporteur**
M.	**Tijani**	**Chahed**	**Rapporteur**
M.	**Philippe**	**Mahey**	**Rapporteur**
M.	**Prosper**	**Chemouil**	**Examinateur**

Remerciements

Ce document synthétise une dizaine d'années de travaux menés au LAAS-CNRS. Mes premiers remerciements ne peuvent donc qu'aller à mes collègues du LAAS, et en particulier à Jean-Marie Garcia, qui m'a guidé dans mes recherches depuis ma thèse, ainsi qu'à Thierry Monteil, Urtzi Ayesta et Balakrishna Prabhu avec lesquels j'ai collaboré sur plusieurs des sujets décrits dans ce mémoire. Je tiens à les assurer de mon amitié et du plaisir que j'ai eu à travailler avec eux.

J'exprime aussi ma gratitude envers mes étudiants, et notamment ceux dont j'ai eu le plaisir d'encadrer la thèse.

Je remercie chaleureusement les membres de mon jury qui ont accepté de rapporter sur cette thèse et d'assister à la soutenance malgré des conditions climatiques particulièrement difficiles. Leur participation au jury fut un grand honneur pour moi.

Enfin, merci à toutes les personnes que j'ai pu côtoyer durant toutes ces années passées au LAAS et avec j'ai eu tant de plaisir à travailler.

Résumé

Les réseaux de communication jouant un rôle de plus en plus important dans nos activités quotidiennes, l'interruption des services qu'ils fournissent, ou même une dégradation significative de leur qualité, deviennent de moins en moins acceptables. La sécurisation des réseaux et le contrôle de qualité de service sont ainsi devenus des enjeux majeurs qui nécessitent de réelles avancées méthodologiques dans plusieurs domaines. Nos contributions en ce sens peuvent se répartir en deux volets : d'une part celui de l'évaluation de performance et d'autre part celui de l'optimisation de réseaux.

Nos travaux en évaluation de performance sont basés sur la théorie des files d'attente et sur la théorie des jeux. Ils visent le développement de modèles stochastiques analytiques ou de simulation permettant le passage à l'échelle pour l'évaluation, en régimes stationnaire et transitoire, des principales métriques de performance au niveau paquet (délai, taux de perte et gigue), mais aussi au niveau flot (nombre de sessions actives, temps de téléchargement, etc). Ils concernent également l'analyse pire cas de la performance des stratégies de routage décentralisées.

Les travaux en optimisation sont quant à eux essentiellement consacrés au développement de méthodes pour la conception de réseaux (d'accès ou de backbone) résilients, leur dimensionnement et l'optimisation de leur routage (avec OSPF ou MPLS). Les problèmes abordés étant généralement des problèmes combinatoires difficiles, avec souvent des fonctions objectifs non-linéaires, les approches heuristiques sont privilégiées. D'autres travaux sont consacrés à l'ordonnancement de tâches strictement périodiques dans les systèmes embarqués avioniques.

Mots-clés: Modèles stochastiques de performance, files d'attente, théorie des jeux, optimisation, réseaux de communication

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. Network survivability and the control of the quality of service have thus become critical issues that require significant methodological progresses in various scientific fields. Our contributions fall in two broad categories : performance evaluation and optimization of networks.

Our research works on the performance evaluation of networks are based on queueing and game theories. They aim at the development of analytical or simulation-based methods for the large-scale evaluation of the main performance metrics, both at the packet level (delay, loss rate and jitter) and at the flow level (number of active sessions, download times, etc). They also deal with the worst-case analysis of the performance of decentralized routing algorithms.

The second category of contributions are mainly devoted to the development of new optimization methods for the design of survivable access and backbone networks, for capacity-planning and for route optimization in OSPF and MPLS networks. Since most of these problems are NP-hard combinatorial problems, often with non-linear cost functions, we usually have to resort to dedicated heuristics. Another work in the applied optimization field deals with the strictly periodic scheduling of tasks in embedded avionic systems.

Keywords: Stochastic performance models, Queueing systems, Game theory, Optimization, Communication networks

Table des matières

1 Introduction **1**

1.1 Evolution des réseaux de communication 1

1.2 Objectifs des travaux de recherche 2

1.3 Un mot sur ce qui n'est pas décrit dans ce document 4

1.4 Organisation de ce document 4

2 Contexte des Recherhes **5**

2.1 Projets de recherche collaboratifs 5

2.2 Encadrement de thèses 8

2.3 Enseignement 9

2.4 Activités de valorisation 10

3 Evaluation de Performances des Réseaux **11**

3.1 Modélisation dynamique des trafics 11

3.1.1 Modélisation des Réseaux à Commutation de Circuits Multiclasses 12

3.1.2 Modélisation des Réseaux à Commutation de Paquets 14

3.1.3 Simulation Hybride Distribuée 16

3.2 Analyse de la file M/D/1/N 17

3.2.1 Solution stationnaire 17

3.2.2 Solution transitoire 18

3.3 Analyse de la gigue des flux audio et vidéo 19

3.3.1 Notations 21

3.3.2 Le cas d'un seul noeud 21

3.3.3 Le cas de plusieurs noeuds 22

3.3.4 Exemple d'application 23

3.4 Modélisation au niveau flot du trafic Internet 24

3.4.1 Modèles de Réseaux d'Accès 25

3.4.2 Allocation de débit équitable de type "Weighted Max-Min" (WMM) . . . 28

3.4.3 Modèle Réseau Global 30

3.5 Prix de l'Anarchie et Equilibrage de Charge Non-Coopératif 32
3.5.1 Formulation du problème 34
3.5.2 Résultats 35

4 Optimisation de Réseaux 37
4.1 Conception de réseaux d'accès avec coûts d'équipements 37
4.1.1 Formulation du problème 38
4.1.2 Calcul des configurations de carte optimales 40
4.1.3 Résolution exacte basée sur du Branch and Bound 41
4.1.4 Extensions 43
4.2 Optimisation des métriques de routage OSPF 44
4.2.1 Formulation du problème 45
4.2.2 Algorithme de recherche locale pour l'optimisation des métriques 45
4.2.3 Quelques résultats 47
4.2.4 Extensions 48
4.3 Placement des LSP dans les réseaux MPLS 48
4.3.1 L'algorithme de placement de LSP en trois phases 49
4.3.2 Quelques résultats 52
4.3.3 Extensions 52
4.4 Ordonnancement multi-processeur de tâches strictement périodiques 53
4.4.1 Ordonnancement uniprocesseur 54
4.4.2 Ordonnancement multiprocesseur 58
4.4.3 Un exemple de résultat 59
4.4.4 Algorithme multi-start 59

5 Prospective de recherche 61
5.1 Ordonnancement pour les systèmes embarqués critiques 61
5.1.1 Contexte scientifique 61
5.1.2 Objectifs scientifiques 62
5.2 Théorie des jeux et routage distribué 63
5.2.1 Contexte scientifique 64
5.2.2 Objectifs scientifiques 65
5.3 Contrôle intelligent des réseaux 66
5.3.1 Contexte scientifique 66
5.3.2 Objectifs scientifiques 66

6 Publications 69

Bibliographie **73**

Chapitre 1

Introduction

Ce chapitre présente le contexte et les objectifs de nos recherches. Le paragraphe 1.1 présente tout d'abord un panorama de l'evolution des réseaux de communication au cours de ces dernières années et situe nos recherches dans ce panorama, les objectifs et thèmes de recherche étant décrits au paragraphe 1.2. Nous signalons brièvement au paragraphe 1.3 les travaux qui ne sont pas décrits dans ce document. Enfin, le paragraphe 1.4 conclu ce chapitre en présentant l'organisation du reste du document.

1.1 Evolution des réseaux de communication

Avec aujourd'hui environ 1,5 milliards d'utilisateurs dans le monde, l'Internet a connu un succès fulgurant. Ce succès planétaire, obtenu en moins de dix ans, est le résultat d'une succession de percées, à commencer bien sûr par le développement incroyable du World Wide Web, suivi par l'adoption par le monde professionnel des applications Internet telles que le courrier électronique et, plus récemment, par le déploiement des logiciels de partage de fichiers pair à pair. Parallèlement à l'augmentation soutenue du trafic liée aux services "traditionnels", l'Internet a connu plusieurs évolutions majeures ces dernières années.

La première de ces évolutions concerne sa transformation en une architecture de communication globale supportant également des services "temps réel", tels que la voix sur IP ou la vidéo. Ces services ayant des contraintes de qualité de service beaucoup plus strictes que les services traditionnels évoqués ci-dessus, les nouveaux réseaux IP intègrent un ensemble de mécanismes pour fournir une qualité de service différenciée aux flux en fonction de leurs besoins. Le développement attendu de ces nouveaux services "temps réel" 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.

Un autre fait marquant est la multiplication des réseaux d'accès, filaires et surtout radio (GPRS, UMTS, WiFi, WiMAX, etc.). Cette évolution conduit vers une réelle convergence des

services fournis aux utilisateurs, quels que soient les modes d'accès. Cette convergence va se traduire pour les utilisateurs par la possibilité de toujours accéder à une multiplicité de services Internet, quel que soit le média utilisé et où qu'ils se trouvent. L'Internet de demain sera ainsi une interconnexion de réseaux d'accès hétérogènes permettant d'accéder, via des cœurs de réseaux IP/MPLS, à un grand nombre de services "universels".

Enfin, on assiste à un développement sans précédent de nouvelles applications avec le succès fulgurant des réseaux sociaux, avec l'émergence de l'informatique en nuage et notamment du concept SaaS (Software as a Service), qui permet à une application logicielle d'être hebergée en tant que service déporté accessible à travers 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'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 qualité de service, sont de moins en moins acceptables. La sécurisation des réseaux et le respect des nouvelles exigences de qualité de service sont ainsi des enjeux majeurs.

Ces nouvelles exigences de qualité de service, l'augmentation continue du trafic Internet et les impératifs de plus en plus forts de sécurisation des réseaux imposent aux fournisseurs d'accès Internet (FAI) et aux opérateurs une adaptation permanente de leurs infrastructures de communication. Cependant, le contexte concurrentiel qui induit des marges bénéficiaires réduites, ne permet plus d'améliorer les performances d'un réseau par un surdimensionnement excessif des équipements.

Une solution consiste à avoir un suivi plus régulier et plus fin du réseau pour mettre en œuvre 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. L'utilisation de ces techniques d'ingénierie de trafic suppose toutefois le développement de modèles et de méthodes théoriques ainsi que d'outils logiciels appropriés pour l'évaluation de performances et l'optimisation des réseaux.

1.2 Objectifs des travaux de recherche

Les développements de ces dernières années ont conduit à une complexité sans précédent des réseaux dont la maîtrise pose plus que jamais de réelles difficultés conceptuelles. Les problèmes de recherche soulevés par ce secteur applicatif sont énormes et nécessitent de réelles avancées méthodologiques dans plusieurs domaines. En particulier, l'exploitation optimale des ressources

des réseaux de communication et le respect des exigences de qualité de service supposent de nouvelles approches de modélisation, de planification et de gestion des ressources des réseaux.

L'objectif de nos recherches est le développement de nouveaux modèles et méthodes pour évaluer et optimiser les performances des réseaux de communication :

- **Evaluation de performances :** pour prédire correctement les performances des flux, la modélisation doit prendre en compte le caractère imprévisible du trafic aussi bien au niveau flot qu'au niveau paquet, les différents traitements opérés dans le réseau et le comportement dynamique et distribué des protocoles de routage ou de contrôle de débit, etc. Devant une telle complexité, les modèles analytiques dits exacts atteignent rapidement leurs limites. Les simulations événementielles quant à elles ont un coût prohibitif pour des réseaux de taille réelle (centaines de noeuds, milliers de flux), en particulier pour des réseaux hauts débits. C'est pourquoi la recherche est actuellement fortement sollicitée par le monde des télécoms pour développer de nouveaux schémas d'approximation permettant d'estimer la qualité de service des flux. Ces approximations doivent être d'un coût algorithmique adapté à la taille des réseaux actuels, prendre en compte le caractère réactif des protocoles et doivent s'intégrer efficacement dans des procédures d'optimisation. Nos travaux dans ce domaine utilisent des outils théoriques issus de la théorie des files d'attente, de la théorie des jeux et de la simulation stochastique.

- **Optimisation de réseaux :** il est nécessaire de développer une approche intégrée des différents problèmes de planification qui se posent à un opérateur sur des horizons multiples (topologie, dimensionnement et routage) ainsi que des méthodes de gestion dynamique des ressources des réseaux. L'abscence de données fiables sur le trafic, les contraintes de QoS ou de sécurité et la complexité des réseaux (taille, contraintes technologiques ...) conduisent à des problèmes d'optimisation extrêmement difficiles. Pour aboutir à des méthodes de résolution efficaces, on est souvent contraint de développer des techniques heuristiques basées sur la programmation mathématique linéaire ou non-linéaire ou sur la théorie des flots dans les réseaux, et d'explorer de nouvelles approches d'optimisation.

Nous décrivons dans ce document nos contributions dans ces deux grands domaines, qui constituent les deux axes essentiels de nos travaux de recherche. Plutôt que de faire un catalogue "à la Prévert", nous avons choisi de présenter une sélection de ces contributions, chacune pouvant ainsi être décrite en quelques pages. Nous disons toutefois un mot dans le paragraphe suivant sur les travaux qui ne sont pas présentés dans ce document.

1.3 Un mot sur ce qui n'est pas décrit dans ce document

Nous avons choisi de ne pas revenir sur nos travaux concernant l'algorithmique parallèlle [194, 47, 40, 19] ou les environnements de calcul parallèle [94, 85, 87, 41], qui ne correspondent plus à nos activités.

Concernant nos travaux sur la modélisation des réseaux, nous avons choisi de ne pas présenter nos travaux sur la modélisation des sources de trafic [109, 107, 113] ou du protocole SIP [108, 110, 111]. Certaines approches de modélisation des réseaux Internet ne sont pas non plus présentées [86, 27, 112, 106].

Enfin, en ce qui concerne nos travaux sur la planification des réseaux, les travaux menés sur l'estimation de matrices de trafic ne sont pas présentés [75, 76]. D'autres travaux sont seulement évoqués [22, 23, 21].

1.4 Organisation de ce document

L'organisation du présent mémoire est la suivante :

- Le chapitre 2 présente brièvement le contexte des recherches menées : les projets de recherche collaboratifs auxquels nous avons participés, les thèses que nous avons encadrées, nos activités d'enseignementt et de valorisation.

- Le chapitre 3 décrit une sélection de nos travaux portant sur l'évaluation de performances des réseaux : modélisation dynamique des rśeaux de files d'attente, analyses transitoire et stationnaire des files M/D/1/N, approximation de la gigue, modélisation au niveau flot des réseaux Internet et prix de l'anarchie dans les fermes de serveurs.

- Le chapitre 3 présente une sélection de nos travaux portant sur l'optimisation des réseaux : conception de réseaux d'accès, problèmes de monoroutage dans les réseaux, optimisation des métriques de routage IP et, même s'il ne s'agit par vraiment d'optimisation de réseaux, ordonnancement multiprocesseur de tâches strictement périodiques.

- Le chapitre 4 présente notre prospective de recherche pour les années à venir. Elle concerne la modélisation des systèmes temps réel, les applications de la théorie des jeux au routage distribué et le développement d'un système de supervision/contrôle des réseaux.

- Enfin, le chapitre 5 dresse la liste de nos publications scientifiques.

Chapitre 2

Contexte des Recherhes

La plupart de nos travaux de recherche se sont déroulés dans le cadre de projets de recherche collaboratifs. Nous décrivons succintement ces projets au paragraphe 2.1. Certains de nos résultats ont d'autre part été obtenus en collaboration avec les doctorants que nous avons co-encadrés. On trouvera la liste des étudiants co-encadrés ainsi qu'une brève description de leur sujet de recherche au paragraphe 2.2. Nos travaux de recherche alimentent les cours que nous donnons à l'INSA de Toulouse dont nous donnons un aperçu au paragraphe 2.3. Enfin, nous précisons au paragraphe 2.4 nos activités de valorisation de ces travaux de recherche.

2.1 Projets de recherche collaboratifs

Projet ESQUIMAUX (2000-2003)

Estimation de la Qualité de service d'une architecture Internet audio-visuelle sur Modélisation des flUX

Partenaires : Cegetel, CS, Delta Partners, ENST, INRIA Sophia Antipolis, LAAS.

Rôle : participant.

Résumé : Les objectifs du projet étaient l'estimation de la QoS des flux dans une architecture IP DiffServ/MPLS et le dimensionnement de l'infrastructure.Nos travaux ont porté sur la conception et le développement d'un outil général d'évaluation de performances de réseaux IP DiffServ (DHS) basé sur la modélisation Différentielle Hybride, sur la modélisation des sources de trafic (audio, vidéo, data) et sur la modélisation analytique des mécanismes d'ordonnancement des paquets.

Projet OPIUM (2001-2003)

Optimisation de la Planification des Infrastructures des réseaUx Mobiles

Partenaires : Bouygues Telecom, Delta Partners, LIMOS, LAAS.

Rôle : participant.

Résumé : Les objectifs du projet étaient la conception d'un modèle analytique complet pour décrire l'interconnexion de plusieurs Systèmes Autonomes IP et un cœur de réseau MPLS et la modélisation du problème d'optimisation lié à la création des LSP (Label Switched Path). Nos travaux ont porté sur l'évaluation de performance des flux, sur la conception d'algorithmes pour le routage optimal des LSP, sur le re-routage des LSP en cas de panne et sur le dimensionnement des infrastructures.

Projet CASP (2002-2004)

Cluster for Application Service Provider

Partenaires : Alinka, Delta Partners, LAAS.

Rôle : participant.

Résumé : Le projet visait la création de services Internet déportés sur une machine parallèle de type Cluster pour des grosses applications industrielles comme le logiciel NetQUAD de Delta Partners. L'architecture Cluster a été utilisée pour ses capacités d'extension, de modularité et d'allocation de ressources afin de mettre en place un "portail" Web ayant pour ambition d'offrir à une entreprise un accès personnalisé à des applications. Nos travaux ont porté sur la construction de l'infrastructure logicielle nécessaire à la gestion du cluster pour réaliser de l'ASP (Application Server Provider) : administration du cluster, gestion des ressources (avec garantie de qualité de service), portail internet, ainsi que sur la parallélisation du noyau d'évaluation de performances de NetQUAD pour le démonstrateur final.

Projet LAAS STM (2005-2006)

Sources de Trafic & Métrologie

Partenaires : groupes MRS et OLC du LAAS.

Rôle : responsable.

Résumé : Le projet STM s'inscrivait dans le cadre de la simulation et de l'émulation de grands réseaux, et notamment l'Internet, avec l'objectif de concevoir une méthodologie permettant de mettre en œuvre des scenarii réalistes en termes de topologie, de tomographie et de trafic. STM a permis de développer un système pour la conception, la génération et l'analyse de trafic multimédia.

Projet ANR AVIPS (2006-2008)

Aide à la Virtualisation et l'Interfonctionnement des Plates-formes de Service

Partenaires : SFR, Index Multimedia, QoS Design, ENST, LAAS.

Rôle : responsable LAAS.

Résumé : La prise en compte des usages, la prédominance de la personnalisation, « l'user centrix» activent le paysage des services et conditionnent l'activité économique des réseaux de télécommunications avec une TTM et le ROI omniprésents. La virtualisation des services, leur interfonctionnement, l'adaptabilité des plates-formes de service et le contrôle de la qualité de service sont des points centraux à étudier. Le premier volet du projet AVIPS concerne la modélisation architecturale (trans-génération) à des fins de virtualisation des plates-formes de service et la spécification de solutions organisationnelles de plus en plus dynamiques et de plus en plus complexes. Le deuxième volet vise la conception et le développement d'un outil de simulation et de planification de plates-formes de service. Cet outil, d'aide à la décision permettra de consolider la validation de la modélisation proposée et sera capital pour l'introduction de nouveaux services et l'étude économique du TTM et ROI.

Projet ANR SATRIMMAP (2008-2011)

Safety and Time Critical Middleware for future Modular Avionics Platform

Partenaires : Airbus, CEA LIST, IRIT, LAAS-CNRS, ONERA, QoS Design.

Rôle : participant.

Résumé : Les architectures embarquées connaissent depuis une dizaine d'années une mutation profonde avec l'apparition des architectures modulaires intégrées (IMA). L'objectif de ces architectures était d'offrir aux applications embarquées un support d'exécution et de communication standard et mutualisé. Le gain escompté en a été une réduction du poids et plus généralement de la complexité de l'architecture physique. Cette réduction de la complexité au niveau bas s'est cependant traduite par une difficulté accrue de conception et d'intégration des applications. L'objectif du projet est de lever cette difficulté en étudiant et définissant un middleware embarqué au-dessus d'une architecture IMA. Nos travaux ont porté sur l'ordonnancement de tâches strictement périodiques sur les processeurs d'une architecture avionique embarquée et sur le routage optimal des liens virtuels dans un réseau AFDX.

Projet LAAS OSEC (2011-2013)

Ordonnancement des Systèmes Embarqués Critiques

Partenaires : groupes Mogisa, MRS et OLC du LAAS.

Rôle : coordinateur.

Résumé : Le projet OSEC vise à développer de nouvelles approches méthodologiques pour la gestion des incertitudes dans la conception des systèmes embarqués critiques, en particulier dans les domaines de l'automobile et de l'avionique. Le projet prend en compte deux types d'incertitudes : celles sur les paramètres des tâches temps réel et celles sur l'état du système. En ce qui concerne l'incertitude sur les paramètres des tâches temps réel, le projet OSEC vise au

développement d'algorithmes d'optimisation stochastique pour la conception des systèmes embarqués critiques dont les tâches présentent une forte variabilité, de manière à dépasser les limites de la conception basée sur l'analyse au pire. Le projet a également comme objectif d'améliorer la fiabilité des systèmes embarqués critiques grâce à des des mécanismes de reconfiguration en prenant en compte dès leur conception l'incertitude sur l'état du système, liée en particulier aux pannes.

2.2 Encadrement de thèses

Depuis 2003, j'ai co-encadré les thèses de plusieurs étudiants :

Thèse de Charles Bockstal

Titre : Modélisation différentielle du trafic et simulation hybride de réseaux IP-MPLS Diff-Serv

Etablissement : Université Paul Sabatier, Toulouse, 4 Novembre 2005.

Encadrement : JM.GARCIA (80 %) et O.BRUN (20 %).

Jury : G.AUTHIE, H. TIJMS, P. NAIN, P.BRICK, JM.GARCIA, O.BRUN.

Résumé : les travaux de Charles Bockstal portaient sur la modélisation différentielle des réseaux à commutation de paquets et sur la modélisation hybride. Ces recherches ont notamment permis de développer de nouveaux modèles pour les ordonnancements PQ et WFQ et pour les files d'attente multiclasse, pour la modélisation du protocole TCP. Il a également travaillé sur la superposition hybride et l'évaluation de la gigue.

Thèse de Hassan Hassan

Titre : Modélisation et analyse de performances du trafic multimédia dans les réseaux hétérogènes

Etablissement : Université Paul Sabatier, Toulouse, 18 Décembre 2006.

Encadrement : JM.GARCIA (80 %) et O. BRUN (20 %).

Jury : C. BES, D.KOUVATSOS, Z.SUN, G. AUTHIE, JM.GARCIA, O.BRUN.

Résumé : les travaux de Hassan Hassan portaient sur la modélisation des sources de trafic et sur l'étude des propriétés du trafic Internet. Il a également travaillé sur le développement des modèles agrégés permettant de représenter une agrégation de flux TCP.

Thèse de Cédric Fortuny

Titre : Estimation du trafic, planification et optimisation des ressources pour l'ingénierie des réseaux IP/MPLS

Etablissement : Université Paul Sabatier, Toulouse, 1er Juillet 2008.

Encadrement : JM.GARCIA (20 %) et O. BRUN (80 %).

Jury : C. BES, B. FORTZ, P. MAHEY, P. CHEMOUIL, JM.GARCIA, O.BRUN.

Résumé : Cédric Fortuny a travaillé sur l'optimisation des métriques de routage IP, sur l'estimation de matrices de trafic à partir de mesures SNMP et sur le dimensionnement de réseaux.

Thèse de Mohamed Zied Ben Hamouda

Titre : Conception et optimisation robuste des réseaux de télécommunications.

Etablissement : Université Paul Sabatier, Toulouse, 25 Juin 2010.

Encadrement : JM.GARCIA (20 %), O. BRUN (80 %).

Jury : C. BES, B. FORTZ, P. MAHEY, P. CHEMOUIL, M. DIAZ, O. KLOPFENSTEIN, JM.GARCIA, O. BRUN.

Résumé : Ses travaux de recherche ont porté sur la conception de topologie de backbone et sur les problèmes de routage avec incertitude sur la demande.

Thèse de Ahmad Al Sheikh

Titre : Resource allocation in hard real-time avionic systems – scheduling and routing problems.

Etablissement : INSA de Toulouse, Toulouse, 28 Septembre 2011.

Encadrement : O. BRUN (50 %), P.E. HLADIK (50 %).

Jury : F. BONIOL, Y. SOREL, S. BARUAH, J. GOOSENS, O. BRUN, P.E. HLADIK.

Résumé : s'inscrivant dans le cadre du projet SATRIMMAP, les travaux d'Ahmad Al Sheikh ont porté sur l'ordonnancement strictement périodique de fonctions avioniques sur les processeurs embarqués d'une architecture IMA, ainsi que sur le routage des liens virtuels dans un réseau AFDX.

2.3 Enseignement

Depuis 2003, j'effectue environ 20 heures [1] d'enseignement par an au département de Génie Electrique et Informatique de l'INSA Toulouse. J'ai la responsabilité de deux cours en 5ème année "Réseaux et Télécommunications" :

1. Le volume exact a varié en fonction des années et de la disponibilité des intervenants de TP et TD.

– Cours "Evaluation de Performances & Application aux Réseaux de Télécommunications" (15 heures CM) : rappels sur la théorie des probabilités, les processus stochastiques et les chaînes de Markov ; simulation événementielle ; files d'attente ; modélisation des réseaux à commutation de circuits ; modélisation des réseaux à commutation de paquets.

– Cours "Planification de Réseaux" (8 heures CM) : rappels sur l'optimisation convexe, les conditions d'optimalité, la programmation linéraire mixte et la dualité ; formulations noeud-lien et lien-chemin de problèmes de planification de réseaux ; algorithmes de directions admissibles pour l'optimisation du routage ; optimisation des poids OSPF.

En plus des cours magistraux, j'organise et j'interviens dans les TP et les TD de ces enseignements.

De 2006 à 2009, j'ai également assuré la moitié des enseignements (10h CM) du cours " Architectures Distribuées et Applications Parallèles " dans le cadre du M2R SAID de Toulouse, l'autre moitié étant prise en charge par Jean-Marie Garcia.

2.4 Activités de valorisation

Lauréate du 5ème concours national de l'innovation du Ministère de la recherche, la jeune pousse QoS Design a été lancée en avril 2004. Je fais partie des fondateurs de cette société, aux côtés de Jean-Marie Garcia (chercheur LAAS) et de David Gauchard (IR au LAAS). QoS Design conçoit et commercialise des logiciels de modélisation, simulation et d'optimisation des réseaux de télécommunications. Les produits de QoS Design sont utilisés par plusieurs opérateurs de réseaux et constructeurs mondiaux : Alcatel, British Telecom, DIRISI (opérateur interne des réseaux de la Défense), SFR, Maroc Telecom, Tunisie Telecom. Créée pour valoriser les savoir-faire et les compétences du LAAS, QoS Design reste associée à celui-ci en termes de recherche afin de continuer à faire évoluer les technologies appliquées au sein des logiciels de QoS Design. On peut d'ailleurs signaler que les thèses de Cédric Fortuny et Zied Ben Hamouda ont bénéficié d'un financement CIFRE avec QoS Design.

Par ailleurs, nos travaux sur la simulation hybride ont permis de développer un nouveau principe hybride qui a fait l'objet d'un brevet (brevet US N°60/660 439 intitulé « *Method for the digital evaluation of a data transmission network* »). Les inventeurs sont : JM Garcia, D. Gauchard, O. Brun.

Chapitre 3

Evaluation de Performances des Réseaux

Ce chapitre décrit certains de nos travaux sur l'évaluation de performance des réseaux. La présentation suit un ordre chronologique, ce qui permettra au lecteur d'apprécier l'évolution de nos préoccupations scientifiques, qui correspond peu ou prou à celle de la communauté. Le paragraphe 3.1 traite de la modélisation dynamique des trafics, tandis que le paragraphe 3.2 présente nos travaux sur l'analyse des régimes stationnaire et transitoire des files M/D/1/N. Nous présentons au paragraphe 3.3 une approximation pour l'évaluation de la gigue dans les réseaux IP. Le paragraphe 3.4 est consacré à nos travaux sur la modélisation au niveau flot des réseaux TCP/IP. Enfin, nous présentons au paragraphe 3.5 nos travaux sur l'analyse du prix de l'anarchie dans les fermes de serveurs.

3.1 Modélisation dynamique des trafics

La théorie des files d'attente se focalise presque exclusivement sur l'évaluation des performances en régime stationnaire. On dispose ainsi de peu de résulats analytiques sur les performances des files d'attente en régime transitoire.

Les résultats stationnaires permettent de caractériser les performances d'un système sur une échelle de temps longue et sont donc principalement utilisés pour sa conception et son dimensionnement. Cependant, du fait de l'incertitude sur le trafic offert au système et de sa forte variabilité dans le temps, il est difficile d'obtenir à la fois une utilisation satisfaisante des ressources et des performances robustes. Ainsi, des actions de gestion des ressources doivent être prises pour adapter l'utilisation des ressources au trafic offert. Ces actions peuvent souvent être basées sur un modèle transitoire simplifié du comportement du système. Par exemple, [48] a proposé l'utilisation d'un modèle M/M/C pour la gestion adaptative de la capacité des virtual paths (VP) dans les réseaux ATM.

En dépit de son intérêt évident, une approche anlytique pour l'évaluation des performances transitoires des files d'attente est notoiremment difficile. Bien que de nombreux efforts aient été consacrés à ce sujet, peu de résultats généraux sont disponibles et la plupart concernent des approximations pour des systèmes markoviens [2, 118, 119, 144, 143, 166, 169, 187]. Signalons toutefois l'approche développée par Chia-Li Wang pour la file M/G/1 [198].

Nous avons développé une méthodologie générale de modélisation analytique fluide pour l'analyse du régime transitoire des files d'attente. Cette méthodologie s'inscrit dans le cadre de la théorie différentielle du trafic qui a été développée dans les années 80 par JM Garcia pour les réseaux téléphoniques [90]. Nos travaux ont permis d'étendre la théorie du trafic différentiel aux réseaux à commutation de paquets et à commutation de circuits multiclasses. Pour prolonger la modélisation différentielle au delà de ses limites théoriques, nous avons également introduit le concept de simulation hybride distribuée.

Le principe général de la théorie différentielle du trafic est de dériver la forme exacte de l'équation différentielle associée à l'espérance mathématique du trafic de chaque flux dans chaque ressources du réseau. Cette équation différentielle fait intervenir des probabilités transitoires du modèle Markovien dont l'expression analytique n'est en général pas connue. Pour estimer ces probabilités transitoires on introduit des approximations basées sur des relations implicites entre les probabilités d'états et le trafic moyen. Le comportement du réseau peut alors être déterminé, en régime transitoire et en régime stationnaire, par intégration numérique de ces équations différentielles autonomes. Des algorithmes de point fixe permettent aussi de trouver directement la solution stationnaire.

3.1.1 Modélisation des Réseaux à Commutation de Circuits Multiclasses

Les réseaux ATM ou IP IntServ utilisent un mécanisme de contrôle d'admission des appels (CAC), basé sur la notion de bande-passante équivalente, pour garantir la QoS des trafics qu'ils transportent. On parle alors de réseau à commutation de circuits multiclasses [182]. La détermination du trafic écoulé et de la probabilité de blocage de chaque flux d'appels est pour l'opérateur un élément clé dans la planification de son réseau multiservice.

Etude d'un faisceau multiservice isolé

Considérons un faisceau de service de capacité C supportant N classes de trafic. Les appels de classe i arrivent suivant un processus de Poisson de taux λ_i, requièrent la capacité c_i et ont une durée moyenne $1/\mu_i$. Ils ont donc un trafic offert $\rho_i = \lambda_i/\mu_i$. L'objectif est d'évaluer pour chaque classe $i = 1, \ldots, N$ la probabilité de blocage b_i des appels de cette classe, ainsi que le trafic écoulé $X_i = \rho_i(1 - b_i)$. On note $\dot{X}_i(t)$ sa dérivée par rapport au temps. Soit $\mathbf{n} = (n_1, \ldots, n_N)$ l'état du système, où n_i est le nombre d'appels de classe i. On sait que $\mathbf{n} \in \mathcal{S}$

où $\mathcal{S} = \left\{\mathbf{n} \in \mathbb{N}^N : \sum_i n_i c_i \leq C\right\}$ est l'ensemble des états admissibles, et que les probabilités stationnaires ont une forme produit

$$\pi(\mathbf{n}) = \frac{1}{G} \prod_{i=1}^{N} \frac{\rho_i^{n_i}}{n_i!}$$

Si la forme produit permet théoriquement le calcul des probabilités de blocage, elle est limitée par l'explosion combinatoire du nombre d'états. Kaufman [116] et Roberts [180] ont donc proposé une solution plus efficace qui repose sur la discrétisation de la bande-passante C en $M = C/\Delta c$ circuits de capacité $\Delta c = pgcd(c_1, \ldots, c_N)$ et sur le calcul de la probabilité $p(m)$ que m circuits soient occupés. On déduit les probabilités de blocage b_i à partir des probabilités $p(m)$ avec

$$b_i = K_i(C, \rho_1, \ldots, \rho_N) = \sum_{m=M-m_i+1}^{M} p(m) \quad \text{, où } m_i = c_i/\Delta c.$$

L'approximation différentielle est basée sur le résultat suivant [36, 43].

Proposition 3.1

$$\dot{X}_i(t) = \lambda_i \left[1 - b_i(t)\right] \; - \; \mu_i X_i(t) \quad i = 1, \ldots, N \tag{3.1}$$

La difficulté est alors de déterminer la valeur de la probabilité $b_i(t)$ en régime transitoire. Pour cela, nous observons que $\rho_i = X_i(\infty)/(1 - b_i(\infty))$ et prolongeons la validité de cette relation au régime transitoire en introduisant un trafic offert fictif $\rho_i(t)$. On obtient ainsi l'approximation suivante :

Approximation 3.1

$$\dot{X}_i(t) = \lambda_i \left[1 - b_i(t)\right] \; - \; \mu_i X_i(t) \quad i = 1, \ldots, N \tag{3.2}$$

où $b_i(t) = K_i(C, \rho_1(t), \ldots, \rho_N(t))$ *et* $\rho_i(t) = X_i(t)/(1 - b_i(t))$, *pour* $i = 1, \ldots, N$.

L'approximation ci-dessus converge vers la solution stationnaire exacte. De plus, les résultats numériques présentés dans [43] indiquent qu'elle est très précise en régime transitoire.

Etude d'un réseau multiservice

Dans le cas dun réseau, si la forme produit des probabilités reste vraie, son utilisation est complètement inenvisageable. On est donc amené à considérer uniquement des approximations des probabilités de blocage. La plupart des méthodes proposées dans la littérature sont basées sur l'hypothèse d'indépendance des probabilités de blocage et sur l'approximation du trafic réduit [57]. On obtient alors un système d'équations qui peut être résolu par un algorithme de point fixe pour calculer les valeurs approchées des probabilités de blocage.

Notons $b_{i,j}(t)$ la probabilité de blocage d'un appel de classe i au temps t sur le faisceau j et Γ_j l'ensemble des flux passant par ce faisceau. Nous avons proposé l'approximation suivante.

Approximation 3.2

$$\dot{X}_i(t) = \lambda_i \left[1 - b_i(t)\right] \; - \; \mu_i X_i(t) \quad \textit{avec } 1 - b_i(t) = \prod_j \left(1 - b_{i,j}(t)\right) \tag{3.3}$$

où $b_i(t) = K_i(C, \boldsymbol{\rho}_j(t))$ *et* $\rho_{ij}(t) = X_i(t)/(1 - b_{i,j}(t))$*, pour* $i \in \Gamma_j$*.*

Là encore, les résultats numériques présentés dans [43] indiquent qu'elle est très précise en régimes transitoire et stationnaire, sous réserve que l'hypothèse d'indépendance des probabilités de blocage soit bien vérifiée.

3.1.2 Modélisation des Réseaux à Commutation de Paquets

Nous avons établi tout un ensemble de modèles différentiels pour des systèmes élémentaires Markoviens et non Markoviens permettant de représenter les organes d'un réseau à commutation de paquets : files $M/M/1$, $M/M/1/N$, $M^K/M^K/1$, $M/G/1$, $M/D/1/N$, $G/M/1$ et même $GI/GI/1$. Une partie de ces travaux a été publiée dans [91].

A titre d'exemple, nous décrivons ci-dessous l'approximation obtenue pour une file M/G/1. On note T la durée moyenne de service, $C_b^2 \neq 1$ son coefficient de variation et on suppose que $\lambda T < 1$, où λ est le taux d'arrivée. Soit $X(t)$ le nombre moyen de clients et $\dot{X}(t)$ sa dérivée par rapport au temps. Le point de départ est l'équation différentielle suivante

$$\dot{X}(t) = \lambda - \frac{1}{T}\left[1 - P_0(t)\right]. \tag{3.4}$$

Cette équation fait intervenir la probabilité transitoire $P_0(t)$ qui est inconnue. En prolongeant au régime transitoire les relations stationnaires $P_0(\infty) = 1 - \rho$ et $X(\infty) = \rho + \frac{\lambda^2 \overline{x^2}}{2\,(1-\rho)}$, on obtient l'approximation suivante.

Approximation 3.3

$$\dot{X}(t) \approx \lambda \; - \; \frac{1}{T} \frac{-2(1 + X(t)) \; + \; \sqrt{4(1 + X(t))^2 \; + \; 8X(t)(C_b^2 - 1)}}{2(C_b^2 - 1)}. \tag{3.5}$$

La Figure 3.1.2 illustre l'approximation transitoire obtenue pour plusieurs valeurs du taux d'utilisation dans le cas d'une loi de service uniforme.

Un des intérêts de ces modèles différentiels est qu'ils permettent de déterminer la réponse transitoire du système à des phénomènes dynamiques tels que par exemple des variations de trafic, comme illustré sur la Figure 3.1.2 dans le cas d'une file M/M/1.

La théorie du trafic différentiel a de plus l'avantage de fournir des approximations du transitoire d'un coût calculatoire bien plus faible que celles obtenues avec des processus de diffusion, et qui se comporte correctement quel que soit le taux d'utilisation. Soit $z(t)$ le processus de

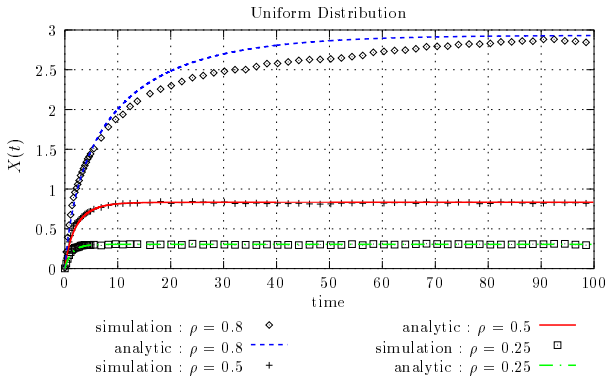


FIGURE 3.1 – Approximation différentielle d'une file M/G/1 avec loi de service uniforme.

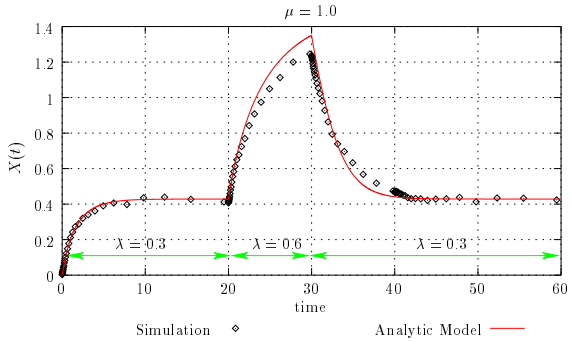


FIGURE 3.2 – Réponse transitoire d'une file M/M/1 à des variations de trafic.

diffusion approximant le nombre de clients dans une file d'attente ayant des durées inter-arrivées de moyenne μ_a et de coefficient de variation au carré C_a^2, et des durées de service de moyenne μ_b et de coefficient de variation au carré C_b^2. La densité de probabilité associée à $z(t)$ sachant que $z(0) = z_0$ est donnée par [126]

$$p(z,t;z_0,0) \quad = \quad (\partial/\partial x)\left\{\Phi\left(\frac{z-z_0-\beta t}{\sqrt{\alpha t}}\right) \; - \; \exp\left(\frac{2\beta z}{\alpha}\right)\Phi\left(-\frac{z+z_0+\beta t}{\sqrt{\alpha t}}\right)\right\}$$

où $\Phi(z) = \displaystyle\int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)\,du$, $\alpha = \dfrac{C_a^2}{\mu_a} + \dfrac{C_b^2}{\mu_b}$ et $\beta = \dfrac{1}{\mu_a} - \dfrac{1}{\mu_s}$. Des comparaisons des réponses transitoires d'une file M/M/1 obtenues avec le modèle différentiel, avec cette approximation diffusion et avec des simulations événementielles sont proposées dans la Figure 3.1.2, quand l'état initial est $x_0 = 0$. Pour $\rho = 0.8$, l'approximation diffusion est légèrement plus précise, mais elle est très grossière pour $\rho = 0.2$. Il s'agit là d'un défaut bien connu des approximations diffusion qui est dû à l'hypothèse sous-jacente d'indépendance des processus d'arrivées et de départs [124]. L'approximation diffusion, utilisée en fort trafic, a néanmoins deux avantages importants : elle est plus précise quand l'état initial $x_0 \neq 0$, et elle offre une approximation de la distribution

de probabilités (et non seulement de sa moyenne).

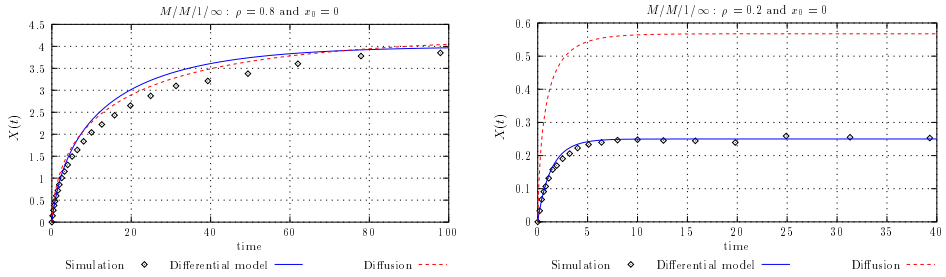


FIGURE 3.3 – Réponses transitoires d'une file M/M/1 obtenues avec le modèle différentiel, l'approximation diffusion et avec des simulations événementielles quand $x_0 = 0$.

Signalons enfin que des approximations différentielles peuvent être développées pour évaluer le comportement transitoire des réseaux de Jackson ou de Gordon-Newell. Ainsi, dans le cas d'un réseau ouvert de Jackson, l'équation différentielle gouvernant le nombre de clients au noeud i peut être approximée par

$$\dot{X}_i(t) \approx \lambda_i + \sum_{j=1}^{n} r_{j,i}\,\mu_j \left(\frac{X_j(t)}{1+X_j(t)}\right) - \mu_i\,(1-r_{i,i}) \left(\frac{X_i(t)}{1+X_i(t)}\right), \tag{3.6}$$

où λ_i est le taux d'arrivée au noeud i depuis l'extérieur et $r_{i,j}$ est la probabilité qu'un client sortant du noeud i aille vers le noeud j. L'approximation ci-dessus converge vers la solution stationnaire exacte et les résultats numériques présentés dans [91] indiquent qu'elle est très précise en régime transitoire.

3.1.3 Simulation Hybride Distribuée

Pour prolonger la modélisation analytique différentielle au delà de ses limites théoriques, nous avons introduit un nouveau concept de simulation hybride qui combine rigoureusement modèles analytiques et modèles stochastiques à événements discrets [91, 88]. Le comportement dynamique de certains noeuds est obtenu par intégration numérique d'un système d'équations différentielles (noeuds analytiques), tandis que celui des noeuds complexes est obtenu par des simulations événementielles (noeuds simulés). L'idée fondamentale est qu'un noeud complexe va estimer (grâce à une simulation stochastique) le comportement de sa propre équation différentielle. Le modèle global du réseau est alors résolu par intégration numérique d'un ensemble d'équations différentielles couplées.

Le modèle global est résolu par intégration numérique d'un système d'équations différentielles couplées. Au cours d'un pas d'intégration, les calculs suivants sont réalisés :

– les équations différentielles des noeuds analytiques sont intégrées,

– des simulations stochastiques des noeuds complexes sont réalisées pendant l'intervalle de temps correspondant,
– l'échange entre les noeuds (ou ensemble de noeuds) des débits instantanés et des variances des flots est réalisé.

De cette façon, la décomposition du réseau en noeuds analytiques et en noeuds simulés permet de réduire les temps de calcul tout en préservant le caractère général de l'approche de simulation. Cette approche, qui a donné lieu à la réalisation d'un prototype dans le cadre du projet Esprit STAR, est complètement parallèle avec un premier niveau de parallélisation sur l'ensemble des noeuds (gros grain) et un deuxième niveau de parallélisation sur l'espace des trajectoires (grain fin).

Par la suite, nous avons également introduit un nouveau principe de simulation hybride permettant de simuler la plupart des flots de manière analytique et quelques flots avec des simulations événementielles de bout-en-bout [93]. Ce principe de simulation hybride a fait l'objet du dépôt d'un brevet qui est exploité par la start-up QoS Design.

3.2 Analyse de la file M/D/1/N

Les traitements réalisés dans les systèmes informatiques ou de communication modernes ont souvent des durées déterministes. De ce fait, l'évaluation de leurs performances fait souvent appel à des modèles de files d'attente à temps de service déterministe (pour la modélisation des multiplexeurs ATM par exemple). Suivant le système à modéliser, différents modèles ont ainsi été considérés (cf. [183], [196] et [100]) : M/D/1, Geo^N/D/1, N*D/D/1 et $\sum D_i$/D/1. La file M/D/1 [123] est de loin le plus simple et le plus général de ces modèles, avec de nombreuses applications aux systèmes de communication, mais aussi aux systèmes de production ou encore en informatique. Ce modèle, déjà étudié par Erlang en 1909, correspond à un système d'attente avec des arrivées suivant un processus de Poisson et une durée de service déterministe (i.e. constante). Bien que ce type de modèle soit bien résolu d'un point de vue numérique [195], il existait finalement assez peu de résultats concernant la file à capacité limitée (modèle M/D/1/N dans la notation de Kendall). En effet, en dehors du cas N=1 (pas d'attente) qui est un cas particulier du modèle d'Erlang, il n'y avait aucune expression explicite pour la distribution du nombre de clients dans le système, ni même pour sa moyenne. Nos travaux ont permis de développer la solution analytique exacte des files M/D/1/N en régime transitoire [89] et en régime stationnaire [42].

3.2.1 Solution stationnaire

Le modèle M/D/1/N correspond à une file d'attente avec un serveur et N-1 positions d'attente. Les clients arrivent suivant un processus de Poisson de taux λ, i.e., la probabilité de k

arrivées en un temps t est $\alpha_k(t) = (\lambda t)^k e^{-\lambda t}/k!$. Pour simplifier, on utilisera la notation abrégée α_k en lieu et place de $\alpha_k(T)$. Les clients sont servis suivant la discipline FCFS, la durée de service étant constante et égale à T unités de temps. Les clients qui arrivent alors que la file d'attente est pleine sont rejetés et n'ont plus d'influence sur le système ("lost customer cleared"), ce qui explique qu'aucune hypothèse *a priori* sur le taux d'utilisation $\rho = \lambda T$ ne soit nécessaire.

Chaîne de Markov incluse

Soit $X_N(t)$ le nombre de clients dans le système au temps t. Notons t_n la date de départ du n^{eme} client. Le processus stochastique $\{X_N(t_n)\}_{n\geq 0}$ est une chaîne de Markov sur $\{0, 1, \ldots, N-1\}$ dont l'analyse par des techniques classiques de transformée en z ne pose pas de problème particulier. On obtient ainsi le résultat suivant :

Proposition 3.2 *A l'équilibre, la distribution de probabilités* $\mathbf{q} = (q_0, \ldots, q_n)$ *du nombre de clients dans le système immédiatement après un départ est donnée par* $q_0 = \frac{1}{b_{N-1}}$ *et* $q_n = \frac{b_n - b_{n-1}}{b_{N-1}}$ *pour* $n \in \{1, \ldots, N-1\}$, *où* $b_0 = 1$ *et*

$$b_n = \sum_{k=0}^{n} \frac{(-1)^k}{k!}(n-k)^k e^{(n-k)\rho}\rho^k \ , \forall n \geq 1. \tag{3.7}$$

Distribution de probabilités stationnaire

Il faut signaler ici que, contrairement au cas de la file $M/D/1/\infty$, pour la file $M/D/1/N$ la distribution stationnaire $\mathbf{q}$ aux instants de départ diffère de la distribution stationnaire à tout instant $P = \big(P_0(N), \ldots, P_N(N)\big)$. Cependant, Cohen a démontré dans le cas plus général des files $M/G/1/N$ [60], que ces deux distributions sont reliées par la relation $q_j = P_j(N)/(1 - P_N(N))$, pour $j = 0 \cdots N-1$. En utilisant cette relation et la Proposition 3.2, on obtient les deux résultats ci-dessous.

ThÈorËme 3.1 *La distribution de probabilités du nombre de clients dans une file* $M/D/1/N$ *est donnée par* $P_j(N) = (b_j - b_{j-1})\ P_0(N)$ *pour* $j = 1 \ldots N-1$ *et* $P_N(N) = 1 - b_{N-1}\, P_0(N)$ *où* $P_0(N) = 1/(1 + \rho b_{N-1})$.

Corollaire 3.1 *Le nombre moyen de clients est* $X(N) = N \ - \ P_0(N) \sum_{k=0}^{N-1} b_k$.

3.2.2 Solution transitoire

Définissons $D_k(t, t+\Delta t)$ comme la probabilité qu'un départ ait lieu entre les instants t et $t + \Delta t$ en laissant k clients dans le système $(0 \leq k \leq N-1)$, ainsi que les taux de départ

$$d_k(t) = \lim_{\Delta t \to 0} \frac{D_k(t, t+\Delta t)}{\Delta t} \quad , k = 0 \cdots N-1.$$

On notera $\mathbf{\Psi}(t) = (d_0(t), \ldots, d_{N-1}(t), 0)$. Enfin, définissons la matrice carrée d'ordre $N+1$

$$\mathbf{W} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \tag{3.8}$$

Notre résultat principal est le suivant.

Proposition 3.3 *La distribution de probabilités $\boldsymbol{\pi}(t) = (\pi_0(t), \ldots, \pi_N(t))$ du nombre de clients dans le système au temps t est donnée par*

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)\, e^{\lambda \mathbf{W} t} \; - \; \int_0^t \mathbf{\Psi}(s)\, \mathbf{W}\, e^{\lambda \mathbf{W}(t-s)}\, ds \quad , \; \forall t \geq 0. \tag{3.9}$$

De plus,

$$e^{\lambda \mathbf{W} t} = \begin{pmatrix} \alpha_0(t) & \alpha_1(t) & \cdots & \alpha_{N-1}(t) & 1 - \sum_0^{N-1} \alpha_k(t) \\ 0 & \alpha_0(t) & \cdots & \alpha_{N-2}(t) & 1 - \sum_0^{N-2} \alpha_k(t) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \alpha_0(t) & 1 - \alpha_0(t) \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \tag{3.10}$$

La Proposition 3.3 permet le calcul de $\boldsymbol{\pi}(t)$ à partir du moment où l'on connaît le vecteur des taux de départ $\mathbf{\Psi}(t)$. Le Lemme ci-dessous décrit comment calculer ces taux de départ dans le cas où le système est initialement vide, i.e. $\pi_0(0) = 1$ (cf. [89] pour le cas général).

Lemme 3.1 *Pour $t \leq T$, on a $d_k(t) = 0$ pour $k = 0 \cdots N-1$. Pour $t > T$, on a*

$$d_k(t) \; = \; \lambda\, \alpha_k\, \pi_0(t-T) \; + \; \sum_{i=1}^{k+1} \alpha_{k+1-i}\, d_i(t-T) \quad , k = 0 \cdots N-2 \tag{3.11}$$

$$d_{N-1}(t) \; = \; \lambda \left[1 - \sum_{i=0}^{N-2} \alpha_i\right] \pi_0(t-T) \; + \; \sum_{i=1}^{N-1} \left[1 - \sum_{q=0}^{N-1-i} \alpha_q\right] d_i(t-T). \tag{3.12}$$

A eux deux, la Proposition 3.3 et le Lemme 3.1 fournissent un moyen efficace de calculer la distribution transitoire $\boldsymbol{\pi}(t)$. Dans la Figure 3.4, nous présentons l'évolution de cette distribution pour une file de capacité $N = 2$ ayant un taux d'arrivée $\lambda = 3$ et une durée de service $T = 1$.

3.3 Analyse de la gigue des flux audio et vidéo

De nombreux codecs voix ou vidéo génèrent des paquets de données de taille constante suivant un processus périodique, i.e. à intervalles réguliers. Une mesure de performance importante

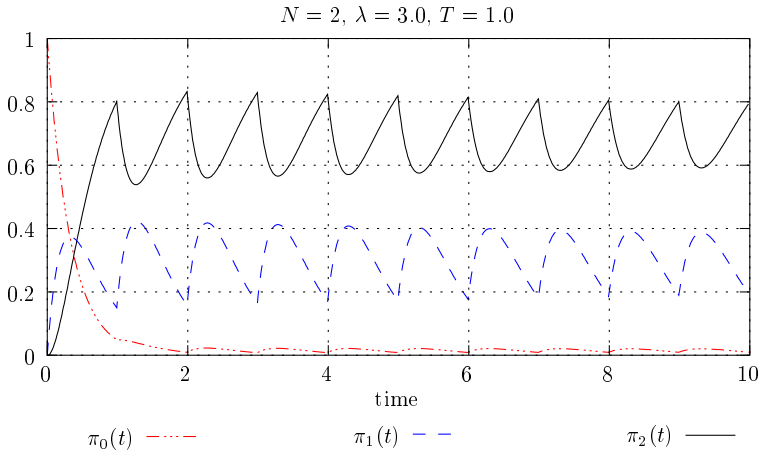


FIGURE 3.4 – Distribution transitoire $\boldsymbol{\pi}(t)$ pour une file M/D/1/2 avec $\lambda = 3.0$ et $T = 1.0$.

pour ces trafics est la gigue qui peut être définie comme la distorsion de la nature périodique du flux de paquets entre la source et la destination. Cette distorsion du signal originel est due aux opérations de multiplexage réalisées par les routeurs traversés.

De nombreuses études ont été consacrées à l'analyse de la gigue dans les réseaux ATM, essentiellement avec des modèles à temps discret. Certains auteurs ont étudié la gigue au moyen de simulations événementielles [35, 6]. D'autres ont obtenu des bornes avec des approches de type Network Calculus, soit dans un cadre déterministe [64, 65, 173, 172], soit dans un cadre stochastique [135, 199]. Enfin, de nombreux auteurs ont proposé des méthodes numériques basées sur la théorie des files d'attente pour l'estimation de la gigue [58, 70, 104, 103, 102, 137, 138, 139, 146, 158, 159, 175, 184, 185]. L'inconvénient principal de ces méthodes est qu'elles ont un coût calculatoire assez important. A notre connaissance, seul [160] propose une approximation analytique simple basée sur des approximations en fort et faible trafics.

De notre point de vue, il est essentiel d'avoir une formule simple et suffisamment précise pour pouvoir : (a) simuler des réseaux avec des milliers de flux temps réel, ou (b) exécuter des algorithmes d'optimisation itératifs intégrant des contraintes de QoS. Dans [39], nous avons proposé une telle formule pour évaluer la gigue de bout-en-bout subie par une trafic périodique ayant une taille de paquet constante. On suppose que ce trafic est multiplexé en chaque noeud avec un trafic de fond, les paquets étant transmis suivant la discipline FCFS. La seule hypothèse concernant ces trafics de fond est que les processus gouvernant les arrivées et les tailles des paquets de ces trafics sont des processus de renouvellement.

3.3.1 Notations

Comme décrit sur la Figure 3.5, nous considérons un trafic passant à travers n files d'attente en série. Ce trafic est initialement périodique, i.e. on suppose une durée constante $T = 1/\lambda_T$ entre deux paquets au noeud 1. Tous les paquets de ce trafic ont la même taille : D unités de données. Le taux de service du noeud $k = 1 \dots n$ est μ_k unités de données par seconde. Dans chaque noeud, le trafic étudié est multiplexé avec un trafic de fond suivant la discipline FCFS. Pour des raisons de stabilité, on suppose le taux d'utilisation de chaque noeud inférieur à 100%.

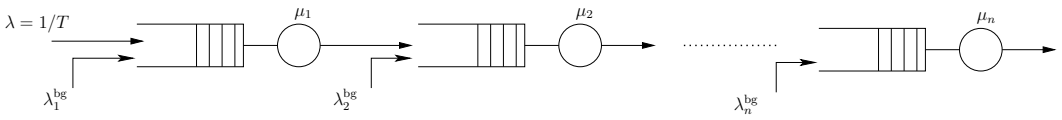


FIGURE 3.5 – Le trafic étudié est multiplexé avec un trafic de fond en chaque noeud.

Considérons deux paquets consécutifs du trafic étudié, disons les paquets 0 et 1. Notons $\tau_j^{\text{in}}(k)$ et $\tau_j^{\text{out}}(k)$ les instants d'arrivée et de départ du paquet $j = 0, 1$ au noeud $k = 1, \dots, n$. On pourra sans inconvénient supposer que $\tau_0^{\text{in}}(1) = 0$. De même, on note $W_j(k)$ le temps d'attente du paquet j au noeud k et $\Delta_k = W_1(k) - W_0(k)$ la différence des temps d'attente au noeud k.

La gigue de bout-en-bout est définie comme $J_{[1\dots n]}(T) = E\left[\ |(\tau_1^{\text{out}}(n) - \tau_0^{\text{out}}(n)) \ - \ T\ |\ \right]$. Puisque $\tau_i^{\text{in}}(k+1) = \tau_i^{\text{out}}(k)$ pour $i = 0, 1$ et $k = 1, \dots, n-1$, on a

$$\begin{aligned} J_{[1\dots n]}(T) &= E\left[\ |(\tau_1^{\text{out}}(n) - \tau_0^{\text{out}}(n)) \ - \ \sum_{k=1}^{n-1} (\tau_1^{\text{in}}(k+1) - \tau_0^{\text{in}}(k+1)) \right. \\ &\qquad \left. + \ \sum_{k=1}^{n-1} (\tau_1^{\text{out}}(k) - \tau_0^{\text{out}}(k)) \ - \ (\tau_1^{\text{in}}(1) - \tau_0^{\text{in}}(1))\ |\ \right], \\ &= E\left[\ |\sum_{k=1}^{n} (W_1(k) - W_0(k))\ |\ \right], \\ &= E\left[\ |\sum_{k=1}^{n} \Delta_k\ |\ \right]. \end{aligned} \tag{3.13}$$

On voit ainsi que la gigue de bout-en-bout correspond à la valeur absolue de la somme $\sum_k \Delta_k$ des variations du délai inter-paquet introduites par chacun des noeuds le long du chemin. Dans la suite, on va obtenir une approximation de la densité de probabilité $f_{\Delta_1+\dots+\Delta_n}(x)$ de cette somme.

3.3.2 Le cas d'un seul noeud

Considérons tout d'abord le cas d'un seul noeud, disons le noeud k. Si la file d'attente n'est pas trop chargée, le paquet 1 entre dans la file alors que le paquet 0 est déjà parti, i.e.

$W_0(k) \leq \hat{T}_k = T - D/\mu_k$. Dans ce cas, on peut voir $W_1(k)$ et $W_0(k)$ comme des variables aléatoires i.i.d. dont la distribution est celle du temps d'attente à l'équilibre de la file d'attente. Il est bien connu que la queue de la distribution du temps d'attente W dans une file GI/GI/1 a souvent une forme exponentielle, i.e., $P[W > x] = \alpha\, e^{-\eta\, x}$ pour x assez grand, η et α étant des paramètres positifs fixés. De telles approximations sont très largement disponibles dans la littérature consacrée aux files d'attente [1, 148, 195, 190] et on sait qu'elles sont souvent d'une qualité remarquable même pour de petites valeurs de x. Dans nos travaux, nous avons utilisé une approximation proposée par Abate et al. [1], qui dépend des trois premiers moments des processus d'arrivées et de service et suppose que ces processus aléatoires sont des processus de renouvellement. Dans la suite, les paramètres η_k et α_k vont résumer toute l'information sur le trafic de fond au noeud k.

Avec cette approximation exponentielle pour la distribution de $W_0(k)$ et $W_1(k)$, la densité de probabilité du temps d'attente s'écrit

$$f_{W(k)}(x) = (1 - \alpha_k)\, u_0(x) \;+\; \alpha_k\, \eta_k\, e^{-\eta_k\, x}. \tag{3.14}$$

où u_0 représente l'impulsion unité. On en déduit l'approximation suivante.

Approximation 3.4 *Posons $\phi_k = 2\,\alpha_k\,(1-\alpha_k/2)$. Si le délai inter-paquet T à l'entrée du noeud k est suffisamment grand, on peut approximer la densité de probabilité de la variation Δ_k de ce délai par*

$$f_{\Delta_k\,|\,T}(x) = \begin{cases} 0 & ,x \leq -\hat{T}_k \\ \frac{\phi_k \eta_k}{2}\, e^{-\eta_k |x|}\, [1 - u_0(x)] \;+\; (1 - \phi_k)\, u_0(x) & ,-\hat{T}_k < x \end{cases}, \tag{3.15}$$

3.3.3 Le cas de plusieurs noeuds

Nous considérons maintenant le cas où le trafic étudié traverse plusieurs noeuds en série. Connaissant les lois gouvernant les arrivées et les tailles des paquets des trafics de fond aux noeuds $k = 1, \ldots, n$, les paramètres α_k, $\phi_k = 2\alpha_k(1 - \alpha_k/2)$ et η_k peuvent être calculés pour chaque noeud. Pour simplifier, nous supposons que $\eta_i \neq \eta_j$ pour $i \neq j$.

L'approximation 3.4 permet de calculer de manière approchée la distribution de la gigue introduite par un noeud quand on connaît le délai inter-paquet à l'entrée de ce noeud. Le calcul de la distribution de la gigue introduite par une série de noeuds se ramène alors à celui d'un produit de convolutions, et il est possible de le réaliser analytiquement du fait de la forme très simple de l'approximation 3.4. Le résultat obtenu est décrit ci-dessous.

Approximation 3.5 *Si le délai inter-paquet T initial est suffisamment grand, la densité de probabilité $f_{\Delta_1+\ldots+\Delta_n}(x)$ peut être approchée par*

$$f_{\Delta_1+\ldots+\Delta_n}(x) \approx \begin{cases} 1 - \sum_{j=1}^{n} \phi_j \, K_j^n & , \; x = 0 \\ \sum_{j=1}^{n} K_j^n \, f_{\Delta_j \,|\, T}(x) & , \; x \neq 0 \end{cases},$$

où

$$K_j^n = \prod_{i=1, i\neq j}^{n} \left(1 - \phi_i \, \frac{\eta_j^2}{\eta_j^2 - \eta_i^2} \right) \quad j = 1 \ldots n.$$

L'approximation 3.5 permet d'obtenir une expression très simple de la gigue moyenne de bout-en-bout, comme décrit ci-dessous.

Proposition 3.4 *En considérant l'approximation 3.5 exacte, la gigue moyenne introduite par les noeuds* $1, \ldots, n$ *est donnée par*

$$J_{[1\ldots n]}(T) = \sum_{j=1}^{n} K_j^n \, J_j(T), \tag{3.16}$$

où $J_j(T)$ *est la gigue moyenne qui serait introduite par le noeud* j *si le délai inter-paquet en entrée de ce noeud était égal à* T*, i.e.,*

$$J_j(T) = \frac{\phi_j}{\eta_j} \left[1 \; - \; \frac{1}{2} \left(1 + \eta_j T'_j \right) e^{-\eta_j \, T'_j} \right].$$

Ainsi, la gigue moyenne de bout-en-bout peut être vue comme une somme pondérée des gigues qui serait introduites par chaque noeud si le délai inter-paquet était T à l'entrée de chacun de ces noeuds.

3.3.4 Exemple d'application

Pour illustrer la qualité de l'approximation obtenue, nous considérons la situation où un flux voix généré avec un codec G.711 traverse 4 liens en série, de capacités $\mu_1 = \mu_4$=1920 kbps et $\mu_2 = \mu_3$=7680 kbps. On suppose que le trafic de fond sur les liens 1 et 4 correspond à l'aggrégation de flux voix générés avec deux codecs : 10% de flux G.726 et 90% de flux G.729. Pour les liens 2 et 3, le trafic de fond est obtenu en aggrégeant 50% de chacun de ces types de flux. De plus, les liens 2 et 3 ayant beaucoup plus de bande-passante, on suppose que le débit de leur trafic de fond est environ trois fois supérieur à ceux du trafic de fond sur les liens 1 et 4 (des nombres entiers de flux G.726 et G.729 sont necéssaires pour les simulations événementielles). Pour le modèle analytique, on approxime ces trafics de fond avec des arrivées suivant un processus de Poisson et une distribution de tailles de paquets D_2 à deux valeurs, les paramètres de ces distributions

dépendant du mélange considéré.

La Figure 3.6 compare la gigue moyenne obtenue avec l'approximation analytique 3.16 et celle obtenue de manière empirique grâce à des simulations événementielles en fonction du débit du trafic de fond sur le premier (et le dernier) lien. L'approximation s'avère assez précise. L'erreur est en dessous de 1% quand le taux d'utilisation du lien 1 est inférieur à 46%. Elle est seulement de 2.08% si ce taux d'utilisation est de 53%, et de 3.46% si le taux d'utilisation est de 60%.

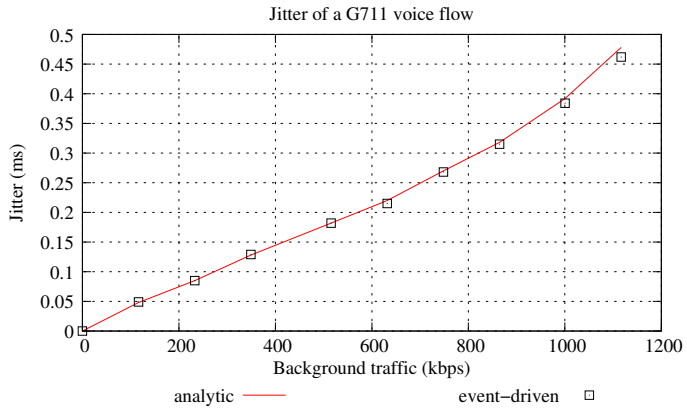


FIGURE 3.6 – Approximation de la gigue d'un flux voix G.711 traversant 4 liens.

3.4 Modélisation au niveau flot du trafic Internet

Un effort de recherche important a été consacré ces dernières années au développement d'une théorie du trafic pour l'Internet. En comparaison avec son équivalent pour les réseaux téléphoniques, cette théorie du trafic Internet est encore très limitée. La principale difficulté de modélisation provient de la nature élastique de la plupart des trafics Internet dont le débit peut être modulé par le protocole TCP en fonction des conditions de trafic dans le réseau. Deux grandes catégories de modèles ont été développés : les modèles de niveau flot et ceux de niveau paquet.

Les modèles de niveau flot sont des modèles idéalisés qui prennent en compte la dynamique aléatoire au niveau des flots (arrivées et départs de connexions), mais qui utilisent un modèle très simplifié du partage de bande-passante réalisé par TCP (voir [181] pour un survey). En se basant sur l'observation que TCP partage la bande-passante de manière à peu près équitable, [114] et [154] ont proposé d'utiliser la file d'attente processor-sharing $M/G/1/PS$ comme modèle de performance pour les réseaux d'accès. Cette direction de recherche a été poursuivie par la suite pour montrer l'insensibilité des résultats obtenus aux caractéristiques fines du trafic [81, 31, 32], pour étudier les questions de stabilité [30], pour prendre en compte de manière plus précise le fonctionnement de TCP [3, 20, 121] et pour généraliser ce modèle à des topologies de réseau

quelconques [155, 33, 32, 142].

Les modèles de niveau paquet intégrent eux de nombreux détails sur le système (Round Trip Times, taille des buffers, etc.), mais suppose un nombre constant de flux persistants. Un résultat important est la formule "square-root" pour le débit d'une connexion TCP [157, 171], qui a notamment permis le développement de plusieurs algorithmes de point-fixe pour le calcul du débit des connexions et des probabilités de blocage des liens (voir par exemple [8] et [49]). Un effort de recherche important a aussi été consacré à l'analyse d'algorithmes de contrôle de congestion tel que celui de TCP sur la base de l'optimisation de fonctions d'utilité [117, 134, 150, 163, 197].

Plusieurs approches ont également été proposées pour tenter de réduire l'écart entre modèles de niveau paquet et modèles de niveau flot [141, 140], mais aussi pour intégrer le trafic de streaming dans ce type de modèles [168, 34]. Malgré tout, et en dépit des avancées significatives que nous avons mentionnées ci-dessus, la théorie du trafic Internet en est encore à ses balbutiements. Il n'existe pas à l'heure actuelle de modèles de performance satisfaisants permettant d'évaluer précisément les performances d'une architecture TCP/IP. Nous avons participé à cet effort en proposant dans [38] :

- de nouveaux modèles de réseaux d´accès représentant de manière détaillée la dynamique des sessions utilisateurs et prenant en compte les limites de débit des terminaux utilisateurs,
- un nouveau paradigme de partage de bande-passante basé sur une allocation de type weighted max-min fair,
- une approximation de type point-fixe permettant d'étendre les modèles de réseaux d'accès pour prendre en compte la congestion dans le coeur de réseau.

3.4.1 Modèles de Réseaux d'Accès

Nous considérons ici un grand nombre d'abonnés ADSL ou UMTS répartis sur une même zone géographique et connectés au backbone réseau d'un opérateur par un même lien d'accès full-duplex de capacité C Mbps, constituant le goulot d'étranglement des flux (cf figure 3.7). Le débit d'un terminal utilisateur est limité à c_{up} Mbps et c_{down} Mbps dans les sens montant et descendant, respectivement, et on note $C = N_{up}\, c_{up} = N_{down}\, c_{down}$.

Les utilisateurs ont accès à un ensemble $\mathcal{R}$ d'applications r, qui se divisent en deux grandes catégories. D'une part, un ensemble $\mathcal{B} = \mathcal{U} \cup \mathcal{D}$ d'**applications de transfert de fichier** (e-mail,ftp), avec lesquelles l'utilisateur télécharge (en upload pour $r \in \mathcal{U}$ ou en download pour $r \in \mathcal{D}$) un seul fichier, de taille aléatoire. D'autre part, un ensemble $\mathcal{I}$ d'**applications interactives** (web browsing), au cours desquelles les utilisateurs alternent entre des périodes de téléchargement de fichiers en download, et des périodes OFF pendant lesquelles ils lisent le contenu des documents téléchargés. De nouvelles sessions sont initiées par les utilisateurs suivant

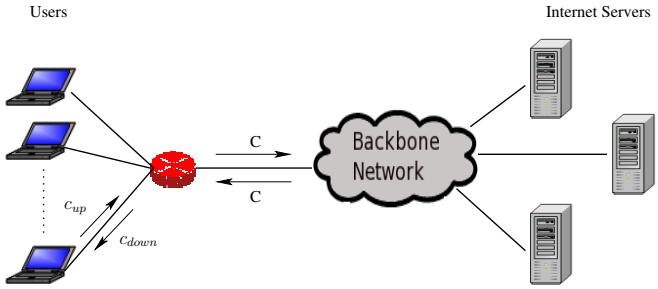


FIGURE 3.7 – Les abonnés Internet partagent un lien de communication à C Mbps.

un processus de Poisson au taux λ (hypothèse confirmée par des mesures sur l'Internet [72]). Une nouvelle session utilise l'application r avec la probabilité γ_r. Suivant le type d'application, elle transfère alors un ou plusieurs fichiers dont les tailles sont des variables aléatoires i.i.d, de moyenne $1/\alpha_r$ (en Mbits). Le nombre de transferts de fichiers au cours d'une session est aussi une variable aléatoire i.i.d., de moyenne $1/p_r$. Par convention, $p_r = 1$ si $r \in \mathcal{B}$. On suppose que le nombre de fichiers téléchargés par une application interactive $r \in \mathcal{I}$ suit une loi géométrique, et nous notons $1/\beta_r$ la durée moyenne de la période OFF pour cette application.

Une de nos contributions majeures est d'observer que le système peut être modélisé par un réseau multiclasse équivalent de files d'attente, comme décrit sur la figure 3.8. Ceci a déjà été observé précédemment (voir [81] par exemple) dans un contexte moins général, mais l'idée n'a pas été complétement exploitée à notre avis. Les clients dans ce réseau de files d'attente représentent des sessions Internet en cours. Les arrivées exogènes correspondent à de nouvelles sessions initiées par les utilisateurs, tandis que les départs du réseau représentent la fin de sessions Internet. Il y a plusieurs classes de client, chacune étant associée à une application $r \in \mathcal{R}$. Chaque noeud dans ce réseau représente une phase possible d'une session en cours. On note $\mathcal{S} = \{up, down, off\}$ l'ensemble des noeuds de ce réseau équivalent, $\mathcal{S}_a = \{up, down\}$ les phases actives, et $\Omega(s)$ l'ensemble des applications qui peuvent être en phase s : $\Omega(up) = \mathcal{U}$, $\Omega(down) = \mathcal{D} \cup \mathcal{I}$ et $\Omega(off) = \mathcal{I}$. Le taux d'utilisation de chaque file d'attente s est $\rho_s = \sum_{r \in \Omega(s)} \rho_s^r$, où $\rho_s^r = \lambda\gamma_r/(p_r\alpha_r)$ pour $s \in \mathcal{S}_a$ et $\rho_{off}^r = \lambda\gamma_r/(p_r\beta_r)$.

Pour prendre en compte les limites des terminaux, nous supposons que les noeuds UP et DOWN sont des files GPS (*Generalized Processor Sharing*) [59]. Quand il y a k sessions en phase $s \in \mathcal{S}_a$, une connexion individuelle obtient le taux de service $f_s(k) = min(c_s, C/k)$, c'est à dire qu'une session en phase s est servie au débit maximal du terminal si $k \leq N_s$, et suivant la discipline processeur sharing ordinaire sinon. La solution du modèle GPS, telle que développée par Cohen [59], fait apparaître les fonctions $\Phi_s(k) = \prod_{j=1}^{k} 1/f_s(j)$, $k \geq 1$ avec $\Phi_s(0) = 1$. Dans notre cas, cela donne pour $s \in \mathcal{S}_a$, $\Phi_s(k) = (1/c_s)^k$ pour $k \leq N_s$, et $\Phi_s(k) = \frac{(C/c_s)^{N_s}}{N_s!}\, k!\, (1/C)^k$ sinon.

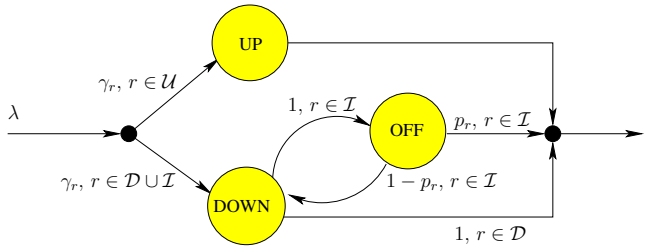


FIGURE 3.8 – Réseau ouvert équivalent.

De même le noeud OFF est modélisé comme une file GPS de capacité infinie ayant la fonction de taux de service $f_{off}(k) = 1,\ k \geq 1$ (voir [59], section 5), et donc $\Phi_{off}(k) = 1,\ k \geq 1$.

Soit x_s^r le nombre de sessions de l'application $r \in \Omega(s)$ dans la phase $s \in \mathcal{S}$. Définissons les vecteurs suivants : $\mathbf{x}_s = (x_s^r)_{r\in\Omega(s)},\ s \in \mathcal{S}$ et $\mathbf{x} = (\mathbf{x}_{up}, \mathbf{x}_{down}, \mathbf{x}_{off})$. Soit $\pi(\mathbf{x})$ la probabilité stationnaire si elle existe que le système soit dans l'état $\mathbf{x}$. Finalement, notons X_s (resp. X_s^r) le nombre total de sessions (resp. de l'application r) en phase s. Le réseau de files d'attente décrit ci-dessus peut être analysé comme un réseau de files d'attente GPS. En utilisant le théorème 7.2 de [59], on obtient le résultat suivant.

ThÈorËme 3.2 *La distribution conjointe stationnaire $\pi(\mathbf{x})$ existe si et seulement si $\rho_s < C$, $s \in \mathcal{S}_a$, et elle a la forme-produit suivante*

$$\pi(\mathbf{x}) = \prod_{s\in\mathcal{S}} \pi_s(\mathbf{x}_s) \quad \text{où } \pi_s(\mathbf{x}_s) = \pi_s(0)\,\Phi_s(|\mathbf{x}_s|) \prod_{r\in\Omega(s)} \frac{(\rho_s^r)^{x_s^r}}{x_s^r!}, \tag{3.17}$$

avec la notation $|\mathbf{x}_s| = \sum_r x_s^r$. La probabilité qu'il n'y ait aucune connexion dans la phase $s \in \mathcal{S}$ est simplement $\pi_{off}(0) = e^{-\rho_{off}}$ pour $s = off$, tandis que pour $s \in \mathcal{S}_a$,

$$\pi_s(0) = \left(\sum_{i=0}^{N_s-1} \frac{(\rho_s/c_s)^i}{i!} \;+\; \frac{(\rho_s/c_s)^{N_s}}{N_s!} \frac{C}{C-\rho_s} \right)^{-1}$$

De plus, pour $s \in \mathcal{S}$ et $r \in \Omega(s)$, on a $Pr[X_s = k] = \pi_s(0)\,\Phi_s(k)\,\frac{(\rho_s)^k}{k!}$.

On notera que les résultats ci-dessus sont insensibles aux caractéristiques statistiques détaillées du trafic (ils ne dépendent que des moyennes). Observons de plus que la distribution de probabilité de X_s est la solution de la file d'attente $M/M/C/\infty$ pour $s \in \mathcal{S}_a$. Par conséquent, la probabilité B_s de congestion du lien d'accès dans la direction s est donnée par la célèbre formule d'Erlang-C

$$B_s = Pr\left[X_s \geq N_s\right] = \pi_s(0)\,\frac{(\rho_s/c_s)^{N_s}}{N_s!}\,\frac{C}{C-\rho_s} \tag{3.18}$$

Le corollaire suivant établit l'expression des principales mesures de performance.

Corollaire 3.2 *Le nombre moyen de session dans la phase $s \in \mathcal{S}_a$ est donnée par le nombre moyen de clients dans la file $M/M/C/\infty$ correspondante*

$$E[X_s] = \rho_s \left(\frac{1}{c_s} + \frac{B_s}{C - \rho_s} \right), \tag{3.19}$$

tandis que $E[X_{off}] = \rho_{off}$. Le nombre moyen de sessions de l'application r en phase s est $E[X_s^r] = (\rho_s^r/\rho_s)\, E[X_s]$ pour $s \in \mathcal{S}$ et $r \in \Omega(s)$. De plus, d'après l'équation (7.27) de [59], la durée moyenne de transfer d'un fichier de x Mbits dans la direction $s \in \mathcal{S}_a$ est $xE[X_s]/\rho_s$.

Remarque 3.1 *Les résultats précédents se généralisent aisément au cas d'un nombre fini d'utilisateurs, pour modéliser par exemple le comportement des salariés d'une agence accédant à l'Internet [37]. On doit alors calculer une constante de normalisation avec une méthode inspirée de celle de Buzen [53, 188].*

3.4.2 Allocation de débit équitable de type "Weighted Max-Min" (WMM)

S'il n'y avait pas de congestion dans le réseau de coeur, les performances obtenues seraient celles décrites pour les réseaux d'accès dans les paragraphes précédents. Pour prendre en compte l'impact des phénomènes de congestion sur les liens du réseau de coeur, il faut s'intéresser au partage de bande-passante opéré par TCP. On considére donc un réseau avec un ensemble $\mathcal{L}$ de liens et un ensemble $\mathcal{F}$ de flots TCP, numérotés $1, \ldots, K$. Soit C_l la capacité du lien l et Γ_l les flots passant par ce lien. On note n_f, θ_f et π_f le nombre de connexions TCP, le RTT et le chemin de bout-en-bout du flot f, respectivement. Les connexions sont persistentes et le nombre de connexions est constant. De plus, on suppose que les délai d'attente ont une influence négligeable sur le RTT (réseau "large delay-bandwidth product" [8]). Enfin, on note $\phi_f(\mathbf{n})$ la bande-passante allouée au flot f, où $\mathbf{n} = (n_1, \ldots, n_K)$ est le vecteur représentant le nombre de connexions de chaque flux. On suppose que RED est utilisé comme politique de gestion de file d'attente [71], et on fait les deux hypothèses suivantes :

- *Hypothèse 1* : la probabilité de perte de paquets est la même pour tous les flux limités en débit par le même lien (i.e., si on augmente la capacité de tout autre lien $k \in \pi_f$, $k \neq j$, le débit aggrégé $\phi_f(\mathbf{n})$ n'est pas augmenté).
- *Hypothèse 2* : si le lien l n'est pas le lien qui limite le débit du flot f, alors la probabilité de perte de paquets du flot f sur le lien l est négligeable.

Sous ces hypothèses, nous montrons dans [38] que si le lien l n'est limitant que pour une partie $B_l \subset \Gamma_l$ des flots le traversant (et donc le taux de perte est nul en ce lien pour les autres flots d'après l'hypothèse 2), on obtient

$$\phi_f(\mathbf{n}) = \frac{\nu_f}{\sum_{k \in B_l} \nu_k} \left[C_l - \sum_{k \in \Gamma_l - B_l} \phi_k(\mathbf{n}) \right] \quad , \; f \in B_l$$

TABLE 3.1 – Allocations de débit pour le réseau de la Figure 3.9.

	ϕ_0	ϕ_1	ϕ_2	ϕ_3
Weighted Max-Min	20	120	60	20
Network Calculus	18.5	121.9	62.5	20.5
TCP Utility	18.3	121.6	61.6	20
ns2	18.5	120.2	61	19.5

où $\nu_f = n_f/\theta_f$ est le ratio du nombre de connexion TCP sur le RTT du flot f. L'équation ci-dessus montre que la capacité résiduelle du lien (la capacité laissée par les flots qui sont limités ailleurs) est partagée entre les flots pour qui ce lien est un goulot d'étranglement suivant une allocation WMM, en fonction des poids ν_f, $f \in \mathcal{F}$. Ce résultat suggère que si le nombre de goulots d'étranglement du réseau est suffisamment faible, une allocation WMM peut être une bonne approximation de ce que TCP fait. Cette allocation peut de plus être calculée très facilement en utilisant un algorithme de type "water-filling" [26, 167], le débit du flot f étant augmenté suivant le pas ν_f.

Pour évaluer cette approximation, nous la comparons avec l'allocation distribuée réalisée par TCP qui est déterminée avec des simulations ns-2 pour le réseau de la Figure 3.9, en supposant la version New Reno de TCP, la politique RED, des paquets de 1024 octets, des délais de lien de 10 ms et des buffers de 1024 paquets. Ces comparaisons incluent également les résultats obtenus avec la méthode de point-fixe “network calculus” proposée dans [8] et l'approche basée sur les fonctions d'utilité développée dans [149]. Le tableau 3.1 résume les résultats obtenus dans le cas de 10 connexions par flot. L'allocation WMM fournit des résultats convenables. Nous ne prétendons bien sûr pas que cette allocation soit une représentation fidèle du comportement de TCP, mais elle fournit une approximation correcte, avec des temps de calcul très faibles, ce qui est un aspect essentiel pour des méthodes itératives d'évaluation de performances.

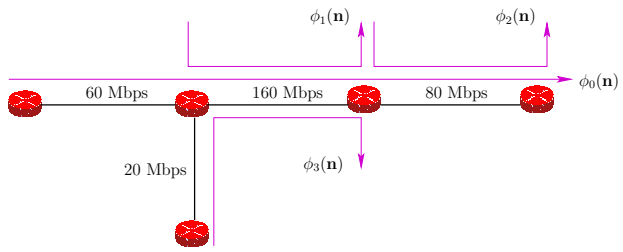


FIGURE 3.9 – Réseau à 4 liens.

3.4.3 Modèle Réseau Global

Nous considérons des utilisateurs situés dans K sites d'accès différents, connectés au backbone par des liens d'accès de capacité $C_1, \ldots, C_K$, qui recoivent du trafic d'un ensemble de serveurs Internet. On suppose que la population de chaque site d'accès peut être considérée comme infinie et que tous les flux de trafic vont de la ferme de serveurs vers les sites d'accès. La Figure 3.10 fournit un exemple du type de réseau considéré. S'il n'y avait pas de congestion dans le réseau backbone, les modèles de performance appropriés seraient ceux de la section 3.4.1. Nous montrons ici comment étendre ces modèles pour prendre en compte la congestion dans le réseau backbone.

Algorithme de point-fixe

Soit C_i^* la capacité de transmission moyenne perçue par les connexions du flot i. L'idée principale est alors de remplacer les liens d'accès de capacité C_i et le réseau backbone par des liens d'accès virtuels de capacité C_i^* connectant directement les sites d'accès à la ferme de serveurs. Notons X_i la variable aléatoire représentant le nombre de sessions actives du site i et soit $\mathbf{X} = (X_1, \ldots, X_K)$. D'après le Théorème 3.2, la probabilité de $X_i = n$ est donnée par

$$\pi_n^i(C_i^*) = \pi_0^i(C_i^*)\, \Phi_i(n, C_i^*)\, \frac{(\rho_i)^n}{n!} \quad , \; n = 0, 1, \ldots$$

Soit $\overline{T}_i(\mathbf{C}^*)$ le trafic moyen genéré par le site i, où $\mathbf{C}^* = (C_1^*, \ldots, C_K^*)$. En utilisant l'allocation WMM comme approximation du partage de bande-passante de TCP, on obtient

$$\overline{T}_i(\mathbf{C}^*) = E\left[\phi_i\left(\mathbf{X}\right)\right] = \sum_{\mathbf{n} \in \mathbb{N}^K} \phi_i(\mathbf{n})\, Pr\left[\mathbf{X} = \mathbf{n}\right].$$

S'il y a suffisamment de capacité dans le réseau backbone, la probabilité que le débit d'un site d'accès soit limité par un lien de backbone va être faible. Dans ce cas, les variables aléatoires $X_1, \ldots, X_K$ pourront être considérées comme indépendantes, et on pourra alors faire l'approximation suivante :

$$\overline{T}_i(\mathbf{C}^*) = \sum_{\mathbf{n} \in \mathbb{N}^K} \phi_i(\mathbf{n}) \prod_{j=1}^{K} \pi_{n_j}^j(C_j^*). \tag{3.20}$$

Cependant, si on interpréte C_i^* comme la capacité d'un lien d'accès équivalent, alors $\overline{T}_i(\mathbf{C}^*)$ peut aussi être écrit de la façon suivante :

$$\overline{T}_i(\mathbf{C}^*) = \sum_{k=1}^{\infty} k\, min(c_i, C_i^*/k)\, \pi_k^i(C_i^*), \tag{3.21}$$

TABLE 3.2 – Valeurs moyennes pour les sites 1 et 2.

Paramètre	site 1	site 2
Taille de fichier (MBytes)	0.5	1
Taux d'arrivée (sessions/s)	1.25	0.5
Nombre de pages (pages/session)	2	1.5
Durée de la période Off (secondes)	30	45

qui est juste ρ_i dans le cas d'une population infinie d'utilisateurs. La capacité C_i^* devrait être telle que les expressions 3.20 et 3.21 coincident. Un vecteur solution $\mathbf{C}$ peut alors être obtenu en utilisant l'algorithme 1.

Algorithm 1 Algorithme de Point-Fixe

procedure APPROXIMATION
  $t = 0$ et $C_i^0 = C_i$, $i = 1, \ldots, K$
  while not convergence **do**
    Calculer $\pi_n^i(C_i^t)$, $n \geq 0$, $i = 1, \ldots, K$,
    Evaluer $\overline{T}_i(\mathbf{C}^t) = \sum_{\mathbf{n}} \phi_i(\mathbf{n}) \prod_{j=1}^{K} \pi_{n_j}^j(C_j^t)$
    Trouver C_i^{t+1} en tant que solution unique en x de,

$$\overline{T}_i(\mathbf{C}^t) = \sum_{k=1}^{\infty} k \, min(c_i, x/k) \, \pi_k^i(C_i^t)$$

    $t = t + 1$
  end while
end procedure

Résultats

Comme exemple test, nous considérons le réseau de la figure 3.10 où C_1=20 Mbps et C_2=10 Mbps. La capacité C du lien de backbone commun est prise pour plusieurs valeurs, allant de 17.5 Mbps à 30 Mbps. Les délais de tous les liens sont fixés à 1 ms. Les utilisateurs sont situés dans les sites d'accès 1 et 2 et téléchargent des données en utilisant une application interactive, mais avec des comportements différents (cf. tableau 3.2).Toutes les distributions de probabilités sont des exponentielles négatives. Les débits crêtes des utilisateurs sont fixés à $c_1 = 3$ Mbps pour le site 1 et à $c_2 = 6$ Mbps pour le site 2.

Nous comparons ci-dessous les résultats obtenus avec l'approximation que nous proposons et ceux obtenus grâce à un simulateur "maison" implémentant l'allocation WMM comme paradigme de partage de bande-passante. Le tableau 3.3 compare les nombres moyens de sessions actives pour les flots 1 et 2 (les intervalles de confiance à 99% sont indiqués pour la simulation). L'approximation fournit des résultats acceptables pour $C \geq 20$ Mbps. Notons que l'algorithme proposé n'est pas limité aux valeurs moyennes. Ainsi, si on considére le cas où $C = 25$ Mbps.,

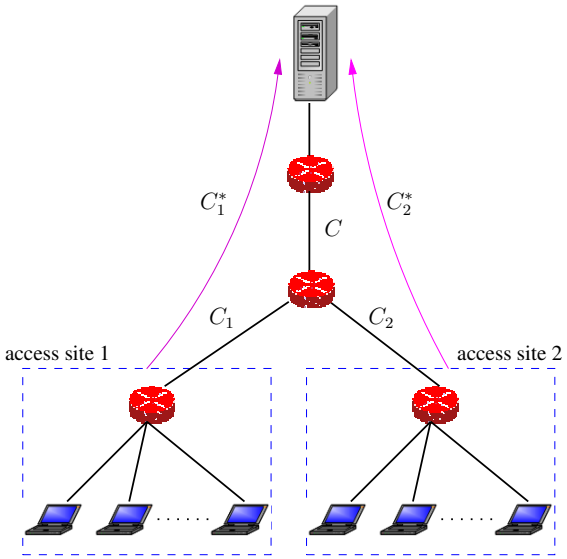


FIGURE 3.10 – Réseau de test à 3 liens.

TABLE 3.3 – Résultats pour $E[X_1]$ et $E[X_2]$

C	Simulation		Approximation	
	$E[X_1]$	$E[X_2]$	$E[X_1]$	$E[X_2]$
17.5	7.882 ± 0.5	4.665 ± 0.33	9.559	7.034
20.0	4.430 ± 0.09	2.517 ± 0.09	4.110	2.676
22.5	3.713 ± 0.05	2.065 ± 0.06	3.642	2.182
25	3.514 ± 0.05	1.907 ± 0.057	3.492	1.933
27.5	3.454 ± 0.03	1.804 ± 0.044	3.440	1.846
30	3.416 ± 0.027	1.770 ± 0.057	3.422	1.797

la Figure 3.11 représente la distribution du nombre de sessions actives du site 1. Signalons enfin que nous avons réalisé les mêmes expériences en utilisant une distribution hyper-exponentielle à deux phases avec des moyennes équilibrées pour les tailles de fichier et que les résultats obtenus suggèrent que notre modèle est approximativement insensible aux caractéristiques détaillés du trafic.

3.5 Prix de l'Anarchie et Equilibrage de Charge Non-Coopératif

Un des problème fondamentaux dans les fermes de serveurs concerne le routage optimal des tâches. Ce problème consiste à déterminer la politique de routage qui permet d'optimiser un certain critère de performance, tel que par exemple le temps de traitement moyen des tâches. En pratique, cette politique optimale ne peut souvent pas être mise en oeuvre de manière centralisée pour des raisons de complexité ou de scalabilité. Dans ce cas, il faut se rabattre sur une mise en

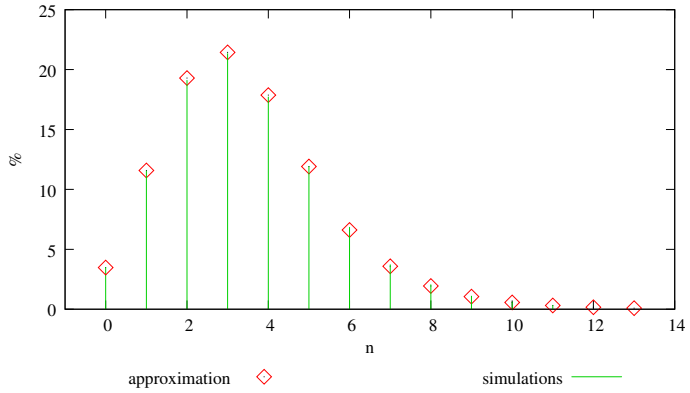


FIGURE 3.11 – Distribution de probabilité du nombre de sessions actives du flot 1.

oeuvre distribuée dans laquelle plusieurs dispatchers sont utilisés. Le passage d'une architecture de décision centralisée à une architecture décentralisée a toutefois un impact important sur la nature du problème. En effet, chaque dispatcher cherchant à minimiser le temps de traitement des tâches qu'il route lui même, l'architecture distribuée implique un jeu non-coopératif entre les dispatchers.

La théorie des jeux fournit le cadre théorique permettant d'étudier de telles situations. Dans le contexte qui nous intéresse, cette théorie est utilisée pour s'interesser à l'existence d'un équilibre de Nash, c'est à dire à l'existence d'une stratégie de routage dont aucun dispatcher n'a interêt à dévier unilatéralement dans la mesure où cela ne lui permettra pas d'améliorer les performances perçues par les tâches qu'il route. En dehors de l'existence et de l'unicité de l'équilibre de Nash, une autre question importante pour le concepteur du système concerne la perte d'efficacité liée au passage à une solution distribuée. En effet, puisque chaque dispatcher réalise une optimisation individuelle pour ses propres tâches, il est clair que la performance globale du système dans l'architecture décentralisée sera pire que dans l'architecture centralisée.

La question que nous avons étudié est alors la suivante : *quelles garanties de performance peut on apporter dans le cas d'un routage décentralisé* ? Nos travaux ont fait suite à une première étude d'Altman, Ayesta et Prabhu [9] qui portait sur l'équilibre de Wardrop, c'est à dire quand chaque tâche peut choisir elle-même le serveur qui va minimiser son coût de traitement (voir également [15, 56]). Dans [17, 18], nous avons étudié les proriétés de l'équilibre de Nash dans le cas d'une fonction coût du type M/M/1. Après avoir montré qu'il existe toujours un unique équilibre de Nash, nous avons montré que si l'on fixe le traffic total offert au système, le pire des cas correspond au jeu symétrique. En d'autres termes, l'équilibre de Nash offrant les plus mauvaises performances globales est obtenu quand chaque joueur route exactement la même quantité de trafic. Ce résultat est important car le jeu symétrique est un jeu potentiel, ce qui implique de

nombreuses propriétés, notamment celle de la convergence de la stratégie "meilleure réponse" vers l'équilibre de Nash. Ce résulat a ensuite été généralisé à une classe beaucoup plus large de fonctions coûts dans [45]. Dans ces deux publications, nous l'avons utilisé pour obtenir des bornes sur le Prix de l'Anarchie, une mesure de l'inefficacité des algorithmes distribués introduites par Koutsoupias et Papadimitriou [133]. Signalons également qu'en termes de méthodologie, nos travaux sont proches de ceux de Orda *et al* [127, 170] et que plusieurs arguments utilisés sont inspirés de ces articles.

3.5.1 Formulation du problème

Notons $\mathcal{C} = \{1, \ldots, K\}$ l'ensemble des dispatchers et $\mathcal{S} = \{1, \ldots, S\}$ l'ensemble des serveurs. Les jobs reçus par le dispatcher i sont dits de classe i. Le serveur $j \in \mathcal{S}$ a une capacité r_j et un coût de traitement c_j par unité de temps doit être payé pour chaque job envoyé à ce serveur. On note $\mathbf{r} = (r_j)_{j\in\mathcal{S}}$ et $\mathbf{c} = (c_j)_{j\in\mathcal{S}}$ les vecteurs de capacités et de coûts des serveurs, respectivement. Les jobs de classe i arrivent dans le système suivant un processus de Poisson d'intensité λ_i et ont des tailles distribuées suivant une loi générale. On suppose que $\sum_{i\in\mathcal{C}} \lambda_i = \overline{\lambda}$, où $\bar{\lambda}$ représente l'intensité totale du trafic, et on note Λ l'ensemble des vecteurs $\boldsymbol{\lambda}$ vérifiant cette contrainte.

Notons $\mathbf{x}_i = (x_{i,j})_{j\in\mathcal{S}}$ la stratégie de routage du dispatcher i, où $x_{i,j}$ est la quantité de trafic envoyée vers le serveur j. Soit

$$\mathcal{X}_i = \left\{ \mathbf{x}_i \in \mathbb{R}^S \ : \ 0 \le x_{i,j} \le r_j \ , \forall j \in \mathcal{S}; \sum_{j\in\mathcal{S}} x_{i,j} = \lambda_i \right\}$$

l'ensemble des stratégies de routage faisables pour le dispatcher i. Ce dispatcher cherche à minimiser son propre coût $T_i(\mathbf{x})$:

$$\underset{\mathbf{x}_i\in\mathcal{X}_i}{\text{minimiser}}\ T_i(\mathbf{x}) = \sum_{j\in\mathcal{S}} \frac{c_j}{r_j}\, x_{i,j}\, \phi(\rho_j)$$

où $\rho_j = \sum_{i\in\mathcal{C}} x_{i,j}/r_j$ est le taux d'utilisation du serveur $j \in \mathcal{S}$. Ainsi, le coût à payer par la classe i sur le serveur j dépend à la fois de la quantité de flot $x_{i,j}$ qu'elle envoie sur ce serveur mais aussi du taux d'utilisation de ce serveur via la fonction ϕ. Sans rentrer dans trop de détails, disons que nous considérons deux grands classes de fonctions de congestion ϕ, qui sont des cas particuliers des fonctions de type B introduites dans [170]. La première classe contient notamment les fonctions du type $\phi(\rho) = (1-\rho)^{-m}$, tandis que la seconde contient les fonctions de délai des modèles M/G/1/PS ou M/Pareto/1/SRPT en fort trafic.

Dans un équilibre de Nash du jeu de routage, aucune classe de trafic n'a interêt à changer unilatéralement sa stratégie. De là, $\mathbf{x} \in \bigotimes_{i\in\mathcal{C}} \mathcal{X}_i$ est un équilibre de Nash si

$$\mathbf{x}_i = \arg\min_{\mathbf{z}\in\mathcal{X}_i} T_i(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{z}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_K), \quad \forall i \in \mathcal{C}.$$

L'existence et l'unicité de l'équilibre de Nash sont prouvés dans [170]. Soit $\mathbf{x}$ l'équilibre de Nash avec K dispatchers. La performance globale du système peut être évaluée en utilisant le coût global

$$D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}) = \sum_{i \in \mathcal{C}} T_i(\mathbf{x}) = \sum_{j \in \mathcal{S}} c_j \, \rho_j \, \phi(\rho_j)$$

où les taux d'utilisation ρ_j sont ceux à l'équilibre de Nash. Remarquons que pour le système centralisé où il y a un seul dispatcher qui contrôle tout le trafic, ce coût s'écrit $D_1(\bar{\lambda}, \mathbf{r}, \mathbf{c})$. Le prix de l'anarchie est alors définie de la façon suivante :

$$PoA(K) = \sup_{\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}} \frac{D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c})}{D_1(\bar{\lambda}, \mathbf{r}, \mathbf{c})}.$$

3.5.2 Résultats

Notre principal résultat affirme que le coût global $D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c})$ atteint son maximum quand $\boldsymbol{\lambda}$ est le vecteur symétrique $\boldsymbol{\lambda}^{=} = \left(\frac{\bar{\lambda}}{K}, \ldots, \frac{\bar{\lambda}}{K}\right)$.

ThÈorËme 3.3

$$\sup_{\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}} D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}) = \sup_{\mathbf{r}, \mathbf{c}} D_K(\boldsymbol{\lambda}^{=}, \mathbf{r}, \mathbf{c}).$$

Ce résultat implique que pour le calcul du prix de l'anarchie on peut se restreindre à considérer le jeu symétrique. Il est bien connu que, dans le cas où tous les dispatchers contrôlent le même trafic, le jeu de routage non-coopératif est un jeu potentiel, i.e., les flots à l'équilibre correspondent au minimum global d'un problème d'optimisation convexe (cf par exemple le Thèoréme 4.1 de [61]). Ceci est écrit formellement dans la proposition suivante.

Proposition 3.5 *Si le vecteur $\boldsymbol{\rho}$ est le minimum global du problème convexe*

$$\begin{aligned} \underset{\boldsymbol{\rho}}{\textit{minimiser}} \quad & \sum_{j \in \mathcal{S}} c_j \rho_j \phi(\rho_j) + (K-1) \int_0^{\rho_j} c_j \, \phi(z) dz \\ \textit{s.t.} \quad & \\ & \textstyle\sum_{j \in \mathcal{S}} r_j \rho_j = \bar{\lambda}, \\ & 0 \leq \rho_j < 1, \ \forall j \in \mathcal{S}, \end{aligned}$$

alors le vecteur $\mathbf{x}$ tel que $x_{i,j} = r_j \, \frac{\rho_j}{K}$, $\forall i \in \mathcal{C}$, $\forall j \in \mathcal{S}$ est un équilibre de Nash du jeu symmétrique.

A partir des résultats précédents, nous avons pu obtenir des bornes sur le prix de l'anarchie pour des fonctions de congestion du type M/G/1/PS, M/Pareto/1/SRPT et pour des fonctions polynomiales dans le cas $K \geq 2$.

Proposition 3.6

1. *Pour* $\phi(x) = \frac{1}{1-x}$, $\frac{K}{2\sqrt{K}-1} \leq PoA(K) \leq \sqrt{K}$
2. *Pour* $\phi(x) = 1 + x^m$, $PoA(K) \geq \frac{(1+m/K)^{-1}}{(1+m)^{-1}+m^{-1}\log\frac{1+m}{1+m/K}}$.
3. *Pour* $\phi(x) = \frac{1}{(1-x)^m}$, $PoA(K) \geq K\left(K^{1/(m+1)} + mK^{1/(m+1)} - m\right)^{-1}$.

Remarquons que pour la fonction délai du modèle M/G/1/PS, le résultat établi que le prix de l'anarchie est de l'ordre de $\sqrt{K}$ indépendamment du nombre de serveurs, et reste donc borné pour un nombre fixe de serveurs. Signalons également que les bornes inférieures ont été obtenues à partir des conditions KKT en définissant les paramètres du système de telle manière que pour $K > 1$ un seul serveur soit utilisé, tandis que pour $K = 1$ plus d'un serveur soit utilisé. Ainsi, d'une part le coût global à l'équilibre des solutions distribuées devient trivial à calculer, mais on permet à la solution centralisée d'avoir de bien meilleures performances en répartissant son trafic sur plusieurs serveurs.

Chapitre 4

Optimisation de Réseaux

Ce chapitre décrit certains de nos travaux sur l'optimisation des réseaux. Au lieu d'une présentation suivant un ordre chronologique, nous avons cette fois ci opté pour une présentation dans l'ordre dans lequel les problèmes se posent à un opérateur de réseau. Ainsi, le premier paragraphe est consacré à un problème de conception de réseaux d'accès. Nous abordons ensuite les problèmes d'optimisation du routage, en commançant par le problème d'optimisation des poids OSPF au paragraphe 4.2, pour ensuite décrire nos contributions sur les problèmes de monoroutage au paragraphe 4.3. Enfin, le paragraphe 4.4 présente une approche que nous avons récemment proposée pour l'ordonnancement multiprocesseur de tâches strictement périodiques.

4.1 Conception de réseaux d'accès avec coûts d'équipements

La conception d'un réseau WAN se fait généralement suivant une approche (sous optimale) par décomposition [174]. La première phase concerne la conception des réseaux d'accès et de cœur, qui impliquent des décisions sur où installer les équipements (nœuds et liens) permettant d'interconnecter différents sites distants. Elle est suivie par le dimensionnement de l'architecture réseau, c'est à dire d'une part le choix des lignes reliant les différents sites mais aussi le choix des équipements (routeurs, cartes de communication) placés dans ces sites. Finalement, la dernière phase concerne le choix d'une stratégie de routage des flux dans le réseau de cœur.

De nombreux travaux ont été consacrés au problème de conception d'une topologie d'accès [7, 14, 25, 80, 95, 96, 98, 99, 151, 156, 178, 125, 13] (cf. [84, 101, 78, 193] pour le problème de conception d'un backbone). La plupart propose des algorithmes d'approximation et prennent en compte différents paramètres ou restrictions, comme par exemple un type spécifique de topologie (en étoile, en arbre ou en anneau) ou une limite sur le nombre de concentrateurs pouvant être utilisés. Si généralement les coûts des liens ne dépendent pas des capacités, certains de ces articles supposent des liens modulaires. Les approches utilisées sont extrêmement variées : relaxation lagrangienne avec optimisation des sous-gradients [156], recuit simulé [151], relaxation basée sur la programmation linéaire [14], heuristiques gloutonnes [99] ou GRAS [13], Branch-and-Bound

avec décomposition de Benders [178, 179], réseaux neuronaux [7] ou encore recherche tabou [98].

L'argument permettant de dissocier la conception d'une topologie de son dimensionnement est que la conception d'une topologie fait intervenir des coûts fixes qui sont de plusieurs ordres de grandeurs supérieurs à ceux des équipements. Ces coûts fixes représentent typiquement ceux des tranchées à creuser pour y installer des fibres optiques, les coûts d'acquisition ou de location des sites ou même les coûts d'installation et de configuration des équipements. Ainsi, dans les travaux précédents, on suppose généralement que le coût d'un lien ne dépend pas du type de ligne qui sera effectivement choisi et que les autres coûts d'équipements peuvent être négligés. Cependant, avec le déployement massif de fibres optiques, les coûts de location des lignes de transmission deviennent de plus en plus faibles et les coûts de routeurs et de cartes de communication représentent aujourd'hui une part significative du coût global d'un réseau. C'est pourquoi nous avons proposé dans [46] une approche intégrée prenant en compte les coûts d'équipements dans les toutes premières phases du processus de conception d'un réseau.

4.1.1 Formulation du problème

On considère N nœuds terminaux qui doivent être reliés à des routeurs d'accès. Ces derniers doivent être placés dans K sites d'accès potentiels. Le problème consiste à déterminer les sites à ouvrir, les équipements à y installer et les nœuds terminaux connectés à chaque site, de manière à minimiser le coût global du réseau.

Soit I le nombre de types de lien/port et r_t la capacité d'un lien de type $t = 1, \ldots, I$. On notera Φ_t l'ensemble des nœuds terminaux devant être connecté au réseau avec un lien de type $t = 1, \ldots, I$. Un lien de type t doit être branché sur un port d'une carte de communication. Soit p_t le nombre de ports d'une carte de type t et ϕ_t le coût d'une telle carte.

Les routeurs à installer dans les sites d'accès doivent être choisis parmi un ensemble de R modèles. Il n'y a pas de limite sur le nombre de routeurs que l'on peut installer dans un site. Pour chaque modèle $r = 1, \ldots, R$, on note $\overline{s}_r$ le nombre de slots où l'on peut brancher une carte de communication, $\overline{T}_r$ le débit maximal du routeur et ψ_r son coût.

Une solution est définie comme un vecteur $\mathbf{x} = [x_{i,j}]$, où la variable de décision binaire $x_{i,j}$ est telle que $x_{i,j} = 1$ si le nœud i est relié au site j et $x_{i,j} = 0$ sinon. Une affectation admissible des nœuds terminaux aux sites vérifie les contraintes suivantes :

$$\begin{aligned} \sum_j x_{i,j} &= 1 \quad i = 1 \ldots N, \\ x_{i,j} &\in \{0,1\} \quad i = 1, \ldots, N,\ j = 1, \ldots, K. \end{aligned}$$

On notera $\mathcal{X}$ l'ensemble des solutions admissibles. Etant donnée une solution $\mathbf{x} \in \mathcal{X}$, on notera $a(\mathbf{x}, i)$ le site d'accès affecté au nœud terminal i, i.e., $a(\mathbf{x}, i) = \sum_{j=1}^{K} j\, x_{i,j}$.

Le coût d'une solution est la somme du coût des liens entres nœuds terminaux et sites d'accès, des coûts d'ouverture des sites et des coûts d'équipement. En notant $c_{i,j}$ le coût du lien permettant de relier le client i au site j (qui peut dépendre de la distance et de la capacité), le coût global des liens s'écrit

$$\sum_{i=1}^{N} \sum_{j=1}^{K} x_{i,j}\, c_{i,j}.$$

En notant L_j le coût d'ouverture du site j et en posant $u_j(\mathbf{x}) = 1$ si $\sum_{i=1}^{N} x_{i,j} \geq 1$ et $u_j(\mathbf{x}) = 0$, le coût d'ouverture des sites s'écrit

$$\sum_{j=1}^{K} u_j(\mathbf{x})\, L_j.$$

En ce qui concerne le coût des équipements, l'observation clef est que le coût minimal des équipements à installer dans un site ne dépend de l'affectation $\mathbf{x}$ qu'au travers du nombre de cartes de chaque type à installer dans ce site. Notons $p_j^t(\mathbf{x})$ le nombre minimal de cartes du type $t = 1, \ldots, I$ nécessaires pour supporter les clients affectés au site j :

$$p_j^t(\mathbf{x}) = \left\lceil \frac{\sum_{i \in \Phi_t} x_{i,j}}{p_t} \right\rceil$$

Soit $\mu^*(\mathbf{p}_j(\mathbf{x}))$ le coût minimal de l'équipement du site j dans la solution $\mathbf{x} \in \mathcal{X}$, où $\mathbf{p}_j(\mathbf{x}) = \left[p_j^1(\mathbf{x}), \ldots, p_j^I(\mathbf{x})\right]$. Le coût total des équipements peut alors être écrit

$$\sum_{j=1}^{K} \mu^* \left(\mathbf{p}_j(\mathbf{x})\right).$$

Finalement, le problème à résoudre se formule de la façon suivante :

$$\min_{\mathbf{x} \in \mathcal{X}} \; \Gamma(\mathbf{x}) = \sum_{i=1}^{N} \sum_{j=1}^{K} x_{i,j}\, c_{i,j} \; + \; \sum_{j=1}^{K} [u_j(\mathbf{x})\, L_j \; + \; \mu^* \left(\mathbf{p}_j(\mathbf{x})\right)]. \tag{4.1}$$

On peut noter que les coût d'équipement et de lien dépendent du nombre de sites ouverts. Si on utilise de nombreux sites, il va falloir payer pour de nombreux "petits" équipements qui comparativement sont plus chers que quelques "gros" équipements (économie d'échelle). Le coût d'équipement a donc tendance a croître avec le nombre de sites ouverts. A l'inverse, avec peu de sites ouverts, il faut des liens plus longs pour connecter les nœuds terminaux aux sites d'accès, ce qui conduit à un budget plus important en terme de coût de lien. Il y a donc un compromis

à trouver entre ces deux aspects.

Il est bien sûr possible de formuler le problème ci-dessus comme un problème linéaire en nombres entiers, mais cette formulation conduit à des temps de calcul prohibitifs même pour de petits réseaux. On propose donc ci-dessous un algorithme exact de type Branch and Bound ainsi qu'une heuristique. Avant cela, nous analysons la fonction μ^*.

4.1.2 Calcul des configurations de carte optimales

Configurations de carte

On a vu que le coût des équipements à installer dans le site j ne dépend de la solution $\mathbf{x} \in \mathcal{X}$ qu'au travers du vecteur $\mathbf{p}_j(\mathbf{x})$. Ce dernier est de la forme $\mathbf{s} = (s_1, \ldots, s_I)$ où s_t est un nombre de cartes de type t. Dans la suite, nous appelons un tel vecteur une configuration de carte. Vu qu'il y a au maximum $|\Phi_t|$ nœuds terminaux qui peuvent être raccordés à un site avec un lien de type t, le nombre maximal de cartes de type t qui peut être requis sur un site est $h_t = |\Phi_t|/p_t$. On en déduit que l'ensemble de toutes les configurations possibles est donné par

$$\Lambda = \left\{ \mathbf{s} = \sum_{t=1}^{I} s_t \, \mathbf{e}_t \mid 0 \leq s_t \leq h_t \,, \; t = 1, \ldots, I \right\}$$

où $\mathbf{e}_t$ est le vecteur $(0, \ldots, 1, \ldots, 0)$ avec un 1 en position t et des 0 partout ailleurs. Le nombre total de configurations est donné par $H = |\Lambda| = \prod_{t=1}^{I} (h_t + 1)$. Remarquons que Λ contient aussi bien la configuration nulle $(0, \ldots, 0)$ que la configuration $(h_1, \ldots, h_I)$ qui permet de connecter tous les clients à un seul site.

Les configurations de carte peuvent être numérotées de 0 à $H - 1$, l'indice $h(\mathbf{s})$ associé à la configuration $\mathbf{s} = (s_1, \ldots, s_I)$ étant donné par

$$h(\mathbf{s}) = s_I + \sum_{t=1}^{I-1} s_t \prod_{u=t+1}^{I} (h_u + 1) \quad \forall s \in \Lambda$$

Dans la suite, $\mathbf{s}(h)$ représentera la configuration associée à l'indice $h = 0, \ldots, H - 1$. De plus, étant données deux configurations $\mathbf{s}, \mathbf{s}' \in \Lambda$, on dira que $\mathbf{s} \leq \mathbf{s}'$ si et seulement si $s_t \leq s'_t$, $t = 1, \ldots, I$.

Coût des configurations

Notre objectif est de construire la fonction $\mu^* : \Lambda \to \mathbb{R}$ associant à une configuration de carte $\mathbf{s} \in \Lambda$ le coût optimal $\mu^*(\mathbf{s})$ des équipements permettant de la supporter. Ce coût optimal a deux composantes : d'une part le coût des cartes elles-mêmes, qui est déjà connu et égal à $\sum_{t=1}^{I} s_t \, \phi_t$, et d'autre part le coût $\mu_r^*(\mathbf{s})$ des routeurs permettant d'accueillir ces cartes :

$$\mu^*(\mathbf{s}) = \sum_{t=1}^{I} s_t\,\phi_t \;+\; \mu_r^*(\mathbf{s}) \quad \forall \mathbf{s} \in \Lambda$$

Le problème est donc de calculer $\mu_r^*(\mathbf{s})$. Pour cela, définissons Λ_j comme l'ensemble des configurations de carte qui peuvent être supportées par un seul routeur de type $j = 1, \ldots, R$,

$$\Lambda_j = \left\{ \mathbf{s} \in \Lambda \,|\, \sum_{t=1}^{I} s_t \leq \overline{s}_j \text{ and } \sum_{t=1}^{I} s_t\, p_t \leq \overline{T}_j \right\}$$

Grâce au principe d'optimalité de Bellman [174], le coût optimal $\mu_r^*(\mathbf{s})$ d'un ensemble de routeurs accueillant les cartes de la configuration $\mathbf{s} \in \Lambda$ peut être écrit de la façon suivante

$$\mu_r^*(s) = \min_{j=1,\ldots,R} \left[\min_{s_j \in \Lambda_j, s_j \leq s} (\psi_j \;+\; \mu_r^*(s - s_j)) \right]$$

Ceci suggère que la fonction $\mu^*() : \Lambda \to \mathbb{R}$ peut être calculée en utilisant un algorithme de programmation dynamique. Les détails de cet algorithme sont donnés dans [46]. Les tests effectués montrent que cet algorithme fourni une solution numériquement très efficace.

4.1.3 Résolution exacte basée sur du Branch and Bound

Une fois la fonction μ^* calculée, on peut résoudre le problème (4.1) de manière exacte avec un algorithme de type Branch and Bound. Ce dernier est décrit dans l'Algorithme 2 (avec la notation $E_j(x) = u_j(\mathbf{x})\, L_j \;+\; \mu^*\left(\mathbf{p}_j(\mathbf{x})\right)$). Cet algorithme récursif prend en entrée une solution partielle $\mathbf{x}$ et son coût, le prochain nœud terminal i à affecter et une borne supérieure ub sur le coût optimal. Il réalise une recherche arborescente en utilisant la borne supérieure et une borne inférieure pour élaguer l'arbre des solutions. La borne supérieure est mise à jour à chaque fois qu'une solution améliorante est découverte. Une borne inférieure sur le coût à venir est générée en chaque nœud de l'arbre. De plus, l'algorithme explore les affectations les plus prometteuses en premier pour obtenir la borne supérieure la plus précise possible et ainsi réduire les temps de calcul. Enfin, des coupes de distance sont utilisées pour éviter l'exploration de sous-arbres ne contenant que des solutions sous-optimales.

La borne supérieure initiale peut être obtenue avec une heuristique efficace pour ce problème. Elle est ensuite mise à jour à chaque fois qu'une solution complète améliorante est découverte.

Une borne inférieure est également générée à chaque appel de la fonction. Soit x une solution partielle telle que les nœuds $1, \ldots, i$ sont déjà affectés. Notons $Subtree(x)$ l'ensemble des solutions complètes dans le sous-arbre de racine x, i.e.,

$$Subtree(x) = \left\{ x' \in X | a(x', k) = a(x, k) \quad k = 1, \ldots, i \right\}.$$

Des coupes peuvent être générées à partir du résultat suivant.

Algorithm 2 Algorithme de Branch-and-bound pour le problème d'affectation

```
1: procedure BC(x, i, ub)
2:     if i > N then                                          ▷ no terminal left
3:         if Γ(x) < ub then
4:             ub=Γ(x) and x* = x
5:         end if
6:         return 0
7:     end if
8:     bestCost=∞
9:     for j ∈ Sites(x, i) do
10:        Let x'_{i,j} = 1 and x'_{n,k} = x_{n,k}   k = 1...K, n < i
11:        costToGo = c_{i,j} + E_j(x') − E_j(x)
12:        lb = costToGo+lowerBound(x') ;
13:        if Γ(x)+lb < ub then
14:            costToGo = costToGo + BC(x',i + 1, ub) ;
15:        else
16:            costToGo = lb ;
17:        end if
18:        if costToGo < bestCost then
19:            bestCost=costToGo
20:        end if
21:    end for
22:    return bestCost
23: end procedure
```

Proposition 4.1 *Pour toute solution $x' \in Subtree(x)$, on a $\Gamma(x') \geq \Gamma(x) + lb(x)$ où*

$$lb(x) = \sum_{k=i+1}^{N} \min_{j} c_{k,j} \; + \; \sum_{t=1}^{I} max \left(h_t - \sum_{j} p_j^t(x) \; , \; 0 \right) \phi_t. \tag{4.2}$$

Si $\Gamma(x) + lb(x)$ est plus grand que la borne supérieure, alors aucune solution améliorante ne peut se trouver dans $Subtree(x)$ et cette branche de l'arbre des solutions n'a pas besoin d'être explorée.

Enfin, d'autre coupes peuvent être générées pour éviter l'exploration de certaines parties de l'arbre en utilisant le résultat suivant.

Proposition 4.2 *Soient i et $i' \neq i$ deux nœuds terminaux tels que $i, i' \in \Phi_t$. Soit $\mathbf{x} \in \mathcal{X}$ une solution telle que i and i' sont affectés aux sites k' et $k \neq k'$, respectivement. Si $c_{i'k} + c_{ik'} > c_{ik} + c_{i'k'}$, alors la solution $\mathbf{x}$ n'est pas optimale.*

Les tests que nous avons effectués montrent que ces coupes accèlerent très sigificativement la résolution [46]. On peut ainsi résoudre des problèmes avec une dizaine de sites d'accès et une quarantaine de nœuds terminaux en une à deux heures. Ces expérimentations ont également révélé que des gains significatifs pouvaient être obtenus en intégrant les coûts et les contraintes des équipements dès la phase de conception de la topologie.

Bien sûr, pour des problèmes de taille plus importante, il faut avoir recourt à des heuristiques. Nous avons donc proposé dans [46] ainsi que dans la thèse d'Anouar Rachdi [176] une approche heuristique basée sur un algorithme de clustering qui permet de prendre en compte le rôle essentiel de l'information géométrique dans ce type de problèmes. Cette heuristique permet d'obtenir des solutions assez performantes en quelques secondes. De plus, elle a l'avantage de pouvoir être combinée à l'algorithme exact dans le cadre d'une recherche à divergences limitées (Limited Discrepancy Search) [105]. On obtient ainsi un compromis intéressant entre qualité de la solution et temps de calcul dans la mesure où cette approche focalise la recherche sur une région pertinente de l'espace des solutions.

4.1.4 Extensions

L'originalité principale de nos travaux sur la conception de réseaux réside dans la prise en compte des contraintes et des coûts liés aux cartes de communication. Outre les travaux sur la conception de réseaux d'accès présentés ci-dessus, ces aspects ont été pris en compte dans :

- un problème de dimensionnement des liens d'un réseau IP que nous avons traité durant la thèse de Cédric Fortuny [77, 73],
- un problème de conception de réseaux dorsaux avec contraintes de connectivité et de délai que nous avons traité durant la thèse de Zied Ben Hamouda [22, 23, 21].

4.2 Optimisation des métriques de routage OSPF

Dans le contexte concurrentiel actuel, une alternative économique à un surdimensionnement excessif des réseaux est l'optimisation du routage, qui permet une utilisation plus efficace des ressources en adaptant les routes aux trafics transportés. Dans un domaine IP, le routage, calculé automatiquement par des protocoles tels qu'OSPF ou IS-IS, consiste à acheminer les trafics sur les plus courts chemins (PCC), avec partage équitable s'il y a plusieurs PCC vers la destination. Les métriques des liens (en réalité des interfaces des routeurs), et donc la longueur des routes, peuvent être changées par l'opérateur. En général, ces métriques sont soit égales à 1, soit inversement proportionnelles à la capacité, comme le propose CISCO, mais ces heuristiques conduisent souvent à une utilisation peu efficace des ressources.

Le problème d'optimisation des métriques, connu pour être NP-complet [79], consiste à trouver un ensemble de métriques des liens optimisant un critère donné. Une première approche pour résoudre ce problème consiste à déterminer un plan de routage, puis à chercher la solution d'un problème de PCC inverse pour obtenir les métriques correspondantes. Le problème de PCC inverse n'ayant pas forcément de solution, des approximations ont été développées [11, 12, 52, 192]. Une autre approche consiste à améliorer itérativement les métriques à partir d'une solution initiale. Ainsi, Fortz et Thorup [79] ont proposé une heuristique de recherche locale et l'ont testée sur un réseau dorsal d'AT&T et sur des réseaux fictifs. D'autres auteurs ont proposé des algorithmes génétiques pour l'optimisation des métriques IP [69, 51].

L'heuristique que nous avons proposé dans [74] permet une optimisation itérative des métriques IP. Elle a plusieurs avantages par rapport aux travaux mentionnés ci-dessus.

1. Tout d'abord, elle a été conçue pour une optimisation incrémentale du routage, par une série de modifications des métriques existantes. D'un point de vue opérationnel, l'opérateur peut donc optimiser le routage initial en plusieurs étapes, chaque étape étant constituée de quelques modifications améliorant l'étape précédente. Cette solution incrémentale, permet d'atteindre la solution optimale sans avoir à changer toutes les métriques du réseau en même temps.

2. Ensuite, l'algorithme est beaucoup plus rapide que ceux connus dans la littérature.

3. Enfin, les tests effectués montrent que, même s'il n'offre que des garanties d'optimalité locale, les solutions qu'il obtient sont en général performantes.

Après avoir rapidement introduit quelques notations dans la prochaine partie, nous présentons notre algorithme dans la partie 4.2.2 et quelques résultats obtenus dans la partie 4.2.3. Une description plus détaillée est disponible dans la thèse de Cédric Fortuny [73].

4.2.1 Formulation du problème

Un réseau IP est modélisé par un graphe orienté ayant N nœuds et M arcs. A chaque arc e du graphe est associé un poids w_e qui représente la métrique de l'interface correspondante et qui peut prendre toute valeur dans l'intervalle $\Omega = \left[1, 2^{16} - 1\right]$.

Etant donnée une solution admissible $\mathbf{w} = (w_1, \ldots, w_M)$, on notera $D_i^x(\mathbf{w})$ la distance du nœud i au nœud destination x, $\delta_{i,j}^x(\mathbf{w})$ l'utilisation du lien $(i,\ j)$ vers x (1 si l'arc (i,j) est sur un PCC vers x et 0 sinon), et $n_i^x(\mathbf{w})$ le nombre d'arcs sortants du nœud i qui sont sur un PCC vers x. A partir de $\mathbf{w}$, la résolution des PCC permet de connaître les valeurs de D, δ, et n pour tous les éléments du réseau.

Le réseau écoule un ensemble de K flots Origine-Destination (OD). Chaque flot $k = 1, \ldots, K$ est caractérisé par sa source $s(k)$, sa destination $t(k)$ et sa demande en bande-passante $\lambda_{s(k)}^{t(k)}$.

Notons $\gamma_i^x(\mathbf{w})$ le trafic reçu au nœud i, direct et en transit, pour la destination x et $y_{i,j}(\mathbf{w})$ le trafic sur le lien (i,j). En supposant un partage de charge équitable, on a :

$$\gamma_i^x(\mathbf{w}) = \lambda_i^x \ + \ \sum_{j \neq x} \frac{\delta_{j,i}^x(\mathbf{w})}{n_j^x(\mathbf{w})} \gamma_j^x(\mathbf{w}) \quad \text{et} \quad y_{i,j}(\mathbf{w}) = \sum_{x=1}^{N} \frac{\delta_{i,j}^x(\mathbf{w})}{n_i^x(\mathbf{w})} \gamma_i^x(\mathbf{w}) \tag{4.3}$$

Ainsi, étant donné un vecteur de métriques $\mathbf{w}$, nous pouvons calculer récursivement le trafic $\gamma_i^x(\mathbf{w})$ reçu au nœud i pour la destination x et la charge $y_{i,j}(\mathbf{w})$ sur chaque lien (i,j). Le problème consiste alors à déterminer le vecteur de métrique $\mathbf{w}$ minimisant le coût total $\Phi(\mathbf{w})$ du réseau, que nous supposerons ici être la somme des coûts des liens :

$$\min_{\mathbf{w} \in \Omega^M} \Phi(\mathbf{w}) = \sum_{(i,j)} \Phi_{i,j}\left[y_{i,j}(\mathbf{w})\right] \tag{4.4}$$

Dans les résultats présentés, le coût retenu représente le délai moyen d'un paquet, calculé à partir des formules de files d'attente M/M/1/N [2].

4.2.2 Algorithme de recherche locale pour l'optimisation des métriques

Pour résoudre le problème précédent, nous proposons un algorithme de recherche locale, dont l'originalité principale réside dans sa structure de voisinage. Ce voisinage est de taille réduite et garantit que tout voisin a un coût différent de la solution courante. Le voisinage de la solution $\mathbf{w} = (w_1, \ldots, w_M)$ est défini par $V(\mathbf{w}) = \left\{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^M\right\}$, avec $\mathbf{w}^e = (w_1, \ldots, w_e + \Delta_e, \ldots, w_M)$, pour $e = 1, \ldots, M$, où :

$$\Delta_e = argmin_{\Delta \geq 1}[y_e\left(w_1, \ldots, w_e + \Delta_e, \ldots, w_M\right) < y_e(\mathbf{w})] \tag{4.5}$$

2. L'algorithme proposé s'adapte facilement à d'autres formes de coût, y compris non additif.

Dès lors, le voisinage d'une solution $\mathbf{w}$ contient au plus M solutions. Chacune d'entre elle est associée à un lien e et est obtenue en augmentant la métrique du lien e de la quantité minimale Δ_e permettant de dévier du trafic de ce lien.

Soit F_e l'ensemble des flots OD qui passent par le lien e. Pour générer $\mathbf{w}^e$ (i.e. trouver la valeur de Δ_e), on fixe la métrique du lien e à $+\infty$ et on recalcule les PCC. La longueur des routes des flots $f \in F_e$ ne peut qu'augmenter. Soit Δ_{min} l'augmentation minimale des longueurs des PCC des flots $f \in F_e$ suite à la modification de métrique,

$$\Delta_{min} = \min_{f \in F_e} \left(D_e^{t(f)}(\overline{\mathbf{w}_e}) - D_e^{t(f)}(\mathbf{w}) \right) \tag{4.6}$$

où $\overline{\mathbf{w}_e}$ est le vecteur de métriques obtenu à partir de $\mathbf{w}$ en affectant $+\infty$ à la métrique de e. Si $\Delta_{min} = 0$, au moins un flot $f \in F_e$ était partagé en amont du lien e et donc l'incrément de métrique $\Delta_e = 1$ suffit à dévier du trafic du lien e. Si $0 < \Delta_{min} < \infty$, l'incrément $\Delta_e = \Delta_{min}$ dévie intégralement au moins un flot du lien e. Enfin, si $\Delta_{min} = \infty$, aucun flot ne peut être dévié du lien e et le voisin $\mathbf{w}^e$ n'est pas généré.

Le pseudocode de l'algorithme 3 décrit l'algorithme de recherche locale proposé. Cet algorithme garantit une décroissance monotone du coût, sauf si la solution courante est un minimum local $\mathbf{w}^*$ auquel cas la prochaine solution est le voisin de coût minimum. L'algorithme s'arrête si, au bout de K itérations après une augmentation de coût, une solution meilleure que $\mathbf{w}^*$ n'est pas trouvée.

Algorithm 3 Algorithme d'optimisation des métriques

procedure MetricOptimisation
  Soit $\mathbf{w} = (w_1, \ldots, w_M)$ la solution initiale et $\Phi(\mathbf{w})$ son coût.
  $\mathbf{w}^* = \mathbf{w}$ ▷ Initialisation de la solution de coût minimum
  while Convergence() = false **do**
    $\Phi_{min} = \infty$
    for $e = 1 \ldots M$ **do**
      Calculer Δ_e et $\mathbf{w}^e = (w_1, w_2, \ldots, w_e + \Delta_e, \ldots, w_M)$
      Calcul des PCC : $\delta_{u,v}^x(\mathbf{w}^e),\ n_u^v(\mathbf{w}^e) \quad u, v, x = 1 \ldots N$
      Propagation des trafics : $\gamma_u^v(\mathbf{w}^e),\ y_{u,v}(\mathbf{w}^e) \quad u, v = 1 \ldots N$
      Calcul du coût $\Phi(\mathbf{w}^e)$
      if $\Phi(\mathbf{w}^e) \leq \Phi_{min}$ **then**
        $\mathbf{w}_{next} = \mathbf{w}^e$ et $\Phi_{min} = \Phi(\mathbf{w}^e)$
      end if
    end for
    $\mathbf{w} = \mathbf{w}_{next}$
    if $\Phi(\mathbf{w}) < \Phi(\mathbf{w}^*)$ **then**
      $\mathbf{w}^* = \mathbf{w}$ ▷ Mise à jour solution de coût minimum
    end if
  end while
end procedure

Les opérations les plus coûteuses lors de l'exploration d'un voisinage sont le calcul des PCC et la propagation des trafics suite au changement d'une seule métrique. L'utilisation d'un algorithme de PCC dynamique permet de réduire considérablement le temps de calcul des PCC par rapport à l'algorithme de Dijkstra. Comme suggéré dans [79], nous utilisons une amélioration [50] de l'algorithme de Ramalingam et Reps [177]. Nous avons également utilisé une technique de propagation dynamique des flots proposée dans [79]. L'idée consiste à ne repropager les flots qu'à partir des nœuds dont un lien entrant ou sortant est apparu ou a disparu des PCC suite au changement de métrique.

4.2.3 Quelques résultats

Nous présentons quelques résultats dans le tableau 4.1. Onze topologies différentes faisant varier le nombre de routeurs, le nombre de liens, le nombre de flots ou leur intensité ont été testées. Les résultats présentent l'écart relatif en % du coût du routage IP par rapport à celui de la solution optimale de partage de charge sur tous les flots, obtenue avec l'algorithme Flow Deviation. Les métriques IP sont soit unitaires (Unit), soit inversement proportionnelles à la capacité (InvCapa), soit calculées par notre algorithme (LAAS), soit calculées par celui de Fortz & Thorup (Fortz). La dernière colonne précise les écarts relatifs en % entre les temps de calculs de l'algorithme de Fortz & Thorup et celui proposé. L'algorithme de Fortz & Thorup a été limité à 100 itérations et à 10 itérations successives augmentant le coût.

# nœuds	# liens	# OD	Unit	InvCapa	LAAS	Fortz	Temps Fortz
7	16	3	9.0%	3.4%	3.3%	3.3%	1570%
8	24	8	144%	150%	19%	19%	4640%
10	28	6	132.0%	1.0%	1.5%	1.5%	560%
12	19	14	8.8%	1.6%	1.6%	1.6%	606%
12	46	13	42.3%	0.15%	0.7%	1.7%	112%
13	48	12	26.6%	25.9%	11.3%	20.2%	1100%
13	48	12	17.5%	16.0%	7%	8.4%	2280%
13	48	12	58.8%	82.1%	36.9%	36.9%	1170%
15	42	12	6400.0%	3.87%	4.5%	3.8%	1640%
21	46	25	14.3%	1.6%	0.03%	0.03%	3690%
30	122	13	84.0%	2.3%	2.3%	2.2%	7770%
50	148	15	30.5%	3.35%	2.4%	1.5%	2300%
108	342	960	321.0%	321.0%	11.4%	21.7%	439%
108	342	6692	368.0%	13.1%	146.7%	85.8%	315%

TABLE 4.1 – Comparaison des résultats

On peut constater que l'algorithme proposé se comporte bien dans la plupart des cas. Les solutions obtenues sont souvent proches de l'optimum de partage de charge (borne inférieure). Dans la plupart des cas, les solutions obtenues ont un coût comparable à celui de l'algorithme de Fortz & Thorup. L'algorithme proposé a l'avantage d'avoir des temps de calcul bien plus faibles

et de permettre une optimisation incrémentale du routage.

4.2.4 Extensions

Dans le cadre de la thèse de Zied Ben Hamouda, nous avons étudié le problème d'optimisation robuste des poids OSPF avec incertitude sur la demande. Le modèle d'incertitude combine un modèle de type "hose" avec un modèle de type intervalle pour chaque flot OD. Les techniques employées sont similaires, la différence principale étant que le calcul du coût requiert la solution de problèmes de flot à coût minimum [21].

4.3 Placement des LSP dans les réseaux MPLS

Pour offrir un service plus évolué que le simple service Best Effort, les opérateurs de réseaux Internet choisissent souvent de combiner la technique de commutation d'étiquettes MPLS avec l'architecture à différentiation de services DiffServ. En effet, elles offrent ensembles une solution évolutive pour la gestion de la qualité de service en aggrégeant les flux IP par classes de service et en traitant ces dernières de manière différenciée dans le réseau. Avec MPLS, des étiquettes sont ajoutées aux paquets IP au niveau des nœuds de frontière du domaine MPLS. Ces étiquettes sont ensuite utilisées par les routeurs de cœur de réseau pour acheminer les paquets IP sur des chemins prédéfinis, appelés LSP (Label Switched Path), en fonction de la classe de service du paquet. Un LSP entre deux routeurs peut correspondre à la route bond-à-bond de niveau 3. Ce LSP peut également être obtenu par un routage contraint prenant en compte non seulement la topologie du réseau mais aussi la disponibilité des ressources réseaux et éventuellement d'autres contraintes imposées par l'administrateur du réseau (par exemple éviter les liens satellites pour les flots sensibles au délai). Enfin, le routeur MPLS d'entrée peut également spécifier un chemin explicite (ER-LSP) pour le LSP. Quelle que soit la façon dont sont établis les LSP, les routeurs d'extrêmité peuvent associer un LSP différent à chaque trafic en fonction de sa classe de trafic (CoS) et des classes d'équivalence de transmission (FEC) associées aux LSP.

La possibilité de mettre en place des ER-LSP calculés hors ligne est une caractéristique très séduisante de MPLS dans la mesure où cela permet de router le trafic sur des chemins optimaux prenant en compte non seulement les contraintes administratives, mais aussi les régles d'ingéniérie de trafic et les exigences de QoS des flux. Cela suppose toutefois de résoudre un problème d'optimisation combinatoire NP difficile visant à router chaque LSP sur un et un seul chemin entre le nœud ingress et le nœud egress [82]. Plusieurs heuristiques ont été proposées pour résoudre ce problème [63, 16, 55, 54, 162, 122, 82]. Les études numériques que nous avons conduites ont montré que notre méthode en trois phases ILSP-OLS-ACO, présentée ci-dessous, est en général plus efficace que ces dernières.

4.3.1 L'algorithme de placement de LSP en trois phases

On considère un réseau de N nœuds et M liens représenté par un graphe $G = (X, U)$. La bande-passante du lien $u \in U$ est C_u Mbps. Ce réseau doit écouler un ensemble de K flots (LSP), le flot $k = 1, \ldots, K$ étant caractérisé par son nœud source $s(k)$, son nœud destination $t(k)$, sa classe de service $c(k)$ et sa demande en bande-passante d^k. On notera $p(k)$ le nombre de chemins possibles pour le flot k et P_i^k le $i$ème de ces chemins.

Soit x_j^k la variable de décision binaire telle que $x_j^k = 1$ si le flot k est routé sur le chemin P_j^k et $x_j^k = 0$ sinon. On pose de plus $\mathbf{x}^k = (x_j^k)_{j=1\ldots p(k)}$ et $\mathbf{x} = (\mathbf{x}^k)_{k=1\ldots K}$. Le vecteur $\mathbf{x}$ représente une solution du problème de routage. Cette solution est admissible si le vecteur $\mathbf{x}$ vérifie les contraintes suivantes :

$$\sum_{k=1}^{K} \sum_{j=1}^{p(k)} P_j^k(u)\, x_j^k\, d^k \leq C_u \ , \ u = 1 \ldots M \quad \text{et} \quad \sum_{j=1}^{p(k)} x_j^k = 1 \ , \ k = 1 \ldots K, \tag{4.7}$$

où nous avons utilisé la même notation pour désigner le chemin P_j^k et le vecteur P_j^k défini par $P_j^k(u) = 1$ si $u \in P_j^k$ et $P_j^k(u) = 0$ sinon.

Le coût d'une solution est lié au trafic sur les liens. Soit $y_u^k = \sum_j P_j^k(u)\, x_j^k\, d^k$ la capacité utilisée par le flot k sur le lien u dans la solution $\mathbf{x}$ et $y_u = \sum_{k=1}^{K} y_u^k$ la capacitée totale utilisée sur ce lien dans cette solution. Le coût de la solution admissible $\mathbf{x}$ est défini par

$$\Gamma(x) = \sum_{u=1}^{M} F(y_u, C_u), \tag{4.8}$$

où la fonction F est en général une fonction non linéaire. Dans les résultats présentés ci-dessous, la fonction F utilisée est une fonction quadratique définie par (A_1 et $A_2 \gg A1$ sont deux paramètres ajustables) :

$$F(y, C) = \begin{cases} A_1\, y^2 \text{si } y \leq C \\ A_1\, y^2 + \ A_2\, (y - C)^2 \quad \text{si } y > C \end{cases} \tag{4.9}$$

Le problème d'optimisation consiste à déterminer la solution admissible $\mathbf{x}$ de coût $\Gamma(x)$ minimal. Pour résoudre ce problème, l'heuristique proposée opére en trois phases : ILSP, OLS et ACO.

Phase 1 : le millefeuille

L'efficacité d'un algorithme de placement des LSP est fortement réduite par le nombre considérable de chemins reliant les extrêmités de chaque flot dans un grand réseau. Cette phase permet de ne selectionner qu'un sous-ensemble de chemins candidats pour chaque flot $k = 1 \ldots K$. Soit $\mathcal{P}_k$ l'ensemble des chemins candidats pour le flot k. L' idée générale de l'algorithme de sélection

des chemins que nous proposons est de charger progressivement le réseau avec des fractions de flots en R itérations (R=10 par exemple). A chaque itération j, un chemin candidat $P^k_{j^\star}$ est trouvé pour chaque flot k. Au début de chaque itération j on pose $\Omega_j = \{k \mid k = 1 \dots K\}$ et $y_u = 0$ pour $u = 1, \dots, M$, puis on effectue les étapes suivantes K fois :

1. Sélectionner un flot $k \in \Omega_j$ suivant une distribution uniforme,
2. Router la demande d^k/R sur son plus court chemin $P^k_{j^\star}$ en prenant pour chaque lien u le poids $F(y_u + \frac{d^k}{R}, C_u) - F(y_u, C_u)$ (algorithme de Dijkstra). Ce chemin est celui qui conduit à la plus petite augmentation du coût.
3. Faire $\mathcal{P}_k := \mathcal{P}_k \cup \{P^k_{j^\star}\}$ et $\Omega_j := \Omega_j - \{k\}$.
4. Mettre à jour les charges des liens y_u en propageant la demande d^k/R le long du chemin $P^k_{j^\star}$, i.e. incrémenter y_u de d^k/R pour tout lien $u \in P^k_{j^\star}$.

Cet algorithme permet de réduire considérablement le nombre de chemins candidats puisqu'il retient au plus R chemins par flot (le même chemin peut être découvert plusieurs fois).

Phase 2 : partage de charge optimal

Cette phase consiste à résoudre le problème initial en relaxant la contrainte de mono-routage, i.e.,

$$\begin{aligned} \text{minimiser} \quad & \Gamma(x) = \sum_{u=1}^{M} F(y_u, C_u) \\ \text{s. c.} \quad & \\ & \sum_{j=1}^{p(k)} x^k_j = 1 \qquad k = 1 \dots K, \\ & x^k_j \geq 0 \qquad j = 1 \dots p(k)\ , k = 1 \dots K, \end{aligned}$$

où $p(k) = |\mathcal{P}_k|$ a été calculé dans la première phase. Ce problème peut être résolu avec une méthode de directions admissibles comme par exemple l'algorithme du gradient projeté qui, à l'itération q, va opérer de la façon suivante :

$$x^k_j(q+1) = x^k_j(q) \ - \ \nu \left\{ (p(k) - 1) \frac{\partial \Gamma}{\partial x^k_j} \ - \ \sum_{i \neq j} \frac{\partial \Gamma}{\partial x^k_i} \right\} \tag{4.10}$$

On détermine ainsi très efficacement la solution optimale $(\overline{x}^k_j)$ de routage par partage de charge (optimale par rapport aux chemins sélectionnés). Cette solution de multiroutage n'est bien sûr pas admissible pour le problème initial, mais elle fournit un bon point de départ pour la recherche d'une solution monoroutée dans la phase suivante, tout en permettant d'éliminer les chemins P^k_j tels que $\overline{x}^k_j = 0$ de l'ensemble $\mathcal{P}_k$ des chemins candidats.

Phase 3 : optimisation par colonies de fourmis

Les algorithmes de colonies de fourmis sont des algorithmes d'optimisation combinatoire inspirés du comportement des fourmis [29, 62, 189]. Dans notre algorithme, les fourmis construisent récursivement des solutions au problème de placement des LSP grâce à une exploration aléatoire de l'espace des solutions. La distribution de probabilités utilisée par les fourmis est initialisée avec les valeurs de partage de charge optimales $\overline{x}_j^k$ déterminées dans la phase précédente. A l'itération t, chaque fourmi $l = 1, \ldots, m$ construit une solution en K étapes en routant un flot après l'autre suivant un ordre aléatoire. Plus précisément, à l'itération t la fourmi l va itérer sur les étapes suivantes jusqu'à ce que tous les flots soient routés (au début de l'itération t, $\Omega_t^l = \{k \mid k = 1 \ldots K\}$).

1. Sélection aléatoire du flot $k \in \Omega_t^l$ à router suivant une distribution uniforme.
2. Choix aléatoire d'un chemin pour le flot k :

$$\Pr[\text{la fourmi } l \text{ route le flot } k \text{ sur } P_j^k] = \frac{\left[\tau_j^k(t)\right]^\alpha \left[\eta_j^k\right]^\beta}{\sum_{i=1}^{p(k)} \left[\tau_i^k(t)\right]^\alpha \left[\eta_i^k\right]^\beta} \tag{4.11}$$

   où :
   - $\eta_j^k = \left(\sum_{u \in P_j^k} F(y_u + d^k, C_u) - F(y_u, C_u)\right)^{-1}$ est l'inverse de l'augmentation de coût résultante du routage du flot k sur le chemin P_j^k. Cette information heuristique, appelée visibilité, représente l'attractivité immédiate de ce choix.
   - $\tau_j^k(t)$ est la quantité de phéromones virtuelles associées au routage du flot k sur P_j^k. Cette quantité de phéromones est mise à jour à la fin de chaque itération et permet de représenter la désirabilité de ce choix apprise depuis le début. Elle reflète l'expérience acquise par les fourmis au cours de la résolution du problème.
   - les deux paramètres ajustables α et β contrôlent le poids relatif de l'intensité des phéromones $\tau_j^k(t)$ et de la visibilité η_j^k.
3. Mise à jour des charges des liens en propageant les demandes d^k le long du chemin $P_{j^\star}^{k,l}(t)$ sélectionné.

Chaque fourmi l répéte K fois les étapes ci-dessus pour construire une solution. Une fois que toutes les fourmis ont construit une solution, les quantités de phéromones associées aux décisions de routage sont mises à jour de la façon suivante :

$$\tau_j^k(t+1) = (1-\rho)\,\tau_j^k(t) \; + \; \sum_{l=1}^{m} \Delta\tau_j^{k,l}(t) \tag{4.12}$$

où ρ est le coefficient d'évaporation des phéromones (pour assurer une exploration efficace de l'espace des solutions), et $\Delta\tau_j^{k,l}(t)$ est la quantité de phéromones associées par la fourmi l au

routage du flot k sur le chemin P_j^k ; cette quantité dépend du coût de la solution globale construite par cette fourmi :

$$\Delta\tau_j^{k,l}(t) = \begin{cases} Q/\sum_{u=1}^{M} F(\sum_{q/u\in P_{j^\star}^{q,l}(t)} d^q,\ C_u) & \text{si } P_{j^\star}^{k,l}(t) = P_j^k \\ 0 & \text{sinon} \end{cases} \tag{4.13}$$

où le dénominateur représente le coût de la solution obtenue par la fourmi l et Q est un paramètre (comme noté par [29] pour le problème du voyageur de commerce, la valeur de Q n'influence que peu le résultat final ; on la fixe au coût de la solution optimale de partage de charge). Initialement, les quantités de phéromones $\tau_j^k(0)$ sont fixées à $\tau_0\,\overline{x}_j^k$, où τ_0 est une petite constante et où $\overline{x}_j^k$ est la fraction du flot k routée sur le chemin P_j^k dans la phase 2.

4.3.2 Quelques résultats

Nous présentons ci-dessous les résultats obtenus pour les réseaux de test décrits dans le Tableau 4.2. Ces résultats sont comparés avec ceux obtenus avec l'algorithme glouton pCalc, aussi connu sous le nom de CSPF (avec un tri des LSP par ordre décroissant des demandes en bande-passante). Les résultats sont donnés dans le tableau 4.3. On constate que sur tous ces exemples notre algorithme permet de placer tous les LSP, ce qui n'est pas toujours le cas de pCalc. De plus, pour les instances où les deux algorithmes arrivent à router tous les LSP, notre algorithme dégage plus de bande-passante résiduelle. Ses temps de calcul sont tout à fait raisonnables pour un algorithme hors ligne. D'autres expérimentations ont permis de montrer que la solution de monoroutage obtenue avec cet algorithme a souvent un coût très proche de la solution optimale de partage de charge [44, 92].

TABLE 4.2 – Caractéristiques des réseaux de test.

	# Nœuds	# Liens	# LSPs
Test1	5	6	5
Test2	15	21	49
Test3	15	22	49
Test4	30	75	92
Test5	30	75	435
Test6	30	75	947

4.3.3 Extensions

Nous avons traité plusieurs extensions du problème décrit ci-dessus. Tout d'abord, dans [92], nous avons considéré une fonction objectif représentant le délai dans un réseau DiffServ combinant des ordonnancements à priorités fixes et WFQ (Weighted Fair Queueing). Dans ce même article, nous avons également décrit comment prendre en compte les protections de bout-en-bout (LSP

TABLE 4.3 – Comparaison de l'algorithme en 3 phases avec pCalc.

	Algo 3 phases				**pCALC**			
	Coût	Temps (sec)	# non routés	BP résiduelle	Coût	Temps (sec)	# non routés	BP résiduelle
Test1	9.3	0.02	0	2	9.3	0.02	0	2
Test2	2010.7	1.11	0	327	1952.6	0.05	3	388
Test4	5014.9	12.29	0	1280	5999.5	0.20	3	1244
Test5	3754.3	100.09	0	1328	4866.9	0.58	0	1324
Test6	17425.1	124.28	0	5570	18025.5	1.20	68	7170

de backup) ainsi que les protections locales (LSP de Fast ReRoute). Ces travaux sont présentés en détail dans la thèse d'Anouar Rachdi [176]. D'autre part, dans le cadre de la thèse de Zied Ben Hamouda [21], nous avons étudié un problème assez proche : celui de la conception robuste des VPN avec incertitude sur la demande de type "hose model". Ces travaux ont été publiés dans [24].

4.4 Ordonnancement multi-processeur de tâches strictement périodiques

Dans le domaine avionique, les architectures embarquées connaissent depuis une dizaine d'années une mutation profonde avec l'apparition des architectures modulaires intégrées (IMA). En offrant aux applications embarquées un support d'exécution et de communication standard et mutualisé, ces architectures ont permis une réduction du poids et de la complexité de l'architecture physique. Cette réduction de la complexité au niveau bas s'est cependant traduite par une difficulté accrue de conception et d'intégration des applications car il faut gérer le partage des ressources au moyen de nombreux paramètres de configuration. Un des principaux problèmes concerne l'ordonnancement des fonctions avioniques sur les unités de traitement disponibles. La particularité de ce problème est qu'il porte sur l'ordonnancement de tâches strictement périodiques, c'est-à-dire des tâches qui s'exécutent de façon non-préemptive à intervalles de temps constants sur un horizon infini.

La plupart des travaux sur l'ordonnancement de tâches périodiques portent sur le cas où la préemption est permise [145, 136, 147]. En comparaison, le problème de l'ordonnancement périodique non-préemptif a reçu beaucoup moins d'attention [97] et a essentiellement été considéré dans le cas particulier d'une périodicité lâche [132]. Un des premiers travaux sur l'ordonnancement non-préemptif de tâches strictement périodiques est celui de Korst pour le traitement temps réel de signaux vidéo [130, 131]. Ces dernières années, plusieurs articles traitant de ce problème sont apparus [152, 161, 120]. Dans [66], Eisenbrand *et al.* cherchent à minimiser le nombre de processeurs pour ordonnancer des tâches strictement périodiques. Ils montrent que si les périodes

sont harmoniques, on peut obtenir une 2-approximation. Dans [67], les auteurs traitent du même problème avec quelques contraintes supplémentaires dans un contexte avionique. En supposant des périodes harmoniques, ils proposent une formulation linéaire en nombres entiers qui, combinée avec une heuristique de type bin-packing, permet de résoudre optimalement des problèmes de tailles réelles.

Nos travaux ont permis d'apporter une solution efficace à ce problème sans hypothèse restrictive sur les périodes (cf. thèse d'Ahmad Al Sheikh [191]). Nous proposons tout d'abord une formulation sous la forme d'un programme linéaire en nombres entiers intégrant de nombreuses contraintes temporelles et de ressource de ce problème NP difficile [5]. Pour permettre le passage à l'échelle, nous proposons également une heuristique inspirée de la théorie des jeux dans laquelle chaque tâche adapte à son tour son ordonnancement pour maximiser sa propre fonction d'utilité (qui est liée aux marges d'évolution des tâches) [4]. Nous montrons la convergence de cet algorithme vers un point d'équilibre dans lequel aucune tâche n'a intérêt à modifier unilatéralement sa stratégie et établissons qu'il existe au moins un équilibre qui est globalement optimal. Les résultats numériques obtenus montrent que cet algorithme est beaucoup plus rapide que la méthode exacte et fournit une bonne approximation. Pour améliorer encore la qualité des solutions obtenues, nous utilisons cette heuristique dans un algorithme multi-start qui permet d'obtenir des garanties probabilistes sur l'optimalité des équilibres atteints.

4.4.1 Ordonnancement uniprocesseur

Etant donné un ensemble $\Pi = \{1, \ldots, N\}$ de N tâches strictement périodiques, nous cherchons un ordonnancement non-préemptif de ces tâches permettant de garantir qu'il n'y a aucun recouvrement temporel dans leurs exécutions. Chaque tâche $i \in \Pi$ est caractérisée par sa période T_i et par son budget de temps b_i, qui représente la durée d'exécution maximale d'une instance (WCET). Posons $\mathcal{T}_i = \{0, 1, 2, \ldots, T_i - 1\}$. Nous notons $t_i \in \mathcal{T}_i$ la date de première exécution de la tâche i, aussi appelée offset. Nous définissons $\mathbf{t} = [t_1, \ldots, t_N]$ comme le vecteur des offsets, et $\mathcal{T} = \times_{i=1}^{N} \mathcal{T}_i$ comme l'ensemble des vecteurs d'offsets possibles. Les tâches étant strictement périodiques, pour un vecteur d'offsets $\mathbf{t}$ donné, l'instance k de la tâche i (ou $k$ème exécution) s'exécute dans l'intervalle

$$I_k^i(t_i) = [t_i + k\,T_i, t_i + k\,T_i + b_i]\,. \tag{4.14}$$

Le problème que nous considérons consiste à affecter une date de première exécution à chaque tâche de telle manière qu'il n'y ait aucun recouvrement dans le temps de leurs exécutions, c'est à dire à déterminer un vecteur d'offsets $\mathbf{t} \in \mathcal{T}$ tel que $I_k^i(t_i) \cap I_l^j(t_j) = \emptyset$ pour tous $k, l \in \mathbb{Z}$.

Puisque nous voulons garantir qu'il n'y a pas de recouvrement entre deux exécutions, il est naturel de s'intéresser à la distance minimale entre deux débuts d'exécution. Le résultat essentiel

est présenté dans le Lemme 4.1, où $g_{i,j}$ est le *plus grand commun diviseur* de T_i et T_j et où le symbole % est utilisé comme notation abrégée de l'opérateur modulo, i.e. $a\%b$ doit être lu comme $a \mod b$.

Lemme 4.1

$$\min_{k,l\in\mathbb{Z}} |(t_j + lT_j) - (t_i + kT_i)| = \min\left[(t_j - t_i)\%g_{i,j}, (t_i - t_j)\%g_{i,j}\right] \tag{4.15}$$

Nous déduisons de ce résultat une condition necéssaire et suffisante pour l'ordonnançabilité de deux tâches qui est décrite dans le théorème suivant [130].

ThÈorËme 4.1 *Les exécutions de deux tâches i et j ne se recouvrent pas si et seulement si $b_i \leq (t_j - t_i)\%g_{i,j}$ et $b_j \leq (t_i - t_j)\%g_{i,j}$, ou, de façon équivalente, si et seulement si $b_i \leq (t_j - t_i)\%g_{i,j} \leq g_{i,j} - b_j$.*

Le terme $\frac{(t_j - t_i)\%g_{i,j}}{b_i}$ représente le facteur multiplicatif maximal par lequel la durée d'exécution b_i de la tâche i peut être multipliée sans interférer avec les exécutions de la tâche j. On peut voir ce terme comme la marge d'évolution sur la durée b_i par rapport à la tâche j, et si ce terme est supérieur ou égal à un pour toutes tâches $i, j \neq i$, cela signifie que le vecteur d'offsets $\mathbf{t}$ est admissible. Une condition suffisante pour que l'ordonnancement $\mathbf{t}$ soit admissible est donc que $\min_{i\neq j} \frac{(t_j - t_i)\%g_{i,j}}{b_i} \geq 1$. Pour déterminer un ordonnancement admissible, il apparaît donc naturel de chercher à maximiser ce minimum. Ainsi, en introduisant

$$d_{ij}(\mathbf{t}) = \min\left(\frac{(t_j - t_i)\%g_{i,j}}{b_i}, \frac{(t_i - t_j)\%g_{i,j}}{b_j}\right), \tag{4.16}$$

le problème d'ordonnancement peut être formulé de la façon suivante :

$$\begin{aligned} &\text{maximiser } \min_{i,j\neq i} d_{ij}(\mathbf{t}), \\ &\text{sous la contrainte } \mathbf{t} \in \mathcal{T}. \end{aligned} \tag{OPT}$$

Remarquons de plus que si la valeur optimale de ce problème est strictement supérieure à 1, cela permet de garantir une marge d'évolution minimale sur les budgets de temps alloués aux tâches qui peut être utile à l'avenir si les traitements effectués par ces tâches évoluent. Il est possible d'écrire ce problème d'ordonnancement uniprocesseur sous la forme d'un programme linéaire en nombres entiers (*cf.* [5] pour plus de détails) :

$$\begin{aligned} &\text{maximiser} && \alpha \\ &\text{sous les contraintes} && \\ & && \mathbf{t} \in \mathcal{T}, \\ & && (t_j - t_i) - q_{j,i}\, g_{i,j} \geq \alpha\, b_i,\ \forall (i,j) \in \Pi^2, \\ & && (t_j - t_i) - q_{j,i}\, g_{i,j} \leq g_{i,j} - \alpha\, b_j,\ \forall (i,j) \in \Pi^2, \\ & && t_i \in [0, T_i),\ \forall i \in \Pi, \end{aligned}$$

où la variable $q_{j,i}$ représente en fait le quotient entier $\left\lfloor \frac{t_j - t_i}{g_{i,j}} \right\rfloor$. Bien que ce programme linéaire en nombres entiers puisse être résolu numériquement, il n'est possible de résoudre que des exemples de tailles modestes en un temps raisonnable – rappelons en effet que le problème est NP complet au sens fort.

Algorithme de la meilleure réponse dans le cas uniprocesseur

L'algorithme de la meilleure réponse que nous proposons pour résoudre le problème d'optimisation (OPT) est inspiré d'un algorithme du même nom en théorie des jeux [83]. Dans cet algorithme, nous identifions les tâches à des joueurs jouant un jeu séquentiel. Dans ce jeu, chaque tâche adapte à son tour sa stratégie (i.e. son offset) en fonction des offsets des autres tâches. Le jeu se poursuit jusqu'à ce que le vecteur d'offsets converge vers un point d'équilibre, appelé équilibre de Nash, dans lequel plus aucune tâche n'a interêt à changer sa stratégie. Notons t_j^n la stratégie de la tâche j au début de l'itération n et supposons qu'à cette itération c'est au tour de la tâche i de jouer. Cette tâche va calculer son offset de manière à maximiser sa distance relative par rapport aux autres tâches en résolvant le problème suivant :

$$\begin{aligned} &\text{maximiser } \min_{j \neq i} d_{i,j}(x, \mathbf{t}_{-i}^n) \\ &\text{sous la contrainte } \ x \in \mathcal{T}_i, \end{aligned} \tag{SCHD-i}$$

où, suivant la notation habituelle en théorie des jeux, $\mathbf{t}_{-i} = [t_1, t_2, \ldots, t_{i-1}, t_{i+1}, \ldots, t_N]$ est le vecteur d'offsets de tous les joueurs autres que i. Le joueur i va alors fixer t_i^{n+1} à la valeur x donnant la solution optimale du problème SCHD-i, c'est-à-dire sa meilleure réponse. Si cette dernière n'est pas unique, nous supposons que la tâche va retenir le plus petit offset parmi ceux donnant la meilleure réponse.

Dans la suite, définissons

$$\alpha_i^n = \min_{j \neq i} d_{i,j}(\mathbf{t}^n) \tag{4.17}$$

$$\mathcal{S}_i(\mathbf{t}_{-i}^n) = \operatorname*{argmax}_{x \in \mathcal{T}_i} \min_{j \neq i} d_{i,j}(x, \mathbf{t}_{-i}^n), \tag{4.18}$$

où α_i^n est l'utilité du joueur i après l'itération n, et $\mathcal{S}_i(\mathbf{t}_{-i}^n)$ est l'ensemble des meilleures réponses de ce joueur. Nous supposons de plus que si le joueur i ne peut améliorer son utilité α_i^n, il ne change pas sa stratégie, i.e. $t_i^{n+1} = t_i^n$. Cette hypothèse, bien que non restrictive, est utile pour démontrer la convergence de l'algorithme. Le pseudo-code de l'algorithme de meilleure réponse est décrit dans l'Algorithme 4. A l'étape 6 de l'algorithme, $n\%N + 1$ donne l'index du joueur dont c'est le tour d'adapter sa stratégie à l'itération n.

Deux propriétés importantes de cet algorithme sont formulées dans les théorèmes suivants.

Algorithm 4 Meilleure Réponse Uniprocesseur

```
1: procedure BestResponse
2:     REQUIRE t^0
3:     n ← 0
4:     while t^n ≠ t^{n−N} do
5:         for i = 1 to N do
6:             if i = n%N + 1 and max_x min_{j≠i} d_{i,j}(x, t^n_{−i}) > α^n_i then
7:                 t^{n+1}_i ← min argmax (SCHD-i)
8:             else
9:                 t^{n+1}_i ← t^n_i
10:            end if
11:        end for
12:        n ← n + 1
13:    end while
14:    Return t^n
15: end procedure
```

ThÈorËme 4.2 *L'algorithme de la meilleure réponse converge vers un point d'équilibre.*

ThÈorËme 4.3 *Il existe au moins un point d'équilibre qui est aussi une solution optimale du problème* (OPT).

En conséquence des résultats précédents, nous déduisons que si le point de départ est choisi de manière appropriée, l'algorithme de la meilleure réponse convergera vers une solution globalement optimale. La proposition suivante fournit de plus une borne supérieure sur le nombre d'itérations.

Proposition 4.3 *Soit* $\alpha_{max} = \max_i \min_{j\neq i} \frac{g_{i,j}}{b_i+b_j}$ *et* $\Delta = \min_{j,k} \frac{1}{ppcm(b_j,b_k)}$. *L'algorithme de la meilleure réponse converge en au plus* $\binom{N+K}{K}N$ *itérations, où* $K = \lceil \alpha_{max}\Delta^{-1} \rceil$.

Cette borne supérieure sur le nombre d'itérations est une estimation pessimiste qui est exponentielle en le nombre de tâches. En pratique, pour toutes les expérimentations effectuées, l'algorithme a toujours convergé en quelques dizaines d'itérations au pire.

Un point critique de cet algorithme concerne le calcul de la meilleure réponse d'un joueur. La meilleure réponse de la tâche i peut bien sûr être calculée en effectuant une recherche linéaire, ce qui nécessite $O(T_i)$ opérations. On peut en fait déterminer la meilleure réponse de façon beaucoup plus efficace en utilisant le résultat suivant.

ThÈorËme 4.4 *Si les offsets peuvent prendre des valeurs réelles,i.e.* $\mathcal{T}_i = [0, T_i)$, *alors* $\mathcal{S}_i(\mathbf{t}_{-i}) \subset \mathcal{I}_i(\mathbf{t}_{-i}) \subset \mathcal{T}_i$, *où*

$$\mathcal{I}_i(\mathbf{t}_{-i}) = \bigcup_{(j,k)\in(\Pi\setminus\{i\})^2} \left\{ x : \frac{(x-t_j)\%g_{i,j}}{b_j} = \frac{(t_k-x)\%g_{i,k}}{b_k} \right\}$$

est l'ensemble des points d'intersection associés à la tâche i.

Ainsi, la solution de (SCHD-i) peut être obtenue en restreignant la recherche aux points de l'ensemble $\mathcal{I}_i(\mathbf{t}_{-i})$. Dans la mesure où le problème original n'est défini que pour des offsets entiers, l'heuristique décrite dans l'Algorithme 4 va chercher la meilleure réponse de la tâche i dans les entiers directement inférieurs et supérieurs aux points de l'ensemble $\mathcal{I}_i(\mathbf{t}_{-i})$, plutot que d'examiner tous les T_i points possibles. Il est de plus possible de génerer très efficacement les points de $\mathcal{I}_i(\mathbf{t}_{-i})$ grâce à la méthode décrite dans [4].

4.4.2 Ordonnancement multiprocesseur

Nous montrons dans ce paragraphe comment l'algorithme de la meilleure réponse pour l'ordonnancement uniprocesseur s'étend naturellement au cas multiprocesseur. Soit $\mathcal{P} = \{1, \ldots, P\}$ un ensemble de P processeurs, le processeur k étant caractérisé par sa capacité mémoire M_k et par le nombre maximal H_k de tâches qu'il peut accueillir. Un ordonnancement n'est plus seulement décrit par la donnée du vecteur d'offsets $\mathbf{t}$, mais également par l'affectation d'un processeur à chacune des tâches. Cette affectation peut être représentée par un vecteur de variables binaires $\mathbf{a} = (a_{i,k})_{i\in\Pi, k\in\mathcal{P}}$ telles que $a_{i,k} = 1$ si la tâche i est affectée au processeur k, et $a_{i,k} = 0$ sinon. On peut alors formuler le problème d'ordonnancement multiprocesseur comme un programme linéaire en nombres entiers de la façon suivante (certaines contraintes spécifiques à l'application avionique sont omises ici, cf [5]) :

$$\text{maximiser}_{\mathbf{a},\mathbf{t}}\, \alpha \tag{4.19}$$

$s.t.$

$$\sum_{p_k\in\mathcal{P}} a_{i,k} = 1 \quad ,\forall i \in \Pi, \tag{4.20}$$

$$\sum_{i\in\Pi} a_{i,k}\, m_i \leq M_k \quad ,\forall k \in \mathcal{P}, \tag{4.21}$$

$$\sum_{i\in\Pi} a_{i,k} \leq H_k \quad ,\forall k \in \mathcal{P}, \tag{4.22}$$

$$(t_j - t_i) - q_{j,i}\, g_{i,j} \geq \alpha\, b_i - (2 - a_{i,k} - a_{j,k})\, Z \quad ,\forall k, \forall (i,j), \tag{4.23}$$

$$(t_j - t_i) - q_{j,i}\, g_{i,j} \leq g_{i,j} - \alpha\, b_j + (2 - a_{i,k} - a_{j,k})\, Z \quad ,\forall k, \forall (i,j), \tag{4.24}$$

$$a_{i,k} \in \{0,1\} \quad ,\forall k, \forall i, \tag{4.25}$$

$$t_i \in [0, T_i) \quad ,\forall i \in \Pi, \tag{4.26}$$

$$q_{j,i} \in \mathbb{Z} \quad ,\forall (i,j), \tag{4.27}$$

Comme dans le cas uniprocesseur, la fonction objectif (4.19) représente le minimum des marges d'évolution des tâches. Les contraintes (4.20) imposent le choix d'un seul processeur par

tâche, tandis que les contraintes (4.21) et (4.22) sont celles associées aux capacités mémoire et en nombre de tâches des processeurs. Les contraintes (4.23) et (4.24) expriment la condition d'ordonnançabilité (cf. Théorème 4.1) pour les couples de tâches affectées au même processeur (Z est une grande constante qui permet de garantir que ces contraintes ne sont actives que si $a_{i,k} = a_{j,k} = 1$). Les contraintes (4.25), (4.26) et (4.27) décrivent le domaine des variables. Cette formulation linéaire n'est évidemment utilisable que pour des problèmes de tailles modestes.

Algorithme de la meilleure réponse dans le cas multiprocesseur

L'algorithme de la meilleure réponse pour l'ordonnancement uniprocesseur s'étend au cas multiprocesseur de la façon suivante. A son tour, une tâche i va calculer sa meilleure réponse sur chacun des processeurs, les uns après les autres. Puis, elle va sélectionner le processeur et l'offset sur ce processeur lui permettant de maximiser son utilité qui est définie de la façon suivante

$$\alpha_i^n = \min_{\{j:j\neq i, a_{i,k}^n = a_{j,k}^n \forall k\}} d_{i,j}(\mathbf{t}^n), \tag{4.28}$$

et qui représente la marge d'évolution de la tâche i par rapport aux tâches ordonnancées sur le même processeur qu'elle et pour des vecteurs d'offsets $\mathbf{t}^n$ et d'allocation $\mathbf{a}^n$ donnés. Comme dans le cas uniprocesseur, on peut montrer que cet algorithme converge vers un point d'équilibre et qu'il existe au moins un point d'équilibre qui est globalement optimal.

4.4.3 Un exemple de résultat

Pour illustrer les résultats obtenus, considérons le problème d'ordonnancement uniprocesseur avec 20 tâches non harmoniques dont les caractéristiques temporelles sont indiquées dans le tableau 4.4. Pour cet exemple, l'algorithme de la meilleure réponse fournit une marge d'évolution minimale égale à $\alpha = 1.41$ en 2.83 secondes, tandis que la résolution avec CPLEX [115] de la formulation linéaire en nombres entiers ne permet d'obtenir qu'une marge de $\alpha = 1.11$ au bout d'une heure de calcul. De manière générale, l'ensemble des expérimentations effectuées ont montré que l'algorithme de la meilleure réponse permet d'obtenir des solutions de bonne qualité, et généralement admissibles, en des temps calculs très inférieurs à une résolution exacte basée sur la programmation linéaire en nombres entiers.

4.4.4 Algorithme multi-start

La qualité des solutions obtenues avec l'algorithme de la meilleure réponse peut encore être améliorée en utilisant une méthode multi-start [153]. En effet, l'espace des solutions du problème d'ordonnancement est divisé en région d'attraction, l'algorithme de la meilleure réponse conduisant vers le même équilibre pour tous les points initiaux appartenant à la même région.

TABLE 4.4 – Exemple uniprocesseur avec 20 tâches non harmoniques (hyperpériode de 756000).

Tâche	1	2	3	4	5	6
Budget de temps	10	30	30	10	10	10
Période	1200	1200	3600	1200	1200	1500

7	8	9	10	11	12	13
10	10	30	10	10	45	40
4200	1000	2000	4000	1200	2400	2000

14	15	16	17	18	19	20
40	60	80	30	10	60	40
4000	3000	3000	2700	200	1800	1800

Une méthode multi-start va permettre de générer aléatoirement des points de départ de l'heuristique appartenant à des régions d'attraction différentes. Des régles d'arrêt Bayesiennes [28] sont utilisées pour arrêter l'exploration aléatoire de l'espace des solutions lorsque des garanties probabilistes suffisantes sont obtenues sur l'optimalité de la meilleure solution trouvée.

Dans le cas uniprocesseur, la méthode multi-start s'exécute en quelques minutes. Elle nous a permis de réduire l'écart relatif moyen à l'optimum de 7.8% avec l'heuristique à 0.25% pour les exemples harmoniques, et de 4.2% à 0% pour les exemples non harmoniques. Dans le cas multiprocesseur, nous avons obtenu un temps de calcul moyen de 16 minutes. L'écart à l'optimum est passé de 14.17% à 3.84% pour les instances harmoniques, et de 18.5% à 9.6% pour les instances non harmoniques. Nous avons observé que dans le cas multiprocesseur les points d'équilibre optimaux peuvent se trouver dans de petites régions d'attraction, ce qui peut nécessiter un critère d'arrêt plus strict que celui que nous avons utilisé.

Chapitre 5

Prospective de recherche

Nous décrivons dans ce chapitre les travaux de recherche qui nous méneront dans les années à venir. Ces travaux concernent trois sujets distincts :

- la gestion des incertitudes lors de la conception des systèmes embarqués critiques (paragraphe 5.1),
- l'analyse et la conception d'algorithmes de routage distribués en utilisant la théorie des jeux (paragraphe 5.2),
- et la supervision et le contrôle en boucle fermée des réseaux de communication (paragraphe 5.3).

5.1 Ordonnancement pour les systèmes embarqués critiques

5.1.1 Contexte scientifique

Les systèmes embarqués automobiles et avioniques ont connus des évolutions importantes ces dernières décennies. Ainsi, les systèmes avioniques actuels intégrent dans un espace réduit de nombreux calculateurs, capteurs, actuateurs ainsi que des unités de contrôle et d'affichage qui doivent fonctionner en "temps réel" tout en répondant à des exigences élevées en termes de sûreté et de fiabilité. De même, les voitures modernes sont devenus des systèmes temps réel critiques et extrêmement complexes intégrant de nombreux composants électroniques et informatiques : la plupart sont aujourd'hui équipées d'un système de freinage anti-blocage, d'une unité électronique de contrôle moteur, d'un régulateur de vitesse programmable ou encore d'un système de navigation GPS, etc.

Que ce soit dans le domaine avionique ou dans celui de l'automobile, les systèmes embarqués traditionnels ont tous été développés en suivant une architecture fédératrice qui conduit à un ensemble d'équipements pour chaque sous-système. Face à l'explosion du nombre des fonctions qu'ils réalisent, et à une augmentation forte de la complexité globale (grand nombre de calculateurs et de bus imbriqués), il a fallu complètement repenser la façon de concevoir ces systèmes. De nouvelles architectures ont ainsi vu le jour. Celles-ci ont permis une homogénéisation des

plate-formes embarquées (IMA pour l'avionique et ECU pour l'automobile) et une standardisation des supports d'exécution (ARINC en avionique et AUTOSAR en automobile). Une des caractéristiques fonfamentales de ces nouvelles architectures est de permettre le partage des ressources de calcul et de communication entre les différents sous-systèmes.

Parmi les gains apportés par les nouvelles architectures, la diminution du nombre de ressources embarquées (par la possibilité offerte de multiplexer ces ressources) et donc la diminution du poids, se traduit par la nécessité de partager temporellement et spatialement une ressource par un plus grand nombre de fonctions. Les choix qui pouvaient être effectués « manuellement »par un expert pour dimensionner un système, sont devenus irréalisables de par la taille des problèmes considérés et la complexité des interactions entre les fonctions. La problématique de l'allocation de ressources et de l'ordonnancement pour les systèmes embarqués devient une réalité à laquelle les industriels du domaine des systèmes embarqués sont confrontés et pour laquelle il n'existe pas de réponse simple.

Ce qui distingue l'ordonnancement des systèmes embarqués critiques par rapport aux problèmes, par exemple, de production ou de tournée de véhicule, réside dans l'aspect récurrent des fonctions à réaliser, dans la garantie forte des contraintes et dans les mécanismes d'ordonnancement dynamiques sur lesquels ils s'appuient. En raison de la nature critique de la majorité des applications embarquées temps réel, la garantie a priori des diverses exigences de ces systèmes est obligatoire. Cela nécessite le développement de techniques et d'outils permettant d'assurer la conformité du système vis-à-vis de sa spécification. Ceux-ci doivent s'inscrire dans un processus de conception et doivent permettre de valider aussi bien les aspects fonctionnels qu'extra-fonctionnels. Parmi ces techniques, la production et la validation des ordonnancements au niveau spatial et temporel est une nécessité. Etant donné les enjeux applicatifs considérables, un effort de recherche important a donc été consacré au dévelopement d'une théorie de l'ordonnancement des systèmes temps réels. De nombreux résultats théoriques ont été obtenus pour garantir l'ordonnançabilité d'un système temps réel, qu'il s'agisse d'un système dirigé par le temps ou d'un système dirigé par des événements, et de nombreux systèmes sont actuellement conçus en s'appuyant sur ces résultats.

5.1.2 Objectifs scientifiques

En dépit des succès obtenus, les approches actuellement utilisées pour concevoir les systèmes temps réel ont de nombreux défauts. Le projet OSEC, dont j'assure la direction scientifique, est un projet de recherche interne au LAAS qui a pour ambition de développer de nouvelles approches méthodologiques pour la gestion des incertitudes lors de la conception d'un système temps réel critique, en particulier dans les domaines automobile et aéronautique. Le projet prend

en compte deux types d'incertitude, celles liées aux paramètres des tâches temps réel et celles liées à l'état du système :

- **Incertitudes sur les dates d'activation et les durées des tâches.** Le bon fonctionnement d'un système temps réel dépend non seulement de l'exécution correcte des tâches, mais aussi du respect de leurs contraintes temporelles. La conception de ces systèmes s'appuie souvent sur une analyse dans le pire des cas pour construire un ordonnancement satisfaisant strictement les exigences temps réel des tâches. Cette analyse est pessimiste car elle repose sur les temps d'exécution dans le pire des cas des tâches (WCET : worst-case execution time). Elle conduit donc généralement à un surdimensionnement excessif des ressources, et impose des contraintes très fortes sur les développements. Par exemple, elle peut conduire à supposer qu'un traitement qui ne se produit que de manière exceptionnelle, est en fait effectué régulièrement. Elle est néanmoins appropriée dans certains domaines applicatifs, comme par exemple l'aéronautique, où les contraintes de sécurité sont primordiales. Dans d'autres domaines en revanche, comme par exemple celui de l'automobile, cette approche n'est pas du tout adaptée dans la mesure où une garantie probabiliste sur le respect de l'échéance peut être suffisante pour la plupart des tâches. Une modélisation plus fine, prenant en compte l'incertitude sur les temps d'exécution, peut alors permettre de réduire le surdimensionnemnt et simplifier les développements. Un des objectifs du projet est de développer des outils théoriques pour la conception des systèmes temps réel prenant en compte l'incertitude sur les durées d'exécution et les dates d'activation des tâches.

- **Incertitudes sur l'état du système.** La conception des systèmes embarqués distribués se focalise en général sur l'état normal du système et la sûreté de fonctionnement est obtenue en introduisant de la redondance. La reconfiguration est un mécanisme de tolérance aux fautes permettant de fournir les fonctionnalités attendues en réallouant les ressources en réponse à une panne. La reconfiguration intervient ainsi en complément de la redondance existante pour réagir aux pannes des modules de traitement et ainsi augmenter la fiabilité du système. Le projet a pour objectif de developper de nouvelles approches d'optimisation pour calculer hors ligne l'ordonnancement périodique nominal et un ensemble de reconfigurations en cas de panne. Ces reconfigurations seront choisies pour minimiser une certaine distance par rapport à la configuration parente.

Le projet OSEC se déroulera sur deux ans. Il fait intervenir des chercheurs des groupes Mogisa, MRS et OLC du LAAS et prévoit l'embauche de deux postdoctorants (un an chacun).

5.2 Théorie des jeux et routage distribué

Les problèmes de routage sont omniprésents dans de nombreux contextes applicatifs, et notamment dans les grands systèmes distribués (les fermes de serveurs, par exemple) et les réseaux

de communication, qu'il s'agisse de l'Internet, de réseaux sans fil ou de réseaux de capteurs. Ces problèmes se posent en effet dès qu'un ensemble de ressources doivent être partagées entre des flux d'activités concurentes. Ils consistent à adapter les ressources utilisées à la demande de manière à optimiser un certain critére, qui est généralement lié à la performance perçues par les utilisateurs du système.

La plupart des travaux sur le sujet se sont intéressés à des algorithmes centralisés permettant le calcul d'une stratégie de routage optimale. Quand la taille du système n'est pas trop grande, un seul agent de routage peut alors mettre en oeuvre cette stratégie, que l'on peut voir comme un optimum social dans le sens où elle optimise la performance du système pour tous les flots. En pratique, il n'est généralement pas possible d'utiliser un seul agent de routage pour des raisons de scalabilité, et il faut se résoudre à utiliser un schéma de routage distribué utilisant plusieurs agents de routage. En dehors du gain en scalabilité, un des avantages est la robustesse face aux pannes ou aux perturbations des conditions de fonctionnement du système. Cependant, la conception et la mise en oeuvre de stratégies distribuées de routage posent de nombreux défis.

Un des principaux défi concerne la perte de performance induite par le passage à une architecture décentralisée. En effet, si dans l'architecture distribuée chaque agent de routage réalise de manière autonome une optimisation pour son propre trafic, on peut s'attendre à ce que la performance globale obtenue soit pire que dans une approche centralisée. Dans ce contexte, les questions principales qui se posent sont les suivantes :

- *Peut-on obtenir des garanties de performance pour ces stratégies de routage décentralisées ?*
- *Comment peut-on améliorer les performances de ces algorithmes ?*

Il s'agit des deux questions fondamentales que nous étudierons dans la thèse de Tatiana Seryogina qui est co-encadrée avec Balakrishna Prabhu (débutée en octobre 2011).

5.2.1 Contexte scientifique

Puisque le comportement égoiste des agents de routage peut conduire à des dégradations de performance inacceptables, une approche naturelle consiste à introduire de la coordination entre eux pour une prise de décision collective. Ce type de problème peut être traité en utilisant des approches d'optimisation globale distribuée [165]. Toutefois, ceci peut induire des surcoûts de communication et de synchronisation non négligeables pour de grands systèmes. Une approche alternative consiste à conserver une architecture basée sur des agents de routage autonomes, mais à les concevoir de manière à ce que la solution décentralisée soit efficace par rapport à sa contrepartie centralisée.

Dans une telle approche, les décisions de routage sont prises par chaque agent de routage

indépendamment, en fonction de ses propres objectifs individuels. Ceci conduit à un jeu de routage entre les différents agents [61, 10]. La théorie des jeux fournit le cadre théorique naturel pour étudier de tels problèmes. En particulier, elle peut être utilisée pour étudier la convergence vers des points d'équilibre (des stratégies de routage), appelés équilibres de Nash, tels qu'une déviation unilatérale ne permet à aucun agent de routage d'améliorer les performances du trafic qu'il route. Une mesure standard de l'inefficacité du routage distribuée est alors le « Prix de l'Anarchie » (PdA) [133], qui est défini comme le ratio dans le pire des cas entre la performance obtenue par l'algorithme distribué et la performance optimale. De nombreux travaux ont été consacrés ces dernières années à l'évaluation du PdA. Toutefois, pour des réseaux ayant des topologies générales, les résultats connus sont des résultats asymptotiques (i.e. lorsque chaque job/paquet peut choisir lui même sa route) limités à des fonctions coûts polynomiales [186]. Pour des fonctions plus générales, les seuls résultats connus concernent des topologies de type "liens parallèles" [170, 68, 127, 17, 18].

Plusieurs méthodes ont été proposées pour converger vers des équilibres de Nash efficaces. Des stratégies basées sur l'affectation de prix aux ressources ont été étudiées dans [128]. Une autre méthode pour améliorer l'efficacité des algorithmes de routage distribués est basé sur ce que l'on appelle les jeux de Stackelberg [129], dans lesquels des agents contrôlés de manière centralisée influencent par leurs décisions de routage les actions prises par les agents egoistes pour forcer la convergence vers une solution optimale.

5.2.2 Objectifs scientifiques

Pour évaluer les performances des algorithmes de routage distribués, il s'agira de calculer ou de borner le « Prix de l'Anarchie ». L'idée principale que nous voulons explorer consiste à aborder cette question dans le cadre des jeux symétriques (par exemple quand tous les flux ont le même trafic). Nous avons en effet montré, pour des topologies simples, que ces jeux sont extrémaux dans le sens où c'est pour eux que les performances les pires sont obtenues (cf. paragraphe 3.5). De plus, ils se prêtent plus facilement à l'analyse, notamment avec l'utilisation de fonctions potentiels [164]. L'objectif sera donc, pour des topologies générales, de caractériser les conditions de symétrie sous lesquelles les performances les pires sont obtenues, puis de borner ces performances et de calculer le prix de l'anarchie. Un second volet de nos recherches concernera la détermination des objectifs devant être assignés aux agents de routage autonomes pour converger vers des stratégies de routage offrant des performances proches de l'optimum social.

5.3 Contrôle intelligent des réseaux

5.3.1 Contexte scientifique

La gestion d'un réseau IP/MPLS requiert aujourd'hui l'utilisation d'une multiplicité d'outils spécialisés. Les équipes chargées de l'exploitation opérationnelle du réseau utilisent des outils de gestion de configurations, de métrologie du trafic et de supervision du réseau. Ces outils fournissent "la vision" de l'état du réseau à un instant donné, mais nécessitent l'expertise de spécialistes pour décider d'éventuelles adaptations du réseau. A l'inverse, les équipes de planification travaillent, à partir de prévisions de trafic, sur des évolutions à moyen ou long terme du réseau en utilisant des outils de simulation et de planification. Ces outils disposent de "l'intelligence" permettant d'estimer la QoS des flux et d'optimiser le réseau, mais n'ont pas la vision de l'état du réseau. En couplant ces deux types d'outils, il serait à la fois possible de réaliser une planification plus fine et plus régulière du réseau et en même temps d'aider les équipes opérationnelles à prendre les décisions adéquates.

Le projet NEC (*Network Engineering & Control*) est un projet soutenu par le pôle Aerospace Valley et accepté par le Fonds Unique Interministériel (FUI) en 2011. Il regroupe les sociétés Cassidian, QoS Design et le LAAS-CNRS. Je suis le responsable LAAS de ce projet qui prévoit le recrutement de deux doctorants et d'un post-doctorant. Le projet NEC a pour objectif de concevoir et de fournir un environnement intégré pour la maîtrise et l'exploitation des réseaux IP/MPLS. Au-delà de la simple supervision, ce qui est visé est la conduite intelligente des réseaux informatiques (commande nominale, scenarii de secours). Il s'agit en fait de concevoir et de développer un centre de Supervision/Contrôle, donnant une vision globale et précise de l'état d'un réseau de télécommunications, avec des données agrégées et cohérentes, et qui soit un vrai système d'ingénierie et d'aide à la décision, basé sur des modèles de simulation (analytiques ou hybrides) de bout en bout, et des algorithmes permettant de prendre des décisions optimales et cohérentes vis a vis de tout le système et des objectifs poursuivis. Ce système, outre le traitement intelligent de l'information en un point unique, permettra d'une part, d'étudier précisément et de manière proactive, l'ensemble des scenarii du système et de son environnement et d'autre part, d'étudier les mécanismes réactifs permettant de contrôler (piloter) le système en situation opérationnelle face à toute sorte de perturbations (dégradation de la QoS, Sécurité ...)

5.3.2 Objectifs scientifiques

Dans le cadre du projet NEC, nous interviendrons sur les sujets suivants :

Simulation en ligne

Etant donnée la forte variabilité du trafic Internet à court terme, liée notamment à celle de l'activité des utilisateurs, il est impossible d'éviter des dégradations ponctuelles du service offert

sans mettre en œuvre des techniques de reconfiguration dynamique des ressources. Ceci suppose toutefois de disposer de modèles réseaux permettant d'évaluer non seulement la QoS des flux dans l'état actuel du réseau, mais aussi son évolution suite aux différentes actions qui peuvent être prises. Une contrainte forte est que ces évaluations doivent être faites en temps réel, ce qui interdit l'utilisation des modèles de simulation à événements discrets ou même de modèles hybrides.

Nos travaux viseront l'obtention de modèles analytiques permettant une estimation précise de la QoS des flots de trafic. Il s'agira de réduire l'écart entre modèles de niveau paquet et modèles de niveau flot, mais aussi d'intégrer le trafic de streaming dans ce type de modèles. Les approches développées seront basées sur la théorie des files d'attente, et en particulier sur les modèles à base de réseaux de files d'attente GPS (Generalized Processor Sharing) comme décrit au paragraphe 3.4.

Estimation de matrice de trafic

L'estimation de la matrice de trafic d'un réseau est un sujet important que nous avons déjà abordé dans le cadre de la thèse de Cédric Fortuny [76, 75, 73]. Il est aujourd'hui largement reconnu qu'il s'agit d'un problème difficile pour lequel aucune méthode efficace n'est disponible. La mesure directe de la matrice de trafic par des outils comme NetFlow ou IPFIX monopolise trop de ressources au niveau des routeurs et génère un trafic de reporting important. Les approches basées uniquement sur les mesures collectées par SNMP conduisent à des problèmes mal posés qui ne peuvent être résolus que par l'adjonction d'informations supplémentaires. Il peut s'agir par exemple d'informations a priori sur la structure de la matrice de trafic (approche de type tomogravité développée au Bell Labs) ou d'informations obtenues a posteriori en corrélant une série de mesures SNMP (approches de type FanOut, PCA ou basée sur le filtre de Kalman). Les taux d'erreurs des méthodes actuelles sont trop élevés pour qu'elles soient réellement exploitables.

L'idée originale que nous proposons pour réduire drastiquement l'erreur consiste à utiliser conjointement les mesures de trafic agrégé SNMP et des mesures détaillées obtenues via des sondes matérielles/logicielles externes. L'utilisation de quelques sondes externes, judicieusement placées et configurées pour filtrer uniquement les informations requises, doit permettre de lever l'indétermination inhérente au problème d'estimation de trafic, sans pour autant surcharger le réseau.

Routage IP adaptatif et robuste

L'objectif de nos travaux sera de développer une méthode de recherche locale pour adapter dynamiquement les poids OSPF aux évolutions de la matrice de trafic. On utilisera une approche d'optimisation robuste pour prendre en compte l'incertitude sur le trafic due aux erreurs inhérentes aux algorithmes d'estimation de trafic. L'ensemble initial d'incertitude sera défini par les

mesures de charge des liens collectées avec SNMP. L'accumulation de ces mesures au cours du temps combinée avec une hypothèse de régularité de la matrice de trafic et surtout avec l'information supplémentaire apportée par les changements de la matrice de routage permettront de réduire progressivement l'ensemble d'incertitude, et ainsi d'avoir un routage de plus en plus efficace.

Reroutage dynamique des LSP

Nous souhaitons également étudier l'adaptation dynamique du placement des LSP. L'idée consiste à analyser régulièrement les conditions de trafic et à identifier les goulots d'étranglement qui peuvent apparaître dans le réseau. Un algorithme permettra de décider de dérouter certains des LSP dont les routes primaires passent par ces goulots d'étranglement, pour les basculer sur des routes de réserve. Cette approche est largement employée dans les réseaux à commutation de circuits ou elle est connue sous le nom de débordement. Ainsi, à chaque LSP sera associé un certain nombre de routes pré-calculées, et pour certains des LSP ne satisfaisant plus les contraintes de QoS, une décision de débordement sur une de ces routes de réserve pourra être prise.

Chapitre 6

Publications

Revues à comité de lecture

1. U. AYESTA, O. BRUN, B.J. PRABHU, ***Price of Anarchy in Non-Cooperative Load Balancing,*** Performance Evaluation, accepté pour publication.

2. O. BRUN, C. BOCKSTAL et JM GARCIA, ***Analytical Approximation of the Jitter Incurred by CBR Traffics in IP Networks***, Telecommunication System Journal, Vol 33, Nos 1-3, Novembre 2006.

3. H. HASSAN, JM GARCIA, O. BRUN, ***Bandwidth allocation and session scheduling using SIP***, Journal of Communications, Vol. 1, No 5, Août 2006.

4. V. TEULIERE, O. BRUN, ***Parallelisation of the Particle Filtering Algorithm and Application to Doppler-Bearing Tracking of Maneuvering Sources***, Parallel Computing, Vol 29, No 8, pp 1069-1090, Aout 2003.

5. O. BRUN , V. TEULIERE, JM GARCIA, ***Parallel Particle Filtering***, Journal of Parallel and Distributed Computing, Vol. 62, No 7, pp 1186-1202, Juillet 2002.

6. JM GARCIA, O. BRUN et D. GAUCHARD, ***Transient Analytical Solution of M/D/1/N Queues***, Journal of Applied Probability, Vol. 39, No 4, pp 853-864, Décembre 2002.

7. JM GARCIA, D. GAUCHARD, O. BRUN, P. BACQUET, J. SEXTON, E. LAWNESS, ***Modélisation Différentielle et Simulation Hybride Distribuée***, Calculateurs Parallèles, Réseaux et Systèmes Répartis, 2002.

8. O. BRUN et JM GARCIA, ***Analytical Solution of Finite Capacity M/D/1 Queues***, Journal of Applied Probability, Vol. 37, No 4, Décembre 2000.

Conférences internationales avec actes et comité de sélection

1. A.AL SHEIKH, O. BRUN, P.E. HLADIK, B. PRABHU, ***A best-response algorithm for multiprocessor periodic scheduling***, 23$^{\text{rd}}$ Euromicro Conference on Real-Time Systems (ECRTS 2011), Porto (Portugal), 6-9 July 2011.

2. A.AL SHEIKH, O. BRUN, P.E. HLADIK, ***Partition scheduling on an IMA platform with strict periodicity and communication delays***, International Conference on Real-Time and Network Systems (RNTS 2010), Toulouse (France), 4-5 Novembre 2010, pp.179-188.

3. U. AYESTA, O. BRUN, B. J. PRABHU, ***Price of Anarchy in Non-Cooperative Load Balancing***, in Proceedings of IEEE Infocom Mini-Conference, San Diego (USA), Mars 2010.

4. M.Z. BEN HAMOUDA, O. BRUN et JM GARCIA, ***Integration of Equipment Constraints in the Network Topology Design Process***, 9th Int. Symposium on Communications and Information Technologies, Incheon (Corée du Sud), 2009.

5. C. FORTUNY, O. BRUN, M.Z. BEN HAMOUDA et JM GARCIA, ***Integration of Equipment Constraints in the Capacity-planning Process***, 21st International Teletraffic Congress (ITC21), Paris (France), 15-17 Septembre 2009.

6. M.Z. BEN HAMOUDA, O. BRUN et JM GARCIA, ***A Tabu Search Heuristic for Capacitated Network Design***, IEEE Symposium on Computers and Communications (ISCC'09), Sousse (Tunisie), 5 – 8 Juillet, 2009.

7. O. BRUN, A. AL SHEIKH, J.M. GARCIA, ***Flow-level Modelling of TCP Traffic Using GPS Queueing Networks***, 21st International Teletraffic Congress (ITC21), Paris (France), 15-17 Septembre 2009.

8. O. BRUN, A. RACHDI et JM GARCIA, ***Access network design with capacity-dependent costs***, ValueTools'2008, Athenes (Grèce), 20-24 Octobre 2008.

9. H. HASSAN, JM GARCIA, O. BRUN, ***A packet streams approach to model Internet traffic***, IADIS Multi Conference on Computer Science and Information Systems (MCCSIS 2007), Telecommunication Networks and Systems (TNS), Lisbonne (Portugal), 3-8 Juillet 2007.

10. C. FORTUNY, O. BRUN, JM GARCIA, ***Fanouts Inference from Link Counts***, 4th European Conference on Universal Multiservice Networks (ECUMN), Toulouse (France), 14-16 Février 2007

11. H. HASSAN, JM GARCIA, O. BRUN, ***Session-based quality of service in SIP Networks***, 2nd IEEE International Conference on Information & Communication Technologies : from Theory to Applications (ICTTA'06), Damas (Syrie), 24-28 Avril 2006.

12. H. HASSAN, JM GARCIA, O. BRUN, ***Internet traffic modeling***, 2nd IEEE International Conference on Information & Communication Technologies : from Theory to Applications (ICTTA'06), Damas (Syrie), 24-28 Avril 2006.

13. O. BRUN, C. BOCKSTA et JM GARCIA, ***A simple formula for end-to-end jitter estimation in packet-switching networks***, 5th IEEE International Conference on Networking (ICN'06), Mauritius, 23-26 Avril 2006.

14. H. HASSAN, JM GARCIA, O. BRUN, ***Session scheduling in SIP based Networks***, IEEE Consumer Communications and Networking Conference, Las Vegas (USA), 8-10 Janvier 2006

15. C. FORTUNY, O. BRUN, JM GARCIA, ***Optimization of routing metrics in IP networks***, 19th International Teletraffic Congress (ITC'19), Beijing (Chine), 29 Août- 2 Septembre 2005.

16. H. HASSAN, JM GARCIA, O. BRUN, ***Generic modelling of multimedia traffic sources***, 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-Nets'05), Ilkley (Grande-Bretagne), 18-20 Juillet 2005.

17. JM. GARCIA, A. RACHDI et O. BRUN, ***Optimal LSP Placement with QoS Constraints in DiffServ/MPLS Networks***, Proceeding of the 18th Teletraffic Congress (ITC-18), Berlin (Allemagne), Sept. 2003.

18. P. BACQUET, O. BRUN, J.M. GARCIA, T. MONTEIL, P. PASCAL et S. RICHARD, ***Telecommunication Network Modeling and Planning Tool on ASP clusters***, International Conference on Computational Science 2003 (ICCS'2003), Melbourne (Australie), 2 au 4 Juin 2003.

19. O. BRUN, JM . GARCIA, ***Resource Allocation in Communication Networks***, 5th International Conference on High-Speed Networks and Multimedia Communications (HSNMC'02), Jeju Island (Corée du Sud), 3 au 5 Juillet 2002.

20. O. BRUN et JM GARCIA, ***Real-Time Parallel Particle Filtering***, 14th International Symposium of Mathematical Theory of Networks and Systems (MTNS'2000), Perpignan (France), 19-23 Juin 2000.

21. JM GARCIA, D. GAUCHARD, T. MONTEIL et O. BRUN, ***Processor Mapping given by Processor and Network Dynamic Load Prediction***, IFIP/ACM Euro-Par'99 Parallel Processing, Springer, LNCS 1685, 1999.

22. JM GARCIA, T. MONTEIL , D. GAUCHARD et O. BRUN, ***Communication Kernel for High Speed Networks in the Parallel Environment LANDA-HSN***, 1st IEEE Computer Society International Conference on Cluster Computing (IWCC'99), Rajkumar Butta et al. Eds, IEEE Computer Society, 1999.

23. O. BRUN, JM GARCIA, T. MONTEIL et P. GUYAUX, ***Communication strategies in the LANDA Environment***, International Workshop on New Technologies, Interconnects and Communications in Distributed and Parallel Systems, Toulouse, France, 16-17 Juin 1997.

Autres publications

1. A.AL SHEIKH, O. BRUN, P.E. HLADIK, B. PRABHU, ***Strictly Periodic Scheduling on an IMA-based Avionic Platform***, AFG'11, Toulouse (France), 19-23 Septembre 2011.

2. A. AL SHEIKH, O. BRUN, P.E. HLADIK, ***Ordonnancement de tâches sous contrainte de périodicité stricte***, Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2011), St Etienne (France), 2-4 Mars 2011, Vol.I, pp.55-56.

3. U. AYESTA, O. BRUN, B. J. PRABHU, ***Prix de l'anarchie et équilibrage de charge non-coopératif***, 11e congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2010), Toulouse (France), 24-26 Février 2010.

4. M.Z. BEN HAMOUDA, O. BRUN et JM GARCIA, ***Une méthode tabou pour l'optimisation des VPN***, 11e congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2010), Toulouse (France), 24-26 Février 2010.

5. H. HASSAN ,O. BRUN ,JM GARCIA et D. GAUCHARD, ***Integration of Streaming and Elastic Traffic : A Fixed Point Approach***, SimuTools 2008, Marseille (France), 3-7 Mars 2008.

6. C. FORTUNY, O. BRUN et JM GARCIA, ***Optimisation des poids DNS dans les réseaux radio de 3eme génération***, 9eme rencontres francophones sur les aspects algorithmiques des télécommunications (AlgoTel'2007), Ile d'Oléron (France), 29 Mai-1 Juin 2007 .

7. C. FORTUNY, O. BRUN et JM GARCIA, ***Estimation des Matrices de Trafic***, 8eme rencontres francophones sur les aspects algorithmiques des télécommunications (AlgoTel 2006), Trégastel (France), 9-12 Mai 2006.

8. H. HASSAN, JM GARCIA, O. BRUN, D. GAUCHARD, ***Caractérisation de l'autosimilarité de trafics WEB et FTP par un outil de simulation***, 7eme rencontres francophones sur les aspects algorithmiques des télécommunications (ALGOTEL'2005), Presqu'ile de Giens (France), 11-15 Mai 2005.

9. C. BOCKSTAL, JM GARCIA, O. BRUN, ***Approximation du régime stationnaire d'un système WFQ***, 6eme rencontres francophones sur les aspects algorithmiques des télécommunications (ALGOTEL'2004), Batz- sur-Mer (France), 26-28 Mai 2004.

10. JM. GARCIA, HA-DUC LE, D. GAUCHARD et O. BRUN, ***Modélisation Analytique Du Protocole Tcp New Reno Et Etude Comparative Avec Le Simulateur Ns (Network Simulator)***, Congrès RIVF'03, Hanoi (Vietnam), 10 au 13 Février 2003.

11. O. BRUN et JM. GARCIA, ***Modélisation Dynamique du Trafic dans les Réseaux Multi-Services***, Congrès Modélisation des Systèmes Réactifs (MSR'2001), Toulouse (France), Juanole G. et Valette R. Eds, Hermes.

12. D. GAUCHARD, C. FOURNIE, T. MONTEIL, JM GARCIA, B. LECUSSAN et O. BRUN, ***Communication mechanisms for Myrinet and MPC in the parallel environment LANDA-HSN***, 1st Myrinet User Group Conference MUG'2000), Lyon (France), 28-29 Septembre 2000.

13. G. AUTHIE, JM. GARCIA et O. BRUN, ***Dimensionnement et Routage Optimal dans les Réseaux Haut-Débit SDH***, 2eme Ecole d'Hiver des Télécommunications de Sophia-Antipolis (ECOTEL'99), Antibes (France), Décembre 1999.

Bibliographie

[1] J. Abate, G. Choudhury, and W. Whitt. Exponential approximations for tail probabilities in queues, I : waiting times. *Operations Research*, 43(5), 1995.

[2] J. Abate and W. Whitt. Transient behavior of the M/M/1 queue : starting at the origin. *Queueing Systems*, 2, 1987.

[3] D. Abendroth, H. van den Berg, and M. Mandjes. A multiple time-scale model for TCP bandwidth sharing under user heterogeneity. In R. B. et al. Eds, editor, *Proc. of Networking 2005, LNCS 3462*, 2005.

[4] A. Al Sheikh, O. Brun, P. Hladik, and B. Prabhu. A Best-Response Algorithm for Multiprocessor Periodic Scheduling. In *Euromicro Conference on Real-Time Systems (ECRTS)*, Juillet 2011.

[5] A. Al Sheikh, O. Brun, and P.-E. Hladik. Partition scheduling on an IMA platform with strict periodicity and communication delays. In *International Conference on Real-Time and Network Systems (RTNS 2010), pp.179-188*, November 2010.

[6] H. Alshaer and E. Horlait. Expedited forwarding end-to-end delay jitter in the differentiated services networks. In *Proceedings of HSNMC 2004*, 2004.

[7] K. Altinkemer and A. Chaturvedi. Neural networks for topological design of local access tree networks. In *Proceeding of Telecommunication Systems, Modelling and Analysis Conference*, pages 256–263, 1993.

[8] E. Altman, K. Avrachenkov, and C. Barakat. TCP network calculus : The case of large delay-bandwidth product. In *Proc. of IEEE INFOCOM*, 2002.

[9] E. Altman, U. Ayesta, and B. J. Prabhu. Load balancing in processor sharing systems. *Telecommunication Systems*, 47(1-2) :35–48, 2011.

[10] E. Altman, T. Boulogne, R. El-Azouzi, T. Jimenez, and L. Wynter. A survey on networking games in telecommunications. *Computers & Operations Research*, 33(2), 2006.

[11] W. B. Ameur, E. Gourdin, and B. Liau. Dimensioning of Internet networks. In *Proceedings of the DRCS'2000*, Munich, 2000.

[12] W. B. Ameur and B. Liau. Calcul des métriques de routage pour Internet. *Annales des Télécommunications*, 56, 2001.

[13] F. R. Amoza. *GRASP heuristics for Wide Area Network design.* PhD thesis, IRISA, Rennes, France, 2005.

[14] M. Andrews and L. Zhang. The access network design problem. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998.

[15] J. Anselmi and B. Gaujal. Optimal routing in parallel, non-observable queues and the price of anarchy revisited. In *22nd International Teletraffic Congress (ITC)*, Amsterdam, 2010.

[16] G. R. Ash. *Dynamic Routing in Telecommunications Networks.* McGraw Hill, 1998.

[17] U. Ayesta, O. Brun, and B. J. Prabhu. Price of anarchy in non-cooperative load balancing. In *Proc. of IEEE Infocom mini-conference*, Mars 2010.

[18] U. Ayesta, O. Brun, and B. J. Prabhu. Price of anarchy in non-cooperative load balancing. *Performance Evaluation*, 68 :1312–1332, 2011.

[19] P. Bacquet, O. Brun, J. Garcia, T. Monteil, P. Pascal, and S. Richard. Telecommunication network modeling and planning tool on asp clusters. In *International Conference on Computational Science 2003 (ICCS'2003)*, Saint Petersbourg, Russie et Melbourne, Australie, Juin 2003.

[20] J. Beckers, I. Hendrawan, R. Kooij, and R. van der Mei. Generalized processor sharing performance models for Internet access lines. In *Proc. of 9th IFIP Conference*, 2001.

[21] M. Z. Ben-Hamouda. *Conception et optimisation robuste des réseaux de télécommunications.* PhD thesis, Université Paul Sabatier, Toulouse, France, 25 Juin 2010. Rapport LAAS No 10462.

[22] M. Z. Ben-Hamouda, O. Brun, and J. M. Garcia. Integration of equipment constraints in the network topology design process. In *Proc. of 9th Int. Symposium on Communications and Information Technologies (ISCIT09)*, Incheon, Korea, 2009.

[23] M. Z. Ben-Hamouda, O. Brun, and J. M. Garcia. A tabu search heuristic for capacitated network design. In *IEEE Symposium on Computers and Communications (ISCC'09)*, Sousse, Tunisia, July 5 - 8 2009.

[24] M. Z. Ben-Hamouda, O. Brun, and J. M. Garcia. Une méthode tabou pour l'optimisation des VPN. In *Proc. of 11e congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2010)*, Toulouse, France, 24-26 Février 2010.

[25] R. T. Berger and S. Raghavan. Long-distance access network design. *Management Science*, 50 :309–325, 2004.

[26] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, 1987.

[27] C. Bockstal, J. Garcia, and O. Brun. Approximation du régime stationnaire d'un système wfq. In *6eme rencontres francophones sur les aspects algorithmiques des télécommunications (ALGOTEL'2004)*, Batz- sur-Mer (France), Mai 2004.

[28] C. Boender and A. Rinnooy Kan. Bayesian stopping rules for multistart global optimization methods. *Mathematical Programming*, 37(1) :59–80, 1987.

[29] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence - From Natural to Artificial Systems*. Oxford University Press, 1999.

[30] T. Bonald, L. Massoulié, A. Proutière, and J. Virtamo. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems*, 53, 2006.

[31] T. Bonald and A. Proutière. Insensitivity in processor-sharing networks. *Performance Evaluation*, 49, 2002.

[32] T. Bonald and A. Proutière. Insensitive bandwidth sharing in data networks. *Queueing Systems : Theory and Applications*, 44(1), 2003.

[33] T. Bonald and A. Proutière. On performance bounds for balanced fairness. *Performance Evaluation*, 55, 2004.

[34] T. Bonald and A. Proutière. On performance bounds for the integration of elastic and adaptive streaming flows. In *Proc. of ACM SIGMETRICS/Performance*, 2004.

[35] P. E. Boyer, F. Guillemin, M. Servel, and J. P. Coudreuse. Spacing cells protects and enhances utilization of ATM network links. *IEEE Network*, 6, 1992.

[36] O. Brun. *Modélisation et Optimisation des la Gestion des Ressources dans les Architectures Distribuées - Parallélisation Massive de la Technique de Filtrage Particulaire*. PhD thesis, Université Paul Sabatier de Toulouse, 2000.

[37] O. Brun. Modélisation des réseaux hétérogènes : du terminal utilisateur aux plates-formes de service. Technical report, LAAS-CNRS, 2009. Projet ANR AVIPS.

[38] O. Brun, A. Al-Sheikh, and J. Garcia. Flow-level modelling of TCP traffic using GPS queueing networks. In *Proc. of 21st International Teletraffic Congress (ITC 21 2009)*, Paris (France), 2009.

[39] O. Brun, C. Bockstal, and J. M. Garcia. Analytical approximation of the jitter incurred by CBR traffics in IP networks. *Telecommunication System Journal*, 33(1-3), Novembre 2006.

[40] O. Brun and J. Garcia. Real-time parallel particle filtering. In *14th International Symposium of Mathematical Theory of Networks and Systems (MTNS'2000)*, Perpignan, Juin 2000.

[41] O. Brun, J. Garcia, T. Monteil, and P. Guyaux. Communication strategies in the landa environment. In *International Workshop on New Technologies, Interconnects and Communications in Distributed and Parallel Systems*, Toulouse, juin 1997.

[42] O. Brun and J. M. Garcia. Analytical solution of finite capacity M/D/1 queues. *Journal of Applied Probability*, 37(4), Décembre 2000.

[43] O. Brun and J. M. Garcia. Modélisation dynamique du trafic dans les réseaux multiservices. In G. Juanole and R. Valette, editors, *Proceedings of Congrès Modélisation des Systèmes Réactifs (MSR'2001)*, Toulouse, 2001. Hermes.

[44] O. Brun and J. M. Garcia. Resource allocation in communication networks. In *5th International Conference on High-Speed Networks and Multimedia Communications (HSNMC'02)*, Jeju Island, Corée, 3 au 5 Juillet 2002.

[45] O. Brun and B. J. Prabhu. Worst case analysis of non-cooperative load balancing. In *Workshop in Algorithmic Game Theory (AlgoGT)*, Bordeaux, France, Juillet 2010.

[46] O. Brun, A. Rachdi, and J. M. Garcia. Access network design with capacity-dependent costs. In *ValueTools'2008*, Athenes, Grèce, 20-24 Octobre 2008.

[47] O. Brun, V. Teulière, and J. Garcia. Parallel particle filtering. *Journal of Parallel and Distributed Computing*, 62(7) :1186–1202, Juillet 2002.

[48] C. Bruni, P. D'Andrea, U. Mocci, and C. Scoglio. Optimal capacity management of virtual paths in an ATM network. In *IEEE Globecom 94 Proceedings*, 1994.

[49] T. Bu and D. Towsley. Fixed point approximations for TCP behavior in an AQM network. In *Proc. of SIGMETRICS'01*, 2001.

[50] L. Buriol, M. Resende, and M. Thorup. Speeding up dynamic shortest path algorithms. Research Technical Report TD-5RJ8B, AT&T Labs, 2003.

[51] L. S. Buriol, M. G. C. Resende, C. C. Ribeiro, and M. Thorup. A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. *Networks*, 46 :36–56, August 2005.

[52] D. Burton. *On the inverse shortest path problem.* PhD thesis, University of Namur, 1993.

[53] J. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Comm. of the ACM*, 16(9), 1973.

[54] P. Chanas. *Réseaux ATM : conception et optimisation.* PhD thesis, CNET, 1998.

[55] P. Chanas, O. Goldschmidt, and M. Burlet. Routing virtual paths in ATM networks. *J. of Heuristics*, 1999.

[56] H. L. Chen, J. Marden, and A. Wierman. The effect of local scheduling in load balancing designs. In *Proceedings of IEEE Infocom*, 2009.

[57] S. P. Chung and K. W. Ross. Reduced load approximations for multirate loss networks. *IEEE Trans. on Comm.*, 4(8) :1222–1231, 1993.

[58] I. Cidon, A. Khamisy, and M. Sidi. Dispersed messages in discrete-time queues : delay, jitter and threshold crossing. In *Proceeding of IEEE INFOCOM 1994*, 1994.

[59] J. Cohen. The multiple phase service network with generalized processor sharing. *Acta Informatica*, 12, 1979.

[60] J. W. Cohen. *The Single Server Queue.* North Holland, 1969.

[61] R. Cominetti, J. R. Correa, and N. E. Stier-Moses. The impact of oligopolistic competition in networks. *Operations Research, Published online in Articles in Advance, DOI : 10.1287/opre.1080.0653*, June 2009.

[62] D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, editors. *New ideas in optimization*. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.

[63] S. Cosares and I. Saniee. An optimization problem related to balancing loads on sonet rings. *Telecommunication Systems*, 3 :165–181, 1994.

[64] R. Cruz. A calculus for network delay, part i : network elements in isolation. *IEEE Trans. on Inf. Theory*, 37, 1991.

[65] R. Cruz. A calculus for network delay, part ii : network analysis. *IEEE Trans. on Inf. Theory*, 37, 1991.

[66] F. Eisenbrand, N. Hahnle, M. Niemeier, M. Skutella, J. Verschae, and A. Wiese. Scheduling periodic tasks in a hard real-time environment. In L. . S. Abramsky et al (Eds.), editor, *ICALP 2010 Proceeding, Part I*, page 2010, 2010.

[67] F. Eisenbrand, K. Kesavan, R. Mattikalli, M. Niemeier, A. Nordsieck, M. Skutella, J. Verschae, and A. Wiese. Solving an avionics real-time scheduling problem by advanced IP-methods. *Algorithms–ESA 2010*, pages 11–22, 2010.

[68] S. El-Zoghdy, H. Kameda, and J. Li. Numerical studies on a paradox for non-cooperative static load balancing in distributed computer systems. *Computers and Operations Research*, 33 :345–355, 2006.

[69] M. Ericsson, M. G. C. Resende, and P. Pardalos. A genetic algorithm for the weight setting problem in OSPF routing. *Journal of Combinatorial Optimization*, 6 :299–333, 2002.

[70] D. Ferrari. Delay jitter control scheme for packet-switching internetworks. *Computer Communications*, 15(6) :367–373, 1992.

[71] S. Floyd and V. Jacobson. Random early detection (red) gateways for congestion avoidance. *IEEE/ACM Trans. Network.*, 1(4), 1993.

[72] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Trans. Network.*, 9(4), 2001.

[73] C. Fortuny. *Estimation du trafic, planification et optimisation des ressources pour l'ingéniérie des réseaux IP/MPLS*. PhD thesis, Université Paul Sabatier, Toulouse, France, 1er Juillet 2008. Rapport LAAS No 08335.

[74] C. Fortuny, O. Brun, and J. M. Garcia. Optimization of routing metrics in IP networks. In *19th International Teletraffic Congress (ITC'19)*, Beijing, China, 29 Août- 2 Septembre 2005.

[75] C. Fortuny, O. Brun, and J. M. Garcia. Estimation des matrices de trafic. In *8eme rencontres francophones sur les aspects algorithmiques des télécommunications (AlgoTel 2006)*, Trégastel, France, 9-12 Mai 2006.

[76] C. Fortuny, O. Brun, and J. M. Garcia. Fanouts inference from link counts. In *4th European Conference on Universal Multiservice Networks (ECUMN)*, Toulouse, France, 14-16 Février 2007.

[77] C. Fortuny, O. Brun, M. Z. B. Hamouda, and J. M. Garcia. Integration of equipment constraints in the capacity-planning process. In *Proc. of the 21th International Teletraffic Congress (ITC21)*, Paris, France, 15-17 Septembre 2009.

[78] B. Fortz. *Design of Survivable Networks with Bounded Rings.* PhD thesis, Université Libre de Bruxelles, 1999.

[79] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of INFOCOM (2)*, pages 519–528, 2000.

[80] L. F. Frantzeskakis and H. Luss. The network redesign problem for access telecommunications networks. *Naval Research Logistics*, 46 :487–506, 1999.

[81] S. B. Fredj, T. Bonald, A. Proutiere, G. Régnié, and J. Roberts. Statistical bandwidth sharing : A study of congestion at flow level. In *Proc. of SIGCOMM'01*, 2001.

[82] C. Frei and B. Faltings. Bandwidth allocation heuristics in communication networks. In *Proc. of ALGOTEL'99*, pages 53–58, Roscoff, France, 1999.

[83] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, 1991.

[84] H. N. Gabow, M. X. Goemans, and D. P. Williamson. An efficient approximation algorithm for the survivable network design problem. In *Proceedings of the 3rd MPS Conference on Integer Programming and Combinatorial Optimization*, pages 57–74, 1993.

[85] J. Garcia, D. Gauchard, T. Monteil, and O. Brun. Processor mapping given by processor and network dynamic load prediction. In Springer, editor, *IFIP/ACM Euro-Par'99 Parallel Processing*, volume LNCS 1685, 1999.

[86] J. Garcia, H.-D. Le, D. Gauchard, and O. Brun. Modélisation analytique du protocole tcp new reno et etude comparative avec le simulateur ns (network simulator). In *Congrès RIVF'03*, Hanoi (Vietnam), Février 2003.

[87] J. Garcia, T. Monteil, D. Gauchard, and O. Brun. Communication kernel for high speed networks in the parallel environment landa-hsn. In R. B. et al., editor, *1st IEEE Computer Society International Conference on Cluster Computing (IWCC'99)*. IEEE Computer Society, 1999.

[88] J. M. Garcia, O. Brun, and D. Gauchard. Modélisation différentielle hybride dans les réseaux multiservices. Atelier CNES Modélisation et Simulation des Systèmes Communicants (Mod'2000), Décembre 2000. Toulouse.

[89] J. M. Garcia, O. Brun, and D. Gauchard. Transient analytical solution of M/D/1/N queues. *Journal of Applied Probability*, 39(4) :853–864, Décembre 2002.

[90] J. M. Garcia, F. L. Gall, and J. Bernussou. A model for telephone networks and its use for routing optimization purposes. *IEEE Journal on selected areas in communication*, 4(6), Septembre 1986. Special issue on communication network performance evaluation.

[91] J. M. Garcia, D. Gauchard, O. Brun, P. Bacquet, J. Sexton, and E. Lawness. Modélisation différentielle et simulation hybride distribuée. *Calculateurs Parallèles, Réseaux et Systèmes Répartis*, 2002.

[92] J. M. Garcia, A. Rachdi, and O. Brun. Optimal LSP placement with QoS constraints in DiffServ/MPLS networks. In *Proceeding of the 18th Teletraffic Congress (ITC-18)*, Berlin, Allemagne, Septembre 2003.

[93] D. Gauchard. *Simulation hybride des réseaux IP-DiffServ-MPLS multi-services sur environnement d'exécution distribuée.* PhD thesis, Université Paul Sabatier, Avril 2003. Rapport LAAS No 03544.

[94] D. Gauchard, C. Fournie, T. Monteil, J. Garcia, B. Lecussan, and O. Brun. Communication mechanisms for myrinet and mpc in the parallel environment landa-hsn. In *1st Myrinet User Group Conference (MUG'2000)*, Lyon, septembre 2000.

[95] B. Gavish. Topological design of telecommunication networks - local access design methods. *Annals of Operations Research*, 33 :17–71, 1991.

[96] B. Gavish and K. Altinkemer. Parallel savings heuristics for the topological design of local access tree networks. In *Proceedings of IEEE INFOCOM'86*, pages 130–139, 1986. Fifth Annual Conference on Computers and Communications Integration Design, Analysis, Management.

[97] L. George, P. Muhlethaler, and N. Rivierre. Optimality and non-preemptive real-time scheduling revisited. Technical report, Rapport de Recherche RR-2516, INRIA, 2006.

[98] A. Girard, B. Sanso, and L. Dadjo. A tabu search algorithm for access network design. *Annals of Operations Research*, 106(1-4) :229–262, 2001.

[99] L. Gouveia and M. J. Lopes. Using generalized capacitated trees for designing the topology of local access networks. *Telecommunications Systems*, 7(4) :315–337, 1997.

[100] A. Gravey, J. R. Louvion, and P. Boyer. On the $geo/d/1$ and $geo/d/1/n$ queues. *Performance Evaluation*, 11(2), 1990.

[101] M. Grotschel, C. L. Monma, and M. Stoer. Design of survivable networks, 1993. Volume on Networks, Handbook in Operations Research and Management Science.

[102] F. Guillemin, P. Boyer, A. Dupuis, and L. Romoeuf. Peak rate enforcement in ATM networks. In *Proceedings of IEEE INFOCOM*, 1992.

[103] F. Guillemin and W. Monin. Limitation of cell delay variation in ATM networks. In *Proceedings of ICCT'92*, 1992.

[104] F. Guillemin and J. W. Roberts. Jitter and bandwidth enforcement. In *Proceedings of IEEE GLOBECOM*, pages 261–265, 1991.

[105] W. D. Harvey and M. L. Ginsberg. Limited discrepancy search. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, volume 1, 1995.

[106] H. Hassan, O. Brun, J. Garcia, and D. Gauchard. Integration of streaming and elastic traffic : A fixed point approach. In *SimuTools 2008*, Marseille, Mars 2008.

[107] H. Hassan, J. Garcia, and O. Brun. Generic modelling of multimedia traffic sources. In *3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-Nets'05)*, Ilkley (Great-Britain), Juillet 2005.

[108] H. Hassan, J. Garcia, and O. Brun. Bandwidth allocation and session scheduling using sip. *Journal of Communications*, 1(5), August 2006.

[109] H. Hassan, J. Garcia, and O. Brun. Internet traffic modeling. In *2nd IEEE International Conference on Information & Communication Technologies : from Theory to Applications (ICTTA'06)*, Damas (Syrie), Avril 2006.

[110] H. Hassan, J. Garcia, and O. Brun. Session-based quality of service in sip networks. In *2nd IEEE International Conference on Information & Communication Technologies : from Theory to Applications (ICTTA'06)*, Damas (Syrie), Avril 2006.

[111] H. Hassan, J. Garcia, and O. Brun. Session scheduling in sip based networks. In *IEEE Consumer Communications and Networking Conference*, Las Vegas (USA), Janvier 2006.

[112] H. Hassan, J. Garcia, and O. Brun. A packet streams approach to model internet traffic. In *IADIS Multi Conference on Computer Science and Information Systems (MCCSIS 2007), Telecommunication Networks and Systems (TNS)*, Lisbonne (Portugal), Juillet 2007.

[113] H. Hassan, J. Garcia, O. Brun, and D. Gauchard. Caractérisation de l'autosimilarité de trafics web et ftp par un outil de simulation. In *7eme rencontres francophones sur les aspects algorithmiques des télécommunications (ALGOTEL'2005)*, Presqu'ile de Giens (France), Mai 2005.

[114] D. Heyman, T. Lakshman, and A. Neidhardt. A new method for analysing feedback-based protocols with applications to engineering web traffic over the Internet. In *Proc. of SIGMETRICS'97*, 1997.

[115] ILOG CPLEX. `http://www.ilog.com/products/cplex/`.

[116] J. S. Kaufman. Blocking in a shared resource environment. *IEEE Trans. on Comm.*, 20(10), 1981.

[117] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks : Shadow prices, proportional fairness and stability. *Journal of the Operat. Res. Society*, 49, 1998.

[118] W. D. Kelton. Transient exponential-erlang queues and steady-state simulation. *Communications of the ACM*, 28(7), 1985.

[119] W. D. Kelton and A. M. Law. The transient behavior of the m/m/s queue, with implications for the steady-state simulations. *Operations Research*, 33(2), 1985.

[120] O. Kermia and Y. Sorel. A rapid heuristic for scheduling non-preemptive dependent periodic tasks onto multiprocessor. In *Proceedings of ISCA 20th international conference on Parallel and Distributed Computing Systems, PDCS*, volume 7. Citeseer, 2007.

[121] A. Kherany and A. Kumar. Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet. In *Proc. of IEEE INFOCOM*, 2002.

[122] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598) :671–680, 1983.

[123] L. Kleinrock. *Queueing Systems. Volume I : Theory.* Wiley Interscience, 1975.

[124] L. Kleinrock. *Queueing Systems. Volume II : Applications.* Wiley Interscience, 1976.

[125] J. G. Klincewicz. Hub location in backbone/tributary network : a review. *Location Science*, 6 :307–335, 1998.

[126] H. Kobayashi. Application of the diffusion approximation to queueing networks ii : Nonequilibrium distributions and applications to computer modeling. *Jour. of the Ass. for Comp. Machinery*, 21(3), 1974.

[127] Y. Korilis, A. Lazar, and A. Orda. Capacity allocation under noncooperative routing. *IEEE Transactions on Automatic Control*, 42(3) :309–325, March 1997.

[128] Y. Korilis and A. Orda. Incentive compatible pricing strategies for qos routing. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 891 –899 vol.2, mar 1999.

[129] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5 :161–173, 1997.

[130] J. Korst. *Periodic multiprocessor scheduling.* PhD thesis, Eindhoven university of technology, Eindhoven, the Netherlands, 1992.

[131] J. Korst, E. Aarts, and J. K. Lenstra. Scheduling periodic tasks. *INFORMS JOURNAL ON COMPUTING*, 8 :428–435, 1996.

[132] J. Korst, E. Aarts, and J. K. Lenstra. Scheduling periodic tasks with slack. *INFORMS JOURNAL ON COMPUTING*, 9 :351–362, 1997.

[133] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS 1999*, 1999.

[134] S. Kunniyur and R. Srikant. End-to-end congestion control schemes : Utility functions, random losses, ecn marks. In *Proc. of IEEE INFOCOM*, 2000.

[135] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proceedings ACM SIGMETRICS'92*, 1992.

[136] J. Labetoulle. Some theorems on real time scheduling. *Computer Architecture and Networks*, pages 285–293, 1974.

[137] S. Lam and G. Xie. Burst scheduling : architecture and algorithm for switching packet video. In *Proceedings of IEEE INFOCOM*, 1995.

[138] R. Landry and I. Stavrakakis. Traffic shaping of a tagged stream in an ATM network : approximate end-to-end analysis. In *Proceedings of IEEE INFOCOM*, 1995.

[139] R. Landry and I. Stavrakakis. Study of delay jitter with and without peak rate enforcement. *IEEE/ACM Transactions on Networking*, 5(4), 1997.

[140] P. Lassila and M. Mandjes. A multi-level TCP model with heterogeneous RTTs. In R. M. et al. Eds, editor, *Proc. of Networking. 2004*. LNCS 3042, 2004.

[141] P. Lassila, H. van den Berg, M. Mandjes, and R. Kooij. An integrated packet/flow model for TCP performance analysis. In *Proc. of ITC 18*. Elsevier, Amsterdam, 2003.

[142] P. Lassila and J. Virtamo. Access network dimensioning for elastic traffic based on flow-level qos. Technical report, Networking Laboratory, Helsinki University of Technology, 2005.

[143] D. S. Lee and S. Q. Li. Transient analysis of a switched poisson arrival queue under overload control. *Performance Evaluation*, 17(1), 1993.

[144] I. J. Lee and E. Roth. A heuristic for the transient expected queue length of markovian queueing systems. *Operations Research Letters*, 14(1), 1993.

[145] J.-T. Leung and M. Merril. A note on preemptive scheduling of periodic, real-time tasks. *Information Processing Letters*, 11 :115–118, 1980.

[146] S. Q. Li and C. Fulton. Delay jitter first-order statistics of general traffic on high-speed networks. In *Proceedings of IEEE ICC*, 1996.

[147] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1) :46–61, 1973.

[148] Z. Liu, P. Nain, and D. Towsley. Exponential bounds with applications to call admission. *Journal of the ACM*, 44(3), 1997.

[149] S. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. on Networking*, 11(4), 2003.

[150] S. Low and D. Lapsley. Optimization flow control i : Basic algorithm and convergence. *IEEE/ACM Trans. on Networking*, 7(6), 1999.

[151] B. Lukic. An approach of designing local access network using simulated annealing method. In *Proc. of ConTEL'99 - 5th International Conference on Telecommunications*, pages 241–247, 1999.

[152] M. Marouf and Y. Sorel. Schedulability conditions for non-preemptive hard real-time tasks with strict period. In *Proc. of International Conference on Real-Time and Network Systems (RTNS 2010)*, pages pp 50–58, November 2010.

[153] R. Martí. Multi-start methods. *Handbook of metaheuristics*, pages 355–368, 2003.

[154] L. Massoulié and J. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15, 2000.

[155] L. Massoulié and J. Roberts. Bandwidth sharing : Objectives and algorithms. *IEEE/ACM Trans. Netw.*, 10(3), 2002.

[156] G. R. Mateus and R. V. L. Franqueira. Model and heuristics for a generalized access network design problem. *Telecommunication Systems*, 15(3-4) :257–271, 2000.

[157] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27, 1997.

[158] W. Matragi, C. Bisdikian, and K. Sohraby. Jitter calculus in ATM networks : single node case. In *Proceedings IEEE Infocom*, 1994.

[159] W. Matragi, K. Sohraby, and C. Bisdikian. A framework for jitter analysis in cell based multiplexors. *Performance Evaluation*, 22, 1996.

[160] W. Matragi, K. Sohraby, and C. Bisdikian. Jitter calculus in ATM networks : multiple node case. *IEEE/ACM Transactions on Networking*, 5, 1997.

[161] P. Meumeu and Y. Sorel. Non-schedulability conditions for off-line scheduling of real-time systems subject to precedence and strict periodicity constraints. In *Proceedings of 11th IEEE International Conference on Emerging technologies and Factory Automation, ETFA06, WIP*, 2006.

[162] M. Minoux. Planification à court et à moyen terme d'un réseau de télécommunications. *Annales des Télécommunications*, 29(11-12), 1974.

[163] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.*, 8(5), 2000.

[164] D. Monderer and L. S. Shapley. Potential games. *Games and Econ. Behavior*, 14 :124–143, 1996.

[165] D. Mosk-Aoyama, T. Roughgarden, and D. Shah. Fully distributed algorithms for convex optimization problems. *Lecture Notes in Computer Science*, 4731, 2007.

[166] J. R. Murray and W. D. Kelton. The transient behavior of the m/ek/2 queue and steady-state simulation. *Comput. Operat. Res.*, 15, 1988.

[167] D. Nace and M. Pioro. A tutorial on max-min fairness and its applications to routing and load-balancing in telecommunication networks. *IEEE Jour. Comm. Surveys and Tutorials*, 10(4), 2008.

[168] R. Nunez-Queija, J. van-den Berg, and M. Mandjes. Performance evaluation of strategies for integration of elastic and stream traffic. In *Proc. of ITC 16*, 1999.

[169] A. R. Odoni and E. Roth. An empirical investigation of the transient behavior of stationary queueing systems. *Operation Research*, 31, 1983.

[170] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multi-user communication networks. *IEEE/ACM Transactions on Networking*, 1 :510–521, October 1993.

[171] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput : a simple model and its empirical validation. In *Proc. of ACM SIGCOMM'98*, 1998.

[172] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks : the multiple node case. *IEEE/ACM Trans. on Networking*, 1, 1993.

[173] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks : the single node case. *IEEE/ACM Trans. on Networking*, 1, 1993.

[174] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Series in Networking. Elsevier/Morgan Kaufmann, 2004.

[175] A. Privalov and K. Sohraby. Per-stream jitter analysis in CBR ATM multiplexors. *IEEE/ACM Transactions on Networking*, 6(2), 1998.

[176] A. Rachdi. *Optimisation des ressources de réseaux hétérogènes avec coeur de réseau MPLS.* PhD thesis, Institut National des Sciences Appliquées, Toulouse, 3 Mai 2007. Rapport LAAS No 07198.

[177] G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the shortest path problem. *Journal of Algorithms*, 21 :267–305, 1996.

[178] C. D. Randazzo and H. P. L. Luna. A comparison of optimal methods for local access uncapacited network design. *Annals of Operations Research*, 106 :263–286, 2001.

[179] C. D. Randazzo, H. P. L. Luna, and P. Mahey. Benders decomposition for local access network design with two technologies. *Discrete Mathematics and Theoretical Computer Science*, 4 :235–246, 2001.

[180] J. Roberts. *A Service System With Heterogeneous User Requirements - Application to Multi-Service Telecommunication Systems.* in Performance of Data Communication Systems and their Applications. North Holland-Elsevier Science Publisher, 1981.

[181] J. Roberts. A survey on statistical bandwidth sharing. *Computer Networks*, 45, 2004.

[182] J. Roberts, U. Mocci, and J. Virtamo, editors. *Broadband Network Teletraffic.* Springer Lecture notes in computer science, 1996. Final Report of Action COST 242.

[183] J. Roberts, U. Mocci, and J. Virtamo, editors. *Broadband Network Teletraffic, Final Report of Action Cost 242*. Springer Berlin, 1996.

[184] J. W. Roberts. Jitter due to an ATM multiplex - application to peak rate policing. Technical report, FT R & D, 1989. Technical Report COST 224-35, RACE Cost Project.

[185] J. W. Roberts and F. Guillemin. Jitter in ATM networks and its impact on peak rate enforcement. *Performance Evaluation*, 16 :35–48, 1992.

[186] T. Roughgarden. How bad is selfish routing ? *Journal of the ACM*, 49(2), 2002.

[187] T. L. Saaty. Time-dependent solution of the many-server poisson queue. *Operations Research*, 8, 1960.

[188] C. Sauer. Computational algorithms for state-dependent queueing networks. *ACM Trans. Computer Systems*, 1(1), 1983.

[189] R. Schoonderwoerd, O. Holland, J. Brutten, and L. RothKrantz. Ant based load balancing in telecommunication networks. *Adaptive Behavior*, 5(2), 1997.

[190] L. P. Seelen, H. C. Tijms, and M. H. V. Hoorn. *Tables for Multi-Server Queues*. North-Holland, Amsterdam, 1985.

[191] A. A. Sheikh. *Resource allocation in hard real-time avionic systems. Scheduling and routing problems*. PhD thesis, Institut National des Sciences Appliquées, Toulouse, Septembre 2011.

[192] A. Sridharan, R. Guérin, and C. Diot. Achieving near optimal traffic engineering solutions in current OSPF/ISIS networks. In *Proceedings of INFOCOM 2003*, San Fransisco, USA, 2003.

[193] M. Stoer. Design of survivable networks. *Lecture Notes in Mathematics*, 1531, 1992. Springer Verlag.

[194] V. Teulière and O. Brun. Parallelisation of the particle filtering algorithm and application to doppler-bearing tracking of maneuvering sources. *Parallel Computing*, 29(8) :1069–1090, August 2003.

[195] H. C. Tijms. *Stochastic Modelling and Analysis : a computational approach*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1986.

[196] N. Vicari and P. Tran-Gia. A numerical analysis of the $Geo/d/n$ queueing system. Technical Report RR151, Institute of Computer Science, University of Wurzburg, 1996.

[197] M. Vojnovic, J.-Y. L. Boudec, and C. Boutremans. Global fairness of additive-increase and multiplicative-decrease with heterogeneous roun-trip times. In *Proc. of IEEE INFOCOM*, 2000.

[198] C. L. Wang. On the transient delays of m/g/1 queues. *J. Appl. Prob.*, 36(3), 1999.

[199] O. Yaron and M. Sidi. Calculating performance bounds in communication networks. In *Proceedings IEEE INFOCOM'93*, 1993.