Vers un Internet programmable offrant des garanties de qualité de service

Géraldine Texier

ANNÉE 2019

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

UNIVERSITÉ DE RENNES 1

HABILITATION A DIRIGER DES RECHERCHES

Université de Rennes 1

Domaine : Informatique

Ecole doctorale MathSTICC

présentée par

Géraldine Texier

préparée à IMT Atlantique

Vers un Internet programmable offrant des garanties de qualité de service

soutenue à Rennes
le 3 décembre 2019

devant le jury composé de :

Olivier BONAVENTURE
Professeur, UCL/ rapporteur

Isabelle GUÉRIN LASSOUS
Professeure. Université Claude Bernard Lyon 1/ rapportrice

Thierry TURLETTI
Directeur de Recherche, INRIA/ rapporteur

Dominique BARTH
Professeur, Université de Versailles Saint-Quentin-en-Yvelines / examinateur

André-Luc BEYLOT
Professeur, IRIT/ENSEEIHT / examinateur

César VIHO
Professeur, Université de Rennes 1 / examinateur

Table des matières

CHAPITRE 1

Introduction

Le trafic global véhiculé sur l'Internet connaît une croissance inouïe. Alors que le trafic annuel représentait déjà 1,5 zettaoctets [1] en 2017, le constructeur CISCO prévoit qu'il aura au triplé en 2022 pour atteindre 4,8 zettaoctets [44]. Une telle évolution s'explique à la fois par la démocratisation des accès au réseau, par l'émergence de l'Internet des Objets (Internet of Things (IoT)), et par de profonds changement des usages de l'Internet. L'étude met ainsi en évidence deux autres tendances. La première porte sur l'augmentation impressionnante de l'usage de mobiles et des équipements sans fils pour accéder aux contenus (d'après Cisco, le trafic mobile représentera 71% du trafic IP global en 2022) alors que le trafic fixe qui constituait la moitié du trafic de l'Internet en 2017 représentera moins d'un tiers du trafic IP total en 2022. La seconde tendance porte sur le changement important dans les usages de l'Internet et en particulier la progression importante de la part du trafic vidéo qui représentera 80% du trafic Internet en 2022 (dont 22% du trafic Internet seront des vidéos Ultra HD et 57% des vidéos HD). Or, la diffusion de la télévision en numérique et l'évolution vers l'ultraHD impliquent maintenant la transmission de volumes de données encore plus importants.

Au fil du développement de l'Internet, les infrastructures réseaux ont évolué vers de plus hauts débits et de meilleures couvertures. La fibre, qui avait permis d'augmenter la capacité des réseaux cœur, a été déployée dans les réseaux d'accès. Ces mesures ont aidé à faire face à l'augmentation des volumes de trafic qui suivait jusqu'à il y a peu une courbe de croissance exponentielle mais elles ne suffiront pas pour garder un fonctionnement acceptable de l'Internet. Il est nécessaire d'adopter une meilleure gestion des ressources réseau (bande passante, capacité de calcul et de stockage des équipements dans les cœurs de réseaux, etc.). À l'heure actuelle, l'Internet fonctionne sur un mode appelé *Best Effort* [37] : aucune garantie de service ou de performance n'est offerte. Par conséquent les opérateurs surdimensionnent leurs réseaux, afin d'offrir un service correct à leurs usagers, n'utilisant qu'une partie de leur bande passante. Or, soutenir les prévisions de trafic annoncées en gardant ce fonctionnement simple du réseau implique des investissements massifs et une augmentation considérable de la capacité des réseaux. L'alternative pour réduire le surdimensionnement du réseau est de lui permettre de mieux gérer son trafic en introduisant des mécanismes de qualité de service (Quality of Service (QoS)), soit par réservation de ressource, soit par des techniques d'ingénierie de trafic. L'importance du tra-

1. 1 zettaoctet = 10^{21} octets

fic vidéo (surtout les vidéo live) et des trafics interactifs (comme la voix sur IP et les jeux) rend l'introduction de QoS dans les réseaux de plus en plus inéluctable. La mise en œuvre des paradigmes de QoS n'est pas simple, ce qui a considérablement freiné leur utilisation et les ont confinés principalement à un usage interne dans les systèmes autonomes.

Parallèlement, les dix dernières années ont vu l'émergence de nouvelles architectures visant à automatiser la gestion des réseaux tout en la rendant dynamique. Ces solutions reposent sur une informatisation du réseau qui devient programmable grâce à l'architecture Software Defined Network (SDN) et la virtualisation des fonctions réseaux. Ces propositions sont toujours en cours de définition et seule leur utilisation pour la création de réseaux privés virtuels (Virtual Private Network (VPN)) d'entreprise est timidement proposée sous le terme de Software-Defined networking in a Wide Area Network (SD-WAN). Les autres usages ne sont pas encore opérationnels dans les réseaux car ils impliquent des changements plus profonds au niveau de leur architecture.

Ainsi, l'Internet est sur le point de subir une transformation majeure afin de soutenir la forte augmentation de trafic et les changements d'usages tant dans les réseaux mobiles, IoT que dans les réseaux fixes. Cela implique une meilleure utilisation des ressources et de rendre les réseaux programmables pour pouvoir en automatiser la gestion et les adapter dynamiquement aux besoins des clients/applications. Ceci est vrai à la fois pour les réseaux fixes et mobiles et tends à transformer le paradigme Best Effort de l'Internet en un Internet programmable et des réseaux dont le comportement doit etre hautement prévisible, voire déterministe, pour des flux importants et exigeants en terme de QoS.

1.1 Les thématiques abordées

Après une thèse dans le domaine des systèmes distribués [39][42][43] en 2000, j'ai rejoint l'IMT Atlantique (anciennement Télécom Bretagne et ENST Bretagne) en 2001 pour travailler sur des solutions de routage contraint et d'optimisation de l'utilisation des ressources afin de garantir la qualité de service dans les réseaux.

Je me suis intéressée dans un premier temps à des mécanismes basés sur la réservation de ressources [33][8][9][6] et à la mesure de la qualité de service dans les réseaux [41][15][16] [12][2]. Je me suis ensuite orientée vers les problèmes de routage contraint et les techniques d'ingénierie de trafic grâce auxquelles un administrateur est en mesure d'optimiser l'acheminement du trafic et l'utilisation des ressources dans son réseau [27][28][29][31][30]. J'ai tout d'abord abordé ces problèmes pour les réseaux d'opérateurs et à l'échelle de l'Internet [4][5]. En particulier, la recherche de chemins contraints dans l'Internet suppose une coopération des différents opérateurs impliqués. Ce problème est NP-Complet du fait de la structure même d'Internet qui est composé de plus de 65000 systèmes autonomes [2]. Outre le facteur d'échelle, chaque système autonome est souverains dans ses choix de gestion et peut être vus comme une boîte noire acheminant du trafic entre ses points d'entrées et ses points de sortie. Cette thématique a fait l'objet de plusieurs thèses (S. Lahoud, G. Bertrand et R. Jacquet) et un post-doctorat (M. Saidi) [38].

Mes travaux sont à la fois d'ordre architectural et protocolaire. Ils s'appuient sur l'utilisation de méthodes d'optimisation adaptées au routage dans les réseaux. La formalisation des problématiques de routage sous forme de programmes linéaires permet de proposer des solutions optimales lorsque la complexité du problème le permet, des approximations ou des heuristiques. L'étude des protocoles et des architectures de réseaux permet de concevoir des solutions réalistes et pouvant être mises en œuvre dans les réseaux par exemple avec

2. Chiffre obtenu en septembre 2019 sur le site `https://www.cidr-report.org/as2.0/`

l'ingénierie de trafic. Des travaux de standardisation sont actuellement menés à l'Internet Engineering Task Force (IETF) pour proposer un nouveau paradigme de routage, par exemple le routage par segment qui a pour objectif de simplifier les mécanismes dans les cœurs de réseaux. Prometteur, le routage par segment simplifie la mise en œuvre de l'ingénierie de trafic dans les réseaux. En appliquant les techniques d'optimisation, nous avons pu proposer des solutions réalistes pour construire un routage à partir de segments facilitant la mise en place du routage contraint (thèse de R. Guedrez) [19][20][17].

La pertinence des solutions proposées est fortement liée aux types de contraintes devant être respectées par le routage. Si dans un premier temps j'ai porté mon attention sur les flux en transit chez les opérateurs, j'ai ensuite étendu les types de contraintes considérées à différents contextes. Grâce à plusieurs projets de recherche Agence Nationale de la Recherche (ANR) ou régionaux, j'ai pu appliquer des techniques d'optimisation pour répondre à des besoins spécifiques (tels que ceux des réseaux sur puce (Network on Chip) ou d'applications multimédia et de diffusion de la vidéo). L'acheminement de flux multimédia nécessite en général une bande passante minimum et de la régularité dans l'acheminement des flux pour éviter un rendu haché ou temporairement suspendu chez les utilisateurs [3][32][13]. L'acheminement de flux multimédia a donné lieu à plusieurs travaux car la nature des contraintes prises en compte est inhérente au type d'applications multimédia considérées. Nous avons poussé ces contraintes à l'extrême en nous intéressant à la diffusion de flux vidéo en direct (live) [26][14][21].

Depuis 2014, je travaille également sur l'optimisation des ressources dans les réseaux de capteurs, des réseaux soumis à des contraintes de ressources extrêmement fortes. J'ai participé à l'étude de mécanismes pour mettre en œuvre des communications déterministes dans des réseaux de capteurs industriels (thèse de P. Thubert) [36][11][25]. Le déterminisme a pour but de respecter des contraintes de délai d'acheminement et de limitation des pertes lors de l'acheminement du trafic sur des équipements ayant des ressources limitées. Récemment, j'ai utilisé des techniques d'ingénierie de trafic, d'optimisation et de routage contraint pour réduire la consommation énergétique d'infrastructures de réseaux de capteurs dans les villes intelligentes. J'ai proposé une solution de déchargement des données sur des smartphones effectuant des mesures par crowdsensing [10]. Le but étant d'augmenter la durée de vie de ces infrastructures qui sont généralement alimentées par batterie.

La mise en œuvre de solutions pour garantir une qualité de service aux flux applicatifs requiert une coordination globale des comportement des équipements. Ce qui devait être fait manuellement auparavant peut maintenant être automatisé grâce à l'émergence de l'architecture SDN rendant les réseaux programmables. De plus, les récents efforts de virtualisation des fonctions réseaux (Network Functions Virtualisation (NFV)) offrent de nouveaux mécanismes pour pousser l'utilisation de l'ingénierie de trafic jusqu'à pouvoir proposer une adaptation dynamique d'infrastructures de réseaux virtualisées. J'explore ces nouveaux mécanismes depuis 2015 dans le cadre de la thèse de Cédric Morin et d'une mise à disposition dans l'IRT $B <> COM$.

1.2 Plan du manuscrit

Dans ce manuscrit, je présente une partie de mes travaux avec pour fil conducteur l'utilisation de techniques d'ingénierie de trafic et l'optimisation du routage dans les réseaux pour permettre une meilleure gestion des ressources et faire face aux imminentes et profondes évolutions de l'Internet. Dans le chapitre 2, nous aborderons les solutions permettant de garantir un niveau de service dans les réseaux, nous verrons que des solutions existent à tous les niveaux des couches protocolaires. Si les applications peuvent contri-

buer à améliorer la QoS, elles ne peuvent qu'atténuer les artefacts survenant lors des communications. Elles doivent donc s'appuyer sur des architectures ou des protocoles permettant de modifier le comportement Best Effort de l'Internet. Dans le chapitre 3, nous nous intéresserons à la garantie de QoS de bout en bout dans l'Internet et en particulier aux respects de contraintes de QoS multiples en inter-domaine. Nous verrons que ce problème complexe nécessite une coopération entre les systèmes autonomes qui peut être difficile à mettre en œuvre dans un environnement concurrentiel. Les solutions proposées n'ont pas été adoptées par les opérateurs réseaux pour plusieurs raisons, la principale étant sans doute la complexité de la coordination à mettre en place et le manque de maturité du marché. Cependant, l'évolution des réseaux vers une automatisation de gestion grâce à SDN et les nouveaux paradigmes d'Iaas (Infrastructure as a Service), de Naas (Network as a Service) et de Paas (Platform as a Service) introduits par la 5G ont permis de développer la virtualisation des réseaux (NFV) et leur adaptation aux besoins d'applications ou de secteurs d'activités particuliers. Nous verrons dans le chapitre 4 que les récentes propositions faites dans les organismes de standardisation amènent de nouvelles solutions pour garantir de la QoS au sein des systèmes autonomes. La question de l'adaptation de ces mécanismes au contexte inter-domaine est de nouveau pertinente afin d'offrir des garantie de QoS à large échelle. La virtualisation des réseaux offre de plus la perspective prometteuse de pouvoir mutualiser une infrastructure physique entre plusieurs réseaux virtualisés tout en adapter leur structure aux besoins spécifiques des clients. Je présenterai dans le chapitre 5 mon projet de recherche pour les années à venir qui s'appuie sur ces nouvelles architectures pour répondre aux défis lancés par l'augmentation du trafic et des exigences des usagers en terme de service et de réactivité, que ce soit dans les réseaux cœurs ou dans le contexte des villes intelligentes.

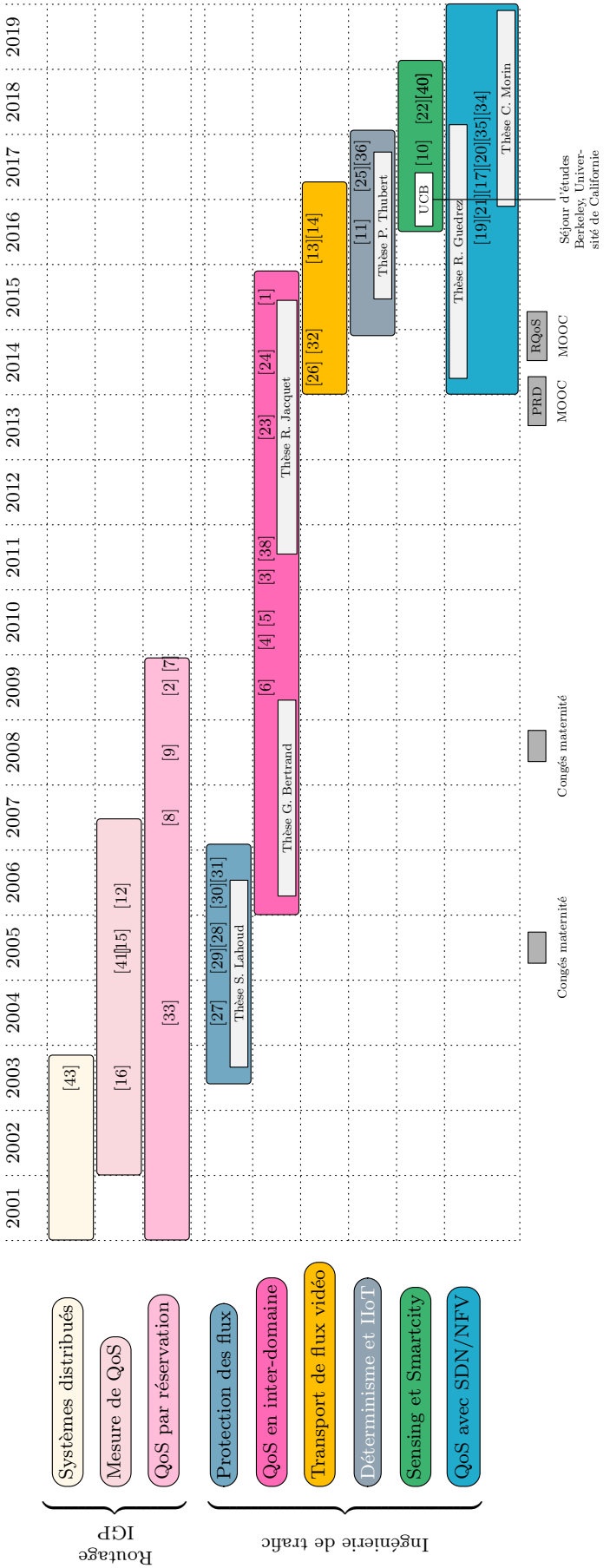


FIGURE 1.1 – Résumé d'activité

CHAPITRE 2

Au-delà de l'Internet Best Effort : la QoS

Pour garder un fonctionnement acceptable de l'Internet, il est nécessaire d'adopter une meilleure gestion des ressources réseau (bande passante, capacité de calcul et de stockage des équipements dans les cœurs de réseaux, etc.). En particulier, l'émergence de nouveaux usages de consommation de média sur les mobiles, l'expansion de l'Internet des Objets et l'évolution des contenus vers la haute définition (HD) (voire l'ultraHD ou 4K) impliquent la transmission de volumes de données encore plus importants. Soutenir les prévisions de trafic suppose des investissements massifs et une augmentation considérable de la capacité des réseaux. Ceci peut être atténué en rationalisant l'utilisation du réseau grâce à des mécanismes de garantie de qualité de service ou d'ingénierie de trafic qui visent à attribuer des ressources aux flux de façon différentiée.

A l'heure actuelle, l'Internet fonctionne sur un mode appelé *Best Effort* [37] : aucune garantie de service ou de performance n'est offerte. Par conséquent les opérateurs surdimensionnent leurs réseaux, afin d'offrir un service correct à leurs usagers. Le passage de cet Internet sans garantie vers un Internet efficace et respectant les contraintes de qualité de service des applications est un problème difficile qui mobilise de nombreux acteurs depuis des décennies. Des solutions sont envisageables au niveau du réseau lui-même, qu'elles soient protocolaires ou architecturale, mais également au niveau des applications. En particulier, les applications de diffusion vidéo ont des besoins fort de régularité d'acheminement des flux et une résistance aux pertes. Elles utilisent des tampons pour stocker un nombre minimum de trames avant de pouvoir jouer un flux afin de réduire l'impact de la gigue du réseau. Cependant ces solutions ne suffisent pas et ne sont qu'un des éléments déployés pour assurer un service de diffusion correct.

Ce chapitre aborde un éventail de solutions possibles pour gérer efficacement l'acheminement de données à travers le réseau. En prenant des cas d'usage extrêmement contraignants, il montre que des actions peuvent être menées à différents niveaux : au niveau applicatif, au niveau transport et au niveau réseau. Le premier cas d'usage, particulièrement exigeant, est la diffusion de vidéos live (en direct) qui doit respecter des bornes de délais ou de gigue. Nous montrons ici que des solutions applicatives et au niveau du transport peuvent améliorer la gestion de tels flux. Le second cas d'usage s'intéresse à l'acheminement des données de l'IoT dans les villes intelligentes et les usines, des réseaux disposant de ressources faibles et finies.

2.1 La QoE et la QoS pour les applications multimédia

La problématique d'offrir des garanties de service à des applications nécessitant des transfert de données sur le réseau génère souvent des discussions animées au sujet de la distinction entre qualité de service et qualité d'expérience. Dans leur article [64], Fiedler *et al.* définissent la qualité de l'expérience comme le lien entre d'une part la perception, l'expérience et les attentes des utilisateurs, et d'autre part les performances des applications et du réseau, généralement exprimée par des paramètres de qualité de service. Les travaux effectués lors de collaborations avec Gwendal Simon, Xavier Corbillon, Ramon Aparicio-Pardo, Nicolas Kuhn, Jiayi Liu, Catherine Rosenberg et Patrice Houzé sur le transfert de flux multimédia (surtout les flux en direct) m'ont permis d'aborder à la fois la qualité de service et la qualité d'expérience. En particulier, l'acheminement des flux multimédia est exigeant du point de vue de plusieurs contraintes : la bande passante, le délai d'acheminement, la gigue (c'est à dire la variation entre les délais d'acheminement de paquets succcessifs). Les utilisateurs des services vidéo en ligne sont sensibles à la qualité globale de la vidéo à l'écran, mais aussi, et surtout, à d'autres facteurs, notamment la latence entre la génération vidéo et la lecture des vidéos en direct. Bien que l'amélioration de la qualité de la vidéo perçue ait été bien étudiée par la communauté multimédia, l'impact du retard, de la latence et de la remise en mémoire tampon n'a pas reçu une attention aussi significative. Les technologies de streaming adaptatif, qui ont été largement adoptées ces dernières années, contribuent à ce manque de considération puisque les fournisseurs recommandent généralement l'introduction de délais supplémentaires importants.

2.1.1 Les CDNs : une solution architecturale

L'introduction des réseaux à diffusion de contenus Content Delivery Network (CDN) dans l'architecture de l'Internet depuis une vingtaine d'années a contribué à améliorer l'acheminement des flux vidéos en rapprochant les contenus des usagers. Cependant, la croissance du trafic vidéo sur Internet pose un réel problème de capacité dans les CDN. Les technologies de streaming s'adaptant au débit offert par le réseau, telles que la norme Dynamic Adaptive Streaming over HTTP (DASH), renforcent ce problème dans l'infrastructure CDN de base, car fournir une seule vidéo signifie fournir plusieurs représentations ou qualités pour un même contenu pour un débit binaire agrégé qui est généralement supérieur à 10 Mbps. Dans l'article [32], nous avons explorés de meilleurs compromis entre le coût de l'infrastructure CDN et la qualité d'expérience (Quality of Experience (QoE)) des utilisateurs finaux pour les applications de diffusion vidéo en streaming "live". Dans le scénario considéré, l'infrastructure CDN est sous-dimensionnée, ce qui signifie qu'il n'est pas possible de transmettre toutes les représentations des contenus demandés aux serveurs de périphérie. Notre objectif était de maximiser la QoE pour la population des utilisateurs finaux hétérogènes malgré le manque de ressources dans les équipements CDN intermédiaires. Nous avons montré que les modèles théoriques antérieurs basés sur les débits élastiques ne conviennent pas à ce contexte. Nous avons proposé un modèle de diffusion en continu discret centré sur l'utilisateur où la satisfaction des utilisateurs finaux est liée au contexte et où un flux doit être diffusé dans son intégralité ou ne pas être diffusé du tout. Pour cela, nous avons formulé le problème sous la forme d'un programme linéaire en entiers (Integer Linear Programming (ILP)) qui permet d'obtenir une livraison optimale grâce à une superposition de livraison multi-arbres. L'évaluation de l'ILP a montré les avantages de ce modèle. Nous avons ensuite conçu un système pratique en revisitant les trois principaux algorithmes implémentés dans le CDN : l'assignation de l'utilisateur au serveur, le placement du contenu et la livraison du contenu. Puis, nous avons utilisé un

simulateur réaliste à grande échelle, piloté par des traces, pour étudier les performances de notre système. En particulier, nous avons montré que la population des utilisateurs est raisonnablement bien desservie (les trois quarts de la population ne subissent pas de dégradation) même lorsque l'infrastructure est gravement sous-dimensionnée (moins de la moitié de l'infrastructure requise).

2.1.2 Des solutions au niveau applicatif

Nous nous sommes intéressés aux limitations apportées par le réseau lui même et en particulier l'impact de la bande passante sur la livraison des flux vidéo. Dans [21], nous nous sommes intéressés au streaming de flux vidéos en direct à faible latence. Notre but était de minimiser la latence entre le moment où une nouvelle image vidéo est générée à la source et le moment où elle est jouée sur l'écran de l'utilisateur final. Nous avons proposé une implémentation de la diffusion vidéo en multi-chemin au niveau applicatif, qui exploite les informations fournies dans la dernière version des normes de diffusion vidéo. Notre implémentation de lecteur vidéo profite du multi-chemin pour permettre la lecture de vidéod transportées par Transmission Control Protocol (TCP) avec une latence inférieure à 100 ms. En initiant le transport en multi-chemin du côté client, notre mécanisme est compatible avec les équipements réseau existants et ne nécessite aucun changement ni au niveau du serveur, ni au niveau des middlebox.

Lorsque le débit binaire vidéo est supérieur à la bande passante réseau disponible, le flux vidéo subit une perte de paquets en raison des paquets qui doivent être abandonnés. L'application peut contribuer à fournir un service correct même quand les conditions du réseau ne permettent pas de les transmettre toutes avec la qualité de service requise grâce à une politique de sélection des données à acheminer. Dans [14] nous avons proposé une stratégie proactive de filtrage de paquets pouvant être mise en œuvre dans un serveur de streaming (ou un proxy réseau). Cette stratégie vise à bloquer volontairement (et à ne pas transférer) certains paquets quand le réseau n'offre pas assez de bande passante. Le défi consiste à décider quels paquets bloquer pour que la qualité de la vidéo côté client soit maximisée par rapport à la bande passante disponible. Les propositions précédentes cherchaient à utiliser les méta-informations issues de l'encodage vidéo ou à pré-traiter les données multimédia. Notre objectif était de concevoir une stratégie simple, qui n'utilise que les métadonnées vidéo disponibles à partir du conteneur de fichiers vidéo. Nous avons démontré sur un ensemble de vidéos High Efficiency Video Coding (HEVC) que notre algorithme léger de filtrage de paquets fonctionne aussi bien que des stratégies plus complexes. De plus, la qualité vidéo reste élevée malgré un grand nombre de paquets bloqués, tandis qu'une sélection aléatoire de paquets perdus entraîne une baisse significative de la qualité.

2.1.3 L'acheminement par des chemins multiples : une solution de la couche transport

Cependant, les applications ne permettent pas toujours l'acheminement sélectif des données, impliquant que toute donnée générée doit être reçue. Les solutions doivent alors venir du réseau. Nous nous sommes intéressés aux protocoles de transport spécifiquement conçus pour permettre une communication plus rapide et plus stable en exploitant simultanément plusieurs chemins réseau, en particulier le protocole Multi-Path Transmission Control Protocol (MPTCP). Ils offrent un délai de transmission moyen élevé, une fiabilité trop stricte et occasionnent des phénomènes fréquents de blocage des vidéos (problème de "*head-of-line blocking*") entraînant des chutes de débit brusques, ce qui les rends mal adaptés aux exigences du streaming vidéo. Dans [13], nous avons abordés

cette inadéquation en introduisant un planificateur multi-couches, qui exploite les informations des couches application et transport pour réorganiser la transmission des données et hiérarchiser les parties les plus importantes de la vidéo. Nous avons exploré des solutions d'ingénierie de trafic afin de choisir les chemins empruntés dans le réseau. Notre proposition s'applique à la fois aux services en direct ("live") et aux services à la demande. Nous avons fait une analyse théorique des potentiels de notre ordonnanceur multi-couche grâce à un ILP et proposé une implémentation basée sur des interactions réalistes entre les couches. Nous avons montré que les technologies disponibles permettent la mise en œuvre de ce planificateur multi-couches sans trop d'effort. La comparaison entre les performances du planificateur multi-couches optimal et celles d'un planificateur traditionnel a permis de justifier notre motivation à mettre en œuvre un planificateur multi-couches dans la pratique. Les utilisateurs bénéficient non seulement des avantages inhérents à l'utilisation de MPTCP (comme une meilleure résilience aux défaillances de chemin), mais aussi d'une meilleure QoE qu'avec le planificateur traditionnel.

Au cours de ces travaux sur l'acheminement des flux multimédia, nous avons évalué l'impact de plusieurs métriques réseaux : la bande passante disponible, le délai de transfert des vidéos et proposé des solutions opérant au niveau applicatif mais également au niveau du transport des données (CDN, protocole de transport multi-chemin).

Les différentes solutions proposées ont été définies pour le cas d'usage particulier du streaming vidéo du fait de son important volume (82% du trafic Internet global en 2022 [1]), cependant on ne peut négliger les autres types de trafic, en particulier celui de l'Internet des Objets qui aura la plus forte progression.

2.2 La QoS pour l'Internet des objets

Avec plus de 7 milliards d'objets connectés à Internet de nos jours, et le double attendu en 2022 [1], l'Internet des Objets pose de nouveaux défis aux opérateurs de réseaux, en particulier lorsqu'il s'agit de prendre en compte les contraintes des objets connectés. La plupart de ces appareils sont mis en œuvre par des systèmes matériels ayant des ressources extrêmement contraintes. Les nœuds de ces réseaux sont en général des capteurs ayant une énergie limitée (ils sont le plus souvent alimentés par pile), une faible puissance de traitement et une mémoire de taille réduite. De plus, ils communiquent par le biais de connexions sans fil assujetties à la perte [108]. D'un autre côté, ils ont un faible coût de production, peuvent être très petits et leur alimentation par pile permet une large gamme de cas d'utilisation tels que les réseaux de surveillance et de contrôle urbains [113], la sécurité industrielle, le contrôle et la surveillance des communications [94] ou la domotique [87]. Ces réseaux de capteurs sont déployés pour assurer des missions de surveillance ou de déclenchement de seuils d'alertes parfois critiques et souvent dans des environnements qui rendent les communications difficiles.

Au cours de différentes coopérations, j'ai considéré la prise en compte de ces contraintes dans deux types de réseaux IoT différents. D'une part les réseaux IoT industriels grâce à l'encadrement de la thèse de Pascal Thubert et à des collaborations avec Georgios Papadopoulos, Nicolas Montavont et Thomas Watteyne. D'autre part les réseaux IoT dans les villes intelligentes à travers une collaboration avec Valérie Issarny (INRIA@Sillicon Valley) (en particulier lors de mon séjour d'études à l'Université de Berkeley en 2016-2017), Françoise Sailhan, Nikolaos Georgantas, Benjamin Billet et Georgios Bouloukakis.

1. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

2.3 Le déterminisme dans les réseaux IoT industriels

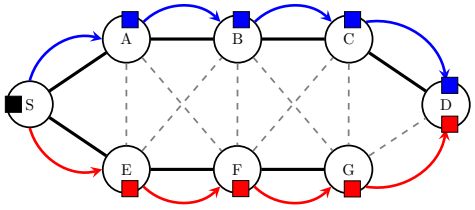
Les communications d'applications critiques telles que le contrôle des processus industriels, le réseau intelligent ou l'automatisation des véhicules reposent sur une fiabilité des communications de bout en bout proche de 100% et sur la ponctualité de la livraison des données. Par exemple, quatre pertes consécutives dans une boucle de contrôle d'automatisation industrielle peuvent suffire pour arrêter une chaîne de production. Selon [94] et [119], les industriels ont défini six classes d'applications classées en trois catégories : les applications de sécurité (classe 0) ne tolèrent pas de latence supérieure à 10ms, les applications de contrôle (classes 1 à 3) supportent des temps de latence maximum compris entre 10 ms et 100 ms et les applications de surveillance (classes 4 et 5) dont le temps de latence maximum est de 100 ms en moyenne. Par conséquent, l'Internet des objets industriel (IIoT) requièrt des réseaux mettant en œuvre des latences très faibles pour une partie des flux de données proches d'un comportement déterministe qu'IP ne sait pas offrir dans son état actuel. Or, les IIoT sont des réseaux de faible puissance et sujets aux pertes. Ils font partie des Low Power and Lossy Networks (LLN), une famille des réseaux dédiés aux communications avec des ressources limitées. Bien que la fiabilité des connexions puisse être améliorée en introduisant la diversité spectrale par saut de fréquence, elle ne protège pas contre les défaillances.

Un certain nombre d'efforts de normalisation ont été entrepris pour permettre la mise en place d'un réseau industriel sans fil basé sur IP. Le standard IEEE 802.15.4-2015, publié en 2016, vise à fournir de la qualité de service aux réseaux sans fil de type industriel. En particulier il traite les pertes de paquets dues aux interférences externes [91] et les évanouissements par trajets multiples (*multipath fading*)[112] en définissant le protocole TimeSlotted Channel Hopping (TSCH) qui tire parti de la diversité des chemins et des fréquences disponibles dans le réseau. Il définit un ordonnanceur qui planifie chaque communication en pré-sélectionnant un ensemble de nœuds et une fréquence. Le temps est divisé en tranches horaires (d'une durée standard de 10 ms), des *timeslots*, pendant lesquelles chaque nœud peut soit recevoir ou transmettre un paquet sur une fréquence donnée, soit se mettre en sommeil pour économiser l'énergie. L'ordonnancement des communications, en attribuant à certains flux de la bande passante et l'accès au canal de communication, peut être assimilé à un mécanisme de réservation de ressources. Ainsi, TSCH facilite l'émergence d'un LLN déterministe car, sans perte et retransmission, le temps de transfert d'une donnée entre deux nœuds est calculable.

Le groupe de travail IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) de l'IETF définit un ensemble de protocoles permettant l'utilisation d'IPv6 au-dessus de la couche MAC. Le but est d'hériter des capacités avancées d'Internet Protocol version 6 (IPv6) pour les mettre en œuvre dans des environnements de type industriel. En particulier, 6TiSCH fournit une couche réseau statistiquement multiplexée basée sur IPv6 (IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)) et sur la technologie IEEE802.15.4-TSCH. La planification des communications utilisée dans 6TiSCH favorise la ponctualité des livraisons de données tout en réduisant les interférences. Bien que 6TiSCH offre un mécanisme d'acquittement et de retransmission des données perdues pour assurer un transport fiable, les retransmissions occasionnées retardent leur réception à la destination. Le groupe DetNet de l'IETF a pour but la normalisation d'un service réseau proche du déterminisme pour des applications ayant des exigences très strictes en matière de qualité de service. Pour ce faire, il se concentre sur trois mécanismes [65] : l'évitement de la congestion, le routage explicite et la réplication, et l'élimination des paquets.

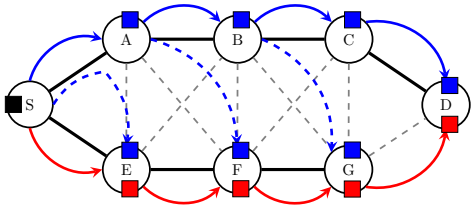
2.3.1 Le déterminisme grâce à la coopération et l'ingénierie de trafic

Dans le contexte des communications ordonnancées, les temps de transfert des données sont calculables, sous réserve d'éviter les retransmissions dues aux pertes de paquets. L'acheminement sur des chemins multiples est un moyen de garantir la livraison des paquets dans un délai borné. Dans le cadre de la thèse de Pascal Thubert et de l'article [36] nous avons proposé un mécanisme de collaboration, nommé Leapfrog, pour améliorer la fiabilité du réseau en tirant parti de la redondance native du support radio pour permettre une diversité (spatiale) supplémentaire dans le plan de données. Le principe est d'effectuer des transmissions parallèles sur deux chemins calculés au préalable et de mettre en œuvre une écoute passive de promiscuité entre les chemins. Les deux chemins sont choisis de façon à garantir que les nœuds d'un chemin sont à une distance leur permettant d'entendre des transmissions effectuées par des nœuds de l'autre chemin.



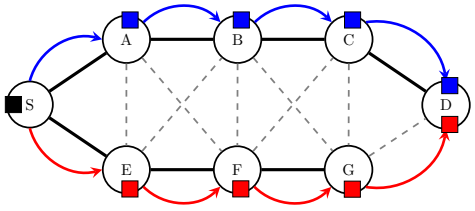
	0	1	2	3	4
0	$S \to A$			$F \to G$	
1		$A \to B$			$G \to D$
2		$S \to E$		$C \to D$	
3			$B \to C$		
4			$E \to F$		

FIGURE 2.1 – Emission multi-chemins



	0	1	2	3	4
f0	$S \to A$			$F \to G$	
f1		$A \to B$			
f2		$S \to E$		$C \to D$	$G \to D$
f3			$B \to C$		
f4			$E \to F$		

FIGURE 2.2 – Ecoute passive des communications du chemin 1



	0	1	2	3	4
0	$S \to A$			$F \to G$	
1		$A \to B$			$G \to D$
2		$S \to E$		$C \to D$	
3			$B \to C$		
4			$E \to F$		

FIGURE 2.3 – Timeslots où E est en état d'éveil pour écouter le trafic

Ce scénario est illustré par la figure 2.1 : les deux chemins parallèles entre la source S et la destination D sont $\{S - A - B - C - D\}$ et $\{S - E - F - G - D\}$. Selon les mécanismes de réplication et d'élimination proposés par DetNet, une trame devant être transférée de S à D est répliquée en S pour être envoyée sur chacun des chemins, charge à D d'éliminer la copie reçue en second. Nous avons proposé que les nœuds augmentent leur chance de recevoir une copie de la trame en écoutant passivement les communications de leurs voisins. Ainsi, le nœud E, voisin de S et A, est en mesure d'écouter l'envoi de la trame fait par S vers A (voir figure 2.2). Ce mécanisme permet d'augmenter la fiabilité du réseau, en effet une perte survenant lorsque S envoie la trame à E en empruntant le second chemin ne perturbe pas la transmission de la trame en multi-chemin, E ayant déjà capté la copie de la trame envoyée de S vers A. Notre proposition permet donc de fiabiliser la transmission sans occasionner de réémission des paquets perdus. Cela suppose d'indiquer

aux nœuds qu'ils doivent se réveiller afin d'écouter passivement des communications que leurs voisins émettent (voir figure 2.3). Dans la pratique, les nœuds devront se réveiller lors des timeslots de l'ordonnancement (*schedule*) correspondant aux communications de leurs voisins pendant lesquels ils seraient en état de sommeil pour une communication classique. Cependant, un nœud qui aurait acquis la copie de la trame par écoute passive n'a plus besoin de se réveiller pour le timeslot qui lui permettait de la récupérer dans l'ordonnancement d'origine. Les résultats de simulation montrent que Leapfrog réduit la latence de bout en bout par rapport à l'approche basée sur la retransmission tout en maintenant une gigue faible. Leapfrog permet d'assurer un très haut niveau de déterminisme mais augmente la consommation d'énergie puisque les nœuds peuvent être amenés à traiter deux fois le même paquet de données. Il faut donc coupler ce mécanisme avec une planification plus fine des réplications et éliminations de paquets sur plusieurs chemins afin de diminuer l'impact de Leapfrog sur la consommation d'énergie.

Nous avons évalué le mécanisme sur la topologie de la figure 2.4 par simulation en utilisant le plugin RealSim3 sur Cooja pour simuler la variation de la qualité des liens radio. Pour chaque simulation, le scénario attribue différentes qualités aux liaisons qui peuvent ensuite être dégradées (par exemple avec un Packet Delivery Ratio (PDR) inférieur à 20%), entraînant des perturbation pour la communication avec les nœuds voisins. Pour souligner la robustesse de l'approche Leapfrog, nous alternons successivement l'état des liens de chaque nœud de l'état défaut à l'état de mauvaise qualité toutes les deux minutes. Nous avons simulé ce comportement sur tous les nœuds intermédiaires dans l'ordre suivant : ID2, ID5, ID6, ID3, ID4 et ID7.

La figure 2.5 illustre la réduction de 28%, 41%, 46% et 54% du délai moyen de bout en bout des transmissions de paquets (qui comprend le temps de propagation du paquet et le délai de retransmission potentiel) par rapport aux approches basées sur la retransmission avec IEEE802.15.4-TSCH (c'est-à-dire RT2, RT4, RT6 et RT8 pour 2, 4, 6 ou 8 retransmissions). La figure 2.6 permet d'évaluer la pertinence de Leapfrog pour l'acheminement déterministe en représentant la gigue (calculée comme la différence entre les heures d'arrivée des paquets consécutifs à destination) observée par les paquets transmis. Avec Leapfrog, la gigue est stable et faible (600 ms en moyenne) alors que les solutions avec retransmission ont une gigue 58%, 71%, 77% et 84% plus élevée pour RT2, RT4, RT6 et RT8 (avec une valeur moyenne de 5000 ms pour RT8). Les bons résultats de RT0 s'expliquent par le fait que la gigue n'est calculée que sur les paquets reçus (comme le montre le PDR sur la figure 2.7). L'analyse du PDR, donnée dans la figure 2.7, complète cette analyse. Pour chaque cas, la perte de paquets est calculée comme $1 - PDR$ (aucune perte de paquets correspond à un PDR de 100%). Leapfrog démontre des performances PDR similaires aux approches basées sur la retransmission car, dans sa version actuelle, Leapfrog ne fait pas de retransmission.

Mise en œuvre du routage multi-chemin et de la supervision

Dans le cadre de la thèse de Pascal Thubert, nous avons travaillé sur un mécanisme d'ingénierie de trafic permettant à la fois d'indiquer le chemin que les paquets doivent suivre et faire la supervision des chemins dans le réseau. Le mécanisme Bit Index Explicit Replication - Traffic Engineering (BIER-TE), en cours de standardisation à l'IETF, permet d'indiquer à chaque nœud s'il doit répliquer, éliminer ou simplement transmettre un paquet tout en respectant l'ordonnancement des timeslots mis en place pour la transmission des données dans le réseau.

BIER-TE s'inspire du protocole de transfert multicast Bit Index Explicit Replication (BIER) proposé à l'IETF [114]. Tous deux définissent une information protocolaire

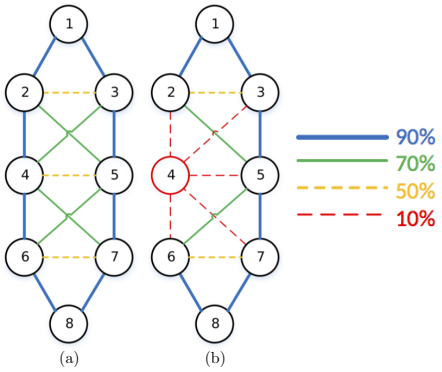


FIGURE 2.4 – Qualité de la liaison réseau émulée avec le plugin RealSim (a) et exemple de liaisons faibles en ID4 (b)

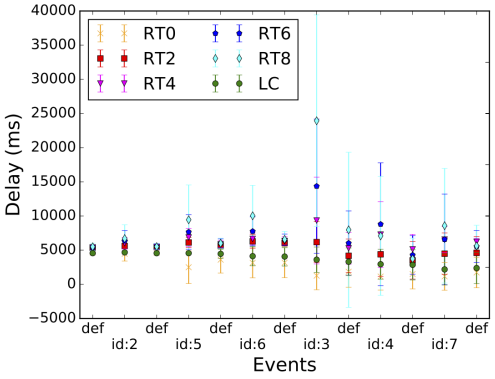


FIGURE 2.5 – Average end-to-end delay : source ID8 to sink ID1.

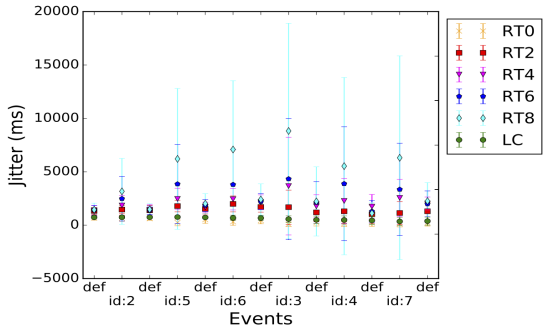


FIGURE 2.6 – Average end-to-end jitter : source ID8 to sink ID1.

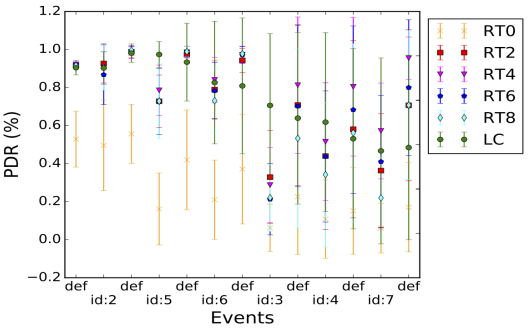


FIGURE 2.7 – Packet delivery ratio performance.

appelée bitmap pour indiquer comment le paquet doit être transféré. Dans BIER, le bitmap est ajouté aux paquets de données multicast dans un en-tête BIER. Chaque bit défini dans le bitmap correspond à une destination spécifique qui doit recevoir le paquet. Un bit du bitmap est mis à 1 si le paquet doit être transféré vers la destination associée, 0 sinon. Comme BIER, BIER-TE introduit un bitmap pour piloter le transfert du paquet. Cependant, pour répondre aux besoins d'un routage explicite, il existe une différence majeure sur la signification de ce bitmap : dans BIER-TE, chaque bit du bitmap représente une adjacence entre deux nœuds. Ceci permet à BIER-TE d'implémenter le transfert saut par saut de façon explicite ou implicite. BIER-TE s'appuie donc sur un contrôleur pour calculer des chemins complexes et attribuer un index de bits à chaque adjacence. En outre, le contrôleur doit installer une table de transfert d'index de bits Bit Index Forwarding Table (BIFT) dans chacun des nœuds pour indiquer comment transférer un paquet en fonction du bitmap qu'il transporte, comme représenté dans la figure 2.1.

Ces travaux offrent de nouvelles perspectives pour des outils d'exploitation, d'administration et de maintenance (Operations, Administration and Maintenance (OAM)) qui, basés sur BIER-TE, permettent de nouveaux degrés de contrôle et de traçabilité dans un réseau déterministe. Pour cela, nous avons proposé un nouvel algorithme de routage multi-chemin qui tire parti de la diversité des routes en dupliquant le flux de données sur des chemins différents et congruents, c'est-à-dire la réplication et l'élimination des paquets (nommé Packet Replication and Elimination (PRE)). Le mécanisme a été mis

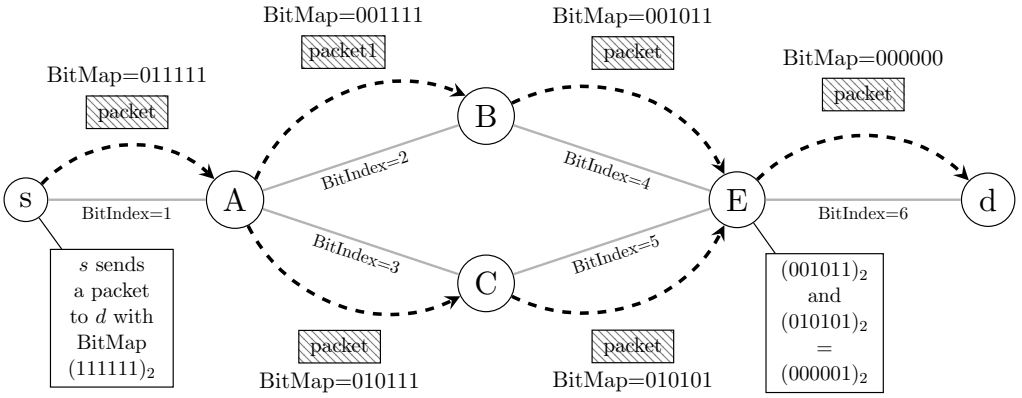


FIGURE 2.8 – Emission multi-chemins

en œuvre sur OpenWSN, une implémentation open-source d'une pile 6TiSCH complète basée sur les standards IoT (IEEE 802.15.4-TSCH, IPv6, 6TiSCH, IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), Constrained Application Protocol (CoAP)). L'étude des performances de BIER-TE lorsqu'il est combiné avec TSCH a montré que le déterminisme peut être atteint ce qui est essentiel lorsque la latence requise est limitée et ne permet qu'une seule retransmission.

Les expérimentations menées confirment que BIER-TE est une solution prometteuse pour mettre en œuvre la communication déterministe [36][25]. Bien entendu, le routage explicite et la réplication/élimination de paquets ont un coût en termes de *timeslots* et d'énergie. Nous avons donc exploité la possibilité d'écouter passivement les communications des autres nœuds du réseau pour augmenter les chances des nœuds de recevoir les messages, ce qui vise à équilibrer les transmissions supplémentaires occasionnées par la réplication et l'élimination des données envoyées. Ainsi, la combinaison de BIER-TE avec TSCH et le modèle de collaboration Leapfrog réduit le coût énergétique de l'acheminement fiable des données. L'ajout d'un contrôleur de type SDN (Software Defined Networking) [11] permet la mise en œuvre d'une boucle de contrôle pour aider à réduire le coût de BIER-TE en n'autorisant la réplication que lorsqu'une protection renforcée de la communication est nécessaire.

2.4 La gestion du routage des réseaux IoT dans les villes intelligentes

Les sections précédentes montrent que les applications peuvent jouer un rôle intéressant pour atténuer les problèmes de qualité de service des réseaux mais que les efforts pour offrir un service de qualité doivent porter sur l'architecture et les protocoles, en particulier lorsque les exigences sont fortes comme dans le cadre du streaming vidéo ou des transmissions dans l'Internet des objets industriel. Cependant lorsqu'il s'agit de réseaux IoT, le respect de la qualité de service ne peut pas se faire au détriment de la consommation d'énergie.

En particulier, nos villes deviennent chaque jour plus intelligentes grâce au déploiement de services numériques tels que l'éclairage intelligent, la surveillance de la pollution environnementale, la gestion de l'énergie et des déchets, le stationnement intelligent, et plus encore (Figure 2.9). Les services urbains intelligents reposent en grande partie sur les infrastructures de l'IoT urbain. Au minimum, les objets connectés sont des capteurs qui

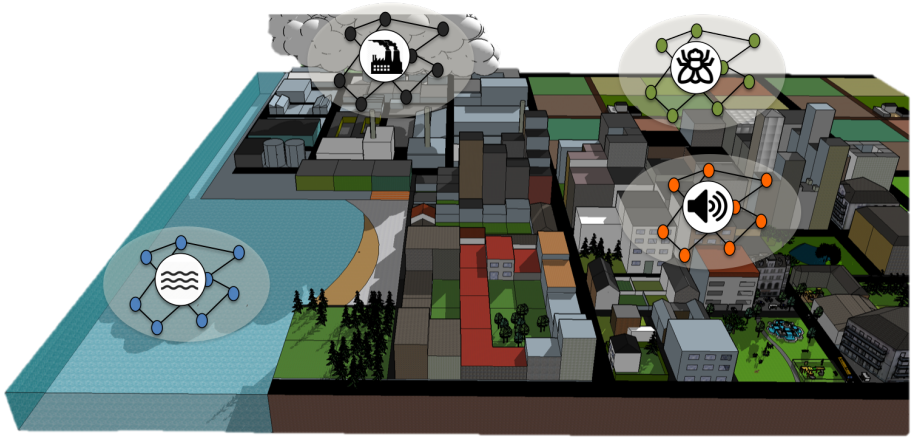


FIGURE 2.9 – Usages des réseaux WSN dans la ville intelligente

effectuent des mesures et les envoient via le réseau de capteurs sans fil sous-jacent (Wireless Sensor Network (WSN)) à un serveur Internet pour une analyse plus approfondie. Or, en raison de contraintes de déploiement dans l'environnement urbain, les objets sont souvent alimentés par pile et leur durée de vie est donc fortement liée à la consommation en énergie de leurs communications.

2.4.1 La problématique de consommation d'énergie dans les réseaux IoT urbains

L'une des raisons de l'épuisement rapide de l'énergie des nœuds du WSN vient de l'utilisation du protocole RPL[115] pour construire et maintenir les adjacences des nœuds. RPL est un protocole à vecteur de distance conçu pour fonctionner sur des centaines de nœuds en construisant un graphe acyclique orienté vers la destination (Destination Oriented Directed Acyclic Graph (DODAG)). Chaque noeud RPL contribue à la création du DODAG de manière distribuée en choisissant son rang et son parent préféré pour transmettre les messages vers le puits racine du WSN (reliée à Internet). Le rang est une métrique calculée dynamiquement et constamment mise à jour pendant la durée de vie du réseau RPL. Il définit la distance virtuelle entre le nœud et la racine du DODAG. Cette distance peut être calculée selon différents types de mesures (par exemple le nombre de sauts, la qualité des liens, le temps de retransmission prévu, etc.). Une fois le DODAG créé, les données sont transmises de manière *convergecast* par le chemin le plus court par rapport à une fonction d'objectif définie par RPL prenant en compte une ou plusieurs métriques. Par conséquent, si l'on néglige le trafic de contrôle en arrière-plan et la fonction de capteur, un nœud WSN passera son énergie à envoyer ses données et à relayer le trafic de ses enfants vers son parent préféré. La réduction de la consommation d'énergie des protocoles de routage dans les réseaux WSN a fait l'objet d'un travail considérable [90]. Plusieurs travaux portent sur l'optimisation de la consommation d'énergie de RPL. Parmi les solutions existantes, [72] prône l'utilisation d'une métrique représentant l'énergie disponible des nœuds. La réduction de la consommation d'énergie moyenne est assurée par la sélection des trajets ayant le budget énergétique disponible le plus élevé. Cependant, l'impact de cette métrique très dynamique sur la gestion du DODAG n'est pas négligeable et crée une grande instabilité de routage due aux changements fréquents des parents préférés.

Une autre façon de réduire la consommation d'énergie due au routage des données dans le WSN est d'introduire des puits supplémentaires qui sont capables d'absorber et

d'envoyer une partie du trafic par leur propre connexion Internet. En effet, la transmission de données à un puits supplémentaire n'augmente pas la consommation d'énergie de l'expéditeur. Au contraire, il s'agit d'économiser l'énergie du nœud parent et des nœud ancêtres en diminuant le volume de données qu'ils relaieraient autrement vers la racine du WSN. Un puits supplémentaire bien placé peut réduire la consommation d'énergie due au relayage et ainsi augmenter la durée de vie du réseau. Plusieurs articles tirent parti des techniques de réseau tolérant aux délais (Delay-Tolerant Networking (DTN)) pour mettre en œuvre la collecte de données avec des puits mobiles. Les techniques diffèrent selon les contraintes qui sont établies en ce qui concerne la mobilité et la disponibilité des puits ([85][50]).

En outre, les infrastructures urbaines de l'IoT servent principalement à la surveillance à long terme des zones urbaines, alors qu'elles sont moins adaptées au suivi de l'expérience des citoyens. La mesure mobile participative (*crowdsensing*), effectuée grâce à des *smartphones*, des téléphones mobiles intelligents, est un moyen efficace de recueillir une grande quantité de données sur l'environnement urbain et d'évaluer l'expérience des citoyens. Cependant, la diversité des smartphones et l'hétérogénéité des capteurs embarqués/connectés rendent la corrélation des observations recueillies plus difficile [73][22]. Nous pensons que les infrastructures IoT en milieu urbain (les WSN urbains) et les appareils de mesures participatives (que nous appellerons *crowdsensors* pour faire court) doivent fonctionner conjointement de manière à : (1) améliorer la qualité des observations sur l'environnement urbain et (2) augmenter la durée de vie des infrastructures IoT urbaines.

D'une manière similaire aux solutions basées sur les DTN, notre travail s'appuie sur des nœuds mobiles servant de puits pour réduire la consommation d'énergie des nœuds du WSN due au relayage des messages. Dans [40], au lieu d'ajouter des puits dédiés, les infrastructures IoT urbaines (des WSN déployés dans la ville) coopèrent avec des applications de crowdsensing que les villes promeuvent de plus en plus et que les usagers adoptent au nom de l'engagement citoyen. Nous nous sommes plus particulièrement intéressés au cas d'usage de la mesure de la pollution sonore (illustré par la figure 2.10), inspiré de l'application Ambiciti [2] avec laquelle les utilisateurs contribuent à la surveillance de l'exposition individuelle et collective à la pollution sonore urbaine. L'application effectue des mesures de bruit et les envoie vers un serveur pour le stockage et l'agrégation des données. Notre solution étend encore la contribution du smartphone pour qu'il offre en plus un service de communication à l'infrastructure de l'IoT. Le smartphone agit alors comme un puits supplémentaire qui recueille les données des capteurs voisins, alors qu'il se déplace au long de la zone de couverture du WSN. Les smartphones des utilisateurs volontaires relaient une partie des mesures effectuées par les nœuds du WSN, ce qui réduit la charge des réseaux IoT urbains et prolonge leur durée de vie.

Afin de pouvoir fonctionner, notre mécanisme nécessite que les les smartphones soient capapbles de communiquer avec les capteurs ce qui n'est pas implémenté actuellement en général. Pour ce faire, les smartphones devraient avoir une interface réseau pour se connecter aux nœuds du WSN (par exemple en Zigbee, 802.15.4 ou Bluetooth Low Energy (BLE)). Bien qu'à l'exception de BLE, la plupart des smartphones n'intègrent pas encore les principales interfaces supportées par WSN, nous considérons que cela va évoluer avec le temps. Nous pensons que le manque de compatibilité est moins un problème technique qu'un problème économique puisque les smartphones implémentant IEEE Std 802.15.4 n'ont pas encore rencontré leur marché. Nous avons donc considéré que la communication directe entre l'infrastructure IoT et les smartphones n'est pas un point de blocage et sera disponible à l'avenir. De plus, dans notre cas d'utilisation l'utilisateur télécharge

2. `http://ambiciti.io/`

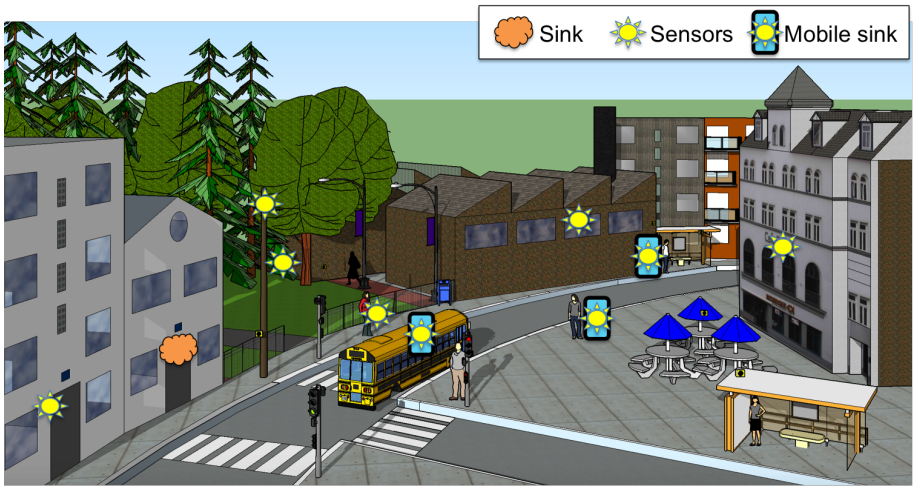


FIGURE 2.10 – Cas d'usage de la mesure de pollution sonore

une application de crowdsensing. Par conséquent, l'appareil de l'utilisateur a obtenu les informations d'identification et les autorisations nécessaires pour soumettre des données au serveur avec des conditions de sécurité suffisante. Notre but n'est pas de développer des considérations sur les questions de sécurité, celles ci étant déjà exposées dans [69].

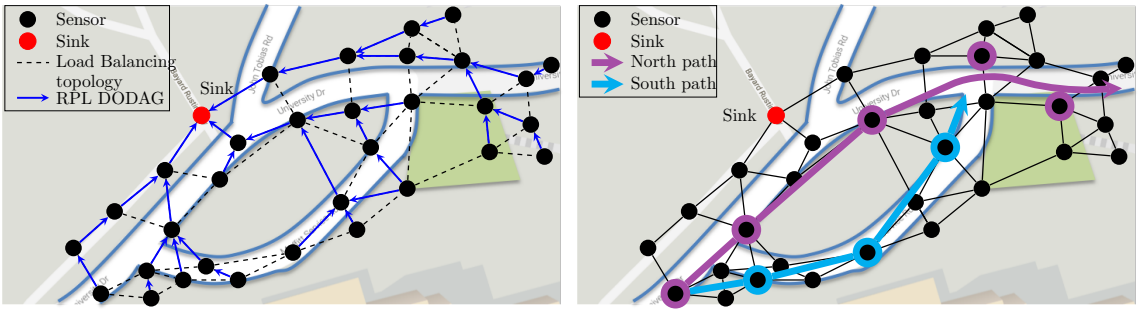
Nous avons adopté les principes de l'ingénierie du trafic pour prolonger la durée de vie du réseau, en combinant les trois approches suivantes :

— **Équilibrage de charge**.
Nous profitons du maillage du WSN pour mettre en place de l'équilibrage de charge dans l'infrastructure de mesure urbaine. Contrairement au routage RPL classique, le trafic est réparti entre plusieurs parents pour réduire la consommation d'énergie sur le chemin vers la racine du WSN, ce qui conduit à étudier comment un nœud devrait répartir son trafic sortant entre ses parents. De même, nous évaluons la quantité de données que chaque nœud peut relayer pour ses enfants. Il faut noter que nous ne nous concentrons pas sur le moment auquel les données doivent être transmises mais plutôt sur un problème de routage de flux qui ne prend pas en compte l'ordonnancement.

— **Optimisation du routage**.
Nous calculons le pourcentage de trafic à envoyer à chaque parent pour maximiser la durée de vie du réseau.

— **Introduction des puits mobiles**.
Nous prenons en compte les nœuds mobiles de crowdsensing qui circulent dans la zone de déploiement de l'infrastructure urbaine de mesure. Nous nous concentrons plus particulièrement sur la quantité de données qu'un nœud mobile devrait recueillir pour prolonger la durée de vie de l'infrastructure. Nous modélisons en outre le déplacement des puits mobiles en définissant un puits comme un nœud ayant une durée de présence limitée (et donc une capacité de transmission correspondante) dans des endroits où le puits mobile reste potentiellement suffisamment longtemps pour pouvoir mettre en œuvre les échanges (typiquement des arrêts de bus, des feux de circulations et des passages piétons, ou des restaurants).

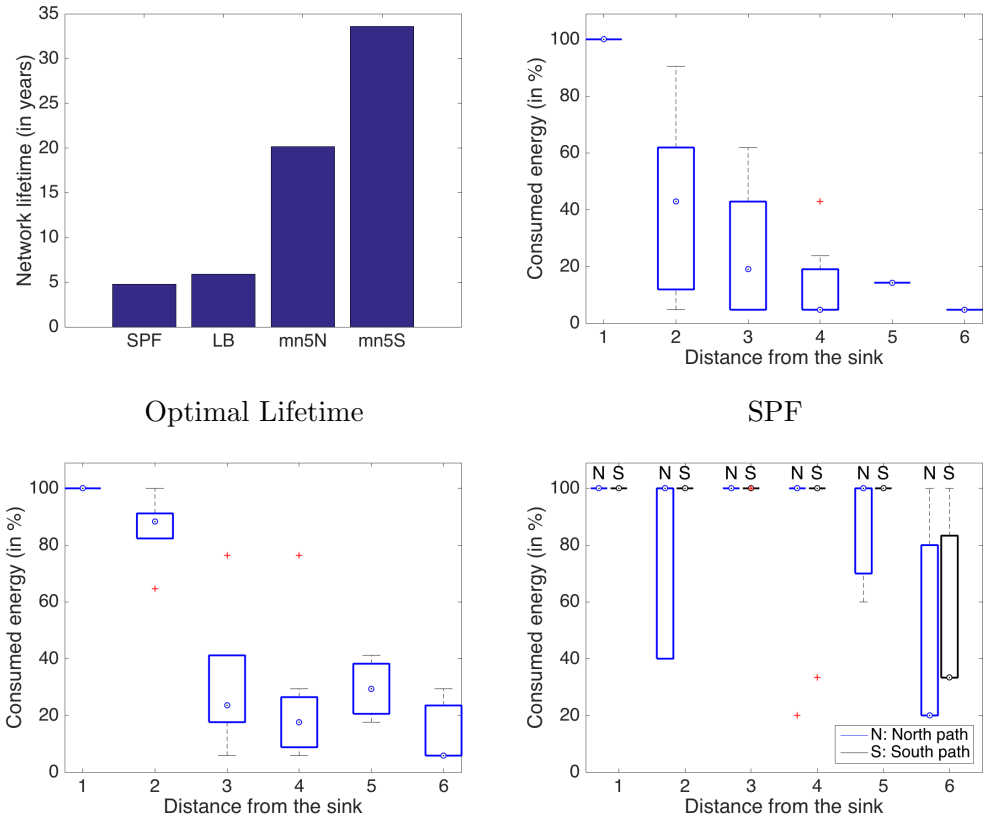
Nous avons défini un programme linéaire pour calculer une stratégie de routage qui détermine pour chaque nœud le pourcentage de trafic qu'il doit acheminer vers chacun de ses voisins - y compris un puits mobile si disponible - dans sa zone de couverture, de



selon RPL avec des puits mobiles

FIGURE 2.11 – Exemple de topologie de WSN urbain

manière à maximiser la durée de vie du réseau. Nous avons choisi la définition la plus stricte de la durée de vie du réseau : elle se termine dès qu'un des capteurs du WSN est à court d'énergie. Les détails du programme linéaire et de l'algorithme de routage qui l'utilise sont donnés dans [40] dont la figure 2.12 présente quelques résultats.



Optimal Lifetime SPF

LB North and south path comparison

FIGURE 2.12 – Sensor lifetime analysis

Les résultats des simulations effectuées montrent la validité de l'approche et le gain en longévité du WSN. En particulier, notre programme linéaire permet de prolonger la durée de vie des nœuds les plus proches de la racine du WSN (ceux qui s'épuisent en premier avec le routage RPL classique). Ainsi une solution simple de partage de charge permet de

solliciter d'avantage les nœuds à une distance 2 de la racine alors que la solution faisant intervenir les puits mobiles prolonge suffisamment la durée de vie du réseau pour épuiser l'ensemble des nœuds (mis à part les feuilles qui ne font pas de relayage). Ceci se traduit dans nos simulations par une multiplication de la durée de vie du réseau par 4 ou 7 en fonction des positions des puits mobiles. Notre travail ouvre également plusieurs questions de recherche. L'un de ces problèmes est lié à la stratégie d'équilibrage de charge proposée qui nécessite de transformer le mécanisme de routage traditionnel pour introduire une table avec routage permettant de faire un partage de charge évoluant dans le temps. Dans le cadre des WSN alimentés par pile, la quantité d'énergie disponible est finie et peut être évaluée avant de concevoir la stratégie de routage. La nouvelle table de routage d'un nœud devrait donc pouvoir indiquer la quantité de trafic qu'il doit envoyer à chacun de ses parents et peut le laisser libre d'arbitrer l'envoi vers chacun d'eux en fonction par exemple des conditions radio actuelles ou d'un ordonnancement des communications (avec TSCH par exemple). Bien qu'une telle table de routage soit plus grande que la table actuelle, elle peut atténuer l'impact des défaillances temporaires de nœuds ou de liens.

2.5 Conclusion

Ce chapitre a présenté différent travaux portant sur l'étude de l'impact de contraintes fortes de qualité de services, en particulier dans les flux qui prennent de plus en plus d'importance dans les réseaux (vidéo et IoT). Lors de ces travaux, nous avons pu aborder des solutions applicatives qui peuvent atténuer les problèmes de qualité de service dans les réseaux. Cependant, nous avons pu constater que les solutions applicatives ne suffisent pas et que des solutions architecturales et protocolaires sont nécessaires. S'il existe cependant des architectures et des protocoles permettant d'améliorer le comportement du réseau en terme de garantie de service, ces solutions sont souvent compliquées ou impliquent une complexité de gestion induisant des coûts d'exploitation importants. Nos travaux sur la QoS pour les réseaux IoT montrent l'intérêt d'utiliser la diversité de routage offerte par les réseaux, ce qui implique une gestion beaucoup plus complexes du routage et un recours à l'ingénierie de trafic.

De plus, prendre en compte une contrainte de qualité de service unique ne suffit pas. La réponse apportées jusqu'à présent était essentiellement de prendre en compte une contrainte sur la bande passante, faisant l'hypothèse que les délais et les pertes observées par les flux sont minimes quand le réseau est peu chargé. Cette hypothèse n'est plus suffisante lorsque l'on considère des applications exigeantes telles que le streaming de flux vidéo en direct ou les boucles de controle des usines. Prendre en compte efficacement la qualité de service impose de considérer en même temps plusieurs contraintes pour effectuer une ingénierie de trafic pertinente. Comme nous le verrons dans le chapitre 3, la prise en compte des contraintes multiples est un problème difficile. En outre, pour être efficace, l'ingénierie de trafic nécessite de repenser le plan de routage dans sa globalité et d'avoir une vision complète de la topologie et des matrices de trafic. Ceci renforce l'intérêt des opérateurs de réseaux pour les solutions plus centralisées. L'évolution depuis quelques années vers l'automatisation des réseaux et leur gestion plus informatisée avec l'émergence de Software Defined Networking et la virtualisation des réseaux vont en ce sens, comme nous le verrons dans le chapitre 4 et la suite de nos travaux.

CHAPITRE 3

La sélection de chemin avec des contraintes multiples dans l'Internet

3.1 Introduction

Bien qu'offrir de la QoS dans l'Internet soit un enjeu suffisamment stratégique pour avoir suscité de nombreux travaux depuis plusieurs décennies, ce n'est toujours pas possible de nos jours. L'une des raisons majeure vient de la structure même de l'Internet qui interconnecte aujourd'hui plus de 65000 systèmes autonomes [1]. Le RFC 1930 [70] définit un système autonome (Autonomous System (AS)) comme un ensemble de routeurs administré par une entité unique qui utilise un ou plusieurs protocoles de routage interne (Interior Gateway Protocol (IGP)), des métriques pour router des paquets et un protocole de routage externe (Exterior Gateway Protocol (EGP)) pour acheminer des paquets vers d'autres AS [2].

La notion de système autonome permet de mettre en œuvre une architecture à deux niveaux : un niveau externe pour le routage dans l'Internet (entre les AS) et un niveau interne pour le routage à l'intérieur de l'AS. Ceci garantit les trois propriétés essentielles pour un mécanisme inter-domaine : l'autonomie, la confidentialité et la scalabilité. Tout AS dispose d'une **autonomie** de gestion et décide de l'organisation de ses réseaux internes ainsi que des protocoles utilisés, en particulier les protocoles de routage et les métriques associées. Un système autonome est donc vu des autres AS comme une boite noire ayant des points d'entrée et de sortie connectés au reste de l'Internet. La propriété de **confidentialité** garantit que le comportement interne, les performances et la topologie des réseaux d'un AS (qui sont des information sensibles) ne sont pas dévoilées en dehors de l'AS. La propriété de **scalabilité** garantit le passage à l'échelle de tout mécanisme inter-domaine qui doit pouvoir s'adapter à l'augmentation constante du nombre d'AS.

Au sein d'un AS (en intra-domaine), l'administrateur peut mettre en place une politique de qualité de service pour garantir les performances de ses réseaux vis à vis d'une métrique de qualité de service. Cela se complique lorsque la QoS doit être offerte de bout-en-bout dans l'Internet car cela fait intervenir plusieurs domaines. Le respect d'une contrainte de QoS impose que cette métrique soit évaluée et respectée de façon cohérente par l'ensemble des domaines traversés sur le chemin entre la source et la destination. Or,

1. source : `http://www.cidr-report.org/as2.0/#General_Status` (visité en septembre 2019)
2. Par la suite, nous nous référons indifféremment aux zones de routage autonomes avec les termes AS ou domaines

pour des raisons de confidentialité ou pour préserver l'évolutivité des protocoles de routage, les informations sur la topologie et les performance des réseaux internes sont confinée à l'AS [109]. Ceci rend le calcul de chemins avec garantie de QoS traversant plusieurs domaines (inter-domaine) plus complexe et requiert une coopération entre les AS concernés. Il faut partager la contrainte de bout-en-bout entre les AS qui sont ensuite responsables d'assurer leur part de garantie de service. Cela nécessite un mécanisme scalable capable de mettre en place cette coopération critique (la défaillance d'un des AS peut entraîner le non respect du contrat de service global), tout en respectant les propriété de confidentialité et d'autonomie de chaque AS.

Restreindre le problème de routage avec garantie de QoS à une seule métrique (telles que le délai ou le coût d'acheminement) n'est pas satisfaisant. Un fournisseur de service Internet a besoin de pouvoir calculer des solutions de routage garantissant plusieurs métriques à la fois. Les métriques considérées dans les réseaux sont en général additives, concaves ou multiplicatives. Une métrique est **additive** si sa valeur pour un chemin est obtenue en ajoutant la métrique de chaque lien composant le chemin (par exemple le délai, la distance ou le coût). Une métrique est **concave** lorsqu'elle indique une valeur minimale respectée par tous les liens composant le chemin (par exemple la bande passante). Une métrique **multiplicative** est le produit de la métrique de chaque lien composant le chemin (par exemple le taux de perte). L'utilisation du logarithme permet de transformer les métriques multiplicatives en métriques additives. La garantie vis à vis d'une métrique concave est mise en œuvre en élaguant les liens qui ne pourraient pas la respecter. Le routage est alors calculé sur une topologie ne comportant que des liens ayant les ressources suffisantes. Après un tel élagage, les problèmes de calcul de chemin respectant des contraintes de qualité de service ne considèrent que des métriques additives. Par la suite, nous nous intéressons à deux problèmes en particulier : *le problème MCP (Multi-Constrained Path)* eet le problème *le problème MC0P (Multi-Constrained Optimal Path).*

Les travaux présentés dans ce chapitre ont été menés à l'occasion des thèses de Gilles Bertrand (soutenue en 2008) et de Romain Jacquet (soutenue en 2015) ainsi que de coopérations avec Emmanuelle Anceaume, Alberto Blanc, Yann Busnel, Paul Lajoie-Mazenc, Samer Lahoud et Miklos Molnar. Nous nous sommes intéressés au problème de recherche de chemins respectant plusieurs contraintes de QoS en inter-domaine. Dans un premier temps, la suite des domaines sollicités pour la recherche du chemin est connue, ce qui simplifie le problème mais réduit l'espace des solutions possibles. Nous avons proposé un algorithme exact, une heuristique et également une solution approchée. Dans un second temps, nous avons considéré que la séquence des domaines sollicités n'est plus connue et nous avons proposé un mécanisme pour explorer un ensemble de domaines restreint autour du plus court chemin. Dans ce cadre, les algorithmes peuvent proposer plusieurs chemins faisant intervenir des séquences de domaines différentes. Le choix du chemin à mettre en œuvre relève de la source. Afin de l'aider à choisir un chemin ayant de bonnes chances de respecter effectivement la QoS demandée, nous avons proposé un mécanisme de réputation attribuant une note à chaque domaine en fonction de son respect des contraintes de QoS lors des demandes antérieures. La combinaison de ces contribution constitue un système global capable de proposer à une source un ensemble de chemin pouvant transporter un trafic tout en garantissant un ensemble de métriques de qualité de service.

3.2 Le problème de routage multi-contraint

Le calcul d'un chemin soumis à de multiples contraintes avec des poids indépendants et additifs est un problème difficile même à l'intérieur d'un seul domaine. Le problème de base s'appelle le problème du chemin multi-contraint (Multi-Constrained Path (MCP)) et consiste à rechercher un chemin réalisable qui satisfasse un ensemble des contraintes donné, alors que la recherche d'un chemin optimal est appelé problème du chemin optimal multi-contraint (Multi-Constrained Optimal Path (MCOP)) [81].

3.2.1 Les problèmes MCP et MCOP

Pour résoudre les problèmes MCP et MCOP, on représente le réseau par un graphe dirigé $G = (V, E)$ pour lequel l'ensemble des sommets V représente l'ensemble des nœuds du réseau et l'ensemble des arcs E représente l'ensemble des liens du réseau. Les métriques de QoS correspondent à des poids additifs et positifs notés $w_k, k = 1..K$ pour chaque arc $e \in E$ (avec au moins un des poids non nul). L'ensemble des chemins entre deux nœuds s et d est désigné par $P_{s\to d}$ et chaque chemin p est associé à K poids dénotés $w_k(p) \equiv \sum_{e\in p} w_k(e)$, $k = 1..K$. Une fonction de longueur $c(p)$ peut être associée au chemin p soumis aux contraintes K. Cette longueur est utilisée pour déterminer le meilleur chemin (c'est-à-dire le chemin le plus court en fonction de la longueur considérée). La fonction de longueur de chemin la plus appropriée, définie dans [124], exprime la valeur critique du chemin par rapport aux contraintes :

$$c(p) = \max_{i=1..K} \left(\frac{w_i(p)}{W_i}\right) \tag{3.1}$$

Intuitivement, $c(p)$ permet de sélectionner le chemin le plus éloigné de la violation des contraintes. Avec cette fonction de longueur, un chemin est réalisable si et seulement si $c(p) \leq 1$. Les deux problèmes fondamentaux MCP et MCOP sont formulés comme suit [78][81]. Toute solution du problème MCP est appellé **chemin faisable**.

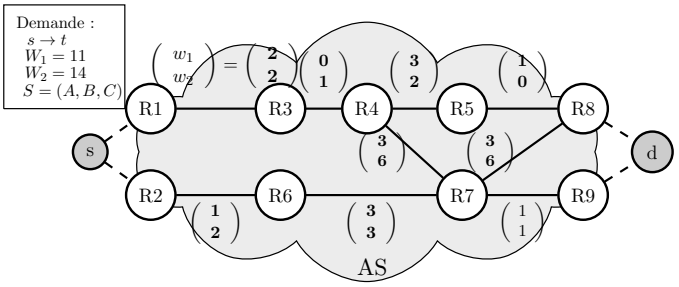
Problème. *MCP (Multi-Constrained Path)*
Soit une source s et d'une destination d, K contraintes W_k, $k = 1..K$, trouver un chemin $p \in P_{s\to d}$ tel que $w_k(p) \leq W_k$, pour $k = 1..K$.

Problème. *MCOP (Multi-Constrained Optimal Path)*
Soit une source s et d'une destination d, K contraintes W_k, $k = 1..K$, trouver un chemin réalisable $p^* \in P_{s\to d}$ tel que pour tout autre chemin réalisable $p \in P_{s\to d}$, $c(p^*) \leq c(p)$.

3.2.2 La notion de dominance et la Pareto-optimalité

La solution du problème MCP peut ne pas être unique car il peut y avoir de nombreux chemins faisables entre deux nœuds. On peut les classer selon une fonction de longueur comme dans le problème MCOP, en particulier la notion de dominance, souvent appelé Pareto-optimalité [106] permet de réduire l'espace des chemins considérés (en éliminant les chemins dominés) pour ne garder que les chemins pouvant conduire à des solutions intéressantes.

Définition 1. *La notion de dominance*
Un chemin p est *dominé* s'il existe un chemin p', ayant la même source et destination, tel que $w_k(p') \leq w_k(p)$ pour tous les poids w_k considérés, avec $k \in [1..K]$ et tel qu'il existe un $k \in [1..k]$ pour lequel $w_k(p') < w_k(p)$. Dans ce cas, nous dirons que p' domine p.



Chemins faisables	Longueur
Non dominés	
s-R1-R3-R4-R5-R8-d	$\binom{6}{5}$
s-R2-R6-R7-R9-d	$\binom{5}{6}$
Dominés	
s-R1-R3-R4-R7-R9-d	$\binom{6}{10}$
s-R2-R6-R7-R8-d	$\binom{7}{11}$

FIGURE 3.1 – Exemple de calcul de la dominance des chemins faisables

Si nous reprenons l'exemple donné dans la Figure 3.1, parmi les quatres chemins faisables, deux sont dominés. Ainsi, $\{s-R1-R3-R4-R7-R9-d\}$ de longueur $\binom{6}{10}$ est dominé sur la seconde métrique par $\{s-R1-R3-R4-R5-R8-d\}$ de longueur $\binom{6}{5}$ et sur les deux métriques par le chemin $\{s-R2-R6-R7-R9-d\}$ de longueur $\binom{5}{6}$. De même, le chemin $\{s-R2-R6-R7-R8-d\}$ de longueur $\binom{7}{11}$ est dominé par les chemins $\{s-R1-R3-R4-R5-R8-d\}$ de longueur $\binom{6}{5}$, $\{s-R2-R6-R7-R9-d\}$ de longueur $\binom{5}{6}$ et $\{s-R1-R3-R4-R7-R9-d\}$ de longueur $\binom{6}{10}$. En revanche, $\{s-R1-R3-R4-R5-R8-d\}$ de longueur $\binom{6}{5}$ et $\{s-R2-R6-R7-R9-Rd\}$ de longueur $\binom{5}{6}$ sont non-dominés. $\{s-R2-R6-R7-R9-d\}$ est meilleur pour la première métrique mais la situation est inversée pour la seconde métrique, aucun des deux chemin n'est meilleur que l'autre sur l'ensemble des métriques.

3.3 MCP et MCOP en inter-domaine

Wang et Crowcroft ont prouvé dans [111] que le problème de routage multi-contraint est NP-difficile, ils montrent également que les problèmes MCP et MCOP sont en fait tous deux NP-complets si les métriques additives considérées sont indépendantes. La référence [124] donne un aperçu des méthodes de calcul de chemins à contraintes multiples. Les solutions du problème MCP peuvent être trouvées par des algorithmes de force brute (par exemple la recherche en profondeur avec retours en arrière). Cependant, complexité de calcul des problèmes MCP et MCOP fait que la complexité temporelle des algorithmes de force brute augmente de façon exponentielle (dans les pires cas), ce qui a suscité la proposition d'heuristiques (par exemple dans [99][75][78]) et de plusieurs algorithmes d'approximation ([117][118]). Dans [123][125], Kuipers et Van Mieghem affirment que le comportement NP-complet du problème MCP n'apparaît que dans des graphes construits avec des poids soigneusement choisis. Néanmoins, des travaux ultérieurs montrent que dans la plupart des cas, les problèmes de calcul de chemins multi-contraints appliqués aux réseaux peuvent être résolus avec précision [123][125]. Par la suite, la communauté de recherche a également exploré l'utilisation de métaheuristiques pour résoudre le problème de routage multi-contraint, comme les algorithmes génétiques [98], la recherche tabu [116], et les colonies de fourmis [71].

Dans [105], De Neve et Van Mieghem ont publié un algorithme exact appelé Self Adaptive Multi-Constrained Routing Algorithm (SAMCRA). Cet algorithme résout le problème MCOP en explorant tous les chemins non dominés, entre un nœud source s et un nœud de destination d, puis renvoie la solution optimale selon une fonction de longueur non linéaire. TAMCRA [59] est une version heuristique de SAMCRA ne gardant que k (un entier prédéfini) chemins faisables et non dominés en chaque noeud. Bien que SAMCRA et ses algorithmes dérivés soient efficaces pour la recherche de chemin multi-contraint,

ils ne considèrent qu'un contexte de domaine unique et doivent être adaptés au contexte inter-domaine.

3.3.1 Le choix de la séquence des domaines considérés

Les problèmes MCP et MCOP sont plus compliqués à résoudre en inter-domaine. Du fait des propriétés d'autonomie, de confidentialité et de scalabilité, la recherche de chemin avec QoS garantie en inter-domaine doit être distribuée. Pour des raisons de simplicité, nous savons supposé que le calcul du chemin vient après une étape préliminaire qui détermine la séquence des AS traversés. Cette hypothèse est un compromis entre la qualité et la complexité de la solution globale. Le calcul de la séquence des domaines et le calcul du chemin lui-même peuvent être considérés comme un problème d'optimisation conjoint. Cependant cela nécessite de tenir compte à la fois des contraintes du calcul de la séquence des domaines (principalement soumis à des contraintes politiques et commerciales) et du calcul de chemins (soumis à des contraintes de performance), augmentant la complexité du calcul. Pour simplifier le problème, nous calculons la séquence des domaines au préalable même si cela ne garantit pas d'atteindre des solutions globalement optimales.

3.3.2 La division des problèmes MCP et MCOP en inter-domaine

Nous avons appliqué les principes de SAMCRA au calcul de MCP inter-domaines : chaque domaine calcule des chemins faisables non dominés depuis ses nœuds d'entrée jusqu'à la destination de la requête. Cela nous permet de transformer le problème Inter-MCP en un problème MCP spécifique pour chaque domaine traversé.

La figure 3.2 représente la division en deux parties des problèmes MCP et MCOP en inter-domaine : un calcul des chemins possibles interne à chaque AS (intra-domaine) puis une phase de propagation et de sélection des chemins possibles de bout-en-bout (inter-domaine).

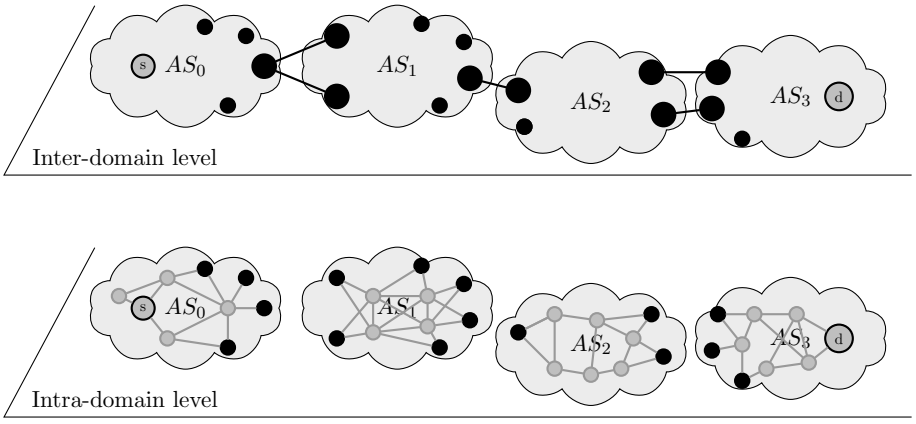


FIGURE 3.2 – Décomposition des niveaux inter- et intra-domaine

Dans **le problème intra-domaine**, les entités de calcul se basent sur leur vision de l'état du domaine afin de trouver un chemin entre deux nœuds de l'AS (des nœuds de bordure pour les AS de transit) qui satisfasse des contraintes données tout en optimisant une fonction objectif.

Le problème inter-domaine repose sur une phase de propagation et de combinaison des représentations abstraites des chemins calculés dans la phase intra-domaine. Les

AS y sont vus comme des boites noires interconnectées par des nœuds de bordure et disposant d'une entité de calcul capable d'effectuer des calculs de chemins complexes et de communiquer avec les entités de calculs d'autres AS.

3.3.3 Approches en ligne (online) ou autonome (offline)

Nous avons proposé deux approches, dites en ligne (online) et autonomes (offline), pour résoudre le problème des MCP inter-domaines.

L'approche de **calcul en ligne** consiste à servir les demandes de calcul de chemin à la volée. L'avantage de cette approche est que pour répondre aux demandes, les entités de calcul de chemins utilisent les informations les plus récentes de l'état du réseau. Cependant, ces entités sont sensibles au facteur d'échelle (en particulier aux arrivées fréquentes de demandes de calcul).

Alternativement, avec l'approche de **calcul autonome**, chaque domaine calcule de façon périodique une représentation agrégée de son état : un ensemble de portions de chemins qui permet aux domaines voisins de calculer efficacement des chemins de bout-en-bout. L'avantage d'une approche autonome est que les opérations de calcul de la représentation du domaine ne sont pas répétées pour chaque requête ; ainsi, le délai total d'établissement du chemin est réduit. Cependant, le calcul du chemin peut s'appuyer sur une connaissance obsolète de l'état du réseau.

Ces approches distribuées reposent sur trois blocs de base :

1. la formulation d'un problème par domaine,
2. un algorithme qui résout le problème par domaine,
3. une méthode de propagation et de combinaison des résultats de calcul par domaine pour déterminer le cheminement de bout-en-bout.

Nous avons formulé le problème inter-domaine en une combinaison de problèmes locaux par domaine dans le but de répartir les opérations de calcul du chemin entre les domaines traversés. Dans cette décomposition, l'efficacité des chemins inter-domaines calculés dépend du résultat des calculs locaux. Ainsi, la conception des problèmes par domaine est guidée par la recherche de portions de chemin qui mènent à une solution inter-domaine efficace sans violer les propriétés de confidentialité et de scalabilité. Par conséquent, le processus de calcul intègre une méthode d'évaluation de la qualité des chemins calculés par chaque domaine reposant sur le concept de dominance. Cela permet de simplifier les opérations de calcul en écartant très tôt des portions de chemin ne pouvant contribuer à la solution finale.

3.3.4 Problème intra-domaine

Les travaux antérieurs sur le problème des MCP intra-domaine [125] montrent que pour calculer les MCP optimaux, les nœuds intermédiaires ne devraient mémoriser que les chemins intermédiaires non dominés réalisables. En particulier, ce résultat nous permet de formuler les calculs MCP inter-domaines sous forme de problème par domaine : chaque domaine doit calculer un ensemble bien choisi de chemins non dominés réalisables pour permettre la sélection de MCP inter-domaines efficaces. Nous avons proposé deux formulations différentes du problème par domaine.

Nous avons proposé un nouvel algorithme de Dijkstra inverse (Reverse Dijkstra's Algorithm (RDA)) étendant [48] qui résout les problèmes de MCP par domaine en mémorisant tous les chemins non dominés réalisables à partir de plusieurs sources jusqu'à une destination. Dans l'approche en ligne, les routes faisables doivent être transmises de la destination

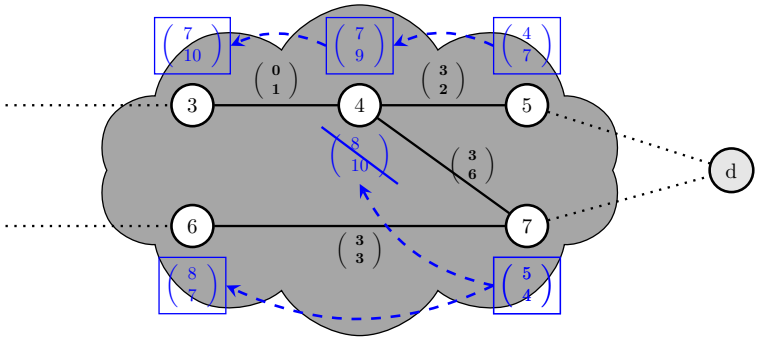


FIGURE 3.3 – Calcul de chemin au sein d'un domaine

vers la source, ce qui justifie la nature inverse de l'algorithme. La figure 3.3 donne un aperçu simplifié des opérations de notre algorithme : le but est de trouver les chemins réalisables non dominés dans le domaine entre les nœuds d'entrée 3 et 6, et les nœuds de sortie 5 et 7. Les contraintes considérées sont les suivantes : $W_1 \leq 15$ et $W_2 \leq 18$.

L'algorithme utilise une structure de file d'attente qui contient les chemins les plus courts de chaque nœud intermédiaire. Supposons que la file d'attente soit initialisée avec deux chemins des nœuds 5 et 7 jusqu'à une destination d extérieure au domaine. Notre RDA étendu exécute une boucle. Lors de chaque itération, il choisit un chemin p de la file d'attente dont les poids sont les plus éloignés des contraintes. L'algorithme relâche p, ce qui signifie qu'il évalue les poids des chemins des nœuds voisins de la source qui ont p comme suffixe. L'algorithme ajoute les chemins découverts à la file d'attente s'ils sont réalisables et ne sont dominés par aucun chemin déjà présent dans la file d'attente, puis il élimine de la file d'attente tout chemin dominé par l'un des nouveaux chemins. La boucle s'exécute tant que la file d'attente contient au moins un élément qui n'a pas été relâché ou jeté. Puisque l'algorithme n'écarte aucune voie faisable non dominée, il garantit de trouver les solutions aux problèmes de MCP par domaine, si elles existent. Sa terminaison est garantie parce que l'algorithme relâche un chemin à chaque itération de la boucle et que le nombre de chemins dans un domaine est fini. A la fin de l'itération, l'algorithme a trouvé les chemins non dominés réalisables à partir des nœuds d'entrée 3 et 6.

3.3.5 Problème inter-domaine : propagation et combinaison des résultats

La procédure de calcul du chemin inter-domaine propage et combine les segments résultant du calcul intra-domaine de chaque AS, pour constituer l'ensemble des chemins possibles de bout-en-bout pour une requête spécifique. La figure 3.4 illustre les opérations des procédures de propagation et de combinaison mettant en œuvre l'algorithme Backward Recursive PCE-based Computation (BRPC). Le domaine de destination annonce au domaine amont un ensemble de chemins vers la destination et leurs performances. Chaque domaine analyse l'information propagée pour déterminer les chemins qu'il annoncera aux domaines voisins en amont. Cette opération consiste à combiner les résultats du problème par domaine et les chemins annoncés par les domaines en aval. Enfin, le domaine source détermine les chemins de bout-en-bout réalisables pour sa demande grâce aux annonces de chemins reçues.

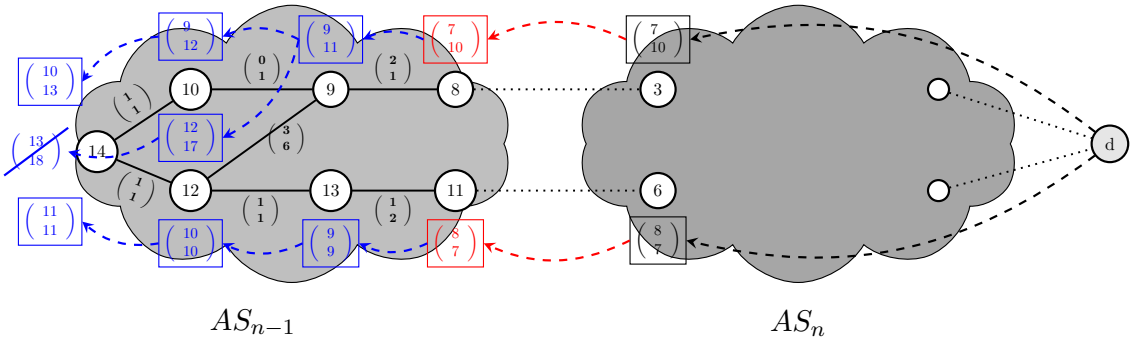


FIGURE 3.4 – Calcul de chemin inter-domaine avec $W_1 \leq 15$ et $W_2 \leq 18$

3.4 Des algorithmes de calcul des chemins en inter-domaine lorsque la suite de domaines est donnée

3.4.1 ID-MCP : une solution exacte offline

Nous avons tout d'abord élaboré un algorithme capable de fournir une solution exact au problème MCOP. Dans [7], nous avons proposé ID-MCP (Inter Domain MCP), le premier algorithme capable de résoudre de façon exacte et distribuée le problème MCP en inter-domaine quel que soit le nombre de contraintes considéré. Notre solution définit les calculs par domaine du problème MCP inter-domaine pour chaque domaine versé. Le résultat de ces calculs par domaine est un ensemble de chemins que notre algorithme utilisera pour trouver un chemin de bout-en-bout réalisable. De ce fait, nous transformons le problème Inter-MCP en un problème MCP spécifique pour chaque domaine traversé. Pour les opérations par domaine d'ID-MCP, nous avons utilisé RDA pour mémoriser tous les chemins non dominés réalisables à moyen terme.

Par construction, ID-MCP offre des garanties de performance prouvables. En particulier, il garantit de trouver un chemin satisfaisant les contraintes de la demande si un tel chemin existe. De plus, comme ID-MCP calcule tous les chemins faisables non dominés, il offre à la source l'opportunité de sélectionner le chemin qui fournit la plus grande marge de performance par rapport aux contraintes de la demande si elle le souhaite. Cette fonction est intéressante pour maximiser les chances qu'un chemin calculé puisse être configuré avec succès, car l'état du réseau peut changer entre le calcul d'un chemin et sa configuration.

Bien que la complexité de cet algorithme reste raisonnable sur des topologies assez petites (25 nœuds), la complexité des calculs MCP à l'intérieur d'un domaine dépend du nombre maximum de chemins mémorisés pour un seul nœud. Nous avons proposé plusieurs stratégies pour améliorer les temps d'exécution de notre algorithme. La première solution, pID-MCP utilise la notion de pré-calcul pour simplifier le travail de l'algorithme, puis nous avons proposé une solution heuristique (kID-MCP) et une solution approchée (aID-MCP) présentées dans les sections suivantes.

3.4.2 Des algorithmes pour une solution plus rapide

pID-MCP : une solution exacte avec précalcul

Le pré-calcul consiste à préparer à l'avance des chemins ou des portions de chemins qui peuvent être utilisés plus tard pour chercher des chemins de bout-en-bout. Cela permet de réduire la charge de calcul sur les éléments du réseau et les temps de réponse des algorithmes de recherche de chemins en réutilisant les chemins pré-calculés mais aussi

d'améliorer la fiabilité du réseau en pré-calculant les chemins de détours à utiliser en cas de panne.

Dans un schéma de pré-calcul, le routage avec qualité de service est effectué en deux phases : la première phase, exécutée hors ligne, calcule à l'avance des chemins faisables pour différentes classes de service. Les résultats des calculs obtenus (qui peuvent être vus comme des offres globale d'acheminement offertes par l'AS) sont stockés dans une base de données pour une utilisation ultérieure. La seconde phase intervient lorsqu'une demande de QoS arrive, il suffit de choisir l'une des solutions pré-calculées disponibles dans la base de donnée. Par exemple, lorsque l'on traite des demandes de qualité de service avec des contraintes de délai, la première phase peut pré-calculer des chemins réalisables pour un large éventail de contraintes de délai possibles, tandis que la deuxième phase doit simplement sélectionner un chemin approprié à partir de l'ensemble pré-calculé, c'est-à-dire trouver la concaténation de portions pré-calculées dont la somme des délais partiel reste inférieur à la contrainte. Cependant, d'autres calculs peuvent être effectués pour considérer des métriques plus précises. Le temps d'exécution de la deuxième phase a un impact immédiat sur les performances de la solution ; il est donc hautement souhaitable de maintenir sa complexité de calcul aussi faible que possible.

Le pré-calcul a été introduit pour le routage QoS par Orda et Sprintson dans [88] mais plusieurs algorithmes ont été proposés dans la littérature. Les stratégies de pré-calcul jouent en général sur la façon de considérer les multiples contraintes, soit en les traitant indépendamment, soit en les combinant en une métrique unique et ainsi pouvoir appliquer un algorithme de calcul plus simple. Par exemple, CDP (Clustering-based distributed pre-computation for Quality of Service routing) pré-calcule un ensemble de chemins non dominés en utilisant une technique de partitionnement (voir [56]) pour diminuer la taille de la table de routage. Dans [57], MEFPA (Multi-constrained Energy Function-based Pre-computation Algorithm) pré-calcule plusieurs plus courts chemins à partir de chaque nœud dans le réseau en minimisent respectivement les différentes combinaisons de poids des liens dans une fonction de longueur linéaire. NM-MCP (Normal Measure-based Multiple Constrained Path) [121] pré-calcule k chemins primaires à l'avance, où k est le nombre de poids des liens.

En outre, grâce à la propriété d'autonomie, chaque domaine choisit les algorithmes utilisés en interne, y compris pour le pré-calcul. Il était donc particulièrement important de concevoir un mécanisme MCP avec pré-calcul polymorphe pouvant opérer dans une situation où les domaines peuvent utiliser des algorithmes de pré-calcul hétérogènes.

Dans [4], nous avons proposé pID-MCP, qui correspond à ID-MCP avec du pré-calcul et kpID-MCP, une version heuristique de pID-MCP qui, inspiré de TAMCRA, ne mémorise que k chemins non dominés à chaque nœud du domaine. Lors de la réception d'une demande, la combinaison des segments de chemin pré-calculés est effectuée en ligne *de manière ordonnée ou indépendante*. La combinaison ordonnée progresse de la destination vers la source. Dans la stratégie indépendante, chaque domaine peut combiner ses données pré-calculs à n'importe lequel de ses voisins, obtenant ainsi une structure agrégée. Le pré-calcul et la sélection de chemin de bout-en-bout sont donc totalement décorrélés . Ainsi, les pré-calculs effectués dans les domaines sont indépendants de la séquence du domaine : c'est l'une des contributions de ce travail qui renforce l'autonomie des domaines. Des domaines ayant des accords plus forts (comme une alliance, *cf.* section 3.5.1) peuvent adopter une stratégie de collaboration plus poussée et pré-calculer une combinaison indépendante de leurs portions de chemins pour cacher leur topologie et être vus comme un seul domaine.

kID-MCP : la version heuristique de ID-MCP

L'algorithme kID-MCP [6] est une adaptation de TAMCRA. Il ressemble à ID-MCP sauf que, pour chaque nœud, on ne mémorise au maximum que k éléments attachés à ce nœud dans la file d'attente de calcul. kID-MCP réduit à la fois la complexité de l'algorithme et la surcharge de signalisation introduite par les échanges d'informations entre domaines en bornant à k le nombre de MCP retenus pour chaque nœud jusqu'à la destination. En, particulier, cela permet de réduire le nombre de chemins communiqués aux autres domaines pour chaque nœud d'entrée.

Pour choisir les chemins qui sont retenus, kID-MCP considère la longueur de chemin non linéaire $c(p)$ (voir l'équation 3.1). L'heuristique kID-MCP représente des cas intermédiaires entre l'algorithme mémorisant chaque chemin faisable non dominé (ID-MCP) et kID-MCP, k=1 (au maximum un chemin est mémorisé pour chaque noeud).

L'évaluation analytique et l'étude des simulations présentées dans [6] mettent en évidence l'excellente performance de cette proposition : kID-MCP trouve des chemins satisfaisant les contraintes dans la plupart des situations et passe bien à l'échelle. Cela montre que de tels algorithmes (ayant une complexité raisonnable) peuvent être utilisés pour déterminer des MCP inter-domaines dans la pratique, sans sacrifier la QoS des chemins, ce qui ouvre des perspectives intéressantes pour le routage QoS inter-domaine et l'ingénierie de trafic.

aID-MCP : l'approximation de ID-MCP

L'heuristique kID-MCP permet de rechercher plus rapidement des solutions au problème MCP mais ne donne aucune indication sur la qualité des solutions éventuellement délivrées. Afin d'avoir un meilleur compromis entre la vitesse de réponse de l'algorithme et la qualité des solutions trouvées, nous avons proposé aID-MCP, un algorithme d'approximation qui fournit une solution proche de l'optimum. Or, pour que notre algorithme soit une approximation $(1 + \epsilon)$, il doit :

1. garantir que, parmi les chemins réalisables, la valeur de la fonction objectif pour le chemin retourné ait un facteur d'au plus $(1 + \epsilon)$ avec la valeur optimale de cette fonction,
2. borner son temps d'exécution par un polynôme de l'ordre de la taille d'encodage de l'instance de problème.

En particulier, nous nous sommes intéressés aux Fully Polynomial-Time Approximation Schemes (FPTASs) car ils fournissent des $(1 + \epsilon)$-approximations dont la complexité temporelle est bornée par un polynôme de l'ordre de la taille de l'instance du problème et de $\frac{1}{\epsilon}$, pour tout $\epsilon \in \mathbb{R}$. Or dans [120], Yuan montre que les instances de problème MCP avec une valeur réelle et $K - 1$ entier peuvent être résolues en temps polynomial. Ce résultat a été repris par la communauté et a donné naissance à des techniques pour transformer des instances de problèmes généraux de MCP en instances spécifiques avec des poids entiers.

Nous avons adapté les techniques d'approximation qui ont été développées pour le problème MCP [67] au problème MCP en inter-domaine. Nous avons défini le problème par domaine et l'objectif d'approximation pour le problème de bout-en-bout.

Approximation du le problème par domaine Nous avons choisi d'utiliser la méthode de *rounding and scaling* pour réduire la complexité de des calculs par domaine en mettant à l'échelle et en arrondissant les poids des liens et les contraintes grâce à un paramètre $\theta \in \mathbb{R}$ qui détermine la précision de l'approximation. Ainsi le problème de MCP devient un problème de Mcpp (Multi-Constrained Path with Positive rounding). MCP et Mcpp

ne sont pas équivalents, la résolution du problème de Mcpp au lieu du problème original introduit une perte de performance quantifiable qui dépend de la valeur du paramètre de transformation θ.

Cette formulation par domaine permet de trouver un chemin inter-domaine réalisable qui vérifie la première contrainte W_1 et qui minimise $\max_{k\in[2..K]} w_k(p)$.

Approximation de la recherche de chemin contraints de bout-en-bout En nous appuyant sur le problème d'approximation par domaine, nous avons proposé un algorithme d'approximation de recherche de chemins de bout-en-bout soumis à des contraintes multiples en inter-domaine nommé aID-MCP. Les détails de la démarche (les pseudo-codes, analyses et preuves) sont présentés dans le manuscrit de thèse de Gilles Bertrand [51].

Evaluation des algorithmes

Nous avons validé ces approches par simulation. Dans un premier temps, nous avons comparé les résultats des algorithmes pID-MCP, kID-MCP, kID-MCP, k=1 et kpID-MCP ($k > 1$) à ceux de l'algorithme exact ID-MCP sur plusieurs topologies (voir le détail dans la thèse de Gilles Bertrand [51]). Les résultats présentés ici ont été obtenus sur les topologies LaticeSL (voir figure 3.5) avec une distribution uniforme des poids de lien intra-domaine et inter-domaine avec $10 \leq w_k \leq 1023$, $k = 1..K$ ou $k = 1..K-1$, et LaticeFM (Full mesh) qui reprend les trois domaines de LaticeSL mais dont chaque nœud d'un domaine est connecté à chaque nœud du domaine suivant et du domaine précédent de la chaîne. Les topologies Latice introduisent une grande diversité de chemins, pénalisant les algorithmes exacts en augmentant significativement leur complexité temporelle. En particulier, dans la topologie LatticeFM(25,3) le nombre de liens inter-domaines est extrêmement important (chaque domaine est connecté par 625 liens inter-domaines). Cette topologie a été conçue pour illustrer un inconvénient des algorithmes basés sur pID-MCP : le fait qu'ils doivent mémoriser dans ce cas un grand nombre de chemins.

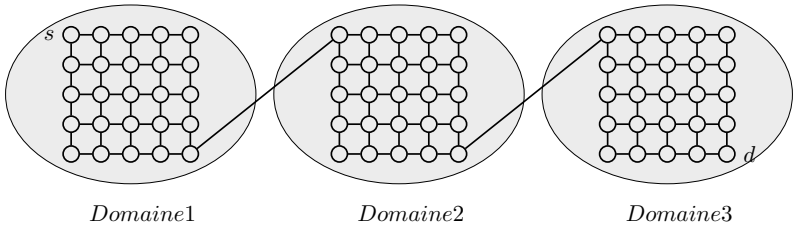


FIGURE 3.5 – Topologie fictive LaticeSL(25,3) comportant 3 domaines identiques de 25 nœuds

Les résultats ont montré l'efficacité de nos algorithmes mais aussi que différents opérateurs peuvent utiliser des algorithmes de pré-calcul différents sans remettre en question le calcul de bout-en-bout du chemin.

Le taux de réussite absolu (Absolute Success Rate - ASR) correspond au pourcentage de réussite des algorithmes pour trouver un chemin réalisable lorsqu'une solution existe (Comme ID-MCP et pID-MCP sont exacts, leur ASR est de 100%). Le taux de réussite (Success Rate - SR) est le pourcentage de réussite des algorithmes pour trouver un chemin réalisable pour les demandes considérées. Lorsque le nombre de demandes simulées n'est pas fourni explicitement, cela signifie qu'il est suffisamment important pour fournir des résultats statistiquement significatifs. Le nombre de chemins retournés par les algorithmes est indiqué par #P. Les deux algorithmes exacts retournent tous les chemins non dominés

réalisables de bout en bout, donc #P(ID-MCP)=#P(pID-MCP). Le nombre de chemins retournés par kpID-MCP peut dépasser k, car les segments calculés par un domaine sont combinés à ceux des domaines en aval. Notre implémentation de kID-MCP, $k = 1$ retourne un chemin unique par requête. Le coût (C) est la valeur la plus faible de la fonction de longueur de chemin (selon la métrique de SAMCRA) parmi les chemins calculés, donc, il prend la même valeur pour tous les algorithmes exacts. Nous définissons une fonction de longueur de chemin supplémentaire c' comme $\mu_{i=1..K}\left(\frac{w_i}{W}\right)$, où μ désigne l'opérateur arithmétique moyen. Nous appelons coût multidimensionnel (MC) la valeur de c' pour le chemin de bout en bout calculé avec la valeur la plus basse de c'. MC permet d'évaluer la qualité des trajectoires retournées en considérant l'ensemble des métriques, alors que C indique leur qualité par rapport à la métrique la plus restrictive. Les coûts (C et MC) des chemins ne sont pris en compte que pour les requêtes pour lesquelles heuristiques et algorithmes exacts parviennent à trouver un chemin réalisable, de sorte que la comparaison des algorithmes est significative.

Nous dérivons la complexité temporelle relative des algorithmes de la mesure du nombre maximum (α) de chemins reliés à un nœud et mémorisés dans la file de calcul. La valeur α donne également une indication de la complexité spatiale des algorithmes. Par définition, α(kID-MCP, k=1) = 1.

	Avec des contraintes lâches $(49100, 49100)^T$ pour SL et $(3000, 3000)^T$ pour FM									
	SR [%]		C [%]		MC [%]		α		#P	
Lattice	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*
ID-MCP	100	100	19.2	13.9	18.7	11.4	10	7	7	3
pID-MCP	100	100	19.2	13.9	18.7	11.4	5	52	7	3
kpID-MCP, k=3	100	100	19.3	26.8	18.8	23.8	3	3	6	1
kID-MCP, k=1	100	100	19.5	13.9	19	11.9	1	1	1	1
	Avec des contraintes strictes $(9800, 9800)^T$ pour SL et $(400, 400)^T$ pour FM									
	SR [%]		C [%]		MC [%]		α		#P	
Lattice	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*	*SL*	*FM*
ID-MCP	66	56	89.3	72	86.5	60.1	8	2	5	1
pID-MCP	66	56	89.3	72	86.5	60.1	5	8	5	1
kpID-MCP, k=3	66	50	89.4	72	86.5	60.1	3	3	4	1
kID-MCP, k=1	64	56	89.8	72	87.9	60.1	1	1	1	1

TABLE 3.1 – Résultats des simulations pour les topologies LatticeSL et LatticeFM

Le Tableau 3.1 presente les résultats de simulation pour les topologies LatticeSL(25,3) (SL) et LatticeFM(25,3) (FM) avec des contraintes lâches ou strictes. Ces résultats illustrent l'inconvénient de pID-MCP lorsque la connectivité inter-domaines est importante. Comme prévu, dans la topologie LatticeSL α pour ID-MCP est supérieur à α pour pID-MCP, alors que dans la topologie LatticeFM α pour ID-MCP est beaucoup plus bas que pour pID-MCP. Ceci souligne la nécessité de limiter α dans l'algorithme pID-MCP et justifie l'heuristique kpID-MCP. Dans la topologie LatticeSL kpID-MCP, k=3 donne de meilleurs résultats (valeur inférieure de C et MC) que kID-MCP, k=1. Cependant, on constate l'inverse pour la topologie LatticeFM. Ce problème est résolu en permettant des valeurs plus grandes de α dans kpID-MCP.

3.4.3 Discussion de l'hypothèse d'une suite d'AS connue au préalable

Dans cette section, nous avons vu un ensemble d'algorithme permettant de trouver des chemins respectant des contraintes multiples de bout-en-bout en inter-domaine. Bien que le problème soit NP-complet, nous avons proposé un algorithme exact exécuté offline (ID-MCP) et une variante plus rapide utilisant du pré-calcul (pID-MCP), une solution approchée (aID-MCP) et deux heuristiques (kID-MCP et kpID-MCP).

Un des enjeux de la recherche de chemin en inter-domaine est de limiter la suite de domaines interrogés et d'obtenir au moins un chemin possible dans un temps raisonnable. Cela nécessite un compromis entre le nombre de domaines explorés, la rapidité des algorithmes et la qualité des chemins obtenus. Nos algorithmes (comme ceux de la littérature) ont en commun la nécessité de connaître à priori la suite des domaines impliquée dans le chemin. S'il semble raisonnable dans un premier temps de choisir le plus court chemin d'AS entre une source et une destination pour trouver une solution, il n'est pas forcément le meilleur pour garantir de la QoS. Ceci nous a poussé à relâcher cette hypothèse et à considérer que la suite des domaines impliqués n'est plus connue.

3.5 MCP lorsque la suite de domaines considérée n'est pas fixée

Se limiter à la recherche de chemins pouvant garantir une QoS demandée en explorant uniquement les AS traversés par le plus court chemin entre une source et une destination pose plusieurs problèmes. Premièrement, si les politiques d'interconnexion le permettent, cette suite d'AS est sollicitée pour la plupart des flux entre ces deux nœuds du réseau. Deuxièmement, certaines applications ont besoin d'utiliser plusieurs chemins, disjoints afin de résister aux pannes. Relâcher la contrainte sur la prédétermination de la suite d'AS à traverser peut aider à trouver des chemins moins chargés (disjoints ou non) ou ayant les ressources disponibles nécessaires pour répondre à une demande de transmission avec une QoS garantie. Explorer l'ensemble des AS composant l'Internet permettrait de trouver le meilleur chemin pour une fonction objectif donnée mais ne passe pas à l'échelle. Il faut donc un compromis entre la qualité de la solution et le nombre de domaines sollicités.

Nous avons mené des travaux en ce sens lors de la thèse de Romain Jacquet. Le but de la thèse était de proposer une méthode pour déterminer une suite d'AS pertinente à interroger pour trouver des chemins respectant des contraintes de QoS. Pour des raisons évidentes de passage à l'échelle, il est exclu de solliciter l'ensemble des AS qui composent l'Internet. Il s'agit donc déterminer un sous-ensemble des AS de l'Internet qui sera considéré lors de la recherche de chemin. Nous verrons dans la section suivante que nous avons cherché à étendre la zone de recherche dans un voisinage déterminé du plus court chemin pour augmenter le nombre de chemins potentiels.

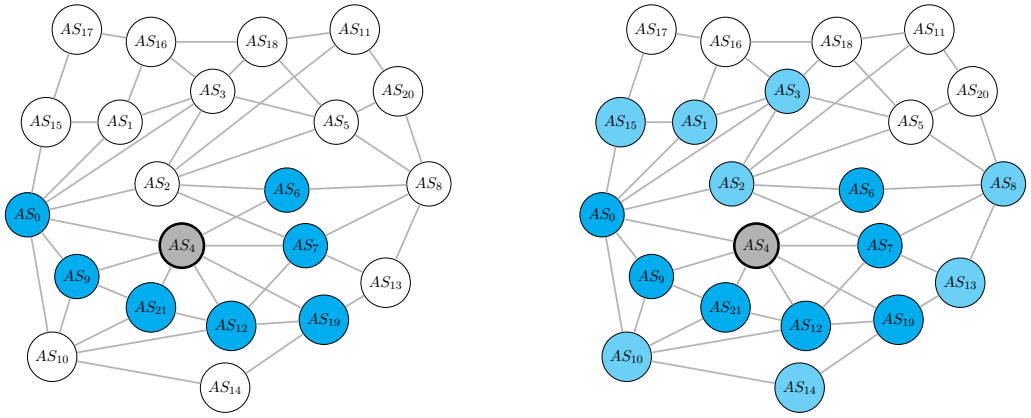
3.5.1 MCP basé sur la notion de voisinage en inter-domaine

Nous avons proposé SANP (Sub-Graph Algorithm for finding feasible Non dominated Path) [23], un algorithme qui peut résoudre le problème général de MCP en respectant la confidentialité et l'autonomie des AS alors que la séquence des AS impliqués dans le calcul n'est pas connue. Il construit un sous-graphe d'AS autour de la route entre la source et la destination sélectionnée par les protocoles de routage. Ce sous-graphe est utilisé pour calculer un ensemble de chemins non dominés réalisables entre la source et la destination. Pour résoudre les problèmes de scalabilité qu'un tel mécanisme peut comporter, nous avons

mis en place deux heuristiques qui limitent la taille du sous-graphe. Lors de l'évaluation de SANP, nous avons comparé les chemins trouvés avec tous les chemins faisables non dominés qui existent dans le graphe. Nos simulations montrent que SANP trouve un nombre raisonnable de chemins proches de la solution optimale.

Le principe de SANP est d'explorer un espace plus grand que la suite d'AS traversés par le plus court chemin entre la source et la destination. Le choix des AS impliqués dans le calcul repose sur la notion de voisinages combinés dans un sous-graphe qui sera exploré par SANP (voir la figure 3.6).

Définition 2. *Définition de la notion de voisinage*
Soit $G = (V, E)$ un graphe d'AS, le voisinage du nœud i est l'ensemble $N_i = \{V_i' \cup i, E_i'\}$ où $V_i' = \{x | d(i,x) \leq r_0, x \in V\}$ et $E_i' = \{e | e(j,k) \in E, j \in V_i', k \in V_i'\}$ où $d(i,j)$ est la longueur (nombre de sauts) du chemin le plus court entre les nœuds i et j, et $e(j,k)$ est le lien entre les nœuds j et k.



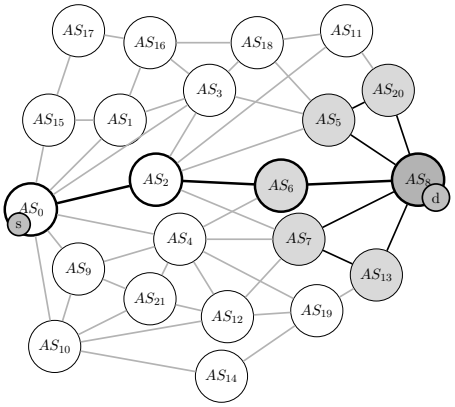
avec $r_0 = 1$ avec $r_0 = 2$

FIGURE 3.6 – Voisinages de l'AS_4

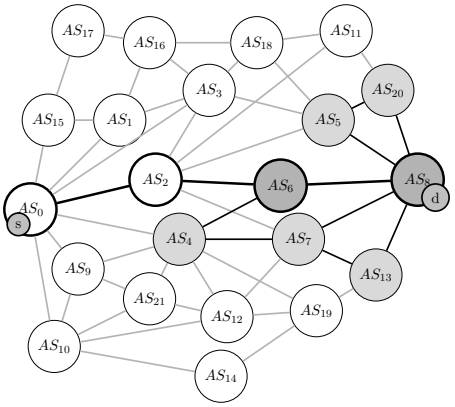
Chaque nœud (représentant maintenant un AS) peut aisément calculer son voisinage à l'aide de plusieurs techniques, par exemple par une inondation limitée de l'information de l'état de lien. L'algorithme SANP suppose que tous les nœuds connaissent et mettent à jour régulièrement leur voisinage respectif. Il suppose également que le réseau sous-jacent est capable d'acheminer les paquets vers n'importe lequel des nœuds du réseau.

Fonctionnement de l'algorithme SANP

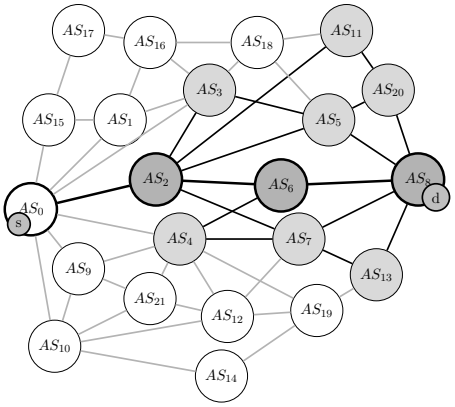
Un nœud source s souhaitant établir un chemin avec des garanties QoS avec un nœud destination d envoie une requête à d. d envoie alors un message contenant le sous-graphe de son voisinage vers s. Lorsqu'il progresse dans le réseau (en suivant le chemin dicté par les protocoles de routage), chaque AS traversé fusionne son propre voisinage avec le sous-graphe véhiculé par le message. Ainsi, le sous-graphe s'étoffe jusqu'à ce que le message atteigne la source. A ce stade, la source dispose donc d'un sous-graphe contenant à la fois elle-même et la destination. s peut alors calculer un ensemble de chemins faisables non dominés dans ce sous-graphe grâce à un algorithme, par exemple un des algorithmes présentés dans la section 3.4. La figure 3.7 montre un exemple de construction du sous-graphe pour chercher un chemin de s à d quand le rayon $r_0 = 1$.



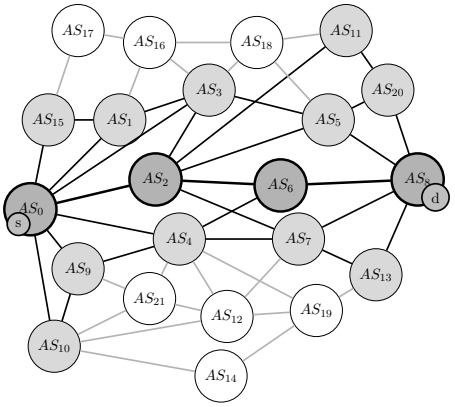
Voisinage de d



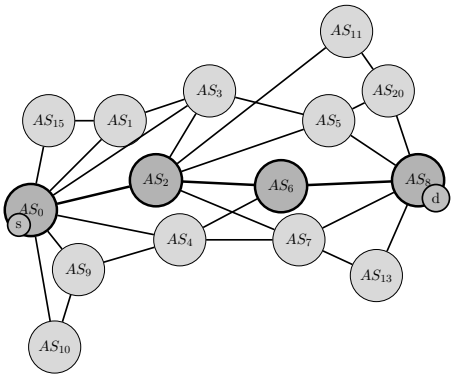
intégration du voisinage de l'AS_6



intégration du voisinage de l'AS_2



intégration du voisinage de l'AS_0



Sous-graphe utilisé par l'algorithme MCP

FIGURE 3.7 – Construction du sous-graphe des voisinages entre d et s avec $r_0 = 1$

Nous avons évalué les performances de SANP par simulation. [23] présente l'approche SANP et montre qu'il est toujours capable de trouver des chemins non dominés, lorsqu'ils existent. Il est possible d'augmenter le nombre et la qualité des chemins trouvés par SANP en augmentant le rayon des voisinages utilisé pour calculer le sous-graphe, mais même pour des valeurs raisonnablement petites du rayon (4 ou 5) les résultats sont assez bons. Dans le cadre de la thèse de Romain Jacquet [74], nous avons également développé des heuristiques pour limiter la taille du sous-graphe en bornant le nombre de nœuds gardés à chaque étape. Nous avons proposé plusieurs méthodes pour choisir les nœuds à garder dans

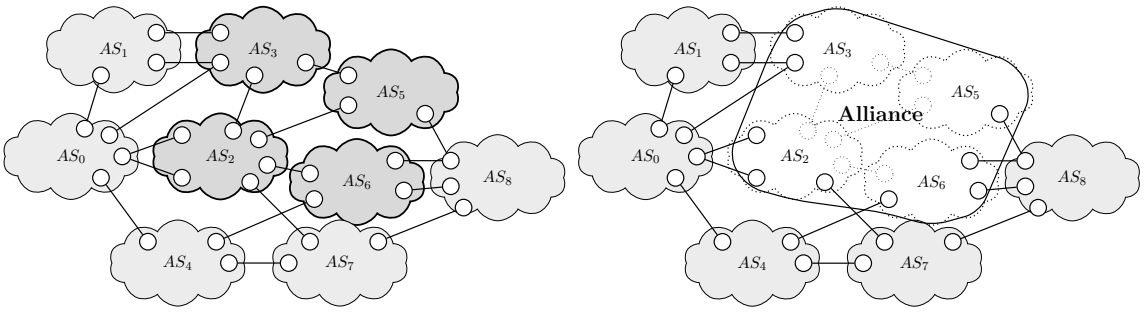
le sous-graphe, par exemple en ne gardant que les nœuds de plus haut degré (l'intuition étant qu'ils apportent plus de diversité que des nœuds moins bien connectés).

Recherche de chemins à QoS garantie en présence d'alliances d'AS

Le nombre d'AS dans l'Internet est très important et ne cesse de croître, de plus certains d'entre eux ont une couverture géographique non disjointe. Il est donc concevable qu'un certain nombre d'AS décide de former des alliances (ou des fédérations) pour mieux répondre à des demandes de chemins à QoS garantie. Plusieurs travaux ont déjà proposé différentes variantes d'un tel système [49][84][82][96][95]. Nous définissons une alliance comme un groupe d'AS qui se font confiance et qui acceptent de partager des politiques commerciales et/ou techniques, à l'instar de ce qui a été proposé par le projet ETICS [3] (Economics and Technologies for Inter-Carrier Services). Étant donné le nombre d'AS dans l'Internet, nous supposons qu'ils créeront plus d'une alliance, chaque AS n'appartenant qu'à une seule alliance et certains AS restant indépendants. Les offres publiées par une alliance sont un exemple d'association collaborative (*"open association"*) telle que définie par le projet ETICS. Une offre d'alliance est la composition (ou la juxtaposition) de plusieurs offres AS au sein de l'alliance. Nous avons considéré que l'alliance dispose de plusieurs possibilités d'instancier ses offres, donc chaque offre est valable pour une période assez longue (au moins de l'ordre de quelques jours, si ce n'est des semaines ou des mois) et son coût peut être calculé à l'avance.

Contrairement aux travaux précédents qui traitent de la recherche de chemins de QoS au sein d'une alliance unique, nous avons proposé ACQA, une extension de l'algorithme SANP, capable de trouver des chemins de QoS de bout-en-bout impliquant plusieurs AS et/ou alliances.

Représentation des alliances



Graphe d'AS — Alliance des AS AS_2, AS_3, AS_5, AS_6

FIGURE 3.8 – Représentation de la topologie des AS avec alliance

Nous avons repris la représentation d'un AS par un graphe dans lequel toutes les interfaces de ses nœuds de bordure sont inter-connectées. Ces liens intra-domaine représentent les offres (Service Level Agreement - SLA) publiées par les AS (pour rappel, ils ne représentent pas la topologie interne de l'AS). Le graphe d'alliance $G_0 = (V_0, E_0)$ dans lequel les nœuds représentent soit des AS soit des alliances (voir la figure 3.8) est construit à partir du graphe $G = (V, E)$ représentant l'interconnexion des AS. Un nœuds de bordure d'alliance est un nœud appartenant à l'alliance et connecté à un nœud de bordure d'un AS extérieur à

3. `https://cordis.europa.eu/project/rcn/93071_en.html`

l'alliance. Nous représentons une alliance par un graphe maillant complètement ses nœuds de bordure. Comme les AS, les alliances publient des offres (offres d'alliance) qui décrivent ce que des clients peuvent demander pour faire transiter des flux entre deux nœuds de bordure de l'alliance.

Comparaison entre ACQA et SANP

ACQA reprend l'idée de base de SANP et construit un sous-graphe d'une région limitée autour d'un chemin d'AS (en général le plus court) de la destination à la source grâce à la notion de voisinage. [24] expose comment ACQA étend SANP pour pouvoir traiter les alliances et présente l'évaluation par simulation et la comparaison des deux algorithmes. Il montre qu'ACQA surpasse SANP car les chemins qu'il trouve ont tendance à dominer les chemins trouvés par SANP (en se basant sur la distance générationnelle (*generational distance*) [110] et la métrique zitzler [122] métriques). Les bonnes performances d'ACQA peuvent être justifiées par le fait que les alliances ont une coopération plus poussée que des AS indépendants. Ainsi, les AS de l'alliance sont capables de partager (dans une certaine mesure) plus d'informations sur leur topologie et leurs métriques. Cela permet de calculer des chemins plus précis et potentiellement d'offrir une plus grande diversité de chemins. Pourtant, cette solution est aussi scalable que SANP, étant donné que le nombre de nœuds dans le sous-graphe est du même ordre de grandeur dans ACQA et SANP. Quelle que soit la configuration (avec ou sans alliance), la source obtient un ensemble de chemins faisables capables de garantir les contraintes de QoS énoncées. Le choix du chemin retenu lui incombe. Maintenant que la séquence d'AS n'est pas fixée, la source obtient des chemins potentiels passant par des ensembles différents d'AS. Le choix peut s'appuyer sur des politiques de partenariat, de tarification ou sur tout autre indicateur. Lorsque l'on considère la garantie de QoS, il nous semble important de pouvoir avoir une estimation du sérieux des AS vis à vis du respect des garanties de service annoncées. Pour cela, nous avons travaillé sur un mécanisme de réputation donnant une indication sur le sérieux des AS et sur leur respect des garanties de QoS vendues par le passé.

3.5.2 Mécanisme de réputation pour le choix des AS impliqués dans le calcul de chemin

L'un des principaux obstacles au déploiement de garanties de qualité de service de bout-en-bout dans l'Internet est qu'il nécessite la coopération de systèmes autonomes. Souvent, les utilisateurs finaux ne sont pas connectés au même opérateur et tous les AS impliqués dans la communication entre ces deux parties doivent promettre et délivrer la QoS souhaitée. Les mécanismes que nous avons proposés dans ce chapitre se concentrent uniquement sur le calcul de chemins contraints en inter-domaines et supposent que chaque AS est prêt à contribuer au mieux de ses capacités pour acheminer le trafic. Cependant, la réussite de la sélection et de l'établissement d'un chemin ne signifie pas que les différentes entités impliquées sur ce chemin se comporteront comme annoncé, ou que la QoS promise sera effectivement livrée. Ces défaillances peuvent résulter soit de changements inattendus dans les réseaux sous-jacents (par exemple une défaillance de liaison ou une augmentation inattendue du trafic), soit de comportements malveillants (par des AS non coopératifs).

Une possibilité pour détecter les chemins peu fiables est de s'appuyer sur des mécanismes de réputation qui aident à identifier les AS qui violent souvent les garanties de QoS qu'ils promettent. Les mécanismes de réputation tendent à être un outil efficace pour encourager la confiance et la coopération dans les systèmes distribués dans lesquels les entités (c'est-à-dire les utilisateurs, les domaines, les opérateurs) ont un comportement ration-

nel [60]. En fournissant les moyens d'évaluer chaque entité impliquée, un mécanisme de réputation agrège ces notations et en déduit les scores de réputation publiquement disponibles calculés avec une fonction bien spécifiée. Ainsi, les entités qui acquièrent une bonne réputation sont celles qui fournissent des services corrects aux autres et qui honorent les engagements qu'elles ont pris. Des mécanismes de réputation ont déjà été proposés pour exploiter les connaissances locales d'un AS donné afin de déterminer s'ils peuvent faire confiance à leurs voisins [83], mais, dans la mesure de nos connaissances, aucun n'avait été proposé pour recueillir les connaissances du système sur les voies fiables - et non fiables.

Dans leur environnement hautement concurrentiel, il est possible qu'un ensemble non négligeable d'AS tentera de tromper ses clients, que ce soit par collusion ou non. Ils peuvent également essayer d'attirer plus de trafic qu'ils ne peuvent raisonnablement traiter, ou d'obtenir un type particulier de trafic. Lorsqu'un AS ne respecte pas ses engagements, il est important que les utilisateurs finaux soient conscients d'un tel échec afin de choisir des chemins de routage fiables vis à vis des promesses de QoS. Les utilisateurs préféreront très probablement éviter de traiter avec des AS qui ne peuvent pas livrer la QoS promise. Cependant, l'identification fiable et correcte des AS vertueux et des AS indélicats est un véritable défi. La raison en est que chaque AS est clairement incité à blâmer les autres afin d'éviter les conséquences négatives de ses échecs. Même si chaque AS correspond à une entreprise bien identifiée, des comportements indélicats existent. Par conséquent, il est obligatoire de fournir un mécanisme utilisable et capable de décourager un AS ou une collusion d'AS de promettre des garanties de qualité de service intéressantes sans les délivrer. Empêcher les AS de rejeter la responsabilité de leurs propres échecs sur les autres est une condition nécessaire pour un routage inter-domaines avec garantie de QoS.

Dans [1], nous avons présenté les avantages que les mécanismes de réputation peuvent offrir au routage inter-domaine avec QoS afin de produire un environnement où la confiance entre les AS et l'utilisateur final prospère. Nous avons identifié les propriétés requises lors de l'utilisation d'un tel mécanisme dans notre contexte, cartographié des briques de base et proposé un mécanisme de réputation pour le routage inter-domaines garantissant les exigences de qualité de service.

3.6 Conclusion

Nous avons vu qu'offrir de la qualité de service aux flux est un problème qui peine à trouver des solutions. Bien que des mécanismes de réservation de ressource et d'ingénierie de trafic existent, ils sont en général confinés à un AS. Or la garantie de QoS vis à vis d'applications est nécessaire de bout-en-bout. Cela requiert une coopération entre tous les AS impliqués sur le chemin des flux. Ainsi, un premier enjeu est de disposer à la fois de mécanismes intra- et inter-domaines qui respectent les propriétés de confidentialité, d'autonomie et de scalabilité inhérentes aux protocoles inter-domaines. De plus, se contenter de garantir les flux face à une métrique de QoS (délais, perte, bande passante, coût, ...) est réducteur, les opérateurs ont besoin de pouvoir prendre en compte plusieurs métriques en même temps, ce qui rend le problème extrêmement complexe. Dans ce chapitre, nous avons présenté les problèmes MCP (Multi-Constrained Path) et MCOP (Multi-Constrained Optimal Path) qui visent à calculer des chemins multi-contraints dans les réseaux. Nous avons proposé plusieurs algorithmes pour adapter ce problème au contexte inter-domaine et proposé un mécanisme en deux étapes qui sépare les actions à effectuer en intra-domaine et la phase de combinaison de ces calculs locaux pour obtenir un calcul inter-domaine. Nous avons proposé un algorithme exact, deux heuristiques et une approximation quand la suite des AS impliqués est connue à l'avance. Nous avons également proposé une adaptation de la

solution exacte et d'une heuristique dans le cas ou la suite des domaines n'est pas indiquée au préalable. Pour ne pas interroger tous les AS de l'Internet, ces solutions explorent un voisinage du plus court chemin d'AS entre la source et la destination du flux. Nous avons adapté ces propositions pour des AS indépendants ou regroupés sous forme d'alliances. Nous avons enrichi ce travail avec un mécanisme de réputation pour aider l'émetteur de la demande de chemin à choisir le chemin à établir (parmi les différentes propositions de chemins faisables) en se basant sur la faculté des AS à respecter leurs engagements par le passé. La combinaison de ces propositions offre un mécanisme global qui pourrait être utilisé avec les protocoles actuellement disponibles dans l'Internet. Cependant cela repose sur la capacité des AS à gérer efficacement la QoS à l'intérieur de leurs réseaux grâce à l'ingénierie de trafic. Dans le chapitre 4, nous montreront que l'évolution vers l'automatisation de la gestion des réseaux et leur virtualisation offre de nouvelles possibilités pour généraliser le recours à l'ingénierie de trafic mais que la dimension inter-domaine reste problématique et constitue une perspective de recherche à venir (voir le chapitre 5).

CHAPITRE 4

La qualité de service par ingénierie de trafic dans les réseaux programmables

Les nouvelles normes 5G qui émergent (telles que des débits de pointe élevés, un débit de données élevé pour l'utilisateur ou une latence très faible) obligeront sous peu les fournisseurs de service réseau à mettre en place un contrôle fin et dynamique des paramètres de QoS (bande passante, latence, perte ou gigue). Les réseaux traditionnels sont confrontés à de nombreuses difficultés pour tenter de répondre à ces besoins. Le protocole DiffServ est utilisé pour gérer la qualité de service des trafics en fonction de leurs exigences et de leur nature : les transferts de fichier sont sensibles aux pertes mais peu à l'augmentation du délai d'acheminement, contrairement aux trafics de voix ou de vidéo live qui sont sensibles aux délais mais pourront supporter un nombre raisonnable de pertes grâce à des mécanismes d'encodage et de décodage plus robustes. Reposant sur la définition de classes de services et un dimensionnement au préalable des ressources réseau attribuées à chaque classe, ce mécanisme ne modifie pas le routage mis en œuvre et les flux continuent à emprunter le plus court chemin alors que des chemins alternatifs permettant de bénéficier de la qualité de service demandée peuvent exister.

4.1 La mise en œuvre de la qualité de service dans les cœurs de réseaux

L'ingénierie de trafic vise à changer ce comportement en permettant aux administrateurs de gérer finement la façon dont les flux sont acheminés au sein de leur réseau. Plus largement, l'ingénierie de trafic permet d'optimiser le fonctionnement du réseau selon des fonctions objectif déterminées par l'administrateur, ce qui en fait une solutions essentielle pour implanter une politique garantissant la QoS. Ce mécanisme repose sur un routage spécialement calculé pour mettre en œuvre la politique du réseau. Le problème est formalisé sous forme de problèmes d'optimisation en faisant appel à des approches et des algorithmes en ligne et/ou hors ligne issus de disciplines comme la théorie des graphes, la recherche opérationnelle, la théorie des files d'attente, ou encore de la programmation linéaire. Ces stratégies doivent ensuite être déployées et appliquées sur le réseau. Pour cela, des protocoles de routages et des mécanismes de transport de données (par exemple OSPF, IS-IS et MPLS) ont été déclinés en version supportant l'ingénierie de trafic (OSPF-TE, IS-IS-TE,

MPLS-TE) afin de considérer des métriques plus complexes que le nombre de saut ou le débit des interfaces des équipements. L'établissement des chemins et la réservation de ressources sont ensuite déclenchés à l'aide du protocole de signalisation Resource Reservation Protocol (RSVP) ou Resource Reservation Protocol - Traffic Engineering (RSVP-TE), sa version supportant l'ingénierie de trafic. Cependant RSVP-TE consomme une quantité importante des ressources des équipements du réseau dans de grands réseaux mettant en place de nombreux chemins par ingénierie de trafic : de la mémoire pour stocker et maintenir des millions d'états, mais aussi des cycles de processeurs pour traiter ces énormes tables d'états et synchroniser les protocoles de contrôle. Après un redémarrage d'un nœud, le nombre de messages échangés pour repeupler les tables d'état peut provoquer une congestion dans les nœuds et par conséquent augmenter le temps de convergence du réseau. L'impact sur les ressources et la complexité de gestion et de configuration des informations d'ingénierie du trafic a été un frein majeur à l'utilisation de ces techniques. Or l'évolution vers des offres de services plus dynamiques et personnalisées a récemment imposé de nouveaux travaux de simplification globale du plan de contrôle des réseaux d'opérateurs conduisant à l'apparition du routage par segment (Segment Routing (SR)).

4.2 Ingénierie de trafic simplifiée par le routage par segment

Récemment standardisé par le groupe de travail SPRING à l'IETF, le routage par segments est une architecture instanciée sur les plans de données Multi-Protocol Label Switching (MPLS) (MPLS Segment Routing (SR-MPLS)) et IPv6 (IPv6 Segment Routing (SR-IPv6)), offrant un plan de contrôle simple en reposant sur le concept de routage par la source. Nos travaux sur Segment Routing sont le fruit d'une collaboration avec Olivier Dugeon (Orange Labs) et Samer Lahoud dans le cadre de la thèse de Rabah Guedrez et, étant destinés aux réseaux d'opérateurs, ils portent sur l'utilisation de Segment Routing sur un plan de données SR-MPLS.

L'idée de Segment Routing est de simplifier le fonctionnement dans le cœur de réseau : le nœud d'entrée détermine la route à suivre pour chaque paquet et encode les instructions de relayage dans l'en-tête du paquet. Tel un piéton demandant son chemin, un paquet obtient une suite d'instructions lui disant quoi faire à chaque intersection (tourner à gauche ou à droite, aller tout droit) pour atteindre sa destination. La différence avec le fonctionnement classique du routage est que les équipements intermédiaires (dont les routeurs en cœur de réseau) n'ont plus de décision locale à prendre pour transmettre le paquet vers le prochain saut. À la place, ils exécutent une instruction d'acheminement placée dans l'en-tête du paquet comme le montre la figue 4.1 : pour atteindre $R10$, le paquet $P1$ doit traverser les équipements $R3$, $R6$ et $R8$ alors que $P2$ doit passer par $R5$.

L'avantage majeur est l'élimination des états (tables de routage) stockés pour chaque flux par les routeurs de cœur de réseau. Dans Segment Routing, ces états par flux ne sont maintenus que par les nœuds d'entrée du réseau, concentrant la complexité sur les équipements de périphérie du réseau. Un chemin est directement utilisable par n'importe quel routeur sans nécessiter de configuration/signalisation préalable, contrairement au fonctionnement de MPLS Traffic Engineering (MPLS-TE) pour lequel un tunnel doit être installé et entretenu à l'aide de protocoles tels que RSVP-TE. De plus, l'architecture Segment Routing étend les protocoles de routage interne déjà déployés (OSPF, IS-IS et BGP Link State) pour échanger des informations Segment Routing, révoquant le besoin d'un protocole de distribution d'étiquettes tel que Label Distribution Protocol (LDP) ou RSVP-TE.

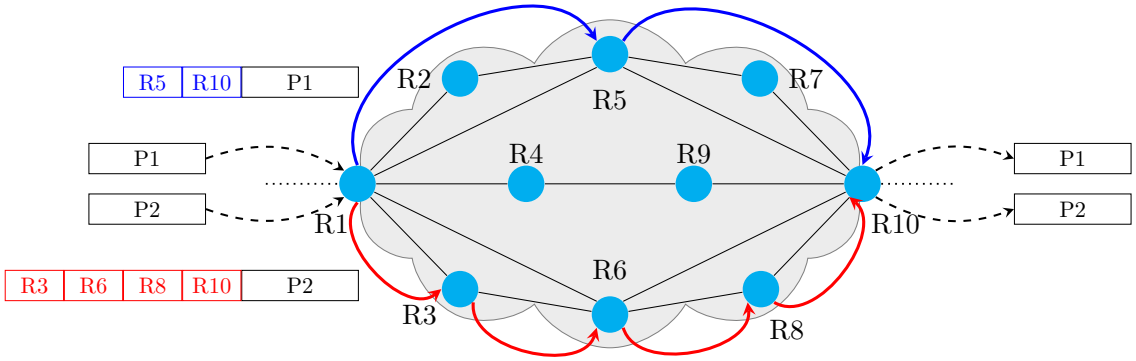


FIGURE 4.1 – Exemple de routage par la source strict en rouge, lâche en bleu

4.2.1 Expression de chemins dans SR-MPLS

Après avoir déterminé le chemin à suivre, le nœud d'entrée encode ce chemin en une séquence de segments à parcourir. Un segment peut représenter un nœud, une liaison, une adjacence de peering pour Border Gateway Protocol (BGP), un chemin MPLS (Label Switch Path (LSP)), ou même un service. Chaque segment individuel est associé à une instruction de transfert correspondante dans le plan de données (par exemple POP, PUSH, SWAP). Ainsi, un chemin de routage de segment (Segment Routing Path (SRP)) est composé d'une succession d'identifiants de segment (Segment IDentifier (SID)). L'architecture Segment Routing définit plusieurs types d'identifiants de segment dont nous n'évoquons que les deux principaux :

— Un **Node-SID** est un identifiant unique global dans le domaine Segment Routing affecté à un nœud spécifique. Dans sa table de routage, chaque nœud dispose d'une entrée pour chaque Node-SID du domaine. Lorsqu'un nœud reçoit un paquet ayant un Node-SID comme SID actif, il transfère le paquet sur le chemin donné par le protocole de routage interne (IGP).

— Un **Adj-SID** identifie une adjacence, c'est-à-dire l'interface à utiliser pour atteindre un routeur voisin. Il est utilisé pour forcer le transfert de paquets à utiliser une interface de sortie spécifique de l'équipement courant.

Dans SR-MPLS, un chemin est codé comme une pile d'étiquettes. Les SRP peuvent être exprimés exclusivement avec des Node-SIDs, des Adj-SIDs, des Adj-SIDs globaux (annoncés dans le réseau) ou une combinaison de ces types de SID. Ils peuvent être exprimés avec un routage par la source stricte ou lâche ("*loose*") :

— Un SRP est dit **strict** si tous les liens et nœuds que le paquet va traverser sont indiqués par des Adj-SIDs, ce qui produit une pile d'étiquettes correspondant exactement au chemin demandé. C'est le cas du paquet $P2$ dans la figue 4.1 qui porte la pile d'étiquettes suivante : (Adj-SID $R1 - R3$, Adj-SID $R3 - R6$, Adj-SID $R6 - R8$, Adj-SID $R8 - R10$) pour aller de $R1$ à $R10$.

— Un SRP est dit **lâche** si son en-tête ne contient qu'un sous-ensemble des liens et nœuds que le paquet va traverser. Pour encoder le chemin, l'approche lâche n'utilise que des Node-SIDs. Deux Node-SIDs successifs dans l'en-tête peuvent ne pas être voisins. Pour aller de l'un à l'autre, on utilise le plus court chemin prévu par l'IGP. Cette portion de chemin sera amenée à changer si la métrique IGP entre ces deux nœuds change, la pile d'étiquettes ne représentera alors plus le chemin initialement

calculé par le nœud Segment Routing d'entrée. Par exemple, un encodage lâche du chemin à emprunter par le paquet $P1$ dans la figue 4.1 sera exprimé par la pile d'étiquettes (Node-SID $R5$) pour aller de $R1$ à $R10$. Ceci n'est valable que si le chemin désiré correspond bien au chemin le plus court entre les Node-SID indiqués ($R1$, $R5$, $R10$), ce qui ne convient pas si le chemin souhaité était ($R1$, $R2$, $R5$, $R7$, $R10$).

Un SRP encodé avec des Adj-SID globaux peut être strict ou lâche : il est strict si tous les liens que le paquet doit traverser sont listés dans la pile d'étiquettes, lâche sinon.

4.2.2 MSD : la limitation de la taille de la pile de label dans SR-MPLS

Afin d'obtenir un traitement de paquets à la vitesse du lien, les constructeurs de matériel utilisent des circuits intégrés spécifiques aux applications (ASIC) conçus pour exécuter des tâches spécifiques de manière très efficace mais qui sont limités vis à vis de la taille et du type d'opérations qu'ils peuvent traiter. Les routeurs actuellement sur le marché ont une limitation sur le nombre d'étiquettes qu'ils sont capables ajouter à un en-tête de paquet MPLS.

Dans Segment Routing, cette limitation est connue sous le nom de Maximum SID Depth (MSD). Aujourd'hui, le MSD est faible, bornant en général le nombre d'étiquettes maximum à 5 (mais pouvant aller de 3 à 10 suivant les routeurs). La longueur du SRP varie en fonction du diamètre du réseau, des exigences de qualité de service et de la disponibilité des ressources réseau. Par conséquent, la pile d'étiquettes pour exprimer un SRP peut être très grande et potentiellement violer le MSD du routeur d'entrée, empêchant l'utilisation de ces chemins en Segment Routing. L'effet du MSD est donc de réduire le nombre de chemins exploitables dans le réseau, ce qui conduit à une utilisation sous-optimale des ressources disponibles. Par conséquent, il est nécessaire de disposer d'un algorithme d'encodage efficace pour minimiser la taille de la pile d'étiquettes.

4.2.3 Les algorithmes SR-LEA et SR-LEA-A pour le calcul d'une pile minimale d'étiquettes pour un SRP strict

Un réseau d'opérateur peut être composé de centaines, voire de milliers de nœuds, augmentant considérablement la taille des en-têtes de paquets et rendant l'utilisation d'un encodage strict incompatible avec la contrainte du MSD (en particulier pour les chemins longs). Bien qu'il soit le pire des scénarios pour l'encodage de SRP, le codage strict peut être indispensable pour accomplir certaines tâches telles que l'exploitation, l'administration et la maintenance (OAM). Nous avons proposé deux algorithmes de codage de SRP qui expriment une pile d'étiquettes par une composition de Node-SIDs et de Adj-SIDs. Notre algorighme Segment Routing paths Label Encoding Algorithm (SR-LEA) calcule la pile d'étiquettes minimale lorsque les Adj-SID sont annoncés comme segments locaux (comme c'est le cas dans les déploiements actuels de Segment Routing). Toutefois, les normes indiquent que les Adj-SID peuvent également être annoncés comme des segments globaux. Nous avons donc proposé l'algorithme SR-LEA-A qui adapte SR-LEA en ce sens.

L'algorithme SR-LEA

L'algorithme prend en entrée le chemin initial exprimé sous la forme d'une liste d'adresses IP, calculée au préalable (manuellement, par une entité centralisée telle qu'un contrôleur SDN, ou par un élément de calcul de chemin (Path Computation Element (PCE)) [97]). La pile d'étiquettes résultante est une combinaison de Node-SIDs et de Adj-SIDs locaux

représentant exactement le chemin initialement calculé dans l'état actuel du réseau. SR-LEA comporte deux étapes principales :

— le *splicing* : le SRP est divisé en une succession de plus courts chemins (les sous-chemins)

— l'*expression des sous-chemins* : chaque sous-chemin composé d'au moins trois nœuds est remplacé par le Node-SID du dernier nœud, tandis que chaque sous-chemin de deux nœuds est remplacé par le Adj-SID entre ces deux nœuds.

Dans le meilleur cas, le SRP demandé suit le plus court chemin IGP, SR-LEA génère alors une pile ne contenant que le Node-SID du nœud de sortie.

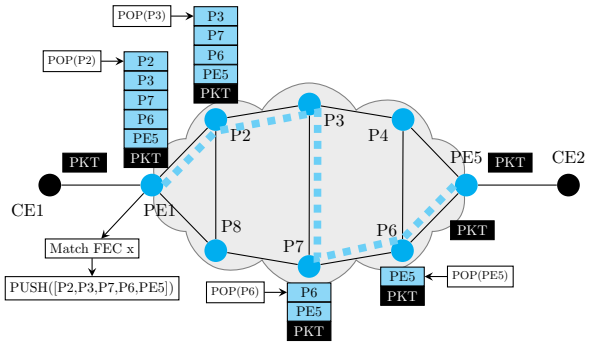


FIGURE 4.2 – Encodage de P avec un chemin strict

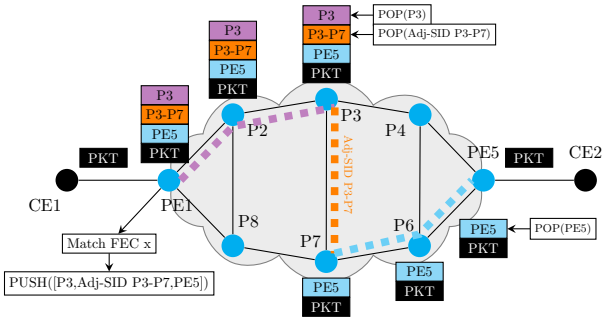


FIGURE 4.3 – avec SR-LEA

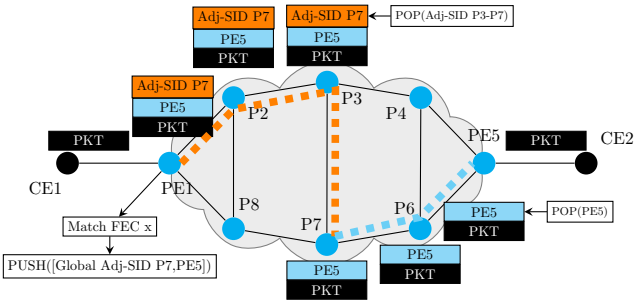


FIGURE 4.4 – avec SR-LEA-A

Les figures 4.2, 4.3 et 4.4, montrent le résultat de l'encodage du chemin P avec un SRP strict, avec SR-LEA et avec SR-LEA-A. L'encodage avec SR-LEA est obtenu grâce aux étapes suivantes :

— P est décomposé en sous-chemins $\{(PE1,P2,P3),(P3,P7),(P7,P6,PE5)\}$.

— Chaque sous-chemin est remplacé par le SID approprié :

  — $\{PE1,P2, P3\}$ étant composé de trois nœuds, il est remplacé par le Node-SID de $P3$,

  — $\{P3,P7\}$ étant composé de deux nœuds, il est remplacé par le Adj-SID $P3$-$P7$,

  — $\{P7,P6,PE5\}$ étant composé de trois nœuds, il est remplacé par le Node-SID de $PE5$.

La pile d'étiquettes résultante est $[P3, P3-P7, PE5]$. Au moment du routage, un paquet suit le chemin le plus court pour atteindre $P3$ en utilisant le Node-SID de $P3$. En $P3$, l'Adj-SID $P3$-$P7$ est utilisée. En $P7$, le Node-SID de $PE5$ permet d'acheminer le paquet par le plus court chemin IGP pour atteindre $PE5$. $PE5$ enlève cette dernière étiquette avant de transmettre le paquet IP à $CE2$ (voir la figure 4.3).

L'algorithme SR-LEA-A

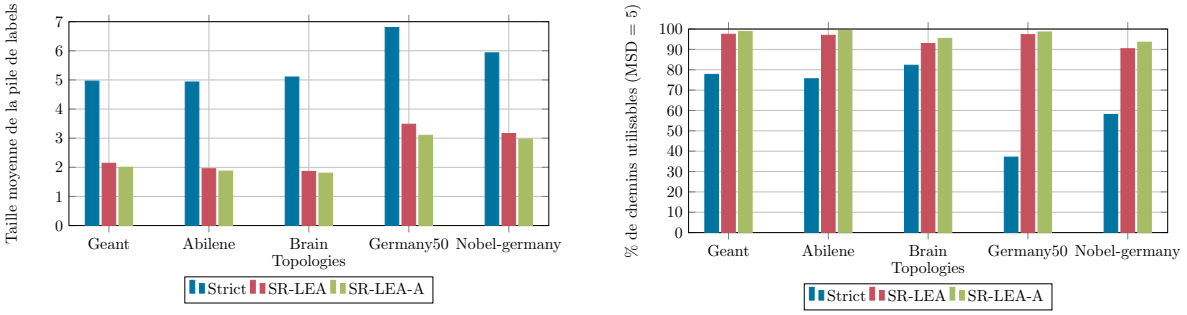
Les spécifications de l'architecture de Segment Routing prévoient qu'une adjacence puisse être annoncée comme un segment global et donc être routable dans le domaine Segment Routing. Les nœuds transfèrent alors le paquet par leur plus court chemin IGP vers le nœud annonçant l'Adj-SID global, puis celui-ci le relaye par l'interface de sortie associée. SR-LEA-A adapte SR-LEA pour prendre en compte des Adj-SID globaux. La décomposition du chemin en sous-chemins reste identique, seule l'expression des sous-chemins change : la succession d'un sous-chemin de taille 3 par un sous-chemin de taille 2 sera codée en n'utilisant que l'Adj-SID global de l'adjacence entre le dernier nœud du premier sous-chemin et le premier nœud du second.

Dans l'exemple de la figure 4.4, $P3$ annonce aux autres nœuds du réseau son adjacence avec $P7$ par un Adj-SID global. P a été décomposé en sous-chemins {($PE1$,$P2$, $P3$), ($P3$,$P7$), ($P7$,$P6$, $PE5$)}. Avec SR-LEA-A, les deux sous-chemins {($PE1$,$P2$, $P3$), ($P3$, $P7$)} sont encodés en utilisant le Adj-SID global $P3$-$P7$. La pile d'étiquettes pour le chemin P est donc [$P3$-$P7$, $P5$]. Lors du routage, le paquet est acheminé par le plus court chemin pour atteindre $P3$. $P3$ retire l'étiquette $P3$-$P7$ et transfert le paquet par l'interface qui le relie à $P7$. $P7$ transfert alors le paquet en utilisant le Node-SID de $PE5$ par le plus court chemin IGP pour atteindre le nœud $PE5$.

Evaluation de performance de SR-LEA et SR-LEA-A

Nous avons évalués la performance des algorithmes SR-LEA et SR-LEA-A sur plusieurs topologies de réseaux SNDlib [89]. Nous avons déterminé les chemins optimaux pour satisfaire une matrice de la demandes donnée par la résolution d'un problème de *multicommodity flow*. Ces chemins stricts sont ensuite encodés par des Adj-SID, par SR-LEA et par SR-LEA-A. Pour chaque topologie, nous avons calculé la taille moyenne de la pile d'étiquettes et le pourcentage de chemins réseau encodés en respectant un MSD de 5 étiquettes. L'ensemble des résultats sont consultables dans [19] et résumés dans la figure 4.5. Nous avons constaté que l'encodage strict produit des piles pouvant atteindre jusqu'à 14 étiquettes. Le pourcentage de chemins exprimables peut alors être très faible, par exemple, seuls 37% des chemins sont utilisables avec Segment Routing pour la topologie Germany50. SR-LEA réduit la taille de la pile d'étiquettes de 52% à 65% par rapport à l'encodage strict (le gain observé varie en fonction de la conception et du diamètre du réseau). Par conséquent, SR-LEA augmente considérablement le nombre de chemins utilisables (à 97% pour la topologie Germany50). Cependant, le codage de la pile d'étiquettes à l'aide de SR-LEA-A donne les meilleurs résultats en réduisant la taille moyenne des piles d'étiquettes de 57% à 67% par rapport à l'encodage strict et augmente le nombre de chemins utilisables à 99% pour Germany50.

Bien que les algorithmes proposés soient très efficaces pour réduire la taille de la pile d'étiquettes et atténuer l'impact du MSD, ils n'éliminent pas le problème car il reste des chemins qui ne peuvent pas être exprimés avec une pile d'étiquettes respectant le MSD. Pour résoudre complètement le problème du MSD, nous avons étudié la fragmentation des SRP en utilisant le Targeted-SID (TSID), un nouveau type de segment assigné à une portion du SRP.

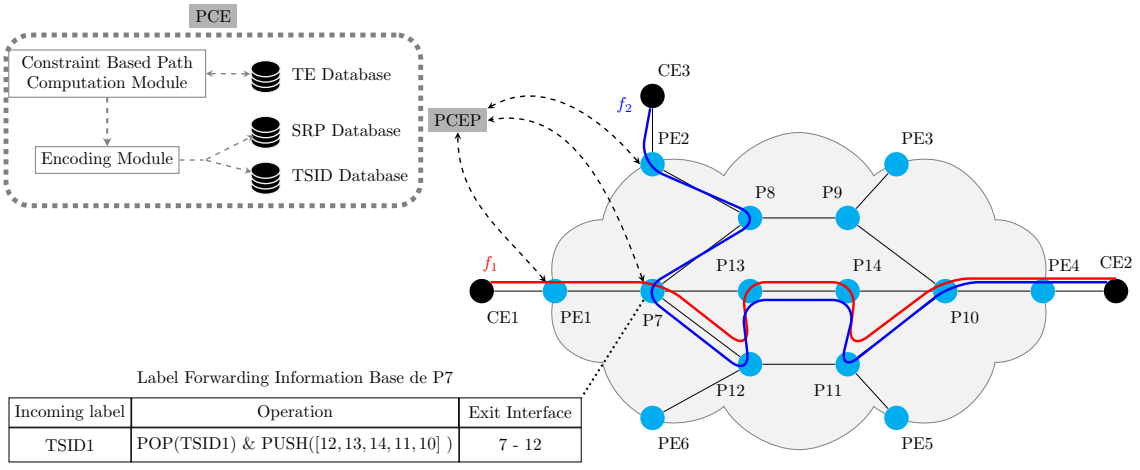


Taille moyenne de la pile de labels — Nombre de chemins utilisables

FIGURE 4.5 – Evaluation des performances de SR-LEA et SR-LEA-A

4.2.4 Le Targeted SID (TSID) et l'approche par segmentation des chemins

Afin de rendre tous les chemins du réseau exprimable par Segment Routing en respectant la contrainte du MSD, nous avons proposé une solution de fragmentation des chemins dans [20] reposant sur l'utilisation d'un nouveau type de segment : le TSID. Un TSID peut être vu comme un identifiant de segment temporaire qui sera remplacé par une pile de segment dans le nœud sachant l'interpréter. Un TSID est donc associé à une sous-pile d'étiquettes qui décrit une portion de chemin. Il est installé sur des nœuds spécifiques du réseau. Comme l'Adj-SID, un TSID est local à un nœud. Dans cette approche, la pile d'étiquettes initiale exprimant le SRP est découpée en plusieurs piles, remplacées par un TSID.



Label Forwarding Information Base de P7

Incoming label	Operation	Exit Interface
TSID1	POP(TSID1) & PUSH([12, 13, 14, 11, 10])	7 - 12

FIGURE 4.6 – Exemple de TSID

Considérons le réseau illustré par la figure 4.6. Le réseau doit acheminer deux flux f_1 (en rouge) et f_2 (en bleu) de 100 Mo à destination de $CE2$, le flux f_1 est envoyé par $CE1$ alors que le flux f_2 est émis par $CE3$. Le calcul des chemins respectant la bande passante demandée donne les piles suivantes :

— $[PE1 - P7, P7 - P12, P12 - P13, P13 - P14, P14 - P11, P11 - P10, P10 - PE4]$ pour le flux f_1,

— $[PE2 - P8, P8 - P7, P7 - P12, P12 - P13, P13 - P14, P14 - P11, P11 - P10, P10 - PE4]$ pour le flux f_2.

L'encodage strict de ces SRPs ne peut pas être utilisé avec MSD limité à 5. L'administrateur installe donc le TSID $TSID1$ pour le sous-chemin $P = [P12 - P13, P13 - P14, P14 - P11, P11 - P10]$ dans le nœud $P7$, utilisable pour les deux flux. Ceci se traduit par une nouvelle entrée dans la base de donnée des étiquettes (Label Forwarding Information Base - LFIB) de $P7$ et par les codage de piles d'étiquettes suivant :

— $[PE1 - P7, TSID1, P10 - P4]$ pour le flux f_1,

— $[PE2 - P8, P8 - P7, TSID1, P10 - P4]$ pour le flux f_2.

L'introduction du $TSID1$ permet de pouvoir exploiter les chemins choisis. Ainsi, en créant autant de TSID que nécessaire, on peut obtenir une taille de pile d'étiquettes respectant le MSD. Cependant, les TSID doivent être pré-installés sur le réseau avant que le trafic ne soit acheminé sur les SRPs.

Considérations architecturales

L'architecture Segment Routing a été conçue pour éviter l'installation d'états dans les nœuds du réseau et ainsi réduire la complexité du plan de contrôle. L'ajout de la pile d'étiquettes dans l'en-tête des paquets évite la mise en place de signalisation et permet de supprimer les protocoles RSVP-TE et LDP. Bien que l'introduction des TSID aille à l'encontre du principe de suppression des états à maintenir dans les équipements de cœur du réseau, cela reste un compromis intéressant pour pouvoir exprimer tous les chemins possibles dans le réseau sans violer la contrainte du MSD, en attendant l'évolution des équipements. Afin de limiter l'impact des TSID sur la gestion du réseau, nous avons cherché à minimiser le nombre de nœuds pour lesquels un nouvel état doit être introduit en proposant un algorithme d'optimisation pour réduire le nombre de TSID installés.

Optimisation de l'installation de TSID

Les nœuds SR-MPLS maintiennent considérablement moins d'états par rapport au fonctionnement traditionnel de MPLS. Toutefois, l'approche de segmentation des chemins ajoute un surcoût à l'architecture Segment Routing. Les TSID sont des entrées supplémentaires dans la table de routage des nœuds, sachant que chaque nœud peut avoir à maintenir une base de données des TSID si les TSID sont annoncés dans l'IGP.

Une solution pour atteinuer ce problème est d'introduire un contrôleur SDN et/ou un serveur PCE et une base de données locale pour l'ingénierie de trafic (Traffic Engineering Database (TED)) pour indiquer la disponibilité des ressources. Le PCE maintient également une base de données des TSID, ce qui permet de réutiliser des TSID existants pour les futurs SRP. Pour cela, nous avons proposé l'architecture décrite dans [17]. Lorsqu'une requête atteint le PCE, il calcule un chemin respectant les contraintes spécifiées en fonction des informations contenues dans la TED. Le chemin calculé est ensuite envoyé au module d'encodage qui décide si un TSID est nécessaire ou non. Pour installer les TSID, le PCE maintient une session Path Computation Element communication Protocol (PCEP) avec tous les nœuds du réseau.

Dans [20], nous avons cherché à résoudre les deux problèmes d'optimisation suivants :

— réduire le nombre global de TSID installés,

— réduire le nombre de sessions PCEP dans le réseau.

Nous avons proposé deux programmes linéaires hors ligne. Les deux modèles prennent un ensemble de chemins en entrée et s'appuient sur la matrice de trafic existante. Un

ensemble réaliste de chemins est généré en résolvant le problème *multi-commodity flow* pour un réseau donné et une matrice de trafic donnée.

Nous avons également proposé deux algorithmes en ligne, appelés OTO pour Online TSID Optimization :

— OTO pour la minimisation des TSID favorise la réutualisation des TSID existants. Il n'en crée de nouveaux que s'il n'existe aucune solution pour réduire le chemin demandé avec les TSID disponibles,

— OTO pour la minimisation des sessions PCEP favorise les solutions qui nécessitent l'installation de TSID sur les nœuds qui maintiennent déjà une session PCEP active avec le PCE tout en favorisant l'utilisant de TSID existants.

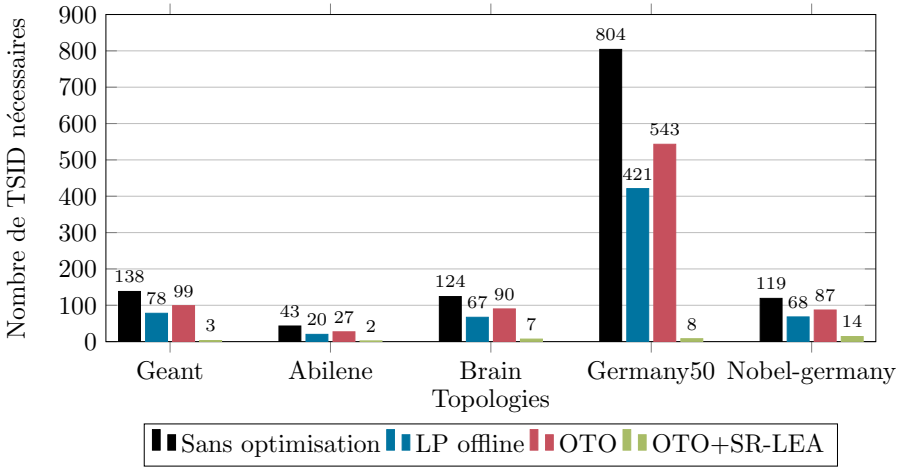


FIGURE 4.7 – Evaluation de l'utilisation des TSID

Ces algorithmes en ligne, plus pratiques avec des matrices de trafic inconnues, sont basés sur les deux modèles hors ligne. La figure 4.7 présente une comparaison des performances des différentes solutions proposées pour exprimer les chemins des topologies étudiées lorsque le MSD est limité à 5. Nous avons comparé le nombre de TSID créés pour chaque topologie dans le pire cas (l'installation en ligne de TSID sans optimisation) par rapport à la solution optimale calculée au préalable grâce à un programme linéaire donnant le nombre minimal de TSID nécessaires pour exprimer chaque chemin de la topologie en respectant le MSD. Nous comparons ces résultats à ceux obtenus avec OTO et avec une combinaison de OTO et SR-LEA. En fait, l'algorithme OTO n'est appelé que si l'algorithme SR-LEA ne parvient pas à calculer une pile d'étiquettes de taille inférieure au MSD. Les résultats expérimentaux montrent l'efficacité des deux variations de l'algorithme OTO qui atteignent des performances proches de celles des modèles hors ligne. En particulier, le couplage de l'algorithme OTO avec SR-LEA permet de réduire considérablement le nombre de TSID installés. Nous avons effectué des tests similaires pour évaluer les performances de nos propositions face à la réduction de sessions PCEP. Ces résultats sont détaillés dans [20].

Dans ces travaux, nous avons proposés des solutions pour mettre en œuvre l'ingénierie de trafic plus simplement avec l'utilisation du Segment Routing. En particulier, nous avons conçu des mécanismes pour les grandes topologies, grâce à différents encodage de la pile de labels, afin de contourner la limitation des matériels disponibles qui ne peuvent traiter que des piles de labels de faible taille. Ces propositions sont transitoires et lèvent un frein

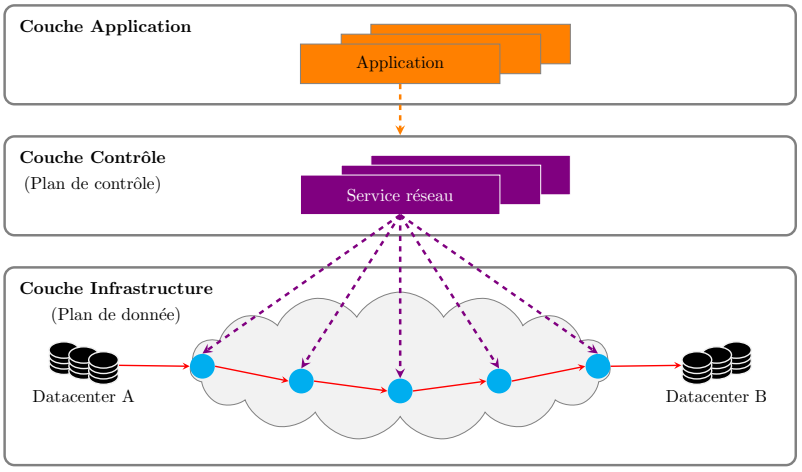


FIGURE 4.8 – L'architecture SDN

essentiel au déploiement de l'architecture Segment Routing dans les réseaux actuels. Bien qu'à terme nous pouvons espérer que le parc des équipements pourra traiter des piles de labels de taille raisonnable, le problème du MSD persistera tant que tous les équipements limitants n'auront pas été remplacés, rendant nos mécanismes indispensables pendant ces années de transition.

4.3 Garantie de QoS dans un réseau programmable

Les opérateurs réseaux ont recours à l'ingénierie de trafic pour faire face à la croissance du trafic et à des besoins de qualité de service de plus en plus pressants. L'évolution de la standardisation d'architectures tels que Segment Routing facilite la mise en place d'un routage optimisant une politique de gestion du réseau pour soulager les plus courts chemins traditionnellement utilisés par les protocoles de routage interne. Cependant, cela nécessite l'introduction de nouveaux éléments (tels que le PCE), plus intelligents pour surveiller la disponibilité des ressources et faciliter les calculs d'optimisation du routage et l'installation des routes choisies.

Au cours des dernières années, l'apparition de l'architecture SDN a permis d'aller plus loin dans l'automatisation de la gestion des réseaux en les rendant programmables [45]. Elle introduit une couche de contrôle dans laquelle l'intelligence des routeurs est déportée, simplifiant les équipements du plan de donnée et les dédiant au relayage des paquets (voir figure 4.8).

Le plan de contrôle est organisé autour d'un élément appelé contrôleur. Son rôle est de calculer les routes à mettre en œuvre pour chaque flux et à envoyer les instruction de configuration aux équipements du plan de contrôle. Une fois configurés, ils sont ensuite en mesure de relayer le trafic, appliquant ainsi une politique de routage décidée par le contrôleur.

Le contrôleur SDN étant logiquement centralisé (bien qu'il puisse être distribué pour des raisons de scalabilité par exemple), il dispose d'une vision holistique de la topologie du réseau et des ressources (bande passante, délai de liaison, CPU...). Ceci ouvre une perspective intéressante lorsqu'il s'agit de mettre en place des chemins à QoS garantie. Avec cette perception globale, le plan de contrôle est capable de résoudre les problèmes de routage contraint de bout-en-bout, sans avoir à faire face aux problèmes de latence des calculs et de convergence des règles de routage, inhérents aux réseaux distribués. Cela

permet une gestion fine des ressources du réseau par le contrôleur SDN et apporte une solution élégante pour gérer efficacement la QoS dans les réseaux et par extension en inter-domaine.

Nous avons exploré ces nouvelles possibilités dans le cadre d'une collaboration avec l'IRT $B <> COM$ [1] (dans lequel je suis partiellement détachée) et dans le cadre de la thèse CIFRE de Cédric Morin avec la société Telediffusion de France (TDF) [2], fournisseur d'infrastructures réseaux spécialisés dans les services de transmissions pour la radio, la télévision, et plus largement, les acteurs du multimédia. Dans cette étude, nous nous sommes appuyés sur un cas d'usage concret apporté par TDF (représenté dans la figure 4.8) pour offrir dynamiquement des garanties de QoS personnalisées à leurs clients. Les clients adressent une requête d'acheminement de trafic spécifiant la quantité de bande passante souhaitée pendant une période de temps limitée. Cette situation peut se produire lorsqu'un client souhaite organiser un transfert de données massif et ponctuel entre deux centres de données, ou lorsqu'un client (tel qu'un stade, un théâtre ou tout autre acteur de l'industrie événementielle) a besoin d'une connexion temporaire entre son emplacement et un centre de données. Actuellement, les paramètres pris en charge sont la bande passante, le nombre de sauts ou le coût, mais des limites sur la latence ou les pertes de paquets peuvent être ajoutées par la suite.

4.3.1 Le module STEM (SDN Traffic Engineering Management)

Dans [35], nous avons identifié les possibilités offertes par SDN pour mettre en place des mécanismes de QoS. Nous avons proposé le module SDN Traffic Engineering Management (STEM), intégré au plan de contrôle, qui fournit une allocation de bande passante à la volée pour les utilisateurs, sans intervention de l'opérateur. Cette solution pour garantir la QoS à tout moment et fournir une allocation de bande passante à travers un réseau SDN implique l'utilisation d'un élément de calcul de chemin (PCE) pour identifier les chemins appropriés et réserver les ressources dans le plan de données.

Le module STEM interprète les demandes de création de tunnel et les transmet au PCE qui calcule un chemin dans le réseau qui puisse satisfaire la demande. STEM traduit ce chemin en règles de flux transmises au contrôleur. Le contrôleur SDN applique les règles de flux dans les commutateurs du plan de données. Ces commandes sont reçues et interprétées par les équipements du réseau (commutateurs ou routeurs). On peut noter que ce mécanisme est compatible avec les approches de garanties de QoS basées sur la réservation de ressource tel que DiffServ mais également avec l'ingénierie de trafic, le contrôleur décidant du chemin emprunté par les paquets et configurant tous les équipements concernés dans le plan de données.

4.3.2 L'évaluation des ressources disponibles dans SDN

La sélection d'un chemin satisfaisant des contraintes de QoS oblige le plan de contrôle à maintenir une représentation précise des ressources disponibles du réseau. Dans la littérature, la plupart des solutions s'appuient sur les techniques de collecte de statistiques pour construire et mettre à jour périodiquement leurs connaissances du trafic en cours d'acheminement et des ressources consommées [76][102]. Or la fluctuation constante de la charge de trafic implique des mises à jour fréquentes de ces informations, conduisant à une augmentation importante du trafic de contrôle [58]. En plus de son impact sur le temps de réponse des commutateurs, une interrogation excessive des statistiques pour

1. `https://b-com.com/fr`
2. `https://www.tdf.fr`

évaluer les conditions du réseau génère une charge de travail lourde pour le contrôleur et exige beaucoup de puissance de calcul. Enfin, le temps de réaction du système ne garantit pas que les contrats de QoS seront respectés à tout moment, en particulier entre deux interrogations. Ces aspects ne sont en général pas pris en compte car la plupart des travaux antérieurs ne considèrent que les petites topologies avec des débits réduits.

Bien que la littérature sur ce sujet soit très importante pour les réseaux traditionnels, peu de solutions ont été adaptées à l'architecture SDN. Les approches proposées sont soit réactives, soit proactives. Les approches réactives sont basées sur des méthodes d'adaptation à la détection des pénuries de ressources. Elles exploitent en général la diversité des chemins : plusieurs chemins sont pré-calculés pour un flux donné, afin qu'il puisse passer rapidement de l'un à l'autre lorsque la QoS n'est plus respectée. Il en résulte soit un surdimensionnement, soit une qualité de service non garantie. Des approches proactives peuvent être mises en œuvre par la réservation de ressources [53].

Nous avons adopté une approche proactive : puisque le plan de contrôle décide des chemins empruntés par tous les trafics, le PCE est en mesure de mémoriser les ressources allouées à chacun et n'a pas besoin de sonder systématiquement les équipements du réseau pour connaître les ressources dont ils disposent. Cette méthode repose sur l'efficacité du contrôle d'admission pour assurer que le flux respecte bien les caractéristiques de la demande et donc les spécifications qui ont permis le dimensionnement du réseau devant l'acheminer. Il s'agit de mesurer les caractéristiques du flux entrant (le débit moyen par exemple) et si besoin d'effectuer soit une mise en conformité (en retardant certains paquets pour lisser son débit moyen ou en rejetant les paquets en excès), soit de marquer le trafic supplémentaire comme moins prioritaire. Pour cela, on utilise généralement des files d'attente, qui permettent de façonner et de prioriser le trafic, de partager la bande passante et de contrôler la latence, et des seaux à jeton (*token bucket*), qui vont mesurer le débit du flux et évaluer sa conformité. Pour répondre aux problèmes de passage à l'échelle, STEM utilise un nombre prédéfini de files d'attente. La flexibilité du mécanisme est assurée par la classification du trafic à l'aide de l'outil de comptage *Meter*, un élément qui permet de mesurer et de contrôler la fréquence des paquets arrivant sur une interface réseau [79].

Le calcul du chemin

La figure 4.9 présente l'architecture du dispositif dans lequel nous avons principalement adapté le plan de contrôle.

L'un de nos objectifs étant d'éviter l'interrogation constante des équipements pour la collecte de statistiques, nous gardons une trace des ressources allouées. La couche de contrôle s'appuie ainsi sur un PCE à état (*stateful*) [55] pour sécuriser les ressources réservées aux flux. Comme un PCE sans état (*stateless*), il gère une TED, la base de données utilisée en ingénierie du trafic, et l'utilise pour effectuer des calculs de chemins. Cependant, un PCE sans état n'enregistre pas les changements de ressources en fonction des calculs effectués. Par conséquent, les ressources utilisées peuvent être affectées plusieurs fois lors des calculs, de sorte que la qualité de service n'est plus assurée lorsque le chemin est effectivement établi. Au contraire, un PCE à état mémorise les ressources potentiellement allouées en enregistrant les routes calculées et les exigences de qualité de service associées. Ce mécanisme, plus coûteux, garantit que la même ressource ne sera pas allouée deux fois (à moins que la surréservation ne soit explicitement autorisée), quitte à ne pas prendre en compte des ressources qui pourraient au final être disponibles si un chemin calculé n'est pas mis en œuvre. Il nécessite également un suivi des demandes et du cycle de vie de chaque flux. Notre PCE, travaillant au niveau d'un domaine, calcule le chemin grâce à

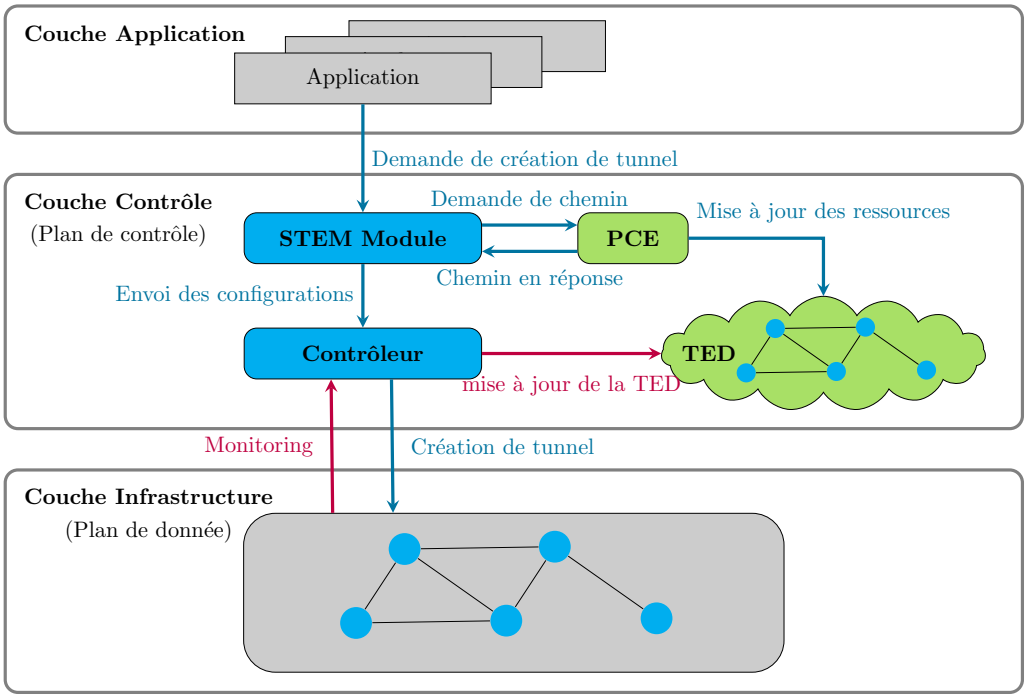


FIGURE 4.9 – Architecture de gestion de l'ingénierie de trafic et de QoS

l'algorithme de Dijkstra lorsque la demande ne comporte qu'une seule contrainte de QoS et SAMCRA [106] lorsqu'elle spécifie des contraintes multiples.

L'implémentation et l'efficacité du module STEM sont détaillés dans [35]. Nous y présentons la plate-forme réalisée pour évaluer notre solution basée sur l'utilisation de l'outil *meter* intégré à OpenFlow 1.3, le contrôleur OpenDaylight (ODL) et un switch matériel PICA8.

Lors de ces travaux, nous avons pu constater que l'architecture SDN est une réelle opportunité pour généraliser l'utilisation de l'ingénierie de trafic dans la gestion des réseaux. Ceci constitue un pas important vers la garantie de QoS et une utilisation plus fine des ressources du réseau. En particulier dans le contexte de de la 5G et des offres Infrastructure as a Service (IaaS) ou Platform as a Service (PaaS), pour lesquelles la création dynamique et automatique de chemins pour acheminer les données est incontournable. Cependant cela ne constitue qu'un des éléments nécessaires pour ces nouveaux paradigmes qui reposent sur la virtualisation des fonctions réseau. Outre la garantie de la QoS lors de l'acheminement des données, il s'agit maintenant de s'assurer que le déploiement des fonctions réseaux virtualisées respecte un ensemble de contraintes, comme nous le verrons dans la section suivante.

4.4 La virtualisation des réseaux

L'émergence de la 5G s'accompagne de nouveaux cas d'usages basés sur la virtualisation des réseaux. En particulier, l'architecture MANO définie à l'ETSI est une réponse aux besoin des fournisseurs de service réseau pour offrir des infrastructures (IaaS), des réseaux (Network as a Service (NaaS)) et des plateformes (PaaS) adaptées à un besoin exprimé par leur client. Le but est de bâtir des infrastructures virtuelles personnalisées de façon dynamique et automatique tout en mutualisant leur infrastructure réseau physique pour

de multiples clients. La grande diversité des services offerts par les réseaux d'aujourd'hui repose sur le déploiement d'un grand nombre de middlebox (tels que les firewalls, proxies ou load balancers), rendant les réseaux plus complexes et plus difficiles à gérer. Le nouveau paradigme de NFV [46] aborde ces questions en transformant les middlebox physiques traditionnelles en logiciels fonctionnant sur des serveurs génériques. Entre autres avantages, les fonctions réseaux virtualisées (VNF) brisent la dépendance vis-à-vis des fournisseurs, permettent des mises à jour fréquentes, réduisent les coûts d'installation et de gestion et introduisent une flexibilité en termes d'évolutivité et de placement.

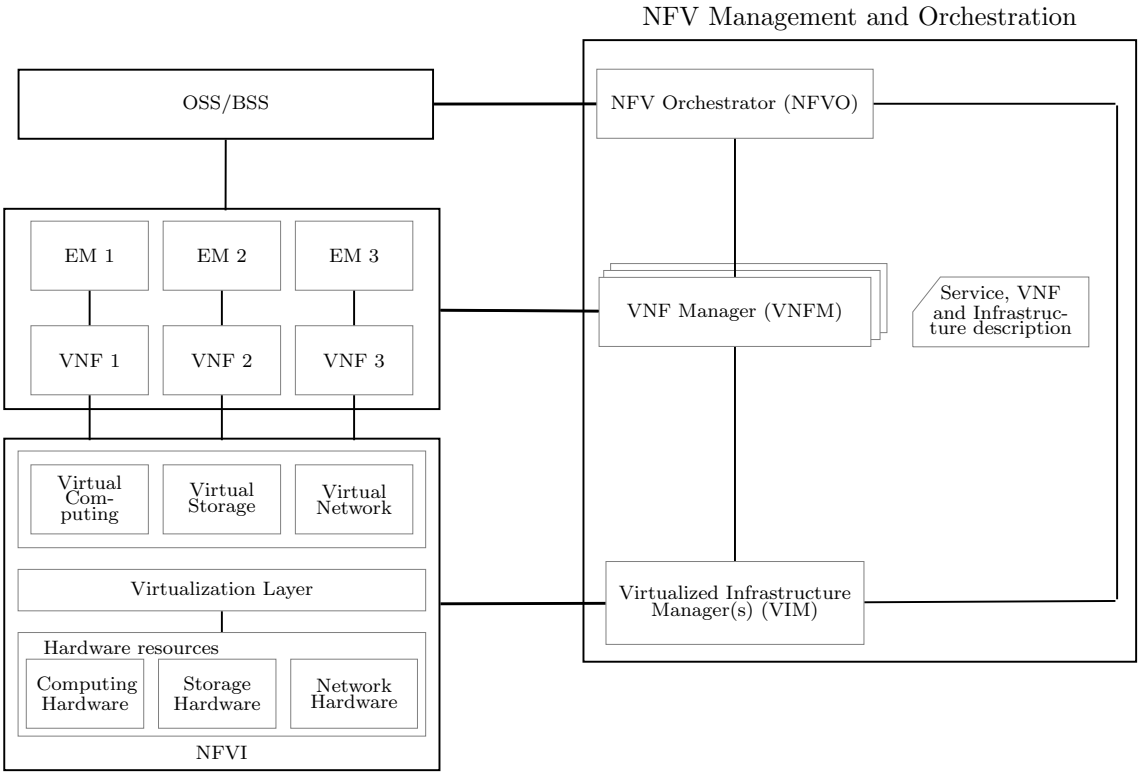


FIGURE 4.10 – Architecture de MANO de l'ETSI [63]

Le standard NFV-Management and Orchestration (MANO) (voir figure 4.10) [63] introduit une nouvelle architecture pour gérer le déploiement de VNF et l'instanciation services réseaux (Network Service (NS)) basée sur des fonctions d'orchestration de service assurées par le Network Functions Virtualisation Orchestrator (NFVO), de gestion de fonctions réseaux virtualisées par le Virtual Network Function Manager (VNFM) et de gestion d'une infrastructure virtualisée par le Virtualized Infrastructure Manager (VIM). Les demandes des clients sont relayées par l'Operations Support System/Business Support System (OSS/BSS) vers le NFVO qui se charge de réaliser le service réseau, décomposé en un graphe ou une chaîne de VNF. Pour cela, le NFVO interagit avec le VNFM pour le choix des VNF et avec le VIM pour les déployer dans l'infrastructure virtuelle (Network Functions Virtualisation Infrastructure (NFVI)).

Le problème de placement d'un graphe de VNF dans le NFVI, appelé Virtual Network Function Graph Placement Problem (VNFGPP) ou Network Function Chain Placement Problem (VNFCPP) pour le placement d'une chaîne de VNF, suscite de nombreuses publications [68]. Les approches diffèrent en fonction de la formulation exacte du problème, du contexte et des aspects qu'il optimise. Dans les datacenters, les travaux (par exemple [100]) cherchent en général une solution de placement de VNF afin de minimiser la consomma-

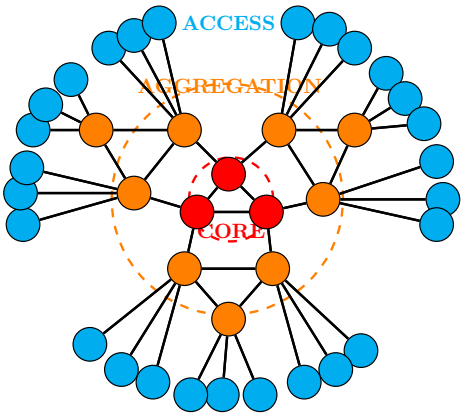
tion d'énergie tout en ignorant les contraintes de liaison telles que les délais, puisque les serveurs sont physiquement co-localisés. Dans les réseaux étendus, une première approche du VNFCPP consiste à traiter séparément le chaînage et le placement des VNF. Dans [47], l'optimisation est priorisée : d'abord pour effectuer l'optimisation sur les liens, puis sur les nœuds. De même, [101] identifie le chemin le plus court entre la source et la destination de la chaîne avant d'y placer les VNF. Une variante consiste à considérer que les VNF sont déjà installées, le problème devient donc de faire correspondre le graphe de VNF à un ensemble d'instances de VNF existantes et à considérer et l'acheminement des nouvelles requêtes par cette chaîne correcte de VNF existantes [61]. D'autres approches tendent à simplifier le problème général. Par exemple, [52] ignore les contraintes des nœuds (comme le CPU) pour se concentrer sur le coût des liens. Peu de contributions considèrent le problème dans toute son ampleur, surtout dans des architectures faisant intervenir plusieurs fournisseurs d'infrastructures virtuelles (cas multi-tenant) qui n'est en général pas traité.

Parallèlement, l'émergence de la 5G conduit à la création de nouveaux services réseau, avec des contraintes de trafic et de latence accrues. En raison de leur proximité avec les utilisateurs finaux, les ressources périphériques sont cruciales pour atteindre des exigences de latence ultra faibles, mais leur rareté impose une gestion prudente. Ainsi, malgré la promesse de flexibilité et d'évolutivité de NFV, nous devons nous concentrer sur les ressources périphériques en plus de celles du cloud [92]. Le placement de VNF en exploitant les réseaux edge a déjà été proposé dans [54][47] et [93] mais sans examiner particulièrement l'influence de la topologie et de la distribution des ressources sur les performances de l'algorithme.

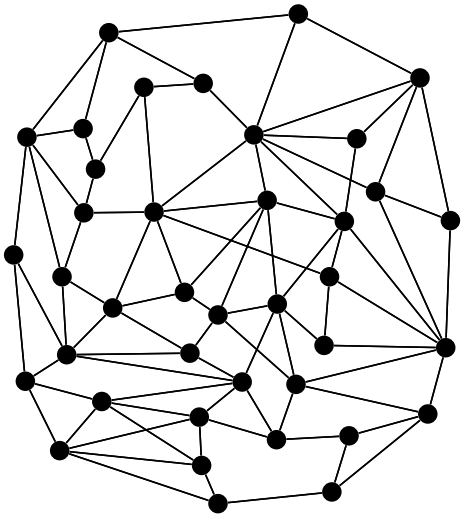
Dans le cadre de la thèse de Cédric Morin, nous étudions le déploiement dynamique et à la demande de nouveaux services réseaux à l'aide de l'architecture MANO et de la virtualisation des fonctions réseaux. Nous avons proposé une stratégie d'optimisation visant à maximiser l'acceptation de nouveaux NS et l'avons formalisée comme un problème de ILP. Nous avons proposé une analyse approfondie de sa performance en fonction des demandes et des caractéristiques topologiques quand l'orchestrateur et l'infrastructure virtualisée appartiennent à une même entité (cas **mono-tenant**). Puis, nous avons proposé une heuristique basée sur l'abstraction réseau pour gérer à la fois les défis de la complexité des calculs et de l'environnement **multi-tenant**, lorsque l'infrastructure et les fonctions d'orchestrations appartiennent à deux acteurs différents (voire concurrents).

4.5 Le placement de chaîne de fonctions réseaux virtualisées dans des réseaux edge et de cœur

À la demande d'un client, le NFVO doit placer un service réseau (décrit par un graphe ou une chaîne de VNF) dans le réseau à la volée. Le NFVO traite les demandes une à une, au fil de leur arrivée, sans avoir connaissance des demandes futures. Dans nos travaux nous avons considéré principalement le placement d'une chaîne de VNF mais nos modèles sont facilement adaptables pour placer un graphe de VNF plus complexe. Une chaîne de VNF se caractérise par ses points physiques d'entrée et de sortie (qui peuvent être identiques), une succession de VNF et deux types de contraintes : sur les capacités des nœuds (le CPU et la mémoire) et sur les capacités des liens (le délai et la bande passante). Certaines VNF (par exemple les pare-feux, les encodeurs et les décodeurs) peuvent faire varier la bande passante requise pour les liens entre les VNF de la chaîne. La latence requise peut également être différente entre deux VNF quand le service offert par certaines VNF tolère des retards, alors que d'autres non. Par conséquent, les besoins en bande passante et en



Edge39 (edge topology)



Giul39 (flat topology)

FIGURE 4.11 – Topologies

délai sont considérés à la fois de bout-en-bout (pour l'ensemble du service) et localement (entre les VNF).

Le placement de VNF étant limité par les capacités des nœuds et des liaisons, la topologie et la distribution des ressources dans le réseau sont susceptibles de jouer un rôle important. Nous considérons deux types d'architectures, comme le montre la Figure 4.11.

Les architectures plates [3], représentatives des réseaux centraux, ne contiennent qu'un seul niveau de nœuds. Les architectures edge (décrites dans [47]) sont composées de 3 couches : coeur, agrégation et accès et sont bien adaptées pour étudier les contraintes de faible latence entre VNF. Outre la répartition spatiale de l'architecture, la répartition des ressources entre les nœuds pourrait également constituer un aspect important du problème. Dans les architectures edge, les ressources entre les nœuds ne sont pas uniformément réparties : les nœuds centraux ont des ressources plus élevées que les nœuds d'agrégation, eux-mêmes ayant plus de ressources que les nœuds d'accès. Dans les topologies plates, cependant, la distribution peut être plus aléatoire, selon l'emplacement des principaux datacenters

Nous abordons le VNFCPP axé sur la maximisation de l'acceptation du nouveau NS dans une architecture dans laquelle l'orchestrateur met en œuvre une coopération totale avec l'infrastructure. Par conséquent, nous réservons les ressources en priorité sur les liens et les nœuds où elles sont abondantes, en les préservant là où elles sont rares pour des demandes futures avec des exigences potentiellement plus fortes. Nous introduisons donc un coût $P(X)$ correspondant à l'inverse des ressources X restantes (CPU, stockage ou bande passante). Cela a pour effet de rendre plus coûteuse l'installation d'une VNF (respectivement d'un chemin) sur un nœud (respectivement un lien) avec peu de ressources. Nous avons formalisé cette stratégie comme un problème de programmation linéaire en nombres entier (ILP) détaillé dans [34]. La fonction objectif de l'ILP cherche à minimiser une somme pondérée de la bande passante, du CPU et du stockage utilisés pour instancier la chaîne de VNF.

3. Ces architectures se trouvent à `http://sndlib.zib.de`

Evaluation de l'ILP

Nous avons évalué l'amélioration apportée par notre ILP en le comparant à une solution de base qui ne prend en compte que la bande passante. Nous avons comparé le nombre moyen de services réseau instanciés par simulation avec le solveur Gurobi et un Intel Xeon E5-2630 ayant 12 cœurs logiques pour 2000 requêtes quand le réseau est proche de la saturation. Nous avons sélectionné cinq topologies de cœur de réseau disponnibles dans SDNLib : Atlanta, Brain, Cost, Germany50 et Giul39 et nous avons conçu deux topologies de type edge : Edge51 et Edge39 (voir la figure 4.11).

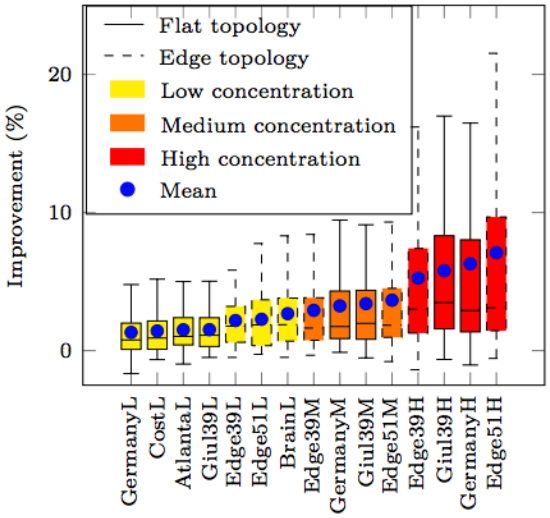


FIGURE 4.12 – Performance de l'ILP comparée à la stratégie n'optimisant que la bande passante

La figure 4.12 représente l'impact de la répartition des ressources à l'aide des trois scénarios suivants. Dans le scénario de **faible concentration des ressources** (L), tous les nœuds ont 80 unités CPU et 100 unités de stockage et tous les liens ont 1000 unités de bande passante. Nous noterons qu'il est peu probable que cela se produise dans la réalité car les datacenters de cœur de réseau ont beaucoup plus de capacité que les datacenters périphériques. Les autres scénarios se concentrent sur des topologies avec un nombre comparable de nœuds et de liens : Edge39, Edge51, Giul39 et Germany50. Le montant total des ressources du réseau est le même que dans le scénario L, mais la répartition des ressources des nœuds change. Les nœuds sont catégorisés en nœuds d'accès, d'agrégation et de cœur (ce qui est immédiat pour Edge39 et Edge51). Nous classons les nœuds de Giul39 (resp. Germany50) en sélectionnant le même nombre de nœuds dans chaque catégorie qu'avec Edge39 (resp. Edge51) en fonction de leur caractère central. La centralité la plus élevée correspond aux nœuds centraux puisque les nœuds de grande capacité sont susceptibles d'être placés à des endroits stratégiques du réseau. Dans le scénario de **concentration moyenne** (M), chaque catégorie de nœud dispose de 33 % du montant total des ressources du réseau alors que dans le scénario de **haute concentration** (H) les nœuds de cœur, d'agrégation et d'accès détiennent respectivement 60%, 30% et 10% du montant total des ressources.

Les observations montrent l'efficacité de l'ILP face à la solution ne considérant que la bande passante, surtout lorsque la concentration des ressources est élevée, ce qui est la situation la plus courante en pratique (les nœuds d'accès ont des ressources limitées

alors que les ressources du datacenter de cœur sont souvent considérées comme infinies). Ceci confirme notre hypothèse initiale : il est crucial d'épargner des nœuds avec peu de ressources afin de ne pas rejeter de futurs NS ayant de fortes exigences de latence en raison d'un manque de ressources suffisamment proches les satisfaire. L'absence d'impact du type topologique (plat ou en bordure) s'explique par leur petite taille : la longueur du trajet d'un nœud d'accès à un nœud central est assez similaire. Enfin, on constate que même pour une concentration élevée, l'amélioration médiane reste inférieure à 5%, alors que l'amélioration moyenne se situe entre 5% et 10%.

L'ILP proposé permet de prendre en compte les ressources edge et de les exploiter de façon plus pertinentes lorsque l'orchestrateur a une vision exacte des ressources exposées par le VIM. Cela est possible lorsque ces deux éléments appartiennent à une même entité mais n'est pas réaliste dans le cas d'usage général de la 5G. Dans la section suivante, nous nous intéressons à l'une des contributions de la 5G qui permet de déployer des services réseaux sur des infrastructures virtuelles offertes par plusieurs acteurs.

4.6 Le placement de fonctions réseaux virtualisées dans des architectures complexes

Dans l'architecture MANO, le NFVO décide où placer et comment connecter les VNF, en se basant sur les informations topologiques fournies par le VIM. Ensuite, les VIMs réservent des ressources dans l'infrastructure virtuelle en fonction de la décision de placement et le VNFM peut installer et exploiter le VNF. Dans une architecture mono-tenant, le NFVO et le VIM sont exploités par le même fournisseur de service réseau. Ce n'est pas le cas dans une architecture multi-tenant dans laquelle le service d'orchestration et l'infrastructure appartiennent à des entités différentes, voire concurrentes. Le NFVO sollicite alors un ou plusueurs VIMs appartenant à d'autres fournisseurs pour implanter des VNF dans leur infrastructure. Le VNFCPP tel qu'il est décrit dans la section précédente suppose la pleine coopération de chaque entité et la mise à disposition du MVNO d'informations précises sur les ressources disponibles dans l'infrastructure. Ceci n'est pas possible dans une architecture multi-tenant où les VIMs peuvent être réticents à divulguer des informations confidentielles telles que leur topologie à des entités concurrentes. Les VIMs présenteront plutôt une abstraction de leur réseau au NFVO. Nous avons introduit un calcul en deux étapes pour prendre en compte les architectures multi-tenant. Premièrement, le NFVO effectue le placement sur les topologies abstraites proposées par les VIMs, ce qui donne lieu à un premier placement "gros grain" des VNF. Ensuite, chaque VIM concerné doit à son tour exécuter l'algorithme pour déterminer le placement final dans son infrastructure et mettre à jour son graphe abstrait et la consommation des ressources.

4.6.1 L'abstraction topologique

Au fil des ans, l'abstraction de la topologie des réseaux a motivé de nombreux travaux de recherche [103], utilisés par exemple pour le routage en inter-domaine. Cependant, comme ils ne considèrent pas le placement de VNF, ces travaux ne tiennent pas compte des ressources des nœuds. Pour être efficace, l'abstraction doit afficher un compromis entre la précision de l'information (pour permettre au NFVO de calculer un chemin correct) et la façon de masquer les détails internes du NFVI.

Les principales abstractions pour une topologie de réseau sont le nœud unique, l'étoile et le maillage complet [86]. L'approche du nœud unique n'est pas assez précise et l'approche par maillage ne représente pas les ressources du nœud. De plus, le calcul d'un

maillage complet pour chaque placement de VNF candidat est impossible à effectuer en un temps raisonnable. Nous sommes donc concentrés sur la division du réseau en sous-réseaux, ou clusters, chacun représenté par une abstraction asymétrique en étoile pondérée. La division en sous-réseaux peut suivre des divisions administratives préexistantes ou un algorithme spécifique. Nous avons utilisé l'algorithme présenté dans [66] qui identifie les clusters en fonction de la métrique de *betweenness* des liens. Le compromis porte à présent sur le nombre de sous-réseaux à considérer et leur taille pour obtenir l'abstraction la plus représentative tout en assurant la non-divulgation des éléments confidentiels. La taille des clusters est arbitraire et doit optimiser à la fois le temps de calcul et la qualité d'abstraction.

Chaque cluster est représenté par une topologie abstraite en étoile comprenant un nœud central (noyau) relié par des liens abstraits (rayons) aux nœuds de bordure du cluster, eux-mêmes reliés aux clusters voisins. Le nœud ayant la plus grande centralité dans le cluster (c'est-à-dire le plus facile à atteindre de n'importe quel point du cluster, en moyenne) devient le noyau. Il est le dimensionné par la somme de toutes les ressources des nœuds qui ne sont pas à la frontière du cluster. Les paramètres de chaque rayon sont déterminés en exécutant plusieurs algorithmes de Dijkstra entre le nœud frontière du rayon et le noyau en se concentrant sur un paramètre à la fois. Malgré sa simplicité, [77] a montré l'efficacité de cette approche agressive (suggérée dans le contexte du routage traditionnel par [86]). Nous y avons ajoutés des liaisons supplémentaires en conservant les liens reliant les nœuds de bordure du sous-réseau pour faciliter la transition d'un cluster à l'autre. Par construction, le noyau central devient un nœud très attractif pour la fonction objectif, cachant potentiellement un grand nombre de nœuds d'accès avec peu de ressources. Nous atténuons ce problème en redéfinissant chaque prix comme étant le prix le plus bas pour cette ressource parmi les nœuds que représente le noyau.

4.6.2 L'heuristique dans une architecture mono-tenant

Le VNFCPP est NP-complet [54], de sorte que les temps de calcul peuvent devenir insupportables lorsque la taille du réseau ou de la chaîne de VNF augmente. Nous avons étendu la technique d'abstraction au scénario multi-tenant pour définir une heuristique qui fonctionne avec toute forme de VNFCPP. Lorsque la topologie est trop grande pour exécuter l'ILP, le NFVO divise artificiellement sa vue topologique en plusieurs clusters de taille limitée. Notez que le processus d'abstraction peut se faire hors ligne si la topologie ne varie pas rapidement dans le temps. L'exécution de l'ILP sur des topologies abstraites donne une première solution de placement qui associe chaque cluster à un ensemble de VNF à héberger. Cela correspond au premier placement fait par le NFVO qui permet de répartir les Virtual Network Function (VNF) sur les NFVI gérées par les différents VIM sollicités. L'algorithme transforme ces résultats en paramètres et exécute l'ILP une seconde fois sur chaque cluster de la topologie abstraite, ce qui correspond au travail effectué par chaque VIM pour placer les VNF qui lui ont été attribuées. Comme l'ILP initial n'est pas modifié, cette heuristique peut être appliquée à n'importe quelle variante du VNFCPP.

Evaluation de l'heuristique

Nous avons évalué l'efficacité de notre heuristique en utilisant le même environnement que dans la section 4.5. Nous avons considéré une large gamme de tailles de topologies edge (basée sur la Figure 4.11), appelées edgeN, composées de nœuds de cœur N, de groupes N de nœuds d'agrégation interconnectés par le noyau avec $2 * N + 1$ nœuds d'agrégation

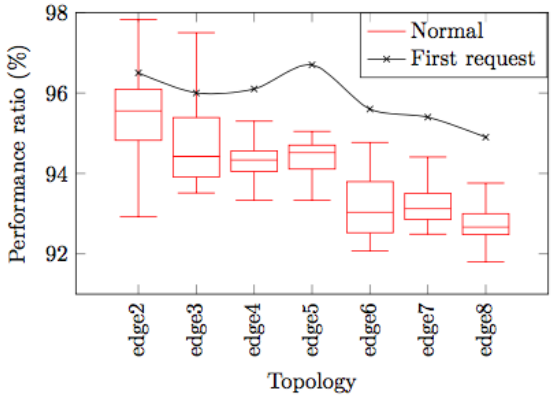


FIGURE 4.13 – Ratio d'acceptation cumulatif de l'heuristique par rapport à une solution holistique

répartis en deux couches dans chaque groupe, et de $4*N$ nœuds d'accès connectés à chaque nœud d'agrégation.

Pour analyser la performance de l'heuristique par rapport à l'ILP introduit en section 4.5, nous avons divisé le réseau en clusters composés d'un maximum de 100 éléments lorsque chaque requête tente de placer 10 VNF. Chaque nœud possède 80 unités CPU et 60 unités de stockage, et chaque liaison possède 800 unités de bande passante. La comparaison du taux d'acceptation cumulatif de l'heuristique lorsque le réseau est vide (première requête) et juste avant la saturation du réseau montre l'efficacité de l'heuristique puisque nous obtenons un ratio de performance entre 93% et 96% entre la première requête et le point de saturation du réseau. Lorsque le réseau est vide, le taux d'acceptation de la première demande n'est que de 96 % en raison de l'agressivité de notre méthode d'abstraction de topologie qui peut conduire à proposer des chemins attrayants qui ne peuvent pas être instanciés dans le NFVI. Ces résultats sont représentés dans la figure 4.13. L'analyse des temps de calcul de l'heuristique et de l'ILP holistique montre que, bien que les durées d'exécution soient comparables sur de petites instances, le temps d'exécution de l'heuristique reste stable lorsque la taille du réseau augmente. Nous avons augmenté la taille des réseaux, doublé le nombre de VNF à placer et permis à l'heuristique de former des clusters pouvant atteindre 3000 nœuds, mettant ainsi en évidence que notre heuristique peut gérer des problèmes à grande échelle sans faire face à des temps de calcul plus longs.

4.7 Conclusion

Dans ce chapitre nous avons abordé l'automatisation de la gestion de la qualité de service dans les réseaux grâce à l'ingénierie de trafic, aux architectures SDN et de virtualisation des réseaux. Les récentes évolutions sur ce sujet permettent maintenant de pallier aux principaux problèmes qui faisaient obstacle au déploiement à large échelle de la garantie de QoS pour des flux clients. En particulier, le Segment Routing offre une solution élégante pour appliquer des politiques de routages décidées grâce à des techniques d'ingénierie de trafic sans impliquer les coûts de gestion et la complexité du protocole de réservation de ressources utilisé jusqu'à présent. Cependant, les équipements actuellement déployés dans les réseaux empêchent la migration des réseaux vers le routage par segment en raison de leur capacité limitée à gérée des piles de labels dans les réseaux MPLS. Nous avons proposé plusieurs algorithmes pour contourner ce problème, y compris dans

des réseaux de grande taille grâce à la définition de programmes linéaires pour réduire l'encodage des chemins de façon à respecter le MSD. Pour être efficace, l'automatisation de l'ingénierie de trafic doit s'appuyer sur des équipements tels que le PCE, capable d'effectuer des calculs d'optimisation plus complexes mais nécessitant une vision globale et pertinente des ressources du réseau. Nous avons donc exploité les nouvelles possibilités offertes par l'architecture SDN qui permet l'automatisation de la gestion des réseaux en les rendant programmables. Couplés à la virtualisation des réseaux (grâce à l'architecture MANO), ces nouveaux paradigmes offrent la possibilité de déployer des services réseaux adaptés aux besoins des applications de façon dynamique. Pour cela, il est nécessaire de disposer d'algorithmes de placement de chaînes de fonctions réseau virtualisées permettant une gestion fine des ressources afin de satisfaire le plus grand nombre de demandes.

Nous avons formulé ce problème sous forme d'ILP capable de prendre en compte à la fois les contraintes des services réseaux (délais et bande passante) et les contraintes des nœuds du réseau (capacité de calcul et de stockage). Bien qu'elle offre une solution utilisable dans des environnements mono-tenant (lorsque l'orchestrateur et le réseau virtualisé appartiennent à la même entité), cette approche se heurte à des problèmes de performance et ne peut être appliquée dans des environnements plus généraux. Nous avons ainsi proposé une heuristique, basée sur notre ILP, qui permet à la fois de passer à l'échelle mais surtout de traiter le cas des réseaux multi-tenant (quand l'orchestrateur sollicite des réseaux virtualisés appartenant à des entités tierces pour déployer des services réseaux). À notre connaissance au moment de la rédaction de ce manuscrit, la prise en compte des problèmes de confidentialité des environnements multi-tenant n'a pas été abordée dans la littérature.

CHAPITRE 5

Bilan et perspectives

La garantie de qualité de service dans les réseaux est un sujet récurrent depuis plus de vingt ans qui n'a toujours pas trouvé de solution totalement satisfaisante. L'augmentation de la bande passante dans les réseaux a longtemps été vue comme une méthode suffisante pour obtenir un service réseau correct. Elle est de moins en moins adaptée pour faire face aux nouveaux usages (et en particulier l'importance du trafic vidéo) et aux contraintes liées aux réseaux d'accès.

Les organismes de standardisation ont imposé la thématique de la qualité de service dans leurs standards depuis de nombreuses années. L'IETF a proposé des architectures et des protocoles pour permettre de garantir la qualité du service réseau par la réservation de ressource (*i.e.* IntServ, DiffServ) ou par l'ingénierie de trafic (*i.e.* MPLS-TE, RSVP-TE). Dans ce manuscrit, nous avons abordé la prise en compte de la qualité de service dans les réseaux fixes et dans les réseaux de capteurs (urbains et industriels). Les outils employés pour cela sont d'ordre architectural, protocolaire et utilisent des méthodes d'optimisation par programmes linéaires afin d'agir sur le routage et d'offrir des garanties de qualité de service aux flux du réseau. Nous avons vu que garantir de la qualité de service dans les réseaux nécessite d'intervenir à plusieurs niveaux du modèle OSI : au niveau application, au niveau transport et au niveau réseau. Même si les applications et l'adaptation du transport des flux peuvent contribuer à améliorer la qualité de service ou la qualité d'expérience ressentie par les utilisateurs, il est nécessaire de traiter la qualité de service au niveau du réseau.

Le problème se complique encore si l'on considère plusieurs métriques de QoS ou si l'on souhaite garantir la QoS sur un chemin faisant intervenir plusieurs systèmes autonomes. Or, le domaine des réseaux subit actuellement une mutation profonde grâce à l'émergence de la gestion automatisée des réseaux et leur virtualisation. Les réseaux définis de façon logicielle (Software Defined Networks - SDN), en séparant les fonctions d'acheminement des fonctions de contrôle, changent la gestion des réseaux et la rendent automatisable et programmable, facilitant le recours à une ingénierie de trafic à large échelle. En parallèle, l'avènement de la virtualisation des réseaux et de la 5G a amené de nouveaux cas d'usages tels que l'IaaS (Infrastructure as a Service), PaaS (Platform as a Service) et le slicing qui visent à offrir un réseau adapté à un client ou à un cas d'usage particulier avec des niveaux de contrôle différents. Ces évolutions conduisent à ne plus voir les réseaux comme des infrastructures sur lesquels implanter des services mais comme des infrastructures permettant

d'implanter dynamiquement des réseaux virtualisés et adaptés soit aux besoins de clients soit à des services réseaux particuliers. Nous verrons dans les sections suivantes que mes perspectives de recherches visent à permettre une définition dynamique et automatique de réseaux virtualisés adaptés à un (ou plusieurs) comportement(s) donné(s).

5.1 Vers des réseaux dynamiques et automatisés offrant des garanties de QoS

5.1.1 Méthodes d'abstraction des ressources des réseaux virtualisés

La virtualisation des services réseaux est un des paradigmes centraux de la 5G. Le standard MANO (voir Chapitre 4), introduit un gestionnaire d'infrastructure virtualisée (le VIM) et un orchestrateur (le NFVO) pour gérer la mise en œuvre de nouveaux services réseaux. La section 4.5 aborde les problématiques liées au placement des fonctions réseaux virtualisées (VNF) dans les infrastructures de réseaux virtualisées en considérant deux cas. Dans le premier, les fonctions d'orchestration de service réseau et de gestion de l'infrastructure virtualisée appartiennent à une même entité (appelé cas mono-tenant), ce qui garantit une coopération totale entre le NFVO et le VIM. Cependant, ce cas n'est pas nécessairement représentatif de la majorité des situations. En effet, l'un des intérêts liés à la virtualisation des réseaux est de permettre l'apparition d'acteurs plus petits dans le monde des opérateurs (à l'instar des MVNO - mobile virtual network operator - pour la téléphonie mobile), capables de mettre en œuvre de services réseaux au dessus d'infrastructures gérées par des opérateurs tiers. Dans ce second cas, nommé multi-tenant, le NFVO et le VIM n'appartiennent pas à la même entité et ne partagent donc pas librement leurs connaissances. L'heuristique proposée dans la section 4.5 offre une première réponse à ce problème en construisant une vision abstraite de l'infrastructure offerte par le VIM au NFVO. Nous avons proposé une abstraction simple basée sur une partition des nœuds du réseau en fonction de leur betweenness, c'est à dire du nombre de plus courts chemins du réseau auxquels ils appartiennent. Cependant, cette approche classique n'est peut être pas la meilleure, il faut étudier **les méthodes d'abstraction d'infrastructures virtualisées** permettant au VIM d'offrir au NFVO une vision de la topologie de l'infrastructure ne divulguant pas d'informations sensibles tout en étant assez précise pour permettre au NFVO de faire des calculs pertinents d'implantation de service réseau. Nous avions effleuré un problème similaire dans le chapitre 3 avec le calcul local des chemins internes à un AS soumis à de multiples contraintes de QoS. Nous avions contourné la difficulté en utilisant une notion d'offres pour représenter les ressources disponibles dans le réseau. Cela reste utilisable dans le cadre d'une architecture MANO multi-tenant en prenant pour hypothèse que les VIMs sont capables d'exposer au NFVO des offres d'acheminement ou d'implantation de VNFs qui représentent une vision abstraite des ressources disponibles. Cependant l'architecture MANO en définissant le VIM a introduit un élément avec lequel le NFVO peut négocier les ressources à mobiliser dans l'infrastructure. Cela ouvre de nouvelles possibilités qu'il faut prendre en compte dans la méthode d'abstraction des ressources des infrastructures virtualisées.

5.1.2 Prise en compte des propriétés inter-domaines dans l'archtecture MANO

Dans le chapitre 3 nous avons vu les difficultés liées aux mécanismes opérant entre plusieurs systèmes autonomes. En particulier le fait que ces mécanismes doivent respec-

ter les propriétés d'autonomie, de confidentialité et de scalabilité. Nous retrouvons ces problématiques lors de la mise en œuvre de services réseaux dans un contexte multi-tenant. La prise en compte du multi-domaine (inter-domaine) dans les mécanismes réseau reste complexe et d'actualité dans les architectures multi-tenant et dans les architectures des réseaux qui apparaîtront dans les prochaines années. Ceci est accentué par l'émergence de la 5G et du concept de slice. Le document [62] montre qu'il existe différentes variations sur la définition du terme slice. L'ETSI définit la notion de slice (ou de slice réseau) comme la description d'un réseau logique adapté à un service et composé de différents éléments, ressources et fonctions de réseau physique ou virtuel. Les slices sont construites en silo, en utilisant des ressources virtualisées indépendantes mais partageant la même infrastructure physique. Les mécanismes utilisés pour gérer les slices et les services réseaux doivent respecter les trois propriétés de l'inter-domaine car un service réseau peut être mis en œuvre en faisant intervenir plusieurs infrastructures virtualisées. L'étude de l'impact des propriétés des protocoles inter-domaine prend ainsi une nouvelle dimension. Dans les efforts de standardisation menés en particulier à l'IETF sur les protocoles de l'Internet, la prise en compte de la dimension inter-domaine qui n'était pas prioritaire (sauf pour BGP) est maintenant essentielle. Ainsi, depuis quelques années, les mécanismes de routage dans les réseaux fixes sont adaptés pour pouvoir transporter des attributs permettant de choisir des routes en inter-domaine. Par exemple, il est maintenant possible de propager les identifiants utilisés pour le routage par segment (voir la section 4.2) vers les domaines voisins grâce à BGP-LS, simplifiant la prise en compte de la qualité de service dans le routage en inter-domaine. L'intégration de l'architecture MANO dans les réseaux de l'Internet ouvre la voie à une meilleure prise en compte de calculs distribués inter-domaines de chemins garantissant des contraintes de qualité de service, ce qui est une perspective d'études importante pour la suite de mes travaux.

5.2 Vers la création de réseaux virtualisés personnalisables

L'une des promesses majeures de la virtualisation des réseaux est la possibilité de définir des infrastructures de communications personnalisées afin de déployer de nouveaux services facilement et à coût raisonnable. Pour cela, les architectures SDN et MANO offrent de nouveaux moyens de contrôle et d'automatisation de la gestion des réseaux qui remplacent les interventions physiques d'équipes pour déployer des équipements réseaux par des déploiements logiciels, réduisant le temps et la complexité de création d'un réseau adapté à un besoin client. De même, l'automatisation du contrôle du réseau rend maintenant la généralisation de l'ingénierie de trafic possible. Un contrôleur SDN est nativement capable de programmer les équipements réseaux pour qu'ils mettent en œuvre des politiques d'ingénierie de trafic. Tout cela ouvre la possibilité de définir en un temps réduit une infrastructure réseau virtualisée adaptée aux besoins d'un service réseau. Cette adaptation du réseau est mis en avant dans la 5G pour intégrer de nouveaux cas d'usage et s'adapter aux besoins de secteurs économiques variés.

Cependant, de nouveaux verrous émergent d'un tel concept. En particulier, la création de slice à la demande (ou Slice as a Service) implique l'adaptation de l'infrastructure de réseau virtuelle grâce au déploiement dynamique de fonctions réseaux virtualisées aux endroits les plus adaptés, mettant l'accent sur la gestion des ressources lors du placement des VNF. Les travaux effectués jusqu'à maintenant ont mis en évidence la complexité d'une gestion fine de ces ressources. Les demandes de placement de service étant traitées au fur et à mesure de leur arrivée, les choix de placement ne sont pas optimaux. Si l'heuristique proposée dans la section 4.6 offre une première solution, elle doit être étoffée pour prendre en

compte la mutualisation des VNF (lorsque le service le permet) et la réoptimisation des placements de fonctions. Comme pour le routage dans les réseaux classique, la réoptimisation du placement des fonctions virtuelles implique des verrous importants. Le premier est **le calcul de la réoptimisation**, c'est à dire d'un nouveau placement des VNF qui a pour but de mieux utiliser les ressources. Ce problème, s'il est également exprimé grâce à des modèles de programmation linéaires, diffère d'un placement originel car le but est généralement de déplacer le moins de VNF possible afin de minimiser l'impact de la réoptimisation sur les services rendus. Le second verrou est lié à **la mise en œuvre de la réoptimisation** qui suppose l'instantiation de nouveaux composants de VNF et le reroutage du trafic avant la destruction de la solution initiale. Ce principe, appelé *make before break*, vise à assurer la continuité du service y compris en phase de migration des VNFs. Le troisième verrrou concerne l'évaluation de l'état du réseau avant la réoptimisation. Afin d'être efficace, le nouveau placement des VNFs doit tenir compte d'une vision à jour des ressources disponibles ou consommées dans le réseau. Cela nécessite **des fonctions de monitoring adaptées au contexte des fonctions virtualisées**. Le monitoring est également utilisé pour quantifier l'aptitude du réseau à accepter de nouvelles demandes de services tout au long de la vie du réseau.

5.3 Gestion de topologies éphémères

Le déploiement à la demande de slices repose sur la création et l'adaptation d'une topologie virtualisée de réseau s'adaptant aux besoins définis par des utilisateurs, des services ou des applications. Grâce aux architectures SDN et NFV, il est maintenant envisageable de déployer un réseau personnalisé à la demande et de le détruire en fin d'utilisation. Cela ouvre la voie à de nouveaux cas d'usage tel que la définition de réseaux éphémères : des réseaux dédiés à courte durée de vie, capables de passer à l'échelle et de s'adapter rapidement à un changement de la demande. Ainsi, le réseau 5G est en cours de virtualisation pour s'adapter à la charge du trafic de façon réactive en déployant à la volée certains composants. Cela illustre la reconfiguration à chaud et l'adaptation dynamique à la charge d'un réseau ayant une grande durée de vie grâce à l'ajout ou à la destruction de fonctions virtualisées ayant une durée de vie réduite pour répondre à un besoin ponctuel.

Ces évolutions récentes vont avoir un impact important sur les réseaux au niveau architectural et protocolaire mais également sur les usages, en particulier dans les environnements urbains (bâtiment, routes,...) qui, en devenant intelligents et connectés, permettant le calcul et la fourniture de nouveaux services à la population. La ville intelligente est donc un terrain d'expérimentation idéal pour mettre en évident les avantages d'une gestion dynamique et automatique de services réseaux virtualisés. La création de réseaux dédiés lors d'évènements sportifs ou culturels est un autre exemple mettant en lumière la définition de réseaux éphémères : des réseaux temporaires ayant une durée de vie courte, proche de celle de l'évènement pour lesquels ils sont déployés. Dans une architecture classique, le déploiement de réseaux éphémères coûte cher et nécessite une préparation importante. Grâce à l'automatisation des réseaux et à la virtualisation, la mise en place de tels réseaux se traduit par la réservation de ressources offertes par une infrastructure virtualisée pour y installer les VNFs nécessaires au fonctionnement des services réseaux. Or, la gestion des infrastructures physiques et virtualisées est complexe et nécessite une bonne connaissance des ressources utilisées, des services implantés et surtout des ressources disponibles. L'architecture MANO peut offrir les fonctionnalités nécessaires mais manque de maturité. Il est donc nécessaire d'étudier la création de réseaux éphémères afin d'en évaluer la complexité et les performances.

5.3.1 Résilience de ces infrastructures de communication

La capacité de déployer des VNFs rapidement offre de nouvelles possibilités pour garantir les propriétés de connexité des réseaux nécéssaires à la plupart des protocoles. Un cas d'usage est la garantie de la résilience des infrastructures de communication, y compris en cas de catastrophe (naturelle ou non). En particulier, il est essentiel d'aider les équipes de secours arrivant sur les lieux d'un sinistre, par exemple en partageant les informations recueillies par les capteurs citoyens ou urbains. Cela nécessite une infrastructure de communication globale regroupant les données issues des infrastructures fixes ou mobiles déployées dans la ville ou par les particuliers. Or les infrastructures existant avant la catastrophe ont pu être considérablement fragilisées ou endommagées. Il s'agit alors en premier lieu de reconstruire une infrastructure réseau résiliente à partir des éléments subsistants. Il faut pouvoir évaluer l'état de la topologie actuelle et la consolider, si besoin, en prévoyant le déploiement rapide de nouveaux relais pour le routage des informations. Le verrou technologique est donc de recréer une infrastructure opérationnelle à partir d'éléments variés et parfois isolés.

Dans la littérature, plusieurs travaux considèrent l'utilisation de robots ou de drones pour recréer une topologie connexe, d'autres prône le déploiement rapide de communications satellites pour relier les portions d'infrastructure réseau restées intactes. Nous proposons de tirer également partie des capacités de relayage des communication et des capteurs des smartphones des personnes présentes sur les lieux. Le routage peut alors utiliser l'ingénierie de trafic et s'appuyer à la fois sur SDN et le routage par segment pour améliorer la résilience du système de détection d'événements liés au sinistre. En effet, Segment Routing et son routage par la source permet de mettre en œuvre des stratégies de routage efficaces et des techniques d'allocation des ressources appropriées dans les nœuds qui ne sont pas compatibles avec SDN. Ces travaux futurs s'inscrivent dans la continuité des travaux menés sur le routage dans l'IoT, sur le segment routing et sur la virtualisation des réseaux.

Références personnelles

[1] E. Anceaume, Y. Busnel, P. Lajoie-Mazenc, and G. Texier. Reputation for inter-domain qos routing. In *International Symposium on Network Computing (NCA)*, page 5. IEEE, 2015.

[2] A. Bellabas, G. Bertrand, S. Lahoud, G. Texier, M. Molnar, et al. Quality of service routing in next generation networks. In *Colloque Francophone sur l'Ingenierie des Protocoles (CFIP)*, 2009.

[3] A. Bellabas, G. Texier, S. Lahoud, and A. K. Najah. Convergent iptv services over ip multimedia subsystem. In *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pages 1–5. IEEE, 2011.

[4] G. Bertrand, S. Lahoud, M. Molnár, G. Texier, et al. Inter-domain path computation with multiple qos constraints. *Recent Advances in Providing QoS and Reliability in the Future Internet Backbone*, 2010.

[5] G. Bertrand, S. Lahoud, M. Molnár, G. Texier, et al. Qos routing and management in backbone networks. *Intelligent Quality of Service Technologies and Network Management : Models for Enhancing Communication*, pages 138–159, 2010.

[6] G. Bertrand, S. Lahoud, G. Texier, and M. Molnár. Computation of multi-constrained paths in multi-domain mpls-te networks. In *Next Generation Internet Networks, 2009. NGI'09*, pages 1–8. IEEE, 2009.

[7] G. Bertrand, S. Lahoud, G. Texier, and M. Molnár. A distributed exact solution to compute inter-domain multi-constrained paths. *The Internet of the Future*, pages 21–30, 2009.

[8] G. Bertrand and G. Texier. Diffserv-aware flow amission control and resource allocation modeling. In *EuroFGI Workshop on IP QoS and Traffic Control, Lisbon, Portugal*, 2007.

[9] G. Bertrand and G. Texier. Intégration du routage pce aux réseaux de prochaine génération avec ims. 2008.

[10] B. Billet, V. Issarny, and G. Texier. Composing continuous services in a coap-based iot. In *AI & Mobile Services (AIMS), 2017 IEEE International Conference on*, pages 46–53. IEEE, 2017.

[11] Z. Brodard, H. Jiang, T. Chang, T. Watteyne, X. Vilajosana, P. Thubert, and G. Texier. Rover : Poor (but elegant) man's testbed. In *Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pages 61–65. ACM, 2016.

[12] J. Choi, G. Texier, Y. Seok, T. Kwon, L. Toutain, and Y. Choi. Interoperability experiences on integrating between different active measurement systems. *Information Networking. Advances in Data Communications and Wireless Networks*, pages 630–638, 2006.

[13] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon. Cross-layer scheduler for video streaming over mptcp. *Proc. of ACM MMSys*, 2016.

[14] X. Corbillon, F. Boyrivent, G. A. De Williencourt, G. Simon, G. Texier, and J. Chakareski. Efficient lightweight video packet filtering for large-scale video data delivery. In *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[15] J. Corral, M. Ourraou, G. Texier, and L. Toutain. Une architecture pour mesurer les performances des classes de service ip. *TSI. Technique et science informatiques*, 24(4) :422–448, 2005.

[16] J. Corral, G. Texier, and L. Toutain. End-to-end active measurement architecture in ip networks (saturne). *Proceedings of PAM2003*, 2003.

[17] O. Dugeon, R. Guedrez, S. Lahoud, and G. Texier. Demonstration of segment routing with sdn based label stack optimization. In *Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on*, pages 143–145. IEEE, 2017.

[18] J.-M. Gilliot, G. Texier, X. Lagrange, G. SIMON, and M. Briand. Intégrer des MOOC dans une formation d'ingénieurs. In *QPES 2015 : questions de Pédagogies dans l'Enseignement Supérieur*, Brest, France, June 2015.

[19] R. Guedrez, O. Dugeon, S. Lahoud, and G. Texier. Label encoding algorithm for mpls segment routing. In *Network Computing and Applications (NCA), 2016 IEEE 15th International Symposium on*, pages 113–117. IEEE, 2016.

[20] R. Guedrez, O. Dugeon, S. Lahoud, and G. Texier. A new method for encoding mpls segment routing te paths. In *Network of the Future (NOF), 2017 8th International Conference on th*, pages 58–65. IEEE, 2017.

[21] P. Houzé, E. Mory, G. Texier, and G. Simon. Applicative-layer multipath for low-latency adaptive live streaming. In *IEEE International Conference on Communications (ICC'16)*, 2016.

[22] V. Issarny, G. Bouloukakis, N. Georgantas, F. Sailhan, and G. Texier. When service-oriented computing meets the iot : A use case in the context of urban mobile crowd-sensing. In *European Conference on Service-Oriented and Cloud Computing*, pages 1–16. Springer, 2018.

[23] R. Jacquet, G. Texier, and A. Blanc. Sanp : An algorithm for selecting end-to-end paths with qos guarantees. In *Future Network and Mobile Summit (FutureNetwork-Summit), 2013*, pages 1–10. IEEE, 2013.

[24] R. Jacquet, G. Texier, and A. Blanc. Computing end-to-end qos paths in the internet considering multiple alliances. In *Telecommunications Network Strategy and Planning Symposium (Networks), 2014 16th International*, pages 1–6. IEEE, 2014.

[25] H. Jiang, Z. Brodard, T. Chang, A. Bouabdallah, N. Montavont, G. Texier, P. Thubert, T. Watteyne, and G. Z. Papadopoulos. Competition : Controlled replication for higher reliability and predictability in industrial iot networks. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, pages 282–283. Junction Publishing, 2017.

[26] L. Jiayi, C. Rosenberg, G. Simon, G. Texier, et al. User-centric discretized delivery of rate-adaptive live streams in underprovisioned cdn networks. *IEEE Journal on Selected Areas in Communications*, 32(4) :706–718, 2014.

[27] S. Lahoud, G. Texier, and L. Toutain. Classification and evaluation of constraint-based routing algorithms for mpls traffic engineering. *6ème rencontres francophones sur les aspects algorithmiques des télécommunications (AlgoTel 2004) ; Batzsur-mer, France*, 2004.

[28] S. Lahoud, G. Texier, and L. Toutain. Fate : a polynomial time framework for flow allocation in mpls-te networks. In *Local and Metropolitan Area Networks, 2005. LANMAN 2005. The 14th IEEE Workshop on*, pages 6–pp. IEEE, 2005.

[29] S. Lahoud, G. Texier, and L. Toutain. Off-line flow allocation for traffic engineering in mpls networks. *LANMAN Greece*, 2005.

[30] S. Lahoud, G. Texier, and L. Toutain. Approximation algorithm for minimum cost flow allocation with varied survivability. In *Next Generation Internet Design and Engineering, 2006. NGI'06. 2006 2nd Conference on*, pages 8–pp. IEEE, 2006.

[31] S. Lahoud, G. Texier, and L. Toutain. Flow allocation with path protection in next generation internet networks. In *Communications, 2006. ICC'06. IEEE International Conference on*, volume 2, pages 860–865. IEEE, 2006.

[32] J. Liu, C. Rosenberg, G. Simon, and G. Texier. Optimal delivery of rate-adaptive streams in underprovisioned networks. *Selected Areas in Communications, IEEE Journal on*, 32(4) :706–718, 2014.

[33] O. Medina and G. Texier. La différenciation de services. *Techniques de l'ingénieur. Télécoms*, (TE7565), 2004.

[34] C. Morin, G. Texier, C. Caillouet, G. Desmangles, and C. Thanh Phan. Vnf placement algorithms to address the mono- and multi-tenant issues in edge and core networks. In *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*, To be published 2019.

[35] C. Morin, G. Texier, and C.-T. Phan. On demand qos with a sdn traffic engineering management (stem) module. In *Network and Service Management (CNSM), 2017 13th International Conference on*, pages 1–6. IEEE, 2017.

[36] G. Z. Papadopoulos, T. Matsui, P. Thubert, G. Texier, T. Watteyne, and N. Montavont. Leapfrog collaboration : Toward determinism and predictability in industrial-iot applications. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[37] P. Rolin, L. Toutain, G. Texier, O. Paul, and C. Chaudet. *Les réseaux : principes fondamentaux*. 2016.

[38] M. Y. Saidi and G. Texier. Hybrid scale-based approximation algorithm for qos routing. In *Proceedings of the 2011 11th International Conference on Telecommunications (ConTEL)*, pages 367–374. IEEE, 2011.

[39] G. Texier. *Gestion des interactions implicites dans des environnements de travail coopératifs*. PhD thesis, 2000.

[40] G. Texier and V. Issarny. Leveraging the power of the crowd and offloading urban iot networks to extend their lifetime. In *LANMAN 2018 : IEEE International Symposium on Local and Metropolitan Area Networks*, 2018.

[41] G. Texier and O. Medina. Métrologie des réseaux. *Techniques de l'ingénieur. Télécoms*, (TE7605), 2005.

[42] G. Texier and N. Plouzeau. Automatic management of sessions in shared spaces. In *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, 1999.

[43] G. Texier and N. Plouzeau. Automatic management of sessions in shared spaces. *The Journal of Supercomputing*, 24(2) :173–181, 2003.

Références extérieures

[44] Cisco visual networking index : Forecast and trends, 2017–2022 white paper.

[45] Software-Defined Networking (SDN) Definition - Open Networking Foundation.

[46] Network Functions Virtualisation, An Introduction, Benefits, Enablers, Challenges & Call for Action. ETSI, Oct. 2012.

[47] B. Addis, D. Belabed, M. Bouet, and S. Secci. Virtual network functions placement and routing optimization. In *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, pages 171–177. IEEE, 2015.

[48] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows : Theory, Algorithms, and Applications*. Prentice-Hall, Inc., New Jersey, 1993.

[49] D. Barth, T. Mautor, and D. V. Monteiro. Impact of alliances on end-to-end qos satisfaction in an interdomain network. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[50] S. Batabyal and P. Bhaumik. Mobility models, traces and impact of mobility on opportunistic routing algorithms : A survey. *IEEE Communications Surveys & Tutorials*, 17(3) :1679–1707, 2015.

[51] G. Bertrand. *Mécanismes de routage inter-domaine multi-critère : vers des services inter-opérateurs à performances garanties*. PhD thesis, Télécom Bretagne, 2009.

[52] F. Carpio, S. Dhahri, and A. Jukan. VNF placement with replication for Loac balancing in NFV networks. In *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017.

[53] R. Casellas, R. Muñoz, R. Martínez, R. Vilalta, L. Liu, T. Tsuritani, I. Morita, V. López, O. G. de Dios, and J. P. Fernández-Palacios. Sdn orchestration of openflow and gmpls flexi-grid networks with a stateful hierarchical pce. *Journal of Optical Communications and Networking*, 7(1) :A106–A117, 2015.

[54] Z. Chen, S. Zhang, C. Wang, Z. Qian, M. Xiao, J. Wu, and I. Jawhar. A Novel Algorithm for NFV Chain Placement in Edge Computing Environments. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2018.

[55] E. Crabbe, I. Minei, J. Medved, and R. Varga. Path computation element communication protocol (pcep) extensions for stateful pce. RFC 8231, RFC Editor, September 2017.

[56] Y. Cui and J. Wu. Clustering-based distributed precomputation for quality-of-service routing. In *International Conference on Computational Science*, pages 551–558. Springer, 2005.

[57] Y. Cui, K. Xu, and J. Wu. Precomputation for multiconstrained qos routing in high-speed networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1414–1424. IEEE, 2003.

[58] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee. Devoflow : scaling flow management for high-performance networks. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 254–265. ACM, 2011.

[59] H. De Neve and P. Van Mieghem. TAMCRA : a tunable accuracy multiple constraints routing algorithm. *Computer Communications*, 23(7) :667–679, 2000.

[60] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *ACM Conference on Electronic Commerce (EC)*, pages 150–157, Minneapolis, Minnesota, USA, Oct. 2000.

[61] A. Dwaraki and T. Wolf. Adaptive Service-Chain Routing for Virtual Network Functions in Software-Defined Networks. pages 32–37. ACM Press, 2016.

[62] ETSI. Network functions virtualisation (nfv) release 3 ; evolution and ecosystem ; report on network slicing support with etsi nfv architecture framework. Technical Report GR NFV-EVE 012 V3.1.1, ETSI ISG, December 2017.

[63] ETSI GS NFV 002. Network functions virtualization (NFV) ; architectural framework v1.1.1. Technical report, ETSI, October 2013.

[64] M. Fiedler, T. Hossfeld, and P. Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2), 2010.

[65] N. Finn, P. Thubert, B. Varga, and J. Farkas. Deterministic networking architecture. Internet-Draft draft-ietf-detnet-architecture-08, IETF Secretariat, September 2018. `http://www.ietf.org/internet-drafts/draft-ietf-detnet-architecture-08.txt`.

[66] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1) :35–41, Mar. 1977.

[67] R. G. Garroppo, S. Giordano, and L. Tavanti. A survey on multi-constrained optimal path computation : Exact and approximate algorithms. *Computer Networks*, 54(17) :3081–3107, 2010.

[68] J. Gil Herrera and J. F. Botero. Resource Allocation in NFV : A Comprehensive Survey. *IEEE Transactions on Network and Service Management*, 13(3) :518–532, Sept. 2016.

[69] S. Gisdakis, T. Giannetsos, and P. Papadimitratos. Security, privacy, and incentive provision for mobile crowd sensing systems. *IEEE Internet of Things Journal*, 3(5) :839–853, 2016.

[70] J. A. Hawkinson and T. J. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930, Mar. 1996.

[71] W. Hua, S. Zhao, G. Anfeng, and Y. Chaoying. An optimized ant colony algorithm based on the gradual changing orientation factor for multi-constraint QoS routing. *Computer Communications*, 32(4) :586–593, 2009.

[72] O. Iova, F. Theoleyre, and T. Noel. Using multiparent routing in rpl to increase the stability and the lifetime of the network. *Ad Hoc Networks*, 29 :45–62, 2015.

[73] V. Issarny, V. Mallet, K. Nguyen, P.-G. Raverdy, F. Rebhi, and R. Ventura. Dos and don'ts in mobile phone sensing middleware : Learning from a large-scale experiment. In *Proceedings of the 17th International Middleware Conference*, page 17. ACM, 2016.

[74] R. Jacquet. *Cooperation in networks and end-to-end quality of service guarantees for the Internet*. PhD thesis, Télécom Bretagne ; Université de Rennes 1, 2015.

[75] J. M. Jaffe. Algorithms for finding paths with multiple constraints. *Networks*, 14 :95–116, 1984.

[76] W. Kim, P. Sharma, J. Lee, S. Banerjee, J. Tourrilhes, S.-J. Lee, and P. Yalagandula. Automated and scalable qos control for network convergence. *INM/WREN*, 10(1) :1–1, 2010.

[77] T. Korkmaz and M. Krunz. Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks. *ACM Transactions on Modeling and Computer Simulation*, 10(4) :295–325, Oct. 2000.

[78] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *IEEE INFOCOM*, volume 2, pages 834–843, 2001.

[79] H. Krishna, N. L. van Adrichem, and F. A. Kuipers. Providing bandwidth guarantees with openflow. In *Communications and Vehicular Technologies (SCVT), 2016 Symposium on*, pages 1–6. IEEE, 2016.

[80] F. Kuipers and P. Van Mieghem. The impact of correlated link weights on QoS routing. In *IEEE INFOCOM*, volume 2, pages 1425–1434, 2003.

[81] F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz. An overview of constraint-based path selection algorithms for QoS routing. *IEEE Communications Magazine*, 40(12) :50–55, Dec. 2002.

[82] N. Kumar and G. Saraph. End-to-end qos in interdomain routing. In *Networking and Services, 2006. ICNS'06. International conference on*, pages 82–82. IEEE, 2006.

[83] M. L. Lamali, D. Barth, and J. Cohen. Reputation-aware learning for sla negotiation. In *NETWORKING 2012 Workshops*, volume 7291 of *Lecture Notes in Computer Science*, pages 80–88. 2012.

[84] N. Le Sauze, A. Chiosi, R. Douville, H. Pouyllau, H. Lonsethagen, P. Fantini, C. Palasciano, A. Cimmino, M. C. Rodriguez, O. Dugeon, et al. Etics : Qos-enabled interconnection for future internet services. *Future network and mobile summit*, 2010.

[85] H. Lee, M. Wicke, B. Kusy, O. Gnawali, and L. Guibas. Data stashing : energy-efficient information delivery to mobile sinks through trajectory prediction. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 291–302. ACM, 2010.

[86] W. C. Lee. Topology aggregation for hierarchical routing in ATM networks. *ACM SIGCOMM Computer Communication Review*, 25(2) :82–92, Apr. 1995.

[87] J. Martocci, P. Mil, N. Riou, and W. Vermeylen. Building Automation Routing Requirements in Low-Power and Lossy Networks. RFC 5867, June 2010.

[88] A. Orda and A. Sprintson. Qos routing : the precomputation perspective. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 128–136. IEEE, 2000.

[89] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3) :276–286, 2010.

[90] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados. Energy-efficient routing protocols in wireless sensor networks : A survey. *IEEE Communications surveys & tutorials*, 15(2) :551–591, 2013.

[91] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel. Thorough iot testbed characterization : From proof-of-concept to repeatable experimentations. *Computer Networks*, 119 :86–101, 2017.

[92] M. Patel, D. Sabella, N. Sprecher, V. Young, and Y. C. Hu. Mobile edge computinga key technology towards 5g. *ETSI white paper*, 11(11), 2015.

[93] C. Pham, N. H. Tran, S. Ren, W. Saad, and C. S. Hong. Traffic-aware and Energy-efficient vNF Placement for Service Chaining : Joint Sampling and Matching Approach. *IEEE Transactions on Services Computing*, 2017.

[94] K. Pister, T. Phinney, P. Thubert, and S. Dwars. Industrial Routing Requirements in Low-Power and Lossy Networks. RFC 5673, Oct. 2009.

[95] H. Pouyllau and G. Carofiglio. Inter-carrier sla negotiation using q-learning. *Telecommunication Systems*, 52(2) :611–622, 2013.

[96] H. Pouyllau and R. Douville. End-to-end qos negotiation in network federations. In *IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps)*, pages 173–176, 2010.

[97] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi. Sdn and pce implementations for segment routing. In *2015 20th European Conference on Networks and Optical Communications - (NOC)*, pages 1–4, June 2015.

[98] W. Shanshan, H. Ying, and Y. Yuan. Approach for Multiple Constraints Based Qos Routing Problem of Network. *Hybrid Intelligent Systems, International Conference on*, 2 :66–69, 2009.

[99] D. Shin, E. Chong, and H. Siegel. A multiconstraint QoS routing scheme using the depth-first search method with limited crankbacks. In *IEEE workshop on high performance switching and routing*, pages 385–383, 2001.

[100] O. Soualah, M. Mechtri, C. Ghribi, and D. Zeghlache. Energy Efficient Algorithm for VNF Placement and Chaining. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 579–588, Madrid, Spain, May 2017. IEEE.

[101] Q. Sun, P. Lu, W. Lu, and Z. Zhu. Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement. In *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016.

[102] S. Tomovic, N. Prasad, and I. Radusinovic. Sdn control framework for qos provisioning. In *Telecommunications Forum Telfor (TELFOR), 2014 22nd*, pages 111–114. IEEE, 2014.

[103] S. Uludag, K.-S. Lui, K. Nahrstedt, and G. Brewster. Analysis of topology aggregation techniques for QoS routing. *ACM Computing Surveys (CSUR)*, 39(3) :7, 2007.

[104] P. Van Mieghem, H. De Neve, and F. Kuipers. Hop-by-hop quality of service routing. *Computer Networks*, 37(3-4) :407–423, 2001.

[105] P. Van Mieghem, H. De Neve, and F. A. Kuipers. Hop-by-hop quality of service routing. *Computer Networks*, 37(3/4) :407–423, 2001.

[106] P. Van Mieghem and F. Kuipers. SAMCRA - Concepts of Exact QoS Routing Algorithms. *IEEE/ACM Transactions on networking*, 12(5) :851–863, May 2004.

[107] P. Van Mieghem and F. A. Kuipers. Concepts of exact QoS routing algorithms. *IEEE/ACM Trans. Netw.*, 12(5) :851–864, 2004.

[108] J. Vasseur. Terms Used in Routing for Low-Power and Lossy Networks. RFC 7102, Jan. 2014.

[109] J. Vasseur, A. Farrel, and A. Ayyangar. A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering. RFC 4726, Nov. 2006.

[110] D. Veldhuizen. Multiobjective evolutionary algorithms : classifications, analyses, and new innovations. 1999. *Air Force Institute of Technology*, page 249, 1999.

[111] Z. Wang and J. Crowcroft. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE J. Sel. Areas Commun.*, 14(7) :1228–1234, 1996.

[112] T. Watteyne, A. Mehta, and K. Pister. Reliability through frequency diversity : why channel hopping makes sense. In *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 116–123. ACM, 2009.

[113] T. Watteyne, T. Winter, D. Barthel, and M. Dohler. Routing Requirements for Urban Low-Power and Lossy Networks. RFC 5548, May 2009.

[114] I. Wijnands, E. Rosen, A. Dolganow, T. Przygienda, and S. Aldrin. Multicast using bit index explicit replication (bier). RFC 8279, RFC Editor, November 2017.

[115] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. Rpl : Ipv6 routing protocol for low-power and lossy networks. RFC 6550, RFC Editor, March 2012. `http://www.rfc-editor.org/rfc/rfc6550.txt`.

[116] Y. Xin and L. Xingming. Heuristic Algorithms for Multi-Constrained Quality of Service Routing. pages 844–853, 2001.

[117] G. Xue, A. Sen, W. Zhang, J. Tang, and K. Thulasiraman. Finding a path subject to many additive QoS constraints. *IEEE/ACM Trans. Netw.*, 15(1) :201–211, 2007.

[118] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman. Polynomial time approximation algorithms for multi-constrained qos routing. *IEEE/ACM Transactions on Networking*, 16(3) :656–669, June 2008.

[119] S. Yamamoto, T. Emori, and K. Takai. Field wireless solution based on isa100. 11a to innovate instrumentation. Technical report, Yokogawa Technical Report English Edition, 2010.

[120] X. Yuan. Heuristic algorithms for multiconstrained quality-of-service routing. *IEEE/ACM Trans. Netw.*, 10(2) :244–256, 2002.

[121] Y. Zheng, T. Korkmaz, W. Dou, and J. Tian. Highly responsive and efficient qos routing using pre-and on-demand computations along with a new normal measure. *Computer Networks*, 50(18) :3743–3762, 2006.

[122] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms : a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4) :257–271, 1999.

Bibliographie

[123] F. Kuipers and P. Van Mieghem. The impact of correlated link weights on QoS routing. In *IEEE INFOCOM*, volume 2, pages 1425–1434, 2003.

[124] P. Van Mieghem, H. De Neve, and F. Kuipers. Hop-by-hop quality of service routing. *Computer Networks*, 37(3-4) :407–423, 2001.

[125] P. Van Mieghem and F. A. Kuipers. Concepts of exact QoS routing algorithms. *IEEE/ACM Trans. Netw.*, 12(5) :851–864, 2004.

CHAPITRE 6

Liste des acronymes

6TiSCH IPv6 over the TSCH mode of IEEE 802.15.4e
6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks
ANR Agence Nationale de la Recherche
AS Autonomous System
BGP Border Gateway Protocol
BIER Bit Index Explicit Replication
BIER-TE Bit Index Explicit Replication - Traffic Engineering
BIFT Bit Index Forwarding Table
BLE Bluetooth Low Energy
BRPC Backward Recursive PCE-based Computation
CDN Content Delivery Network
CoAP Constrained Application Protocol
DASH Dynamic Adaptive Streaming over HTTP
DODAG Destination Oriented Directed Acyclic Graph
DTN Delay-Tolerant Networking
EGP Exterior Gateway Protocol
HEVC High Efficiency Video Coding
IaaS Infrastructure as a Service
IETF Internet Engineering Task Force
IGP Interior Gateway Protocol
IIoT Internet des objets industriel
ILP Integer Linear Programming
IoT Internet of Things
IPv6 Internet Protocol version 6
LDP Label Distribution Protocol
LLN Low Power and Lossy Networks

LSP Label Switch Path
MANO Management and Orchestration
MCP Multi-Constrained Path
MCOP Multi-Constrained Optimal Path
MPLS Multi-Protocol Label Switching
MPLS-TE MPLS Traffic Engineering
MPTCP Multi-Path Transmission Control Protocol
MSD Maximum SID Depth
NaaS Network as a Service
NFV Network Functions Virtualisation
NFVI Network Functions Virtualisation Infrastructure
NFVO Network Functions Virtualisation Orchestrator
NS Network Service
OAM Operations, Administration and Maintenance
OSS/BSS Operations Support System/Business Support System
PaaS Platform as a Service
PCE Path Computation Element
PCEP Path Computation Element communication Protocol
PDR Packet Delivery Ratio
PRE Packet Replication and Elimination
QoE Quality of Experience
QoS Quality of Service
RDA Reverse Dijkstra's Algorithm
RPL IPv6 Routing Protocol for Low-Power and Lossy Networks
RSVP Resource Reservation Protocol
RSVP-TE Resource Reservation Protocol - Traffic Engineering
SAMCRA Self Adaptive Multi-Constrained Routing Algorithm
SDN Software Defined Network
SD-WAN Software-Defined networking in a Wide Area Network
SID Segment IDentifier
SR Segment Routing
SR-MPLS MPLS Segment Routing
SR-IPv6 IPv6 Segment Routing
SRP Segment Routing Path
TCP Transmission Control Protocol
TED Traffic Engineering Database
TSCH TimeSlotted Channel Hopping
TSID Targeted-SID
VIM Virtualized Infrastructure Manager

VNF Virtual Network Function

VNFCPP Network Function Chain Placement Problem

VNFGPP Virtual Network Function Graph Placement Problem

VNFM Virtual Network Function Manager

VPN Virtual Private Network

WSN Wireless Sensor Network

Résumé

Le trafic global véhiculé sur l'Internet connaît une croissance inouïe du fait des profonds changement des usages (Internet des Objets (IoT), le trafic vidéo, ...). À l'heure actuelle, l'Internet fonctionne sur un mode appelé *Best Effort* (sans garantie de service ou de performance) reposant sur le surdimensionnent des réseaux opérateurs qui ne suffira pas pour garder un service acceptable de l'Internet. L'alternative est de mieux gérer le trafic grâce aux mécanismes de qualité de service (QoS) et d'ingénierie de trafic.

Ce manuscrit fait une synthèse de mes travaux de recherche, à la fois d'ordre architectural et protocolaire, avec pour fil conducteur l'utilisation de techniques d'ingénierie de trafic et la formalisation des problèmes de routage sous forme de programmes linéaires pour permettre une meilleure gestion des ressources et faire face aux imminentes évolutions profondes de l'Internet. J'ai porté mon attention sur les flux en transit chez les opérateurs et dans différents contextes (les applications multimédia et de diffusion de la vidéo, les réseaux de capteurs ou les villes intelligentes).

Parallèlement, l'émergence des architectures *Software Defined Networking* (SDN) et la virtualisation des fonctions réseaux (NFV) visant à automatiser la gestion des réseaux tout en la rendant dynamique introduit de nouveaux mécanismes pour pousser l'utilisation de l'ingénierie de trafic jusqu'à pouvoir proposer une adaptation dynamique d'infrastructures de réseaux virtualisées. Cela offre la perspective prometteuse de pouvoir mutualiser une infrastructure physique entre plusieurs réseaux virtualisés tout en adapter leur structure aux besoins spécifiques des clients.

Summary

The global traffic carried on the Internet is experiencing unprecedented growth due to profound changes in usage (Internet of Things (IoT), video traffic, etc.). At present, the Internet operates in a mode called Best Effort (without any guarantee of service or performance) based on the overprovisioning of the operator networks that will not be sufficient to maintain an acceptable Internet service. The alternative is to better manage traffic through quality of service (QoS) and traffic engineering mechanisms.

This manuscript summarizes my research work, both architectural and protocol-based, with the use of traffic engineering techniques and the formalization of routing problems by linear programs to enhance resource management and to cope with the imminent profound changes in the Internet. I have focused on transit flows among operators and in different contexts (multimedia applications and live broadcasting , sensor networks or smart cities).

At the same time, the emergence of Software Defined Networking (SDN) and Network Function Virtualization (NFV) architectures to automate network management while making it dynamic introduces new mechanisms to push the use of traffic engineering to the point where it is possible to propose a dynamic adaptation of virtualized network infrastructures. This promising evolutions offer the opportunity to share a physical infrastructure between several virtualized networks while adapting their structure to the specific needs of customers.