Découverte et agrégation de topologies de réseaux: application au contrôle d'admission

Walid Htira

Université de Toulouse

LAAS-CNRS **École Doctorale EDSyS**

Découverte et Agrégation de Topologies de Réseaux Application au Contrôle d'Admission

THÈSE

présentée et soutenue publiquement le 12 novembre 2008

pour l'obtention du titre

Docteur de L'Université de Toulouse
(spécialité Systèmes Informatiques Critiques)

par

Walid Htira

Composition du jury

Président :	Guy Juanole
Rapporteurs :	Eric Fleury (École Normale Supérieure de Lyon) Jean-Jacques Pansiot (Université Louis Pasteur)
Examinateurs :	Guy Juanole (LAAS-CNRS) Nicolas Navet (LORIA)
Directeurs :	Michel Diaz (LAAS-CNRS) Olivier Dugeon (France Telecom R&D)

Orange Labs

Mis en page avec la classe thloria.

Remerciements

Cette thèse n'aurait jamais vu le jour sans l'aide et le soutien d'un certain nombre de personnes auxquelles j'aimerais exprimer ici toute ma reconnaissance. Je voudrais tout d'abord remercier mon directeur de thèse, Michel Diaz, pour sa confiance et ses encouragements continus tout au long de ces trois années. Ses commentaires et ses remarques m'ont été très précieux et m'ont aidé à développer un esprit critique indispensable pour mener à bien un tel travail.

Je suis aussi profondément redevable à Olivier Dugeon pour la gentillesse et le soutien qu'il a manifestés à mon égard durant cette thèse. Il était beaucoup plus qu'un encadrant, un grand frère.

Dans le contexte de mes travaux de recherche, j'ai eu des discussions très enrichissantes avec plusieurs collègues de France Telecom R&D ; qu'ils en soient grandement remerciés.

Un très grand merci aussi à tous mes amis avec lesquels j'ai partagé de très bons moments : Hamdi Gaaloul, Moez Jerbi, Fabien Batello, Patrick Fleeming, Alae-eddine Adballah, Yassine Mami, Marc Varon, qui n'arrête pas de parler de la beauté de ses enfants Helena et Marius, et tant d'autres, trop nombreux pour être tous cités ici.

Finalement, je tiens à remercier mes parents, mes soeurs et mon beau frère pour leur affection qui m'a été extrêmement précieuse durant ces années loin d'eux.

A mes parents,
A mes soeurs,
A toutes les personnes qui m'ont fait découvrir l'étendu de mon ignorance.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction Générale

Sommaire

1.1 Contexte de Recherche

Au cours de la dernière décennie, l'essor des réseaux à haute vitesse et des ordinateurs à grande puissance a considérablement favorisé le développement des applications multimédia distribuées comme la Video-à-la-Demande (VoD), la Téléphonie IP, la vidéo conférence, etc. Afin de satisfaire les besoins de ces applications en termes de ressources, plusieurs mécanismes de Qualité de Service (QoS) ont été introduits dans les réseaux informatiques et de télécommunication :

- Les mécanismes de contrôle du trafic à l'entrée du réseau : ils consistent à n'accepter de nouvelles demandes de service que si le réseau peut assurer la QoS de toutes les connexions en cours, ainsi que celle de la nouvelle demande. Il faut noter qu'une demande de service peut être de nature diverse : un appel téléphonique, un transfert de clip vidéo, un établissement d'un tunnel, etc. Ces mécanismes sont connus dans la littérature sous le nom de contrôle d'admission (CAC : Call Admission Control),
- Les mécanismes de routage de trafic : ils permettent d'orienter le trafic dans le réseau en fonction de ses caractéristiques et des ressources disponibles dans le réseau. En effet, ceci est accompli via des mécanismes de routage et d'acheminement qui tiennent compte de la QoS requise par les applications, tel que PNNI (Private Network-to-Network Interface), qOSPF (QoS-enabled Open Shortest Path First),
- Les mécanismes de gestion de trafic dans les noeuds : en particulier les mécanismes de rejet sélectifs tels que WRED, les mécanismes de Fair Queueing permettant de garantir un partage équitable de la bande disponible entre flux, ou agrégats de flux, les mécanismes de mise en forme du trafic (espacement ou écrêtage), etc. Plus généralement, ces mécanismes visent à partager, selon une politique choisie par le gestionnaire d'équipement, l'accès à une ressource de transmission (gestion des contentions) et l'accès aux mémoires des noeuds (gestion des pertes),
- Les mécanismes de sur-dimensionnement : l'idée est d'augmenter la bande passante disponible afin d'éviter la congestion et la perte de paquets. Cela est d'autant plus simple à mettre en oeuvre que de nouvelles technologies telles que DWDM[1] (Dense Wavelength Division Multiplexing) [1] permettant d'atteindre des débits de l'ordre de 5 Tbps à un coût relativement faible. Même si certains pensent actuellement qu'il sera toujours possible de sur-dimensionner un réseau, cette approche semble dangereuse car le sur-dimensionnement incite l'usage des applications de plus en plus consommatrices. En outre, cette technique n'est pas applicable pour toutes les technologies, en particulier les réseaux d'accès (DSL[2], WiFi[3], UMTS[4], ...). En ce qui concerne les réseaux fixes, même en cas de surdimensionnement, les fournisseurs d'accès sont désireux de différencier les services (quitte à réduire volontairement la qualité des classes non-prioritaires dans le but d'offrir un meilleur service à ceux qui le nécessitent),

[1] Technologie augmentant la bande passante disponible sur une portion de fibre optique par l'utilisation simultanée de signaux de longueurs d'ondes différentes.

[2] La technologie dite *Digital Subscriber Line* regroupe l'ensemble des mécanismes mises en place pour le transport numérique de l'information sur une simple ligne de raccordement téléphonique.

[3] *Wireless Fidelity* est une technique de réseau informatique sans fil mise en place pour fonctionner en réseau interne ou pour l'accès à Internet. Elle est basé sur la norme IEEE 802.11

[4] Universal Mobile Telecommunications System est un système de téléphonie mobile de troisième génération (3G). Il permet des débits numériques jusqu'à 2 Mbps.

Dans ce mémoire, nous nous restreindrons à l'étude des mécanismes de contrôle de trafic. En effet, le contrôle d'admission est la première ligne de défense pour limiter l'accès aux ressources finies du réseau et garantir ainsi aux connexions déjà acceptées une bonne qualité de service. Si les algorithmes assurant cette fonction divergent, l'objectif est toujours le même, c'est-à-dire n'accepter des connexions que si les capacités du réseau le permettent. Cette piste a été explorée par plusieurs projets de recherche, dont le projet EuQoS [2]. Une des problématiques que ce projet a examinée est la manière dont il est possible d'assurer une fonction de contrôle d'admission efficace sur des topologies étendues. C'est cette problématique que nous développerons tout au long de ce mémoire. Une description plus détaillée est présentée dans la section suivante.

1.2 Problématique

La garantie de la qualité de service vis-à-vis de la demande d'un service est assurée d'une part par la protection du réseau contre les surcharges (fonction de CAC) et par la configuration adéquate des équipements du réseau. Dans les architectures centralisées de QoS, la fonction de CAC est souvent assurée par une entité appelée le Bandwidth Broker. C'est au sein de cette entité que les décisions d'acceptation ou de rejet de nouvelles demandes de connexion sont prises. Par conséquent, la pertinence de ce processus ne peut être assurée que si le Bandwidth Broker détient une cartographie détaillée de son domaine. En effet, dans une telle architecture, la fonction de CAC est fortement dépendante de l'état des équipements, des liens et des ressources dans le réseau. Pour cela, il est nécessaire d'avoir une connaissance fine et à jour du réseau, et plus particulièrement de sa topologie. Aussi, la fonction de CAC est capable de prendre les bonnes décisions sur la base d'une image cohérente entre la description des ressources que le Bandwidth Broker détient et ce qui se passe réellement dans le réseau. Si, dans certains cas d'utilisation, la topologie peut être approvisionnée manuellement, il est impossible d'accomplir une telle tâche sur des topologies étendues. Différentes solutions existent sur le marché. Généralement, elles se basent sur trois techniques :

- Des techniques passives qui reposent sur l'utilisation de SNMP et DNS,
- Des techniques actives qui se basent sur l'utilisation massive d'outils de probing tels que ping et traceroute,
- Des techniques basées sur les informations issues des protocoles de routage.

Si ces techniques sont efficaces dans certains cas d'utilisation, elles sont soit incomplètes soit inadaptées à nos besoins. En effet, la plupart des solutions de découverte de topologie ignorent la couche liaison du modèle OSI[5] (ATM, GigaBit Ethernet, VLAN, MPLS, ...) alors que la gestion des ressources passent également par la vérification des points de contention de niveau 2. De même, la mise à jour en temps réel des modifications qui peuvent survenir dans le réseau de transport n'a pas de solution en l'absence de protocoles réactifs tels que les protocoles de routage IP. Au delà de l'acquisition et de la maintenance de la topologie, il faut aussi pouvoir découvrir les ressources disponibles sur les liens et les équipements et les tenir à jour. A nouveau, si certaines techniques existent (MIBs SNMP, protocoles TE), elles ne sont pas disponibles pour toutes les technologies de transport.

Ainsi, la fonction d'admission d'appel n'est pertinente, et donc efficace, que si elle repose sur une connaissance précise et synchrone du réseau qu'elle contrôle. Comme l'état de l'art

[5]Le modèle OSI (Open Systems Interconnection) est un modèle de communications entre ordinateurs proposé par l'ISO (International Organization for Standardization). Il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

actuel n'offre pas de solutions satisfaisantes, nous nous sommes investis dans la spécification de nouveaux mécanismes assurant la notification en temps réel des informations d'état de liens et de topologie. L'idée alors, est de concevoir une architecture de contrôle d'admission efficace et réaliste. Plus de détails sur cette architecture sont exposés dans la section suivante.

1.3 Contributions de la Thèse

Si nous abstrayons notre problématique, il est possible d'identifier trois éléments clés dont les performances de la fonction de contrôle d'admission sont fortement dépendantes. Il s'agit de : a) la signalisation, b) le processus de décision et c) la propriété de passage à l'échelle. La conception d'une solution efficace de CAC doit nécessairement passer par la prise en compte de ces trois éléments. Pour être conforme à cette logique, dans notre mode fonctionnement, nous avons traité soigneusement chacun de ces éléments dans un module à part ; à savoir un module de signalisation, un module de décision et un module de passage à l'échelle.

Le module de signalisation a pour rôle d'approvisionner en temps réel le module de CAC avec les informations d'état de liens et de topologie. Ceci est très important pour l'efficacité de la fonction de contrôle d'admission dans la mesure où il permet de refléter une photocopie de ce qui se passe réellement dans le réseau. A cette fin, nous avons proposé un nouveau protocole de signalisation : Simple Topology Annoucement and Management Protocol (STAMP). En effet, STAMP permet la diffusion des informations de topologie et de qualité de service dans un domaine. Sa logique de fonctionnement est très simple. Elle se base sur l'échange d'un ensemble de messages "hellos" de niveau 2 entre voisins directs. Le but est d'annoncer des informations sur la présence ainsi que sur les caractéristiques techniques des ports physiques d'interconnexion. Les informations d'adjacence collectées sont par la suite envoyées à une station centrale responsable des fonctions de gestion et de corrélation de données. En adoptant une telle logique, nous avons voulu assurer deux fonctions primordiales pour le module de contrôle d'admission : d'un coté, améliorer la précision du processus de décision par la fourniture d'un mécanisme proactif d'annonce de topologie, et d'un autre coté, explorer les technologies de la couche liaison, plus spécifiquement les technologies 802.3 et ATM souvent déployées dans les réseaux d'opérateur Télécom.

A l'issue de la collecte de données par le module de signalisation, le module de décision est capable d'accomplir la fonction de contrôle d'admission par un simple parcours de la base de topologie. En effet, par de simples opérations d'addition et de soustraction, il est possible, en parcourant la base de topologie, de déterminer si une nouvelle demande de connexion peut être acceptée ou pas sur la base de l'état des ressources disponibles dans le réseau. Cette opération de parcours de la base de topologie peut être, dans certains cas lorsqu'il s'agit de grandes topologies (Backbones IP), coûteuse. Pour cela, nous avons proposé de nouveaux mécanismes d'agrégation de topologies ayant pour but de représenter la base de topologie en une structure plus compacte afin d'optimiser le processus de décision. Ces mécanismes sont implémentés au sein du module de passage à l'échelle. En effet, en réduisant la volumétrie des données de topologie, le module de décision passe moins de temps pour le parcours de la base, ce qui améliore en conséquence les performances de la fonction de CAC et favorise le passage à l'échelle.

1.4 Organisation du Manuscrit

La thèse s'articule autour de trois grands chapitres :

Le chapitre 2 est consacré à la description de l'état de l'art. Dans un premier temps nous introduisons la fonction de contrôle d'admission, ses caractéristiques ainsi que les approches les plus intéressantes qui ont été proposées dans la littérature. Nous étudions par la suite avec plus de détails les approches centralisées, plus spécifiquement l'approche de Bandwidth Broker puisqu'il s'agit de la solution retenue au niveau du projet EuQoS. En effet, dans ce type d'architecture, le Bandwidth Broker doit disposer d'une cartographie exacte et à jour du domaine qu'il gère. Aussi, une connaissance de l'état des ressources dans le réseau est nécessaire. Pour cela, Il est important d'identifier l'ensemble des paramètres et des données de topologie impliqués dans le processus de contrôle d'admission. Plusieurs outils accomplissant cette tâche existent sur le marché. Nous analysons l'ensemble de ces outils vis-à-vis d'un certains nombre de critères d'évaluation. Sur la base de cette analyse, nous étudierons les limites et les avantages de chacune des solutions. Une solution idéale est celle qui permet une remontée en temps réel des données de topologie vers le Bandwidth Broker. Ce sont ces informations qui permettrons plus tard de décider si une nouvelle demande de connexion peut être acceptée ou non. Puisqu'elles sont toutes stockées dans la même structure de données, le parcours des informations de topologie peut s'avérer dans certains cas une tâche très lourde pour le Bandwidth Broker. En effet, plus la topologie est grande plus le parcours de la structure de données est complexe. Comme solution palliative, nous nous intéressons aux mécanismes d'agrégation de topologie. Ces mécanismes permettent de représenter une topologie réelle par une topologie plus compacte. L'idée est de réduire la volumétrie des informations stockées dans la base de topologie afin d'accélérer le processus d'admission d'appel. Nous étudions alors en détails les approches les plus intéressantes proposées dans la littérature.

Sur la base de l'état de l'art exposé dans le Chapitre 2, nous commençons le Chapitre 3 par définir le modèle de données des informations de topologie répondants à notre problématique. Ce modèle présente l'élément de base de notre solution de découverte de topologie : le protocole de signalisation STAMP (Simple Topology Annoucement and Management Protocol). Tous les aspects fonctionnels et techniques du protocole seront étudiés en détail : sa structure, ses messages, ses diagrammes d'état ainsi que son architecture logicielle. Nous terminons le chapitre par une étude des performances du protocole sur la base de l'ensemble des résultats d'émulations enregistrés.

Le Chapitre 4 sera consacré à la description de nos algorithmes de CAC. Nous commençons par présenter le processus d'admission d'appel dans un cadre classique, c'est-à-dire sur la base des données de topologie brute collectées via STAMP (topologie réelle). Ensuite, la fonction de CAC sera étudiée dans le cadre des topologies agrégées. En effet, la topologie réelle est représentée sous la forme d'une modélisation plus compacte afin d'optimiser le processus de contrôle d'admission. Dans un premier temps nous nous intéressons seulement aux réseaux sensibles aux paramètres de délai et de bande passante (contexte mono-service QoS). Nous proposons deux schémas d'agrégation. La façon dont ces schémas sont construits ainsi que l'algorithme de contrôle d'admission seront étudiés avec beaucoup de détails. Une extension pour l'application au contexte multi-services, par l'intégration des paramètres de gigue et de taux de perte, sera aussi proposée. Les règles de CAC seront alors révisées afin d'obéir à la nouvelle modélisation de la topologie. Une autre extension sera apportée aux algorithmes d'agrégation. Il s'agit de la prise en compte du routage multi-chemins. En effet, si nous nous sommes basés dans les premiers schémas d'agrégation sur l'hypothèse qu'une seule route est possible entre deux noeuds, il est possible dans la réalité d'avoir plus d'une route. Nous proposons alors un nouvel algorithme d'agrégation de lien. L'idée est de représenter les paramètres de QoS des différentes routes par une métrique unique. Sur la base de l'ensemble des métriques calculées, il est possible par la

suite de calculer le schéma agrégé. Nous étudions alors la performance de la fonction de CAC sur ce type de schéma.

Finalement, nous concluons par le Chapitre 5 qui résume nos travaux de recherche et dans lequel nous suggérons diverses pistes qui en découlent et qui pourrons faire l'objet de réflexions futures.

Chapitre 2

Contrôle d'Admission, Découverte de Topologie et Agrégation

Sommaire

2.1 Introduction

L'évolution des technologies pour les réseaux de télécommunications et l'accroissement de leurs performances permettent aujourd'hui le déploiement d'applications diverses qui concernent tous les aspects de la vie moderne, telles que la communication de personne à personne, la diffusion de l'information, les jeux, l'enseignement, le commerce etc.

Le déploiement de ces applications s'effectue dans le cadre de réseaux multi-services se caractérisant par leur aptitude à répondre à des besoins importants en bande passante et à des exigences très variées en matière de qualité de service (QoS).

A l'origine, l'ATM semblait être le mode de transfert cible pour les réseaux multiservices par la définition de plusieurs catégories de service (CBR : Constant Bit Rate, VBR : Variable Bit Rate, UBR : Unspecified Bit Rate ...) ayant des besoins en termes de bande passante et de QoS, et IP semblait incapable d'assurer cette tâche. En effet, l'Internet actuel, qui est basé sur le protocole IP, est un réseau à commutation de paquets à l'architecture et aux principes simples. Il ne fournit qu'un service dit "au mieux" (BE pour Best Effort), c'est-à-dire sans aucune garantie sur le débit disponible, le délai subi ou encore les pertes de paquets. Ce service s'avère suffisant pour les applications "élastiques" de transfert de données mais insuffisant pour les nouvelles applications multimédias qui se développent et dont les contraintes temps réel se satisfont difficilement du service BE. Par conséquent, le reproche principal que nous pouvons faire aux réseaux IP actuels est l'impossibilité de contrôler la qualité de service autrement qu'en sur-dimensionnant les ressources. Ce sur-dimensionnement peut être une piste intéressante afin d'améliorer la QoS, cependant, une solution cohérente ne peut pas être simplement basée sur le sur-dimensionnement. En effet, des efforts sont faits visant d'une part à définir des services IP permettant de transporter et d'acheminer des flux de données en fonction de leurs différentes contraintes, et d'autre part à définir des mécanismes et des protocoles permettant une ingénierie de trafic adaptée aux réseaux IP. Un de ces mécanismes est la fonction de contrôle d'admission (CAC : Call Admission Control). Son rôle est de limiter le nombre de flux admis dans un réseau afin de satisfaire les demandes de qualité de service des utilisateurs ou des applications, tout en partageant les ressources du réseau de façon équitable et efficace. À chaque requête, il s'agit de décider si l'occupation du réseau permet d'accepter le niveau de qualité de service demandé.

Dans la section suivante, nous proposons un survol des différentes caractéristiques liées à la fonction de contrôle d'admission.

2.2 Le Contrôle d'Admission dans les Réseaux IP

2.2.1 Identification des Caractéristiques des Fonctions de CAC

Classer les approches de contrôle d'admission est une tâche difficile et subjective due à la diversité des paramètres impliqués dans ce processus. Dans ce paragraphe, nous identifions les différents aspects liés directement à la fonction de contrôle d'admission. Ceci est très important pour mieux débattre et analyser les solutions déjà existantes. Les caractéristiques que nous avons identifiées lors de notre étude sont les suivantes :

– ***La nature du réseau*** : cet aspect identifie le modèle du réseau sur lequel s'applique le contrôle d'admission. Ce réseau peut être défini pour des architectures multiservices ou à service unique (best effort). Cet aspect couvre aussi la portée de la fonction de CAC qu'elle soit à l'intra-domaine, à l'inter-domaine ou de bout en bout,

- ***Le type de service*** : il dépend étroitement du niveau de garantie requis par le service. La terminologie utilisée pour distinguer les types de services ainsi que leur niveau de garantie n'est pas toujours la même : il est soit déterministe[6], statistique[7] ou prédictif[8],
- ***La localisation de la prise de décision*** : elle identifie l'aspect centralisé ou décentralisé du processus d'admission d'appel. Ceci peut être accompli en identifiant les différents noeuds impliqués dans la fonction CAC ainsi que leur rôle. Par exemple, un noeud peut juste avoir le rôle de prise de décision afin d'admettre un nouveau flux ou non, comme il peut juste collecter des informations de réservations à partir d'autres noeuds. Deux facteurs importants sont à étudier avant de favoriser une approche par rapport à une autre, à savoir le taux des informations à stocker et à manipuler dans le cas des architectures centralisées, et les mécanismes de coordination à implémenter et à gérer dans le cas des architectures décentralisées,
- ***La nature de l'algorithme d'admission d'appel*** : cet aspect identifie la façon avec laquelle la décision d'admission d'appel est prise. L'algorithme de CAC peut être basé sur des paramètres, sur des mesures, sur des politiques ou sur une combinaison de ces trois critères (hybride). Cet aspect sera étudié avec plus de détails dans la section suivante.

Ainsi, sur la base de tous les points énoncés précédemment, la performance globale d'une approche d'admission d'appel peut être caractérisée de la façon suivante :

- La capacité de garantir le contrat de qualité de service,
- L'efficacité de l'utilisation des ressources pour les niveaux de service fournis,
- La complexité introduite par le processus d'admission d'appel en termes de temps de calcul et espace de stockage,
- La facilité de déploiement.

C'est à l'ensemble de ces caractéristiques que nous essayerons d'apporter un réponse tout au long de ce mémoire.

2.2.2 Les Critères de Décision

Établir les critères de décision d'une fonction d'admission d'appel consiste à définir un ensemble de règles qui conduisent à l'acceptation ou au refus des demandes de connexion. Les trois critères que nous avons identifiés sont les suivants :

Les Approches Basées sur des Paramètres

Désigné par l'acronyme PBAC (Parameter-Based Admission Control), ce type d'approche se situe parmi les mécanismes de contrôle d'admission "classique", basé sur une description statistique a priori de chaque flux. En effet, le réseau utilise ces paramètres afin d'estimer les ressources à allouer aux flux, et il vérifie leur disponibilité en fonction de tous les autres flux circulant sur le réseau. Le contrôle d'admission se base sur des modèles analytiques de files d'attente pour la détermination des paramètres décrivant le trafic émis. Ce type de contrôle d'admission

[6]Le service garanti[3] se base sur une caractérisation déterministe du trafic et de la qualité de service désirée. Les garanties de type déterministes se basent sur une description du pire comportement possible du trafic.

[7]Le service statistique se base sur une caractérisation stochastique du trafic et de la qualité de service désirée. Les garanties de type statistique se basent sur une description stochastique du trafic, donc une description du comportement moyen du trafic.

[8]On parle de service prédictif car l'on considère que les mesures du passé sont un bon indicateur du comportement futur du réseau.

parait le plus conservateur et orienté garantie de service par sa forte dépendance à la topologie du réseau. Par contre, il se heurte à la difficulté d'estimer rapidement l'impact d'un nouveau flux sur la charge du réseau et sur la QoS. ATM et IntServ s'appuie sur ce type de contrôle d'admission. De plus, si un flot estime mal son débit en excès ou en défaut, le principe peut conduire à un résultat erroné. Pour éviter une sur-allocation, il est alors nécessaire de contrôler individuellement les flux sur la base du débit déclaré ou ajuster le niveau global de réservation par une mesure.

Les Approches Basées sur des Mesures

Ce type de contrôle d'admission (désigné par l'acronyme MBAC : Measurement Based Admission Control), contrairement à l'approche traditionnelle, estime lui-même les paramètres du modèle de trafic, et ne se base plus sur des descripteurs de trafic à priori. D'autre part, le MBAC est intrinsèquement une procédure statistique puisqu'il se base sur des mesures, et il ne peut donc s'appliquer uniquement que dans le cadre de garanties statistiques et de modèles de trafic stochastique [Fig.2.1].

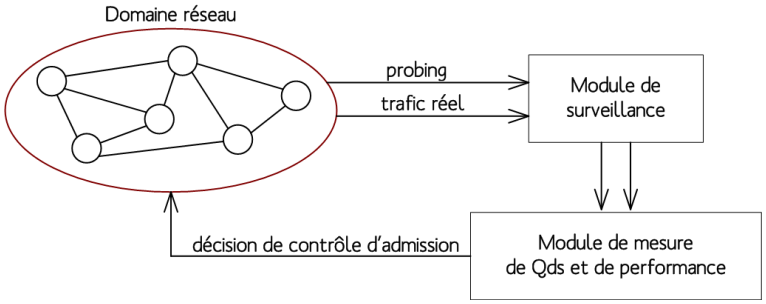

FIG. 2.1: Architecture générale du modèle basé sur les mesures

Le contrôle d'admission à base de mesures est utilisé pour du service prédictif. On parle de service prédictif car l'on considère que les mesures du passée sont un bon indicateur du comportement futur du réseau. Or ceci peut s'avérer dangereux car si cette prédiction du comportement futur du trafic n'est pas bonne, le MBAC risque soit de sous-utiliser les ressources du réseau, soit d'admettre trop de flux et dégrader ainsi les qualités de service demandées par les sessions en cours. Ce type de contrôle d'admission est bien approprié pour le service à contrôle de charge [4]. Le principe ne peut supporté une garantie absolue sur le service demandé, il considère une borne sur le délai nominal puis autorise des variations autour de cette valeur. Le but est d'obtenir une qualité de service semblable à celle que la session aurait eu dans des conditions où le réseau est peu chargé. Un certain nombre d'études ont conduit à l'exploration de différentes techniques de mesure de la charge du réseau. La première technique de mesure est le mécanisme à fenêtre temporelle (time-window) utilisé par Jamin et al. [5]. Ils calculent un débit moyen sur chaque période d'échantillonnage de taille $\bar{b}$. A la fin de la fenêtre de mesure de taille $\bar{t}$, ils prennent la plus grande valeur moyenne calculée sur cette fenêtre comme l'estimation de la charge pour la durée $\bar{t}$ de la fenêtre suivante. Lorsqu'un nouveau flux est admis dans le réseau, cette estimation est immédiatement augmentée de la valeur du débit déclaré par ce flux. En tout temps, si la nouvelle valeur moyenne du débit calculée sur une période $\bar{b}$ est supérieure à la valeur de l'estimation de la charge, la valeur de l'estimation est immédiatement augmentée de la valeur de cette nouvelle moyenne. Le deuxième mécanisme de mesure est le mécanisme *point samples*. Il

prend la valeur moyenne de la charge sur chaque période $\bar{b}$' comme estimation de la charge du réseau. Il s'agit en fait du mécanisme de fenêtre temporelle énoncé précédemment avec $(\bar{t}/\bar{b})$ qui vaut 1. Le troisième et dernier mécanisme est le mécanisme *exponential averaging*. Une valeur moyenne de la charge $\bar{v}^b$ est calculée sur chaque période d'échantillonnage $\bar{b}$. Puis l'estimation de la charge est la valeur $(1 - \bar{w}) \times \bar{v} + \bar{w} \times \bar{v}^b$ calculée à la fin de chaque période $\bar{b}$. $\bar{w}$ est un nombre compris entre 0 et 1, et $\bar{v}$ est la valeur de l'estimation de la charge calculée à la période précédente. Les plus importants algorithmes MBAC se basent sur les techniques précédemment évoqués. Plusieurs comparaisons de ces algorithmes ont été effectuées dans la littérature [6] [7].

Les Approches Basées sur des Politiques

Dans cette dernière famille de contrôle d'admission, la décision est prise à partir d'un ensemble de règles définies par l'opérateur. Ces règles permettant d'ajouter des conditions supplémentaires (sécurité, horaire, identification, classe de services...) par rapport aux autres types de contrôle d'admission. La décision peut être centralisée ou non. Dans le cas centralisé (architecture de type Policy-Based Network)[8], un serveur dédié, le PDP (Policy Decision Point) va prendre la décision d'admission et communiquer sa décision à l'ensemble des routeurs avec un protocole spécifique, par exemple le protocole COPS[9]. Généralement, ce mécanisme de décision est utilisé conjointement avec des mécanismes PBAC et MBAC.

Des exemples de l'ensemble des algorithmes de décision ont été définis et comparés dans [10, 11, 12, 13, 14, 15, 16, 7].

2.2.3 Les Différentes Familles de Contrôle d'Admission

La littérature identifie cinq grandes classes d'approches de contrôle d'admission qui sont les suivantes :

Les Approches Basées sur RSVP

RSVP [17] est un protocole de signalisation qui permet aux applications d'exprimer au réseau leurs besoins en termes de qualité de service. A la réception d'une demande de connexion, les équipements réseaux retournent une indication explicite d'acceptation ou de rejet sur la base de l'état des ressources dans le réseau. Le processus de signalisation se résume en un échange de messages RSVP PATH et RSVP RESV afin d'acheminer la requête et par la suite réserver les ressources demandées si c'est possible. Indépendamment de l'architecture Intserv [18], le problème majeur rencontrer avec RSVP est son application flux par flux, ce qui rend difficile son utilisation dans le cadre de réseaux de grandes dimensions [Fig.2.2]. Ainsi, il se heurte à la difficulté de gérer un grand nombre d'utilisateurs (Passage à l'échelle). Plus il y a d'utilisateurs, plus il y a d'états à créer et à maintenir pour des destinations différentes à chaque demande. Il est encore difficile de dire dans quelle mesure et à partir de combien de flux le problème se pose réellement, mais il est clair qu'une gestion des ressources par flux dans le contexte de l'Internet, dont la dimension ne cesse de croître, impose de très fortes contraintes.

Ce problème de passage à l'échelle a conduit un certains nombres de recherches pour agréger les flux individuels afin de rendre Intserv|RSVP plus extensible. Dans cette nouvelle approche nommée SRP (Scalable resource Reservation Protocol) [19], tous les flux sont agrégés et aucune connaissance des flux individuels n'est conservée dans le réseau. Chaque routeur conserve la réservation totale qu'il a accordée et surveille le trafic. Le protocole utilise trois types de paquets :

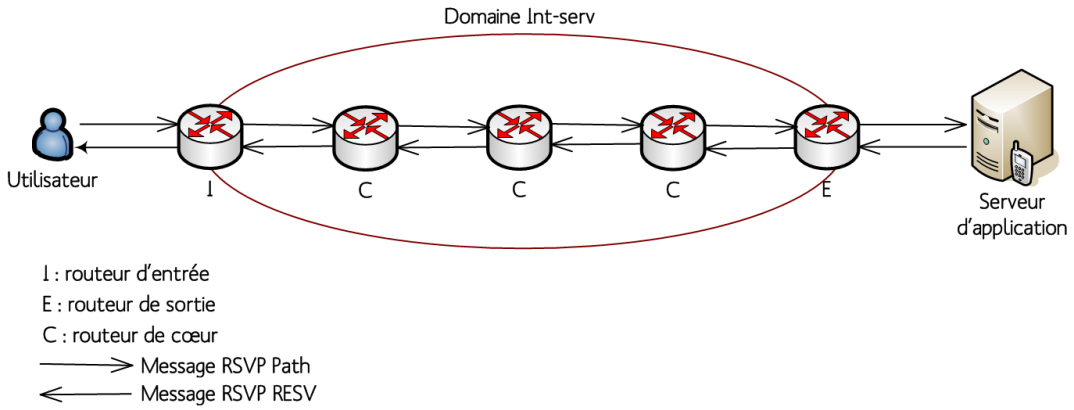

FIG. 2.2: Le modèle RSVP

"best-effort", "request" et "reserved". Brièvement, une application désirant obtenir une garantie de débit émet des paquets "request" au débit désiré. Quand un routeur reçoit des paquets de ce type, si il a assez de ressources il augmente la réservation et transmet les paquets, sinon il change les paquets en best-effort. Le récepteur mesure le débit arrivant en paquets "request" et en déduit la réservation acceptée sur tout le chemin. Il l'envoie par un protocole de réaction (feedback) à l'émetteur qui peut alors émettre des paquets "reserved" au débit accepté. Dans les routeurs, les paquets "request" (non dégradés en best-effort) et "reserved" sont traités prioritairement.

Une autre piste, examinée pour pallier aux problèmes de passage à l'échelle rencontrés avec IntServ sur les très grands réseaux, consiste à utiliser conjointement IntServ et Diffserv dans une architecture hybride. Crée par l'IETF en 1997, le modèle Diffserv tente de dépasser les difficultés rencontrées dans le modèle Intserv. Il introduit la notion de l'agrégation des flux en quelques classes offrant des services spécifiques par agrégat (per Aggregate), mais n'offre aucune garantie sur les flux individuels. Dans ce modèle il n'y a pas de réservation de ressources à travers les noeuds, tel que IntServ, mais un traitement différencié, appelé PHB (Per Hop Behavior) qui est basé sur la priorité par classe pour répondre à la QoS demandée. En plus du best effort, deux services sont définis : Expédié (Expedited Forwarding) et Assuré (Assured Forwarding). Cette architecture hybride IntServ/DiffServ est déployée comme suit : IntServ, basé sur RSVP, opérant au niveau flux individuel est utilisé sur les sites périphériques comme les réseaux d'entreprise ou les réseaux de petite taille (LAN). Et DiffServ, basé sur la priorisation et sur la notion d'agrégats, est utilisé pour les réseaux de transitions ou de coeur (WAN). Dans la perspective de IntServ, les régions DiffServ du réseau sont traitées comme des liens virtuels reliant les routeurs ou les terminaux IntServ, et la visibilité de l'utilisateur est réduite à IntServ [Fig.2.3]. Dans ce contexte hétérogène, IntServ sur DiffServ [20], les flux sont traités à deux échelles différentes : flux individuel et agrégat. De même la qualité de service est assurée suivant deux politiques différentes : réservation de ressources et priorité.

L'Approche DPS (Dynamic Packet State)

Stoica et Zhang proposent dans [21] et [22] un modèle adapté aux garanties de délai. Ce modèle offre les mêmes garanties de service que IntServ mais d'une façon plus extensible. L'originalité de leur approche réside dans le fait qu'elle supprime le besoin de maintenir les états des flux dans chaque noeud. Les informations d'état sont transportées dans quelques champs spécifiques de l'entête IP, selon la technique DPS (Dynamic Packet State).

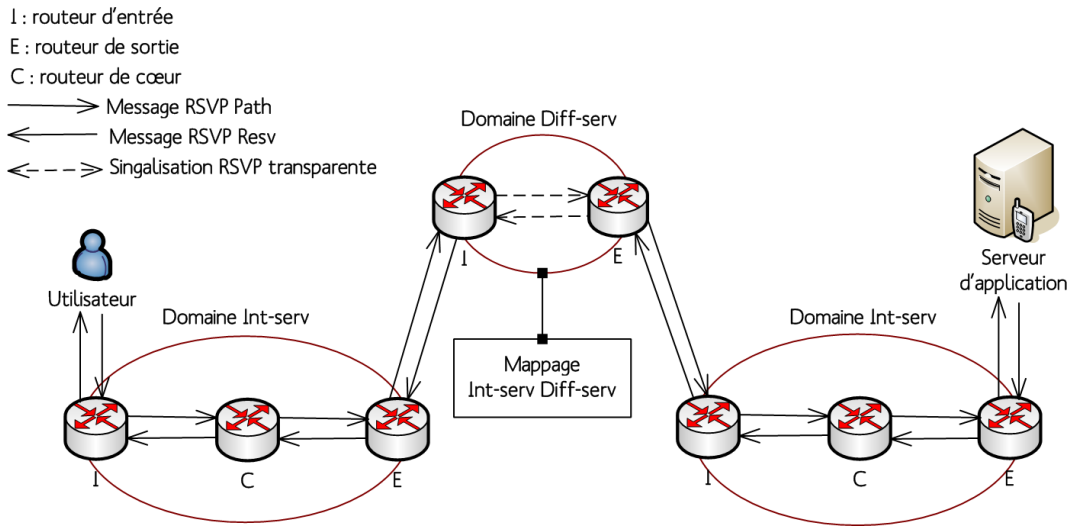

FIG. 2.3: Architecture hybride IntServ|Diffserv

Chaque paquet est traité par les routeurs de coeur suivant son état, ensuite il est mis à jour ainsi que son état avant d'être transmis. L'admission d'appel dans une telle architecture passe par deux phases [Fig.2.4] : Une première phase de signalisation RSVP a lieu entre l'émetteur et le routeur d'entrée du domaine (ingress node) et entre le routeur de sortie et le récepteur (egress node), sachant que les messages RSVP (PATH et RESV) sont transparents au sein du domaine DPS. Lors de la deuxième phase, à la réception d'un message RESV, le routeur d'entrée du domaine enclenche un message de signalisation spécifique vers le routeur de sortie permettant ainsi de vérifier l'admissibilité de la requête sur chaque noeud, de calculer le nombre de saut vers la destination et de calculer un label entre le routeur d'entrée et le routeur de sortie afin de permettre aux paquets d'un même flux de suivre le même chemin. Quand le routeur de sortie est atteint, le routeur d'entrée est notifié afin de prendre la décision finale et par la suite informer l'émetteur de cette décision. Dans ce type d'architecture, l'état des réservations par flux est seulement maintenu au niveau des routeurs de bordure. Le schéma général de cette architecture est décrit par la figure ci-dessous [Fig.2.4].

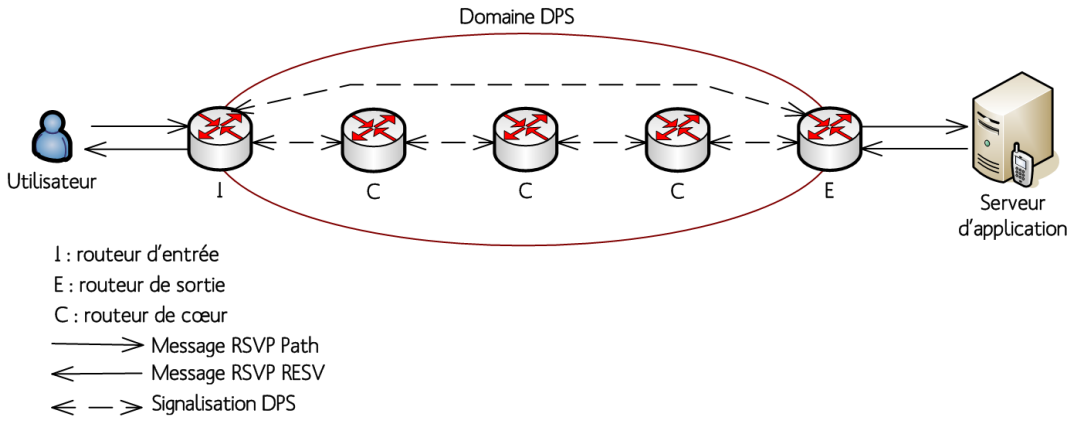

FIG. 2.4: Le modèle DPS

Malgré les réponses qu'il apporte aux problèmes de passage à l'échelle, DPS soufre de quelques limitations qui empêchent son déploiement dans un contexte réel. En effet, pour fonctionner correctement, les spécifications DPS exigent l'implémentation des mêmes mécanismes de scheduling sur tous les routeurs de coeur. Ceci n'est pas réalisable sur des réseaux hétérogènes, voire même homogènes. En outre, l'insertion des informations d'état dans l'entête des paquets n'est pas compatible avec tous les protocoles et les mécanismes déjà existants.

Les Approches Basées sur la Théorie des Enveloppes

Dans le contexte des réseaux multiclasses, Cetinkaya et Qiu [23] [24] proposent un schéma d'admission d'appel basé sur la théorie des enveloppes de trafic [24, 25, 26]. Dans cette proposition, les noeuds de bordures assurent mutuellement les fonctions d'agrégation de trafic et de contrôle d'admission [Fig.2.5]. Pour chaque couple de routeurs d'entrée et de sortie, les noeuds de sortie évaluent deux types d'agrégat d'enveloppes de trafic : i) les enveloppes d'arrivées qui reflètent la bande passante maximale des arrivées ; ii) les enveloppes de services, qui représentent le service minimal disponible. Le premier est basé sur le nombre d'octets reçus tout au long du temps de mesure et le second est basé sur le délai de propagation des paquets.

En ce qui concerne le processus d'admission, les flux spécifient leurs besoins en qualité de service via le protocole RSVP, qui traverse de façon transparente le coeur du domaine pour être traiter par le routeur de sortie. La décision d'admission est prise en se basant sur les caractéristiques du nouveau flux en termes de bande passante et délai, le pic mesuré sur les enveloppes des arrivées et de sa variance, la classe minimale des enveloppes des services et sa variance, ainsi que sur d'autres critères décrits en détail dans [23].

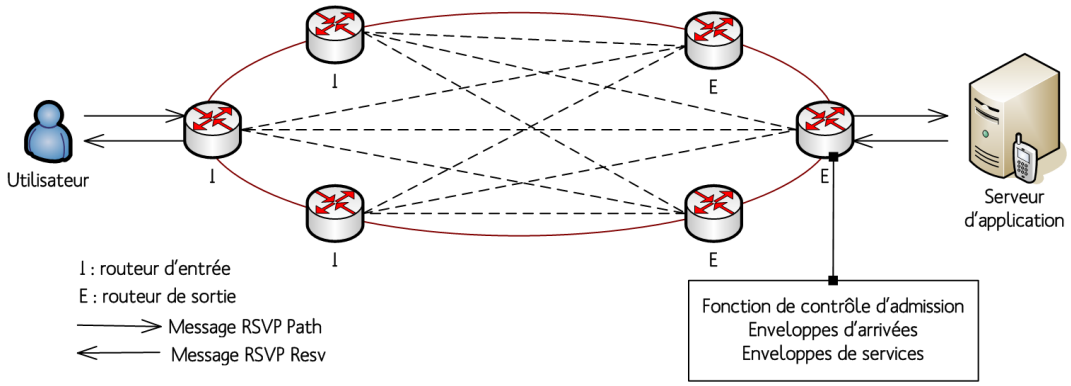

FIG. 2.5: Le modèle basé sur la théorie des enveloppes

Les Approches Basées sur le Probing

Dans ce type d'approche, le noeud d'entrée dans le domaine teste le chemin de donnée, que devra suivre un flux, avec les mêmes caractéristiques que ce flux. Le taux de perte, le délai et la gigue sont mesurés par le noeud récepteur, qui vérifie le niveau de congestion du réseau. Ceci est appelé probing. Si les performances mesurées sont acceptables, le flux est accepté, sinon il est rejeté [Fig.2.6]. Les fonctionnalités de qualité de service dans ce mécanisme sont attribuées aux équipements d'extrémités (*End Points*), excluant ainsi la nécessité du déploiement d'un protocole de signalisation ou d'autres fonctions dans les noeuds de coeur ou de bordure. Le défaut majeur de cette approche est la charge introduite lors de la phase d'initialisation afin de traiter

un nouvel appel.

un nouveau mécanisme basé sur le probing est en cours d'étude à l'IETF : *Congestion and Pre-Congestion Notification* (PCN) [27]. L'idée consiste à effectuer des mesures en continue, puis à renvoyer vers les noeuds d'entrée du réseau (*ingress node*) l'état de congestion des différents chemins. On évite ainsi la phase d'initialisation. Le noeud d'entrée a toute la connaissance nécessaire pour prendre la décision.

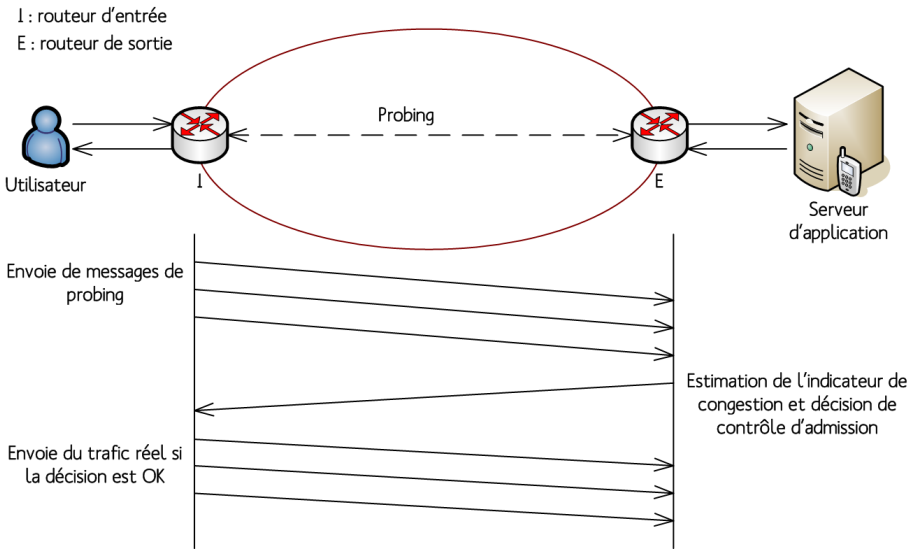

FIG. 2.6: Le modèle basé sur le probing

Les Approches Centralisées

Contrairement aux approches précédentes, les approches centralisées implémentent tous les mécanismes de contrôle d'admission au sein d'une seule unité. C'est au niveau de cette unité centrale que les décisions d'admission ou de refus sont prises. L'approches la plus détaillée dans la littérature, est le Bandwidth Broker (BB). Ce mécanisme a été défini par le RFC 2638 [28] dans le cadre de Diffserv comme étant l'entité qui a la connaissance des capacités du domaine DiffServ en termes de topologie et de ressources disponibles. Il permet par la suite d'allouer la bande passante en fonction des politiques définies dans le réseau et de la disponibilité des ressources.

Le principal module d'un BB est celui du contrôle d'admission. Il est responsable de l'admission d'appel et de la réservation des ressources. Le BB maintient aussi un ensemble de données concernant les informations de topologie, de routage, de flux et de qualité de service sur chaque noeud et lien. Comme le montre la Figure 2.7, un Bandwidth Broker est composé de :

- Un module central, responsable du traitement des requêtes rentrantes. Il contrôle tous les autres modules et réalise la gestion des sessions à QoS.
- Une interface d'accès pour les différentes requêtes des applications, proxys ou administrateurs.
- Un module qui réalise le contrôle d'admission.
- Un composant en charge de la signalisation inter-domaine.

– Une base de données contenant des informations de topologie, des contrats établis avec les clients et les domaines adjacents, des politiques prédéfinies, et une description des ressources disponibles, etc.
– Un système de surveillance et de métrologie.

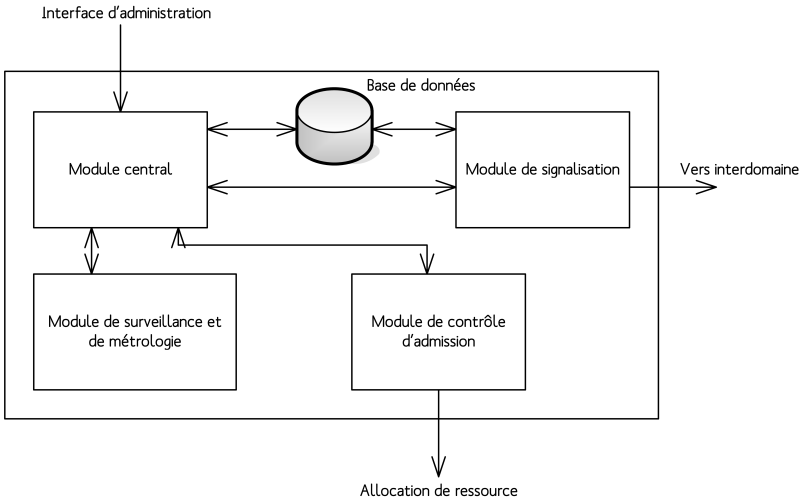

FIG. 2.7: Architecture générale d'un Bandwidth Broker

Généralement un BB est déployé par domaine. Ainsi, pour que la réservation des ressources soit faite de bout en bout, les Bandwidths Brokers doivent communiquer entre eux. Pour cela des contrats de gré-à-gré (peering agreeements) sont mis en oeuvre entre domaines adjacents.

Le processus de contrôle d'admission dans une telle architecture se déroule comme décrit par la Figure 2.8 : A la réception d'une nouvelle demande de connexion, un message de signalisation est envoyé par le routeur d'entrée du domaine au BB spécifiant le profil du flux ainsi que ses exigences en termes de qualité de service. Une fois la requête authentifiée, le BB prend sa décision de refus ou d'acceptation sur la base des politiques du domaine, les SLS établies ainsi que la disponibilité des ressources dans le réseau.

La figure 2.8 illustre le fonctionnement général d'un BB. Dans le cadre de l'architecture Internet2 Qbone [29] [30], une discussion sur l'architecture du BB, sa signalisation et ses exigences sont décrites.

Les avantages d'une telle approche sont divers : a) centraliser les informations d'états et les fonctions de contrôle au sein de la même entité donne une vision globale des capacités du réseau en termes de qualité de service, b) supprimer les fonctions de contrôle au sein du réseau, c) réduire les échanges de messages de signalisation dans le domaine, d) faciliter le changement et/ou la mise à jour des politiques de services et des algorithmes de contrôle d'admission.

Néamoins, l'architecture du Bandwidth Broker présente un défaut majeur. Il s'agit de son coût de déploiement élevé. En effet, le BB doit gérer et stocker une volumétrie très importante d'informations de topologie et d'états, ce qui influe directement sur les performances d'une telle architecture. Pour pallier à ce problème, plusieurs modèles de BB hiérarchiques ont été proposés dans la littérature. Ils consistent à déployer plus d'un BB par domaine réseau. Dans le cas de réseaux de grande échelle, un tel modèle hiérarchique améliore la disponibilité ainsi que le passage à l'échelle et évite la congestion, mais en contre partie il implique des problèmes de coordination et de synchronisation entre les BBs. Il augmente aussi la charge du réseau avec

les messages de signalisation qui transitent entre les différents BBs. Finalement, la fonction de contrôle d'admission du Bandwidth Broker dépend fortement des informations de topologie : si elles sont erronées ou plus à jour, l'admission d'appel sera basée sur une représentation faussée du réseau et produira donc de mauvais résultats.

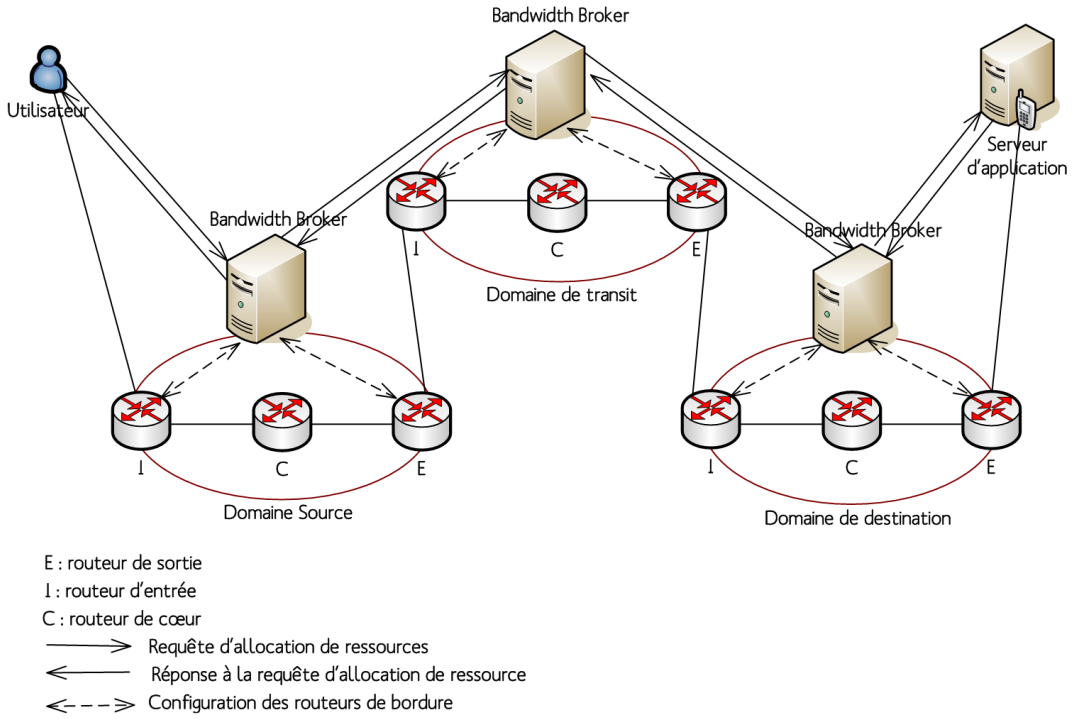

FIG. 2.8: Le modèle d'admission d'appel du Bandwidth Broker

2.2.4 Etude Comparative des Approches de CAC

Sur la base des caractéristiques de CAC identifiées dans les sections 2.2.1 et 2.2.2, le tableau Tab.2.1 résume les aspects pertinents de chacune des approches de contrôle d'admission fournies dans la section 2.2.3, leurs avantages ainsi que leurs limites. Il s'avère que la solution de Bandwidth Broker est la seule à offrir un niveau consistant de garantie de service sans compliquer le plan de contrôle dans le coeur du réseau. En effet, la centralisation des informations de topologie, d'états, de politiques de services et de SLS négociés au sein de la même entité facilite les fonctions de contrôle et de gestion. Le plan de contrôle est découplé du plan de transfert. Les routeurs de coeur sont ignorés du plan du contrôle, ils ne maintiennent aucune information sur les réservations et n'exécutent aucune fonction de contrôle. Si ce découplage peut alléger la charge du réseau, il peut aussi introduire des problèmes de synchronisation responsable d'éventuelles erreurs lors du processus de décision. C'est à ce problème de synchronisation que nous essayons d'apporter une solution tout au long de ce mémoire.

2.3 Paramètres Manipulés par la Fonction de CAC

Dans la section précédente nous avons survolé les différents mécanismes de CAC. Quel que soit le mode de fonctionnement de ces mécanismes, le but est toujours le même : vérifier si les

TAB. 2.1: Evaluation des différentes approches de CAC

Approches de CAC	**Architecture**	**Avantages**	**Limites**
RSVP/Intserv et SRP	Distribuée	- Garantie de service - Approche très conservative	- Problèmes de passage à l'échelle - Difficultés de déploiement - Problème de contrôle de charge
Intserv/Diffserv	Distribuée	Assure un compromis entre la garantie de service et le passage à l'échelle. Ceci dépend de la dimension des régions Diffserv/Intserv et de la fonction de translation entre les deux technologies	
DPS	Distribuée	- Garantie de service - Diminuer le taux des informations d'états maintenues dans le coeur du réseau	- Protocole propriétaire - Difficulté de mise en oeuvre dans un contexte réel - Mise à jour des entêtes des paquets
Envelopes-Based	Distribuée	- Seuls les noeuds de bordures sont sollicités pour la CAC - Garantie de service - Réduire le taux des informations d'états maintenues dans le réseau	- Difficulté de déploiement de bout en bout - Changements fréquents, ce qui induit une charge de calcul très importante
Probing-Based	Distribuée	- Facile à déployer - Supporte le passage à l'échelle - Réduire le taux des informations d'états maintenues dans le réseau	- QoS instable - Un probe par flux - Latence initiale
Bandwidth Broker	Centralisée	- Le domaine réseau est simplifié - Offre une vue globale du domaine - Garantie de service - Plus flexible (facilité de modification/mise à jour des algorithmes et des politiques de QoS)	- Dépend d'une seule entité (Eventuel point de défaillance et de congestion) - Problèmes de passage à l'échelle

demandes de connexion sont vérifiées vis-à-vis de ce que le réseau peut garantir. Cette garantie est identifiée par rapport aux paramètres de qualité de service que le réseau supporte. Ces paramètres sont quantifiables et influent sur la performance des transmissions des données. Traditionnellement, quatre paramètres sont identifiés :

– La bande passante ou débit pic : il s'agit du taux de transfert maximum pouvant être maintenu entre la source et la destination,

– Le délai : il s'agit du temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire. Le délai tient compte du délai de propagation le long du chemin et du délai de transmission induit par la mise en file d'attente des paquets dans les systèmes

intermédiaires. Le délai de propagation est typiquement lié à la topologie du réseau et au routage des paquets. En outre, lorsque le routage est statique, il est constant. Le délai de transmission, quant à lui, est lié à la congestion des liens empruntés, il dépend donc des autres flux circulant sur le réseau. Souvent par abus de langage, le délai de bout en bout désigne plus spécifiquement le délai de transmission,

- La gigue : la gigue désigne la variation du délai de bout en bout au cours de la transmission. Une trop forte gigue affecte en particulier les flux multimédias temps-réel en détruisant les relations temporelles des trains de données transmis régulièrement par le flux multimédia, entravant ensuite la compréhension du flux par le récepteur,
- Le taux de perte : c'est le ratio de la quantité d'information perdue par rapport à la quantité d'information émise.

L'ensemble de ces paramètres est primordial pour l'exécution de la fonction de CAC. Sur la base de leurs valeurs le Bandwidth Broker sera capable de décider de l'acceptation ou non de nouveaux flux dans le réseau. Cependant, une connaissance fine de la topologie est aussi nécessaire pour permettre au Bandwidth Broker d'exécuter correctement son algorithme d'admission d'appel. En d'autres termes, le plan de contrôle (le Bandwidth Broker) doit rester synchrone au plan de transfert (topologie). Par le terme topologie, nous évoquons toutes les informations liées à l'adressage physiques des machines, à l'établissement de circuit logique et au routage. La collecte des informations de topologie et de QoS n'est pas toujours évidente. Du coup, il a fallu trouver les mécanismes nécessaires permettant de remonter en temps réel et avec précision l'ensemble de ces données au Bandwidth Broker. Nous consacrons la section suivante à l'étude des différents outils de découverte de topologie qui ont été proposés dans la littérature.

2.4 Étude des Mécanismes de Découverte de Topologie

2.4.1 Contexte Technologique

La gestion d'un réseau est un sujet très vaste. Il comprend entre autre la gestion du trafic, la gestion des dysfonctionnements, la gestion des configurations, la gestion de l'accès au réseau. Cependant, quelque soit le réseau, il nécessite toujours une connaissance exacte de la topologie. En effet, un administrateur réseau a besoin d'une vue de l'ensemble des éléments (LAN, MAN, WAN) qui entrent en jeu dans le fonctionnement des services fournis par le biais du réseau. En conséquence, toutes les plates-formes commerciales de supervision des réseaux (HP OpenView, Tivoli, etc.) disposent d'un service de recherche dynamique de topologie. Le rôle principal d'un tel service est de collecter depuis le réseau les informations de topologie afin de les corréler au sein d'une entité centrale de gestion.

A ce niveau, plusieurs questions se posent à propos des outils de découverte de topologie existants : sont-ils capables d'opérer dans des environnements hétérogènes là où plusieurs technologies réseaux cohabitent ? sont-ils capables de remonter une topologie physique complète du réseau ? sont-ils capables de notifier en temps réel les changements de topologie qui surviennent dans le réseau ?

Dans ce chapitre, nous essayons d'apporter une réponse à toutes ces questions en se plaçant dans un cas de figure bien défini. Il s'agit des réseaux déployant Ethernet/ATM au niveau de la couche liaison et IP au niveau de la couche réseau. Ceci représente la grande majorité des réseaux de collecte ADSL (Asymmetric Digital Subscriber Line) [31], des réseaux d'agrégation (appelés aussi réseaux de collecte) et des backbones IP. En effet, pour la couche liaison nous

nous intéressons aux technologies GE (Gigabit Ethernet), ATM (Asynchronous Transfert Mode), VLAN (Virtual LAN) [32] et MPLS (MultiProtocol Label Switching) [33] et nous ignorerons FDDI[9] (Fiber Distributed Data Interface), FR[10] (Frame Relay) et X.25[11] du fait qu'elles sont de moins en moins utilisées. En ce qui concerne la couche réseau, nous nous restreindrons à l'étude du protocole IP (IPv4 et IPv6), et nous ignorons les autres protocoles réseaux tel que X.25, IPx[12] (Internetwork Packet eXchange) qui sont peu ou pas utilisés par les services auxquels nous tentons d'apporter une réponse.

La prochaine section sera consacrée essentiellement à l'étude des outils de découverte de topologie que nous jugeons intéressants et originaux.

2.4.2 Etude des Outils Existants

L'étude des mécanismes de découverte de topologie peut être faite suivant deux niveaux d'abstraction : un niveau physique qui concerne la topologie de la couche Liaison, et un niveau logique qui concerne la topologie de la couche IP :

- *La topologie de niveau 2* prend en compte tous les équipements actifs intervenant dans le fonctionnement et la transmission d'information en couche Liaison. Nous rappelons qu'un équipement actif de niveau 2 ne participe pas directement au cheminement des paquets IP dans le réseau.
- *La topologie de niveau 3* représente la cartographie d'un réseau IP. Les équipements actifs intervenant dans cette topologie sont des équipements capables d'effectuer le routage des paquets IP (par exemple un routeur). Les autres équipements sur le chemin (tels que les commutateurs et les concentrateurs) qui ne traitent que des informations de la couche Liaison ne sont pas répertoriés à ce niveau.

Sur la base de ces deux définitions de topologie, nous proposons dans ce qui suit un état de l'art des approches de découverte de topologie que nous jugeons intéressantes. Pour les réseaux IPv4, de nombreux outils de découverte dynamique de topologie existent. Presque toutes les plate-formes de supervision fournissent leur propres outils pour cette fonction. Quelques uns travaillent au niveau des Systèmes Autonomes[13] (AS : Autonomous System), d'autres travaillent à l'intérieur des AS. En ce qui concerne les solutions au niveau AS, la plupart se base sur l'utilisation conjointe d'outils tels que traceroute et les tables BGP [34, 35, 36], et de quelques heuristiques pour la gestion des adresses IP collectées. Les solutions software les plus intéressantes sont Rocketfuel [37], Mercator [38] et Nec [39]. Dans ce mémoire, nous nous intéressons seulement à la découverte de topologie à l'intérieur d'un AS.

[9] FDDI est une technologie d'accès au réseau sur des lignes de type fibre optique. Il fonctionne sur une topologie en anneau double avec un jeton pour la détection et la correction d'erreurs.

[10] Il s'agit d'un protocole à commutation de paquets situé au niveau de la couche de liaison du modèle OSI.

[11] X.25 est un protocole normalisé au sein de l'ITU-T. Il intègre les trois couches basses du modèle OSI et est basé sur un principe de commutation de paquets en mode point à point.

[12] IPX est un protocole datagramme sans connexion implémenté par Novell. Il transmet des paquets à travers un LAN et fournit aux stations Netware et aux serveurs de fichiers des services d'adressage et de routage inter-réseaux. Il s'agit donc d'un protocole de couche 3 du modèle OSI.

[13] Désigne un domaine réseau administré par une même entité et ayant des politiques et une configuration homogènes

Outils de Niveau 3

Solutions Commerciales Basées sur SNMP :

L'architecture SNMP (Simple Network Management Protocol), définit dans le RFC 1157 [40], est la plus couramment utilisée pour la gestion des équipements réseaux. Il s'agit d'un protocole relativement simple englobant un ensemble de fonctionnalités suffisamment puissantes pour permettre la gestion des réseaux hétérogènes complexes.

L'environnement de gestion SNMP est constitué de plusieurs composantes : une station de supervision, des éléments actifs dans le réseau, des variables MIB et un protocole.

- Les éléments actifs du réseau sont les équipements ou les applications que nous cherchons à gérer. Chacun de ces éléments actifs dispose d'une entité dite agent qui répond aux requêtes de la station de supervision dite manager. Les agents sont des modules qui résident dans les éléments réseau. Ils vont chercher des informations de gestion tel que le nombre de paquets reçus ou transmis,
- La station de supervision (manager dans la terminologie SNMP) exécute les applications de gestion qui contrôlent les éléments réseaux. Physiquement, le manager est un poste de travail,
- La MIB (Management Information Base) est une collection d'objets résidant dans une base d'information virtuelle. Ces collections d'objets sont définies dans des modules MIB spécifiques. Nous pouvons distinguer deux types de MIBs :

  1. Celles définies par des RFCs, qui permettent de gérer tout type d'équipement dialoguant via le protocole IP. La principale est la MIB II [41] [42] qui gère principalement les différents protocoles TCP/IP : IP [43], TCP [44], UDP [45] par exemple. D'autres modélisent le comportement de familles d'équipements (par exemple la MIB des ponts [46], etc.),
  2. Celles définies par les constructeurs et spécifiques à leur matériel, par exemple la MIB CISCO [47].

- Le protocole est le mécanisme qui permet à la station de supervision d'aller chercher les informations sur les éléments réseaux et de recevoir des alertes provenant de ces mêmes éléments.

Plusieurs solutions commerciales se sont servis du protocole SNMP, plus particulièrement des MIB, pour intégrer des fonctionnalités de découverte de routeurs et de sous réseaux afin de générer une topologie de niveau 3. Parmi les outils les plus intéressants, nous pouvons citer HP's OpenView [48], IBM's Tivoli [49] et Dartmouth Intermapper [50]. Il est à noter que dans ces outils commerciaux il n'est pas possible d'avoir la ou les méthodes permettant de construire la topologie du réseau.

L'Algorithme de Schönwälder et Langendörfer :

Un autre mécanisme intéressant de découverte de topologie de niveau 3, basé sur ICMP[14], a été proposé par Schönwälder et Langendörfer. Ce mécanisme [53] collecte en premier lieu toutes les informations qu'il peut obtenir du réseau et les analyse par la suite. Ainsi, dans un premier temps, il recherche l'ensemble des adresses en service dans un espace d'adressage précis

[14]Le protocole ICMP (Internet Control Message Protocol) fait partie intégrante du protocole IP. Il permet de contrôler le fonctionnement d'IP par un ensemble de mécanismes : un contrôle de flux, un avertissement en cas d'erreur d'adressage, la redirection d'une machine vers un autre routeur, le test de l'état d'un équipement (fonctions Echo Request et Echo Reply).

et donné au préalable. Cette recherche s'exécute grâce à un ICMP Echo Request exhaustif vers toutes les adresses possibles de l'espace d'adressage donné. Ensuite, il recherche l'ensemble des routeurs permettant de relier ces différentes adresses à l'aide de la fonction traceroute [54]. Pour associer la topologie d'adressage et la topologie de routage trouvées, l'algorithme cherche la liste des préfixes utilisés sur le réseau en envoyant un ICMP Mask Request aux différentes interfaces et routeurs détectés. Comme certains systèmes peuvent posséder plusieurs interfaces, il envoie ensuite un paquet UDP vers un port supposé non utilisé pour essayer d'obtenir les adresses des autres interfaces : l'adresse source retournée dans le paquet d'erreur ICMP donne quelquefois la valeur de l'adresse de l'interface qui a reçu le paquet. Toutes ces informations sont par la suite analysées pour définir les différents sous-réseaux du réseau et pour trouver les routeurs qui relient ces différents réseaux, et les noeuds dans les sous-réseaux ainsi que leurs différentes adresses s'ils possèdent plusieurs interfaces.

L'Algorithme de Lin et al :

Si le service de recherche de topologie précédent consiste d'abord à récolter l'ensemble des informations qu'il peut obtenir avant de construire la topologie du réseau, la solution proposée par Lin [55] construit la topologie du réseau sitôt qu'il la découvre. Connaissant une première liste d'adresses IP, cet algorithme y cherche une interface en service, en utilisant un ICMP Echo Request. Il va ensuite chercher le préfixe du sous-réseau auquel appartient cette interface à l'aide d'un ICMP Mask Request. Puis, il va raccorder ce sous-réseau au noeud où se trouve le service de recherche de topologie par un traceroute ou une requête SNMP. Il obtient alors la liste des routeurs jusqu'à cette interface, et en particulier le routeur par défaut du sous-réseau auquel l'interface appartient. Ce dernier devient le premier élément de la liste des routeurs R. L'algorithme débute ensuite une phase itérative de recherche de tous les noeuds se trouvant sur le réseau :

1. Prendre le premier élément de la liste,
2. Faire la liste des sous-réseaux connectés à ce routeur à l'aide de requêtes SNMP. Ces informations sont obtenues par les objets ipRouteDest, ipRouteMask de la table ipRouteTable de la MIB II,
3. Pour chacun des sous-réseaux trouvés, chercher et mémoriser toutes les interfaces qui y sont connectées (ICMP Echo Request vers toutes les adresses IP possibles du sous réseau),
4. Trouver tous les routeurs directement connectés au routeur courant à l'aide du protocole SNMP. Ces informations sont données par les objets ipRouteNextHop de la table IpRouteTable de la MIB II. Chaque nouveau routeur est inséré dans la liste des routeurs R,
5. Revenir à la première étape.

Si un routeur ne dispose pas d'agent SNMP, la recherche utilise alors l'algorithme de Schönwälder et Langendörfer, pour compléter ses informations.

Cet algorithme présentait l'inconvénient d'induire un trafic important et de découvrir l'ensemble de la topologie dans un délai assez long. Afin de résoudre ces problèmes, H-C Lin a proposé de l'inclure dans une architecture hiérarchique et distribuée [56]. Des serveurs intermédiaires utilisent l'algorithme de Lin et al, afin de trouver la topologie d'une partie du réseau. Ces topologies sont envoyées, sur demande uniquement, à un gestionnaire d'entreprise qui les associe afin d'obtenir la topologie globale. L'implémentation qui en a été faite prouve qu'effectivement le trafic induit est bien moindre et que le temps de calcul de la topologie, puisque réparti entre chaque serveur, est beaucoup plus court.

L'Algorithme de Siamwalla :

L'algorithme proposé par Siamwalla [57] consiste à utiliser conjointement les informations du protocole IP (Ping [58] et Traceroute [54]) et les "zones de transfert DNS". Ces informations de zone de transfert permettent d'obtenir la liste de toutes les machines enregistrées dans le serveur DNS[15] (Domain Name System) afin de construire la topologie désirée. Celle-ci est effectivement trouvée avec plus ou moins de rapidité, suivant le nombre d'hôtes dans la classe d'adresses IP, car l'outil doit tout d'abord déterminer le masque du réseau associé à chaque sous-réseau qui doit être analysé. En effet, la recherche des adresses IP est résolue en diminuant le nombre de bits du masque du réseau (de 31 à 7), puis en envoyant des paquets ICMP en broadcast sur le réseau. Les éléments qui répondent permettent de déterminer le masque de réseau associé aux adresses IP. Chaque élément qui répond est répertorié en utilisant les principes du système de fichier Unix, c'est-à-dire la classification arborescente des répertoires et des fichiers. Chaque élément est enregistré dans le répertoire dont le nom est défini par l'adresse IP du réseau. Ainsi avec cette technique, Siamwalla obtient la représentation d'une topologie sous forme arborescente. Seuls les éléments de types routeurs sont présents dans la topologie obtenue.

L'Algorithme de Pansiot :

Une autre approche intéressante a été proposée par Pansiot [59]. Elle se base sur la collecte d'information de topologie via des requêtes IGMP (Internet Group Management Protocol) dans un environnement multicast. Les résultats de simulation d'une telle approche ont montré son efficacité en termes de précision et de coût : les informations de topologie locale des routeurs sont facilement acquises contrairement aux techniques basées sur traceroute, et le nombre de messages est réduit ce qui permet de collecter fréquemment des informations.

Outils de Niveau 2

Les MIBs Pont :

Elles ont été proposées par l'IETF comme standard pour représenter les MIB des topologies physiques [60] [46]. En effet, une MIB Pont contient les informations nécessaires à la commutation des trames Ethernet sur le réseau. C'est le protocole SNMP qui permet de collecter les informations contenues dans ces MIB afin de faire les associations port/adresse Ethernet des éléments connectés, appelée communément la matrice de commutation (Forwarding Database). En corrélant ces données, c'est-à-dire en associant les adresses Ethernet des éléments du réseau et celles découvertes dans la MIB Pont, on peut arriver à positionner les éléments dans la topologie du réseau. Malheureusement, lorsque le réseau fonctionne avec des équipements actifs non SNMP, une partie de la topologie ne peut être construite, ce qui présente le défaut majeur de ces algorithmes. Plusieurs algorithmes sont basés sur de telles MIBs ; les plus intéressants sont :

L'Algorithme de Lin et al :

La proposition de Lin et al [55] permet la découverte des informations de topologie de niveau 2. Pour cela, elle suppose la présence de SNMP sur tous les équipements du réseau. En effet, l'agent de découverte de topologie va d'abord chercher tous les noeuds en service sur le réseau, en utilisant un ICMP Echo Request exhaustif. Il sépare par la suite les noeuds, les ponts et les commutateurs, en demandant aux agents SNMP de chacun des noeuds trouvés, l'objet sysServices du groupe system de la MIB II [41] [42]. Si cet objet précise que le noeud fournit

[15] Le DNS est un système permettant d'établir une correspondance entre une adresse IP et un nom de domaine et, plus généralement, de trouver une information à partir d'un nom de domaine.

un service de niveau lien, l'algorithme demande alors :

- son adresse physique (objet dot1dBaseBridgeAddress du groupe dot1dBase de la MIB Pont),
- les ports qui sont utilisés (objets dot1dBasePort, dot1dBasePortIfIndex et dot1dBaseNumPorts de la table dot1dBasePortTable de la MIB Pont),
- l'adresse physique des équipements directement connectés à celui étudié (fournis par la table ipNetToMediaTable de la MIB II).

L'Algorithme de Breitbart et al, de Lowekamp et al :

Les algorithmes de Breitbart et al [61, 62, 63] et de Lowekamp et al [64] résolvent le problème de la recherche des connexions physiques entre les noeuds avec des outils mathématiques. Ils modélisent tous les deux le réseau en un graphe mathématique où les noeuds sont les équipements réseau et les arcs sont les liens physiques. Avec des hypothèses de départ plus ou moins fortes, et en utilisant SNMP pour obtenir les informations de niveau 2 dont ils ont besoin, ils peuvent déterminer les arêtes possibles et celles qui ne le sont pas. Les adresses physiques des noeuds et leur type (pont, noeud ou commutateur) sont obtenus grâce à des requêtes SNMP afin d'obtenir l'objet ipRouteNextHop de la table ipRouteTable. Ces algorithmes font la distinction entre un commutateur et un routeur en :

- vérifiant l'existence ou non de la MIB des ponts (MIB Pont),
- lisant l'objet ipForwarding. Si le noeud est un routeur, sa valeur est 1. Si c'est un commutateur ou un simple hôte, sa valeur est 0.

Protocoles Propriétaires :

Certains constructeurs ont implémenté leurs propres protocoles de découverte de topologie. Nous nous restreindrons à présenter les deux protocoles les plus intéressants : CDP (Cisco Discovery Protocol) de Cisco et LLTD (Link Layer Topology Discovery protocol) de Microsoft.

Cisco Discovery Protocol :

CDP (Cisco Discovery Protocol) [65] est un protocole propriétaire indépendant du média qui est utilisé pour la découverte du voisinage réseau. Il s'exécute sur la couche liaison de données, par-dessus le protocole SNAP[16] (Subnetwork Access Protocol) [66]. Chaque équipement configuré pour CDP envoie périodiquement des messages, appelés annonces (advertisements), aux équipements réseau directement connectés. Chaque équipement annonce au moins une adresse à laquelle il peut recevoir des messages SNMP (Simple Network Management Protocol). Les annonces contiennent également des informations de durée de vie ou durée de conservation, indiquant pendant combien de temps les équipements récepteurs doivent conserver les informations CDP avant de les éliminer. De plus, chaque équipement écoute les messages CDP périodiquement envoyés par les autres équipements afin d'identifier ceux qui se trouvent dans le voisinage.

Link Layer Topology Discovery protocol :

LLTD (Link Layer Topology Discovery protocol) [67] est aussi un protocole de niveau 2 propriétaire Windows. Il présente l'un des éléments clé de la plateforme Microsoft Windows Rally. Son rôle est de détecter automatiquement les périphériques présents sur un réseau local.

[16]Protocole réseau placé au dessus des protocoles IEEE 802.2 et 802.3 permettant d'étendre les capacités d'identification de services des champs d'entête du protocole LLC (Logical Layer Control).

Contrairement à CDP, LLTD utilise des mécanismes de sonde (probing) afin de découvrir la cartographie réseau. Il est à noter que ce protocole opère aussi bien avec les équipements sans fil qu'avec les équipements filaires et que sa portée s'arrête au niveau des réseaux locaux (LAN).

Protocole Standardisé :

Le "Link Layer Discovery Protocol"[17] [68] (LLDP) est le seul protocole standardisé. Il est basé sur l'échange de trames spécifiques permettant à des équipements réseaux (station, switch, routeur, téléphone IP) de communiquer leur identité et leur fonction à leurs voisins. Ce protocole est très similaire à CDP (Cisco Discovery Protocole) de Cisco et correspond à son adaptation/amélioration par le monde open-source. Il est implémenté par de nombreux constructeurs tels que HP, Nortel, etc. Les informations échangées entre équipements voisins sont stockées dans des MIBs [60] accessibles via un système de gestion réseau. Afin d'établir une cartographie complète de la topologie, l'utilisation du protocole SNMP est nécessaire. Il permet de remonter les informations de topologie maintenues par les agents à un manager.

Quelques extensions ont été apportées à LLDP via LLDP-MED (Link Layer Discovery Protocol for Media Endpoint Devices) afin d'améliorer les fonctions de gestion et de localisation des périphériques (terminaux Voix sur IP).

2.4.3 Etude Comparative des Outils Existants

Définition des Critères de Comparaison

Quelles que soient les techniques utilisées pour la découverte de topologie et leurs divergences, les critères d'évaluation de tels mécanismes sont toujours les mêmes dans la littérature. Cependant, en réponse à notre problématique (fonction de contrôle d'admission), un protocole de découverte de topologie doit remplir un certain nombre de conditions dont les principales sont (par ordre d'importance) :

- *La précision* : les informations de topologie doivent être récupérées avec précision, c'est à dire sans erreur. Ce critère est d'autant plus important qu'il conditionne la pertinence et la validité des résultats. Toute erreur ou imprécision aura une conséquence directe sur la fonction de contrôle d'admission, à savoir une prise de décision erronée,
- *La réactivité* : un protocole de découverte de topologie doit être capable de notifier en temps réel tous les changements de topologie ou d'informations d'état dans un domaine réseau. De cette façon, la fonction de contrôle d'admission sera informée de toutes les pannes, les points de congestion potentiels (y compris ceux de la couche liaison), les modifications de routage, les ressources réseau allouées, etc., à même d'influencer sa prise de décision dans l'admission d'un nouveau flot,
- *La rapidité* : nous admettons qu'un processus d'acquisition de topologie passe par deux régimes : un régime transitoire qui prend un certains temps avant d'établir une cartographie complète du domaine réseau, suivi d'un régime stationnaire où seuls des changements de topologie sont notifiés. Le critère de rapidité concerne la phase transitoire. Le temps d'établissement de la topologie complète doit être acceptable (pour le régime stationnaire, il s'agit du critère de réactivité),

[17] IEEE 802.1AB

– *L'efficacité* : si nous optons pour l'acquisition de la topologie afin d'améliorer la fourniture de la qualité de service, ce processus d'acquisition ne devra pas être en lui même une entrave devant l'accomplissement de cette tâche. En d'autre termes, une solution idéale d'acquisition de topologie, doit être légère et la moins consommatrice possible en ressources réseau et CPU,
– *La complétude* : la topologie réseau doit être complète, c'est-à-dire ni manque ni ajout. En effet, elle doit prendre en considération tous les équipements réseau quel que soit la technologie déployée ou la couche OSI en question.

Comparaison

D'un point de vue personnel, nous estimons que les outils de découverte de topologies décrits précédemment sont incomplet vis-à-vis des critères d'évaluation. En ce qui concerne les outils de niveau 3, nous distinguons deux grandes familles ; celles basées sur SNMP et celles basées sur ICMP. Le premier type de solution est généralement rapide et efficace. Son problème principal réside dans le fait que SNMP n'est pas toujours activé sur tous les équipements réseau pour des raisons de sécurité. De même, les constructeurs n'implémentent pas toujours toutes les MIBs ou ne les renseignent pas correctement. Sans oublier bien sûr le fait qu'il y a pas mal d'équipements qui n'implémentent pas le protocole SNMP. En dernier lieu, certains équipements ont tendance à disfonctionner en cas de sollicitation intensive de leur MIB via le protocole SNMP. Le deuxième type de solution, basée sur ICMP, malgré sa précision, est très lent, ce qui entraîne par la suite une surcharge du domaine réseau à découvrir. De plus, une telle solution ne peut pas être appliquée dans un cadre réactif ou tout changement doit être signalé dans le moindre délai. Pour les outils de niveau 2, elles sont presque toutes basées sur l'utilisation conjointe de SNMP et des MIBs Pont, ce qui nous ramène aux mêmes limites de SNMP identifiées précédemment.

Le tableau ci-dessous [Tab.2.2] énumère les propriétés des solutions de découverte de topologie vis-à-vis des critères de performances que nous avons identifiés précédemment. Les marques suivantes sont utilisées :

– $\surd$ pour identifier un critère qui est satisfait tout le temps,
– † pour identifier un critère qui n'est pas satisfait tout le temps,
– ‡ pour identifier un critère qui n'est pas du tout vérifiable.

Il est à noter que pour les solutions basées sur SNMP, nous nous mettons dans le cas idéal où les MIBs sont bien remplies.

TAB. 2.2: Evaluation des différents outils de découverte de topologie

Outils de niveau 2 et 3	Précision	Réactivité	Efficacité	Complétude	Rapidité
SNMP	$\surd$	‡	†	†	†
Schönwälder/Langendörfer	$\surd$	‡	$\surd$	†	†
Lin et al	$\surd$	‡	†	$\surd$	‡
Siamwalla	$\surd$	‡	†	†	†
Breitbart/Lowekamp	$\surd$	‡	†	†	†
CDP	$\surd$	†	$\surd$	$\surd$	†
LLTD	$\surd$	†	$\surd$	$\surd$	†
LLDP	$\surd$	†	$\surd$	$\surd$	†

2.4.4 Vers une Approche Réactive et Complète

Le survol des différentes approches de découverte de topologie a mis en évidence l'importance d'avoir une solution complète afin de remplir les quatre critères de performance que nous avons identifiés. Qu'elles soient basées sur SNMP, ICMP ou les MIBs Pont, les solutions existantes sont incomplètes vis-à-vis de nos critères d'évaluation. En effet, SNMP n'est pas déployé sur tous les équipements ; de plus il est consommateur en ressources CPU dans le cadre de réseaux de grande échelle (interrogation de MIBS de grande taille). ICMP est très lent et donc ne peut pas être un bon candidat pour un déploiement dans un contexte très dynamique. Enfin, les MIBs Ponts impliquent les mêmes problèmes que SNMP, puisque généralement c'est ce dernier qui est utilisé pour l'interrogation des MIBs. Les solutions propriétaires, malgré leurs avantages, n'offrent pas un cadre générique d'application et ne sont supportées que par les équipements du constructeur en question (CDP par exemple). Reste LLDP, mais malheureusement il ne s'applique qu'au monde IEEE (Ethernet) et ne prend pas en compte d'autre technologie comme l'ATM.

Au vu des manques de l'état de l'art actuel, nous nous sommes lancés dans le développement de notre propre outil de découverte de topologie et des informations de qualité de service. Cet outil sera capable de :

- Répondre aux critères de performance que nous nous sommes posés,
- Remonter des informations précises et réelles des caractéristiques des équipements ainsi que leur emplacement dans le réseau,
- Notifier tous les changements dans le réseau en temps réel,
- Accomplir la fonction d'acquisition de données de topologie et de QoS sans introduire une surcharge dans le réseau,
- Supporter plusieurs technologies en séparant les couches 2 et 3.

Dans le cadre de nos travaux, nous nous sommes intéressés aux technologies ATM, Ethernet, VLAN, MPLS et IP. Il s'agit des technologies le plus souvent déployées dans les réseaux de collecte et de coeur des opérateurs de Télécom.

Une telle solution présente un candidat très intéressant pour l'approvisionnement du Bandwidth Broker avec les informations de topologie et de qualité de service. En effet, le BB aura alors une vue complète et à jour de la topologie réseau de son domaine, ce qui réduit par la suite la probabilité d'erreur dans le processus d'admission d'appel. En contre partie, la volumétrie d'informations induite par une telle solution peut poser des problèmes dans le cadre de topologies de grande échelle. En effet, sur des réseaux étendus, le taux des informations de topologie et de QoS augmente, ce qui introduit des limitations au niveau des équipements de gestion en termes de capacité de traitement. Dans la section suivante, nous exposons une piste prometteuse à explorer pour pallier aux problèmes de passage à l'échelle, le concept d'agrégation de topologie.

2.5 Application aux Réseaux de Grande Taille

2.5.1 La Notion de Passage à l'Échelle

L'expansion de l'Internet, en taille et en nombre de services a accentué le problème de l'explosion du nombre d'équipements dans le réseau ainsi que la grande quantité d'information qui en découle, ce qui impose des fortes contraintes sur les ressources disponibles. Dans un contexte similaire au notre, où la découverte des informations de topologie implique une charge exponentielle, il devient difficile de tout centraliser au sein d'une seule entité vu les limitations en mé-

moire, en bande passante et en CPU. Une réponse possible à ce problème consiste à appliquer la théorie de la subdivision. Cette solution permet de modéliser les grands réseaux en un ensemble de domaines gérables qui peuvent être des POP[18] (Point of Presence), des aires OSPF (Open Shortest Path First) [69] [70], des régions IS-IS (Intermediate System to Intermediate System) ou d'autres découpes administratives (algorithmes de clustering [71, 72, 73, 74, 75, 76, 77]). De cette façon, un Bandwidth Broker est déployé par découpe réseau, ce qui réduit considérablement le taux des informations de topologie manipulées. Quoique cette approche apporte quelques réponses aux problèmes de passage à l'échelle, elle introduit des problèmes de coordination entre les éléments du domaine découpé (signalisation, mise à jour, etc.).

Une autre solution, prometteuse, est l'application des mécanismes d'agrégation de topologie. Ce concept a été introduit dans les spécifications de PNNI (Private Network-to-Network Interface) [78] dans le cadre des réseaux hiérarchiques. En effet, il est possible de structurer un réseau à grande échelle en un ensemble de domaines agrégés. Chaque domaine détient une cartographie détaillée de sa topologie ainsi qu'une topologie agrégée des autres domaines. C'est sur ce concept que nous avons développé tous nos algorithmes d'admission d'appel dans le cas de réseaux à grande échelle.

Dans la section suivante, nous dressons un état de l'art des différentes méthodes d'agrégation de topologie qui ont été proposées dans la littérature.

2.5.2 Le Concept d'Agrégation de Topologie

Comme nous l'avons vu précédemment, plusieurs fonctions réseau (le Contrôle d'admission, l'ingénierie de trafic, la gestion de ressources, etc.) dépendent de la complétude et de la précision des informations de topologie. Cependant, il est peu pratique de garder toutes les informations de topologie relatives à un réseau étendu de façon centralisée en regard des performances attendues. En effet, les performances d'un BB décroissent exponentiellement en fonction de la taille du réseau. D'une part, le temps de recherche dans la topologie croit linéairement avec la taille du réseau et d'autre part, le nombre d'usagers, donc de requêtes, augmente aussi avec la taille du réseau. Finalement, le nombre d'états à manipuler dans le BB peut aussi influencer sa stabilité. Pour toutes ces raisons, il est souhaitable de travailler sur des tailles de réseau raisonable ou trouver des solutions palliatives. L'agrégation de topologie est l'une de ces solutions.

En parlant d'agrégation de topologie, nous distinguons deux types d'agrégation : une "agrégation de noeuds" qui consiste à abstraire un domaine réseau en une structure plus compacte représentée par un noeud logique au niveau hiérarchique supérieur, et une "agrégation de liens" permettant la représentation de l'ensemble des chemins (physiques ou logiques) entre deux noeuds distincts par un lien logique unique.

L'Agrégation de Noeuds

L'agrégation de noeuds consiste à représenter un domaine réseau par un noeud logique au niveau hiérarchique supérieur [Fig.2.9]. Trois modélisations nodales intéressantes ont été définies dans les spécifications du protocole PNNI :

– Le modèle *symmetric-point* : il consiste à agréger un domaine entier en un seul point [Fig.2.10-(b)]. Généralement ce point représente le chemin de bout en bout ayant la pire

[18] Les POP sont les points de collecte régionaux du réseau d'un opérateur. Leurs rôles est de centraliser les connexions provenant des DSLAM régionaux. Ils sont reliés par des liens très haut débit au backbone d'un réseau d'opérateur.

qualité de service. Ce modèle réduit au maximum la quantité d'informations échangées (espace de complexité après agrégation est de l'ordre de 1) ; par contre il introduit une très grande déformation par rapport à la topologie réelle du domaine,

– Le modèle *full-mesh* : il s'agit de construire une topologie complètement maillée en établissant des liens logiques entre les différents noeuds de bordure d'un domaine [Fig.2.10-(c)]. Dans le cas où tous les liens logiques ont le même coût, le modèle *full-mesh* sera similaire au modèle *symmetric-point*. Malgré que cette méthode introduise une déformation minime comparée au modèle précédent, sa complexité est plus importante (de l'ordre de $o(Card(B)^2)$, où B est l'ensemble des noeuds de bordure, et Card est la fonction cardinal),
– Le modèle *star* : ce modèle présente un compromis entre les deux modèles cités précédemment. Il consiste à définir un point central virtuel appelé *nucleus* connecté aux différents noeuds de bordure via des liens logiques nommés *spokes*. Pour apporter plus de précision au modèle et afin de minimiser la déformation, PNNI définit des liens directs entre les noeuds de bordure appelés *bypasses* [Fig.2.10-(d)]. Le nombre de *bypasses* dans les spécifications PNNI ne doit pas dépasser Card(B)-1. L'espace de complexité de ce modèle est de l'ordre o(Card(B)).

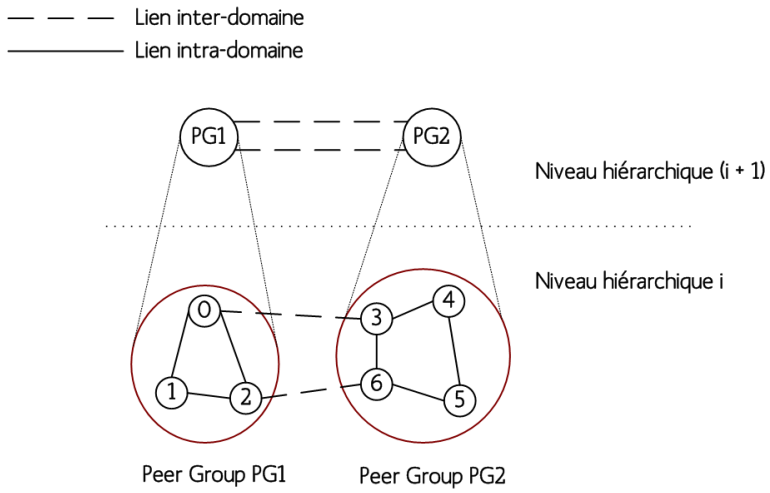

FIG. 2.9: Le modèle hiérarchique PNNI

Guo et Matta [79] ont comparé la performance des trois modèles d'agrégation suscités. Il apparaît que le modèle *star* est beaucoup plus performant que le modèle symmetric point et légèrement moins bon que le modèle *full-mesh*.

Un autre modèle intéressant a été proposé dans [80]. Il permet de construire, à partir d'une représentation *full-mesh*, un spanning tree passant par tous les noeuds de bordures [Fig.2.10-(e)]. Ceci permettra d'agréger la structure en *full-mesh* en Card(B)-1 liens, d'où un espace de complexité de o(Card(B)).

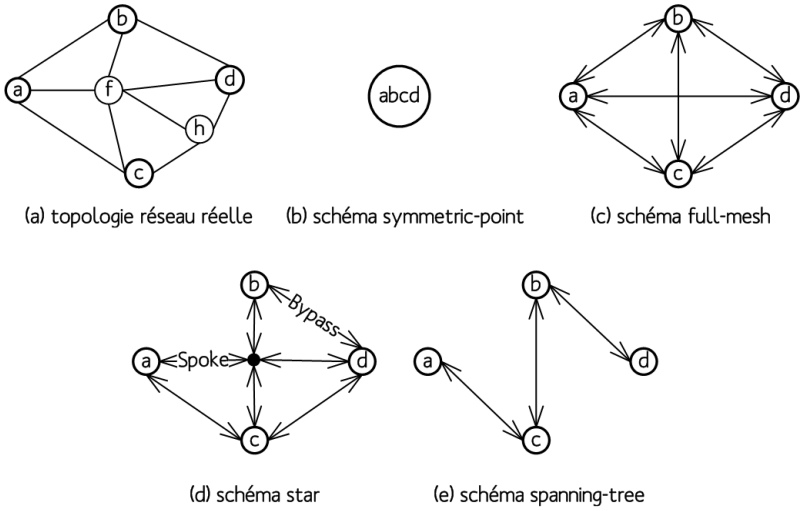

FIG. 2.10: Schémas d'agrégation de noeuds

L'Agrégation de Liens

Une étape fondamentale dans le processus d'agrégation est l'affectation du coût (paramètres de qualité de service) aux chemins logiques. Cette étape est utilisée afin de construire le schéma *full-mesh* d'un domaine réseau donné. En effet, un chemin logique correspond à l'agrégation de tous les chemins possibles entre deux noeuds de bordure. Dans le cas d'un paramètre de qualité de service unique, le poids du chemin logique est égal au coût du meilleur chemin physique qui vérifie le paramètre de QoS en question [Fig.2.11]. Ce coût dépend fortement de la nature du paramètre de QoS, qu'il soit additif, multiplicatif ou concave.

Soit $\widehat{mt}(\ell_i)$ la métrique attribuée à un lien physique ℓ_i. Pour un chemin $\mathtt{L} = \{\ell_i / i = 1, \ldots, n\}$.

Définition 1. $\widehat{mt}$ est dit additif si :

$\widehat{mt}(\mathtt{L}) = \sum_{i=1}^{n} \widehat{mt}(\ell_i)$. Le coût d'un chemin L est calculé à partir de la somme des pondérations des différents liens constituant ce chemin. Le délai et la gigue sont des paramètres additifs.

Définition 2. $\widehat{mt}$ est dit multiplicatif si :

$\widehat{mt}(\mathtt{L}) = \prod_{i=1}^{n} \widehat{mt}(\ell_i)$. Le coût d'un chemin L est calculé à partir du produit des pondérations de différents liens constituant ce chemin. Le taux de perte (loss rate) peut être considéré comme paramètre multiplicatif.

Définition 3. $\widehat{mt}$ est dit concave si :

$\widehat{mt}(\mathtt{L}) = \min | \max\{\widehat{mt}(\ell_i)/i = 1, \ldots, n\}$. Le coût d'un chemin L est calculé à partir de la valeur minimale ou maximale des pondérations des liens constituant ce chemin. La bande passante est un exemple de paramètres concaves où le coût d'un chemin est égal à la bande passante minimale tout au long de ce chemin.

Si l'identification des métriques des chemins logiques est simple dans le cadre d'un seul paramètre de QoS, elle est beaucoup plus compliquée si on tient compte de paramètres multiples et peut diverger facilement. En particulier, le meilleur chemin qui vérifie un paramètre de qualité de service n'est pas nécessairement le meilleur qui vérifie les autres paramètres de QoS. Par exemple, dans la figure 2.12, l'ensemble des chemins logiques entre les noeuds de bordure a et

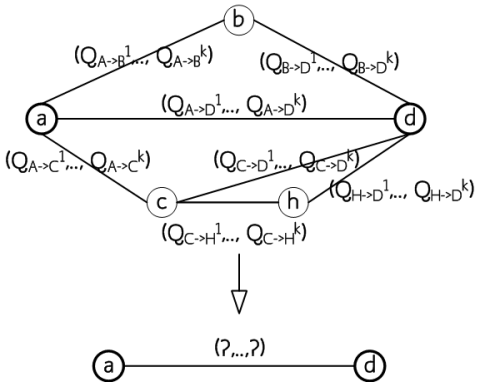

FIG. 2.11: Le principe d'agrégation de liens

d ne vérifient pas tous les mêmes paramètres de QoS : (3,5) est le meilleur chemin en termes de délai et (11,13) est le meilleur en termes de bande passante.

Plusieurs travaux de recherche ont étudié la manière dont les chemins logiques sont calculés avec plusieurs paramètres de QoS. Nous nous restreindrons dans cette section à présenter les travaux les plus intéressants.

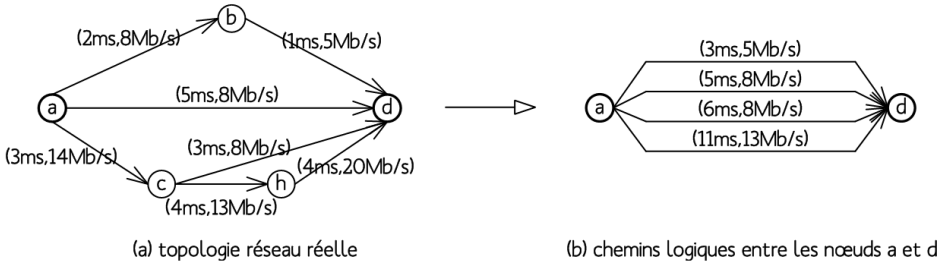

FIG. 2.12: Problématique d'agrégation de liens

Single-Path-Parameters Approach :

Le *SPPA* (Single-Path-Parameters Approach) introduit un ordre préférentiel entre les différents paramètres de qualité de service. Ainsi, l'identification du chemin logique se fait suivant le paramètre de qualité de service prioritaire. En effet, le coût d'un chemin logique est toujours calculé vis-à-vis d'un seul paramètre de QoS. Plusieurs modèles d'agrégation de topologie se sont articulés autour de cette approche [81, 82, 83, 84].

Multiple-Path-Parameters-Best-Case Approach :

Le *MPPBCA* (Multiple-Path-Parameters-Best-Case Approach) consiste à assigner à un chemin logique les meilleures valeurs des paramètres de qualité de service. Dans la figure 2.12, le chemin logique sélectionné sera identifié par le couple délai/bande passante (3,13), où 3 et 13 sont respectivement les valeurs optimales des paramètres de délai et de bande passante. Cette approche est agressive car les chemins réels ne sont pas tous capables de supporter les paramètres de QoS annoncés.

Multiple-Path-Parameters-Worst-Case Approach :

Contrairement au *MPPBCA*, le *MPWBCA* (Multiple-Path-Parameters-Worst-Case Approach) assigne les pires valeurs de qualité de service aux chemins logiques. Dans la figure 2.12, le chemin logique sélectionné sera identifié par le couple délai/bande passante (11,5), où 11 et 5 représentent respectivement les pires valeurs des paramètres de délai et de bande passante. Cette approche engendre une sous utilisation des ressources réseau car il est possible de refuser des appels admissibles par le réseau.

Closest-Single-Path Approach :

Korkmaz et Krunz [85] on introduit un indicateur d'élasticité nommé le s_{factor} permettant de calculer la déviation des paramètres de QoS d'un lien logique par rapport aux paramètres de QoS optimaux.

Etant donné H l'ensemble des chemins possibles entre deux points d'extrémité. Chaque chemin est représenté par z paramètres de qualité de service $\mathcal{Q}$ dont w sont à maximiser (Best$_{\mathcal{Q}^k}$) et (z-w) sont à minimiser (Best$_{\mathcal{Q}^l}$).

$\mathtt{H} = \{\mathtt{H}_1(\mathcal{Q}_1^1, \ldots, \mathcal{Q}_1^z), \ldots, \mathtt{H}_n(\mathcal{Q}_n^1, \ldots, \mathcal{Q}_n^z)\}$

$\text{Best}_{\mathcal{Q}^k} = \max\{\mathcal{Q}_i^k / \, 1 \leq i \leq n\}, \; pour\, k = 1, \ldots, w.$

$\text{Best}_{\mathcal{Q}^l} = \min\{\mathcal{Q}_i^l / \, 1 \leq i \leq n\}, \; pour\, l = w+1, \ldots, z.$

$$s_{factor_i} = \sum_{k=1}^{w} \frac{Best_{\mathcal{Q}^k}}{\mathcal{Q}_i^k} + \sum_{l=w+1}^{z} \frac{\mathcal{Q}_i^l}{Best_{\mathcal{Q}^l}}$$

$$s_{factor} = \min\{s_{factor_i} / \, 1 \leq i \leq n\}$$

Sur la base des paramètres Best$_{\mathcal{Q}^k}$ et Best$_{\mathcal{Q}^l}$, le meilleur lien logique est celui qui représente l'indicateur d'élasticité minimal.

Les mêmes auteurs ont proposé une amélioration de l'approche *MPPBCA* qui consiste à assigner à un lien logique l'ensemble des paramètres de qualité de service optimaux ainsi que l'indicateur s_{factor}. Par conséquent, un lien logique est défini par les z+1 valeurs ($Best_1$, $Best_2$, ..., $Best_z$, s_{factor}).

Contrairement aux approches citées précédemment, Lui et Tang [86] [87] ont formulé le problème d'agrégation d'une façon différente basée sur un modèle géométrique. En effet, elle consiste tout d'abord à représenter l'ensemble des chemins entre deux noeuds de bordure par des points dans un repère euclidien $\mathcal{P}_{(\mathtt{X},\mathtt{Y})}$ où X est l'axe représentant le paramètre de qualité de service additif et Y celui représentant le paramètre concave. Ensuite, seuls les points représentatifs sont sélectionnés afin de construire une courbe que nous appellerons tout au long de ce document la fonction "Escalier".

Définition 4. Un point (x,y) est dit plus représentatif [86] qu'un point (x',y') si :

- $x \neq x'$ *ou* $y \neq y'$, et
- $x \leq x'$ *et* $y \geq y'$.

Définition 5. Soit R l'ensemble des points dans le plan $\mathcal{P}_{(\mathtt{X},\mathtt{Y})}$. Un point (x,y) est représentatif dans R, si :

$\forall\, (x', y') \in \mathtt{R}, \; x < x'$ *ou* $y > y'$.

Comme le montre l'exemple dans la Figure 2.13, les points représentatifs définissant la fonction escalier dans le plan cartésien $\mathcal{P}_{(delay,\, bandwidth)}$ sont (5,3), (8,5) et (13,11). (8,6) n'est pas un point représentatif.

Sur la base de cette modélisation géométrique, plusieurs approches ont été proposées dans la littérature. Dans le reste de ce mémoire, nous utiliserons $\widehat{dl}$ pour dénoter le paramètre de délai et $\widehat{bw}$ pour dénoter le paramètre de bande passante.

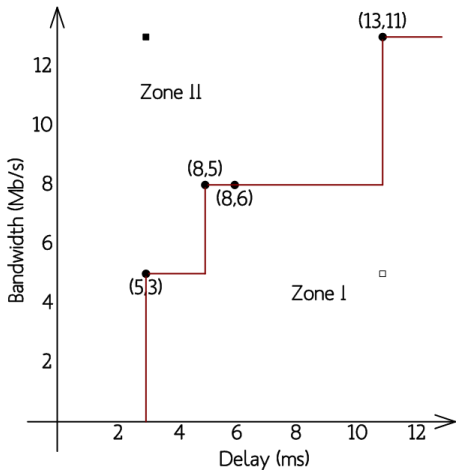

FIG. 2.13: Agrégation de liens dans le plan cartésien $\mathcal{P}_{delay,bandwidth}$

La Méthode Line Segment :

La méthode LS (Line Segment) [86] étudie le problème d'agrégation de topologie dans le cas des réseaux sensibles aux paramètres de délais et de bande passante. Elle consiste à représenter l'ensemble des chemins physiques entre deux noeuds de bordure par des couples $(\widehat{dl}, \widehat{bw})$ dans un repère euclidien et de trouver par la suite à l'aide de la méthode des moindres carrés, le segment qui approxime le mieux la fonction escalier. En effet, il s'agit de minimiser la somme des carrés des distances entre les points représentatifs et le segment. Ainsi, un chemin logique sera représenté par un segment identifié par deux points d'extrémité [Lower endpoint, Upper endpoint].

Méthodes de Tang et al :

Tang et al [87] s'articulent sur une fonction $\mathcal{K}(\widehat{bw})$ afin d'approximer la fonction escalier définie par l'ensemble des points représentatifs R. Cette fonction $\mathcal{K}(\widehat{bw})$, n'est que la représentation du délai en fonction de la bande passante. Elle est définie de la façon suivante :

$$\mathcal{K}(\widehat{bw}) = \begin{cases} \widehat{dl}_1 & ; \widehat{bw} \leq \widehat{bw}_1 \\ \widehat{dl}_i & ; \widehat{bw}_{i-1} < \widehat{bw} \leq \widehat{bw}_i \quad ; \forall i, 1 < i \leq \mathtt{R}, \\ \infty & ; \widehat{bw} > \widehat{bw}_1 \end{cases}$$

Sur la base de la fonction d'approximation $\mathcal{K}$, Tang et Chen ont proposé trois formulations différentes.

Polynomial Curve Approximation :

Ce modèle d'approximation utilise une fonction polynomiale de la forme,

$$\mathcal{K}_{pc}(\widehat{bw}) = \begin{cases} \widehat{dl}_l & ; \widehat{bw} \leq \widehat{bw}_l \\ \sum_{i=0}^{n} \alpha_i \times \widehat{bw}^i & ; \widehat{bw}_l < \widehat{bw} \leq \widehat{bw}_h \\ \infty & ; \widehat{bw} > \widehat{bw}_h \end{cases}$$

Il est à noter que ($\widehat{dl}_l$, $\widehat{bw}_l$) désigne le point de la fonction escalier ayant la bande passante minimale et ($\widehat{dl}_h$, $\widehat{bw}_h$) le point ayant la bande passante maximale.

Nous remarquons que la méthode *line-segment* présentée dans [86], n'est qu'un cas spécial de l'approximation *Polynomial Curve* avec n=1. Avec une grande valeur de n, ce modèle atteint une meilleure précision, par contre les traitements sont plus coûteux et la charge augmente. La méthode du moindre carré est utilisée afin de trouver les meilleurs coefficients α_i permettant de minimiser l'erreur.

Cubic Spline Approximation :

Cette approche fait recours au spline[19]. Elle approxime la fonction escalier segment par segment en utilisant l'approche précédente $\mathcal{K}_{pc}$. Pour cela l'intervalle de bande passante [$\widehat{bw}_l$, $\widehat{bw}_h$] est divisé en un nombre pair g de partitions λ tel que $\widehat{bw}_l = 0 < 1 < \cdots < g = \widehat{bw}_h$. La fonction définie est formulée de la façon suivante :

$$\mathcal{K}_{cs}(\widehat{bw}) = \begin{cases} \widehat{dl}_l & ; \widehat{bw} \leq \widehat{bw}_l \\ \sum_{j=0}^{g} \alpha_i \times \mathcal{F}_j(\widehat{bw}^i) & ; \widehat{bw}_l < \widehat{bw} \leq \widehat{bw}_h \\ \infty & ; \widehat{bw} > \widehat{bw}_h \end{cases}$$

Avec, $\mathcal{F}_j(\widehat{bw}) = (\lambda_{j+2} - \lambda_{j-2}) \sum_{s=-2}^{2} \frac{(\lambda_{j+s} - \widehat{bw})^3}{\prod_{k=-2, k\neq s}^{2} (\lambda_{j+s} - \lambda_{j+k})}$.

Comme pour l'approche précédente, la méthode du moindre carré est utilisée pour trouver les coefficients α_i permettant de minimiser l'erreur.

Polyline Approximation :

Cette approche repose sur l'utilisation des *corner points*[20]. En effet, ces points contiennent plus d'informations de QoS que les autres points. La logique derrière la fonction *PolyLine* consiste à relier un nombre k de points présentatifs par une fonction polynomiale $\mathcal{K}_{pl}$. Pour ce faire, Tang et Chen ont introduit une heuristique permettant de trouver les *corner points* les plus distants de la fonction *Polyline* $\mathcal{K}_{pl}$. Initialement, la fonction $\mathcal{K}_{pl}$ est représentée par un segment reliant les deux points d'extrémités p_1-p_n. Le coin ayant la plus grande distance de la fonction $\mathcal{K}_{pl}$, est inséré dans le segment initial p_1-p_j-p_n. Par conséquence, la fonction *Polyline* n'est plus représentée par un segment mais par la concaténation de deux segments. Ce traitement est répété jusqu'à ce que la fonction *Polyline* contienne k *corner points*.

[19]Dans le domaine de l'analyse numérique, une spline est une fonction définie par morceaux par des polynômes. Elle est souvent utilisée dans des problèmes d'interpolation.

[20]Il s'agit de tous les points de coin de la fonction escalier, à savoir les coins entrants et sortants.

2.6 Bilan

Ce chapitre a survolé les différents aspects liés au contrôle d'admission dans les réseaux d'opérateurs. Notre étude a commencé par identifier les principales caractéristiques liées à la fonction de contrôle d'admission. Nous avons ensuite survolé les plus intéressantes approches de CAC qui ont été proposées dans la littérature. L'approche de Bandwidth Broker a été étudiée avec beaucoup plus de détail. Il s'agit de la solution retenue par le consortium du projet EuQoS [2] pour assurer la fonction de contrôle d'admission. Malgré les avantages multiples qu'un tel mécanisme apporte, le découplage du plan de contrôle du plan de transfert pose des problèmes de synchronisation qui peuvent mettre la fonction de CAC en défaut. De ce fait, une très grande importance doit être allouée au choix des mécanismes d'approvisionnement de la topologie réseau maintenue par le Bandwidth Broker.

Après identification des différents paramètres de qualité de service que le BB gère, nous avons exploré les différents outils et algorithmes de recherche de topologie qui existent dans la littérature et sur le marché. Parmi ces outils, certains opèrent au niveau liaison et d'autres au niveau réseau. Les outils de niveau 2 prennent en compte tous les équipements actifs intervenant dans le fonctionnement et la transmission d'informations en couche Liaison. Quand aux outils de niveau 3, ils représentent la cartographie d'un réseau IP, et seuls les équipements capables d'effectuer le routage des paquets IP seront représentés. A la fin du chapitre, les limites des outils existants ont été dégagées afin de mettre en évidence l'intérêt d'avoir une nouvelle solution satisfaisant les critères de performance que nous avons identifiés. Il s'avère que la majorité des outils existant sur le marché sont basés sur des mécanismes de probing ICMP ou SNMP. En effet, ce n'est pas l'équipement qui annonce sa topologie, c'est une station externe qui l'interroge pour envoyer ses informations de topologie. Ceci est intéressant dans le cadre des applications d'inventaire, mais n'est pas adapté à un contexte réactif tel que celui de la fonction de contrôle d'admission. Les autres solutions, non basées sur le probing, sont soit propriétaires tel que CDP, soit restreintes à une technologie bien définie tel que Ethernet avec le protocole LLDP. Ils offrent cependant un concept intéressant : c'est l'équipement lui même qui s'annonce et qui découvre ses voisins. C'est à partir de ce principe que nous avons élaboré notre proposition. Outre la façon d'acquérir la topologie, la manipulation des données collectées peut poser des problèmes dans le cadre de topologies de grande échelle. Si nous nous référons à l'architecture du Bandwidth Broker évoquée précédemment, la manipulation d'une grande volumétrie d'informations de topologie peut causer des problèmes de passage à l'échelle. En effet, sur des réseaux étendus, le taux des informations de topologie et de QoS augmente, ce qui introduit des limitations au niveau des équipements de gestion en termes de capacité de traitement. Pour cela, nous avons exploré une piste prometteuse afin de pallier à ces problèmes de passage à l'échelle, il s'agit du concept d'agrégation de topologie. Ce concept a été introduit par les spécifications ATM dans le cadre des réseaux hiérarchiques. Malgré l'originalité de l'idée et de sa consistance, le sujet d'agrégation de topologie n'a été exploré que dans le contexte du routage à QoS. Le principe de base d'un tel mécanisme est de rendre plus compacte une topologie réseau. En effet, juste quelques informations jugées nécessaires seront présentes dans la topologie agrégée. Nous pouvons distinguer deux types d'agrégation : une basée sur les liens et une autre basée sur les noeuds. Le premier type d'agrégation consiste à réduire la volumétrie des informations de topologie d'un domaine réseau en une seule entité logique. Plusieurs approches ont été présentées dans l'état de l'art, les plus intéressantes étant le schéma *full-mesh* et le schéma *star*. Une étude comparative faite dans [79] a montré la supériorité du modèle *full-mesh* en termes de précision, alors qu'en termes de réduction de volumétrie d'informations le modèle *star* s'avère plus intéressant. Le deuxième

type d'agrégation permet de représenter un ensemble de chemins physiques entre deux noeuds par un seul chemin logique de bout en bout. Nous nous sommes concentré sur les approches les plus intéressantes présentées dans la littérature. L'analyse de ces différentes solutions a montré les bonnes performances du schéma *PolyLine*. En effet, l'agrégation de lien commence par établir une fonction escalier définissant les différentes métriques des chemins physiques. Cette fonction est par la suite approximée par des segments sélectionnés suivant un ensemble de règles prédéfinies. Par conséquent, le lien logique agrégé sera représenté par une succession de segments sur des intervalles contigus. Ainsi, sur la base des approches d'agrégations recensées, nous essayerons dans le reste de ce mémoire d'apporter une réponse à l'ensemble de ces questions : De quelle façon ces types d'agrégation pourront nous aider à accomplir nos objectifs ? Quels sont les modèles les plus appropriés à notre cas d'utilisation ? Quels seront nos critères de choix ?

Chapitre 3

STAMP : Un Protocole de Découverte de Topologie réactif

Sommaire

3.1 Introduction

Nous avons vu dans le chapitre précédent qu'il n'existe pas de solution satisfaisante pour l'acquisition de topologie vis-à-vis des critères de sélection et de performance nécessaires pour une fonction de CAC basée sur la topologie. En effet, la plupart des solutions étudiées se basent sur des mécanismes de probing, ce qui est principalement inadaptée au critère de réactivité. Cependant, deux techniques semblent avantageuses mais incomplètes. Il s'agit des protocoles CDP (Cisco Discovery Protocol) et LLDP (Link Layer Discovery protocol). Ces deux protocoles offrent un modèle fonctionnel très intéressant. Deux phases sont nécessaires pour l'acquisition de la topologie. Une première phase de découverte de voisinage et une deuxième phase de remontée de données de topologie à une station de gestion. Lors de la première phase, les équipements réseaux annoncent leur présence dans le voisinage direct. Aussi, ils récupèrent les informations de topologie annoncées par leurs voisins directs. La deuxième phase consiste à remonter toutes les données de topologie à la station de gestion. Ceci est accomplit via le protocole SNMP. La station de gestion interroge les équipements réseau pour collecter toutes les informations acquises lors de la phase de découverte de topologie. Malgré les avantages que CDP et LLDP offrent, malheureusement ils sont inadaptés à notre cas d'utilisation. CDP est une solution propriétaire. Elle n'offre pas un cadre générique d'application, et n'est supportée que par les équipements Cisco. LLDP, lui, ne s'applique qu'au monde IEEE (Ethernet) et ne prend pas en compte d'autre technologie comme l'ATM. En outre, les deux solutions se basent sur le protocole SNMP pour la remontée des données de topologie. De telles approches passives représentent une entrave majeure pour un système d'acquisition de topologie réactif. En effet, les équipements réseaux n'on pas d'autonomie pour pouvoir annoncer leurs caractéristiques techniques. Même l'utilisation des traps (Alertes SNMP) ne peut pas être une solution efficace dans un environnement très dynamique.

Comme solution palliative, nous proposons dans ce chapitre un nouveau mécanisme d'acquisition de topologie. Il s'agit du protocole STAMP (Simple Topology Annoucement Protocol). Nous gardons le même modèle fonctionnel que celui des protocoles CDP et LLDP, et nous réétudions la façon avec laquelle les deux phases sont accomplies. L'idée est de ne pas se restreindre seulement à la technologie Ethernet lors de la phase de découverte de voisinage et de rendre l'équipement plus autonome pour la remontée de ces données de topologie. Mais avant d'élaborer le protocole en lui-même, il est important de déterminer les données de topologie dont la fonction de CAC a besoin. Une des difficultés de la description de la topologie est de pouvoir désigner de façon unique tous les objets ainsi que les liens connectant ces objets. La modélisation des données doit donc nous permettre de déterminer quel type d'identifiant il faut manipuler ainsi que la structure de donnée permettant de les stocker avec leurs relations. La section suivante sera consacrée à la description détaillée du modèle de données.

3.2 Le Modèle de Données de la Topologie

Dans notre cas d'utilisation, nous avons considéré les hypothèses suivantes afin de réussir la conception d'un modèle de données complet, générique et facilement extensible :

- Pour la couche 3, nous nous restreindrons à l'étude du protocole IP (IPv4 et IPv6), et nous ignorons les autres protocoles réseaux tel que X.25, IPx, etc. qui sont peu ou pas utilisés par les services et les ressources auxquels nous essayons d'apporter une réponse,
- Pour la couche 2, nous nous intéressons aux technologies GE (Gigabit Ethernet), ATM (Asynchronous Transfert Mode), VLAN et MPLS (MultiProtocol Label Switching) et nous

ignorerons FDDI (Fiber Distributed Data Interface), FR (Frame Relay), X.25 etc. du fait qu'elles sont de moins en moins utilisées. Le cas du POS (Packet Over SONET/SDH) sera cité dans les perspectives en tant qu'illustration de la flexibilité de STAMP à l'étendre à d'autres technologies.

En s'appuyant sur l'ensemble de ces hypothèses, nous avons élaboré un modèle exhaustif des schémas d'encapsulation qui peuvent exister entre les différentes technologies prises en considération. L'idée est d'identifier, sur chacun des équipements, le plan de transfert ainsi que les paramètres de configuration nécessaires pour la fonction de CAC. Ce modèle d'encapsulation est bâti sur les deux technologies de liaison Ethernet et ATM au dessus des quelles les technologies IP, VLAN, MPLS sont montées. En effet, 23 schémas d'encapsulation ont été identifiés : IPv6 sur Ethernet (1), IPv4 sur Ethernet (2), IPv4 sur IPv6 (3), IPv4 sur IPv4 (4), IPv6 sur IPv4 (5), IPv6 sur IPv6 (6), IPv4 sur MPLS (7), IPv6 sur MPLS (8), MPLS sur MPLS (9), MPLS sur ATM (10), ATM sur MPLS (11), IPv4 sur VLAN (12), IPv6 sur VLAN (13), VLAN sur MPLS (14), VLAN sur ATM (15), VLAN sur Ethernet (16), IPv4 sur ATM (17), IPv6 sur ATM (18), ATM sur ATM (19), Ethernet sur ATM (20), ATM sur Ethernet (21), Ethernet sur MPLS (22) et MPLS sur Ethernet (23).

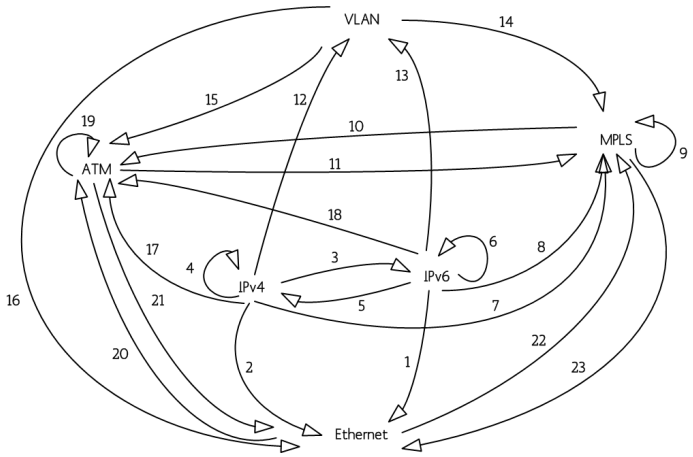

FIG. 3.1: Schéma des encapsulations

Une telle granularité dans la description des différents schémas d'encapsulation est nécessaire pour pouvoir identifier de manière précise l'ensemble des technologies configurées sur un port donnée et de réduire par conséquent la marge d'erreur lors de la phase de remontée des données de topologie.

Sur la base du schéma des encapsulations nous avons bâti notre modèle de données de la topologie. L'idée est de situer les différentes technologies identifiées au coeur de l'équipement réseau. Pour cela, nous nous sommes lancés dans l'identification des différents composants et liens qui décrivent un équipement et qui le situe par rapport à ses voisins dans la topologie réseau. Nous pouvons distinguer deux vues possibles : une vue opératoire décrivant toutes les caractéristiques techniques de l'équipement ainsi que son plan d'adressage [Fig.3.2], et une vue descriptive détaillant toutes les propriétés organisationnelles ainsi que le type de l'équipement [Fig.3.3]. Chacune de ces deux vues sera examinée avec plus de détails dans les sections suivantes.

3.2.1 La Vue Opératoire

Le but derrière la définition de la vue opératoire est d'identifier les caractéristiques fonctionnelles d'un équipement réseaux. Il s'agit de déterminer toutes les données de topologie nécessaires pour la fonction de contrôle d'admission. Comme une grande partie de ces informations est déduite à partir du plan d'adressage, nous avons identifié l'équipement par l'ensemble des interfaces qui le composent et qui le situe dans la topologie via une multitude de technologies de niveau 2 et 3 (schéma des encapsulations) auxquels des identifiants ou des adresses peuvent être attribués. En effet, un équipement réseau peut contenir une ou plusieurs interfaces. Il est à noter que dans notre modèle, l'interface ne correspond pas à la carte physique que l'on insère dans l'équipement réseau. Une telle carte peut comporter plusieurs interfaces, c'est-à-dire plusieurs ports physiques. Les constructeurs font d'ailleurs la distinction en utilisant le terme *slot* pour désigner l'emplacement des cartes. Comme l'interface est un point de raccordement physique de l'équipement réseau, elle doit être nécessairement identifiée par un et un seul port physique. Il est aussi possible de configurer sur l'interface plusieurs ports logiques. En effet, par le terme port, nous désignons des entités physiques ou logiques auxquelles il est possible d'attribuer des identifiants de commutation ou des adresses. Le port logique représente un point de communication identifié par une adresse non figée de niveau 3. Il s'agit d'une adresse IP de type v4 ou v6. Le port logique peut aussi faire référence directement à l'équipement réseau, ce qui est le cas des adresses de loopback. Ces adresses ne sont pas rattachées à une interface physique, elles font partie intégrante de l'équipement. Il est à noter qu'il est possible de configurer plusieurs ports logiques sur l'équipement réseau. En ce qui concerne les ports physiques, le rattachement d'une adresse figée à ce type de port garantit son unicité dans le réseau. En effet, un port physique est identifié de manière unique soit par son adresse MAC (Medium Access Control) dans le cas de la technologie Ethernet, soit par son adresse ESI (End System Identifier) dans le cas de la technologie ATM. Il est à noter que la valeur de ces deux types d'adresse est attribuée par le constructeur.

La distinction entre port physique et port logique a pour but de mettre en évidence deux types de plan d'adressage et de séparer par la suite les technologies identifiées dans le schéma des encapsulations suivant les couches 2 et 3 du modèle OSI. En effet, nous distinguons un plan d'adressage physique figé par le constructeur et un autre plan logique définit par l'administrateur réseau suivant ses besoins et ses politiques de routage et de commutation. Selon le niveau OSI (couche 2 ou 3) et la technologie ciblée (Ethernet, ATM, IP, ...), la jointure entre les ports logiques et le port physique est faite via les deux composants "identification couche 2" et "identification couche 3". L'idée est de situer les technologies du schéma des encapsulations par rapport au couche OSI 2 et 3. Le composant "identification couche 2" est identifié par une adresse unique de type MAC ou ESI ainsi qu'un ensemble de technologies de commutation de niveau 2. De telles technologies sont montées au dessus de l'adresse physique figée. Elles peuvent être des labels MPLS, des identifiants de VLAN ou des identifiants de VP/VC. Aussi, il est possible de monter autant de technologies sur l'adresse physique de niveau 2. Ces technologies peuvent être encapsulées dans elles mêmes. Il n'y a pas de restriction sur la cardinalité des encapsulations dans le cas des technologies MPLS et VLAN. Pour ATM, un seul niveau d'encapsulation est possible. En effet, nous pouvons travailler directement au niveau VP ou bien encapsuler les VCs dans le VP et donc travailler au niveau VP/VC. Le composant "identification couche 2" est aussi en relation directe avec le composant "identification couche 3". Il est possible de rattacher à une adresse physique de niveau 2 un ensemble d'adresses logiques de niveau 3, c'est-à-dire un ensemble d'adresses IP. Ces adresses peuvent être considérées comme des alias au niveau du port physique. Comme dans le cas des technologies MPLS, VLAN et ATM, IP peut aussi être

encapsulé dans lui-même. Il est à noter que le schéma des encapsulations [Fig.3.1] fournit des informations complémentaires sur la description des liens mettant en relation les différentes technologies au niveau du composant "identification couche 2".

D'un point de vue fonctionnel, un équipement réseau peut implémenter plusieurs protocoles de routage et/ou d'ingénierie de trafic (TE : Trafic Engineering). Ces protocoles sont utiles pour la collecte des informations de routage, d'état de liens et de QoS. Il est aussi possible que l'équipement soit passif ou un simple équipement de commutation qui n'implémente aucun de ces protocoles.

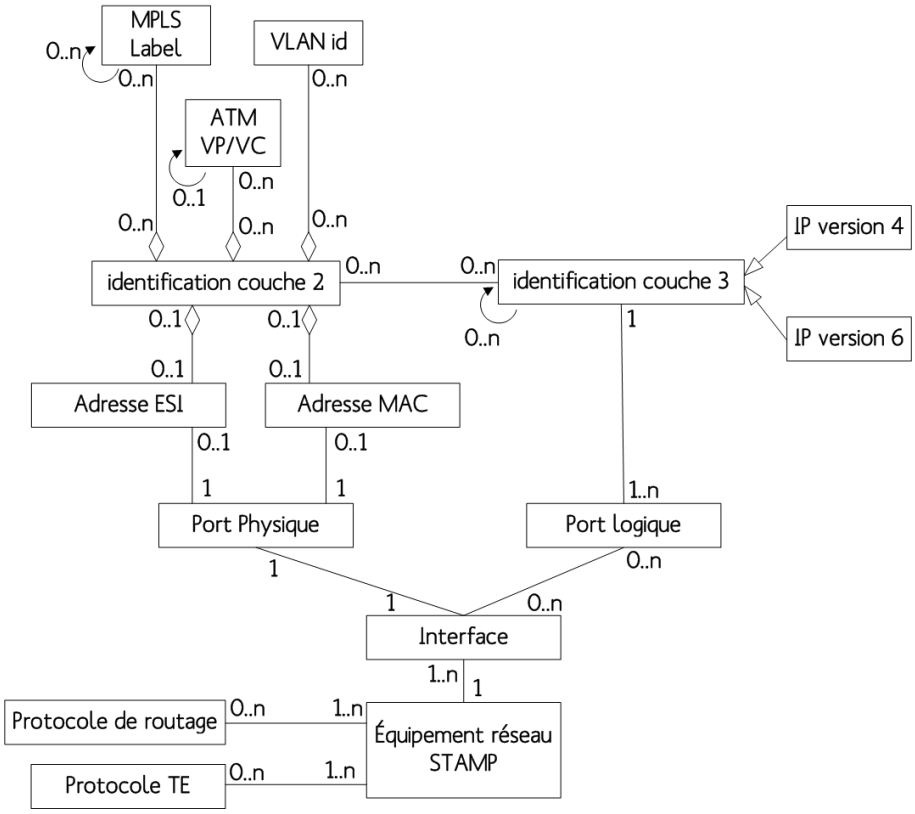

FIG. 3.2: STAMP : La vue opératoire

3.2.2 La Vue Descriptive

Si la vue opérationnelle est capable de remonter des informations de connectivité et d'adressage, elle est incapable de mettre en évidence les propriétés de l'équipement ainsi que son rôle dans le domaine. Toutes ces informations font partie de la vue descriptive. En effet, chaque équipement a un rôle bien déterminé qui identifie sa fonction dans le groupe STAMP. Par le terme groupe, STAMP définit une découpe organisationnelle du domaine réseau. Elle identifie un ensemble d'équipements réseau rattaché administrativement à une même autorité. En effet, un réseau est toujours organisé selon une découpe administrative, logique, hiérarchique que le protocole de découverte de topologie doit pouvoir reconstituer. Le groupe peut être un POP (Point of Presence), une aire OSPF, une région IS-IS, un backbone IP ou même un Système Autonome AS. La périphérie d'un groupe est fixée par l'administrateur, qui devra vérifier qu'il n'y a pas de chevauchement entre les groupes STAMP et les autres découpes qui existent déjà dans le réseau.

Il est à noter qu'un équipement réseau peut être un élément de bordure ou pas. Par élément de bordure, nous évoquons la propriété d'adjacence directe avec d'autre groupe, c'est-à-dire les équipements ayant au moins une interface dans le groupe et une autre interface dans le groupe voisin. Aussi, nous avons identifié les types des équipements possibles dans un réseau d'opérateur Télécom. Un équipement ne peut être identifié que par un et un seul type. Généralement, le type donne une idée sur la fonction de l'équipement dans le groupe. Cependant, il ne désigne pas le rôle attribué à l'équipement. En effet, dans la terminologie STAMP, le rôle identifie la tâche d'un équipement réseau dans le processus de découverte de topologie. Un et un seul rôle est attribué par équipement. Dans notre mode de fonctionnement, nous avons défini trois rôles :

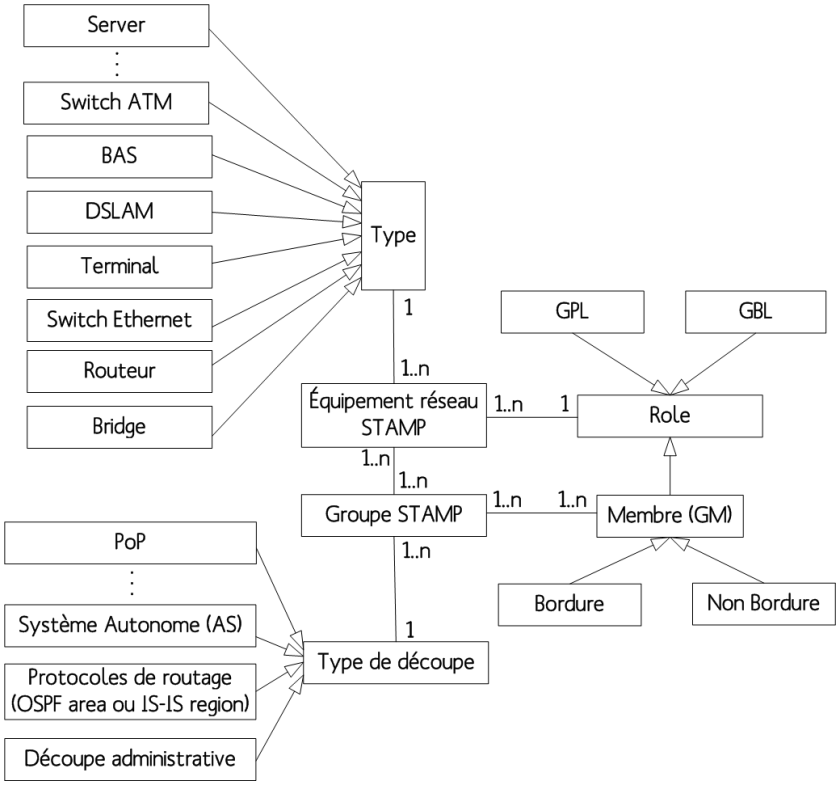

FIG. 3.3: STAMP : La vue descriptive

Le GM (Group Member)

Il s'agit du rôle attribué à tous les équipements au sein du groupe. Ce sont ces équipements dont nous cherchons à établir la topologie. L'échange des données de topologie, identifiées par la vue opératoire [Fig.3.2], lors de la phase de découverte de voisinage permet à chaque GM de maintenir une structure d'adjacence *adb* (adjacency database) décrivant la topologie de son entourage. Cette structure identifie pour chaque port physique le port physique distant directement connecté ainsi que l'identifiant de l'équipement auquel ce dernier appartient. La grammaire de l'*adb* est définie de la façon suivante :

<adb> :=
<Port-NIC-Address>
<Adjacent-MAC-Management-Address>

<Adjacent-Port-NIC-Address>
<Link-Status>

Aussi, chaque GM détient une deuxième structure de données, la *cdb* (characteristics database). Cette structure définit l'ensemble des caractéristiques techniques de l'équipement, à savoir ses informations de routage, ses données de QoS, les encapsulations qu'il supporte, etc. Ces données peuvent être déduites à partir des tables de routage ou des MIBs déjà configurées sur l'équipement. Les constructeurs peuvent aussi approvisionner la *cdb* par le biais d'algorithmes spécifiques implémentés au niveau de l'OS (Operating System). L'ensemble de ces caractéristiques techniques sont envoyées au manager qui collecte toutes les informations. Il est à noter qu'un groupe STAMP peut contenir de 1 à plusieurs GMs. Aussi un GM peut appartenir à plusieurs groupes s'il est un élément de bordure.

Le GPL (Group Primary Leader)

Il s'agit de l'entité qui centralise toutes les informations de topologie du groupe. Dans nos spécifications, nous considérons que le GPL peut accomplir les mêmes fonctionnalités qu'un Bandwidth Broker. Les principales fonctionnalités assurées par le GPL sont les suivantes :

- Déterminer et valider la connectivité entre les différents équipements réseau au sein du groupe,
- Maintenir une cartographie mise à jour des informations de topologie, de routage et de QoS du groupe,
- S'assurer que le GPL de back-up à une image réelle et uniforme de la topologie du groupe.
- Accomplir des fonctions de gestion et de contrôle tel que la fonction de CAC à la quelle nous nous intéressons dans ce mémoire.

La base de topologie résultante stockée dans le GPL est nommée la *tdb* (topology database). En effet, toutes les informations reçues de la part des GMs, sont stockées, après corrélation, dans la tdb afin de maintenir une vue globale des informations de topologie du groupe. Chaque GM est décrit au sein de la tdb par l'ensemble des informations suivantes :

- L'identifiant de l'équipement (device-id), correspond à l'identifiant unique attribué au GM au sein du groupe. Les spécifications de cet identifiant seront détaillées dans la section suivante,
- Le certificat d'authentification (certificate-path) : ce champ est utilisé dans le cas où les équipements sont configurés en mode sécurisé. En effet, STAMP supporte deux modes de fonctionnement, à savoir un mode *safe* et un mode *unsafe*. Dans le mode *safe*, un certificat d'authentification est nécessaire pour l'authentification des GMs. Le chemin de ce certificat est défini par le champ "certificate-path",
- Le type de l'équipement (device-type) : identifie la fonction assurée par le GM conformément à la vue descriptive du modèle de donnée [Fig.3.3],
- La localisation dans le groupe (is-GBN) : ce champ positionne le GM dans le groupe. En effet, il indique s'il s'agit d'un équipement de bordure (br) ou pas (nbr),
- L'état de l'équipement (is-Actif) : ce champ renseigne l'état de l'équipement vis-à-vis de la signalisation STAMP. Il identifie si un équipement est actif ou inactif dans la tdb,
- L'ensemble des propriétés propres aux ports sont décrites par les deux champs "eth-port" et "atm-port" suivant le type de la technologie déployée, Ethernet ou ATM.

Que le port soit de type Ethernet ou ATM, deux types d'informations clé sont nécessaires pour la corrélation des données au sein de la *tdb*. Ces informations sont l'identifiant du port ainsi que l'ensemble des encapsulations qu'il implémente. Le premier type d'information permet d'identifier de manière unique dans le groupe un port physique, soit par son adresse MAC dans le cas d'un équipement 802.3, soit par son adresse ESI dans le cas d'un équipement ATM. Le deuxième type d'information décrit l'ensemble des technologies de niveau 2 et/ou 3 configurées sur le port physique. Ainsi, l'identification des encapsulations dépendra du type des technologies déployées. Comme mentionnée précédemment, nous nous sommes restreint aux technologies MPLS, IP (v4 et v6), VP/VC et VLAN. IP est identifié par l'adresse logique de l'interface, l'adresse du broadcast s'il s'agit d'une adresse de type v4 ainsi que le préfixe réseau. MPLS est identifié par le label entrant et sortant du LSP. Les couples VP/VC identifient les chemins logiques qui peuvent être établies entres des équipements ATM. Ils sont identifiés par un identifiant VP (VP-id) et un identifiant VC (VC-id). Enfin, l'encapsulation VLAN identifie les VLAN-id configurés sur une interface. L'ensemble de toutes ces encapsulations sont corrélées et organisées suivant le champ (encapsulation-type) qui permettra au GPL de dresser des topologies de type logiques. La grammaire ci-dessous décrit en détail la structure de la *tdb*. Nous utilisons le signe (+) après le nom du champ de donnée pour indiquer qu'il s'agit d'au moins un champ de ce type. Le signe (*) indique qu'il peut y avoir zéro ou plusieurs champs de ce type. Le signe (?) indique que le champ est optionnel. Sinon, le champ est obligatoire et doit figurer une seule fois.

<tdb> :=
<device-id>
<certificate-path> ?
<device-type>
<is-GBN>
<is-actif>
<eth-port>*
<atm-port>*

<eth-port> :=
<MAC-address>
<symmetry>
<rate>
<metric>
<mtu>
<adjacent-port-id>
<adjacent-device-id>
<link-status>
<ipv4>*
<ipv6>*
<vlan>*
<mpls>*
<vpvc>*

<atm-port> :=
<ESI-address>
<NSAP-address>
<rate>
<mtu>

<adjacent-port-id>
<adjacent-device-id>
<link-status>
<ipv4>*
<ipv6>*
<vlan>*
<mpls>*
<vpvc>*

<ipv4> :=
<alias>
<ipv4-address>
<ipv4-broadcast>
<ipv4-prefix>
<encapsulation-type>

<ipv6> :=
<alias>
<ipv6-address>
<ipv6-prefix>
<encapsulation-type>

<vlan> :=
<alias>
<valn-id>
<encapsulation-type>

<mpls> :=
<alias>
<local-label>
<outgoing-label>
<encapsulation-type>

<vpvc> :=
<alias>
<vp>
<vc>
<encapsulation-type>

En plus de la tdb, le GPL détient une autre structure de données nommée la *qdb* (quarantine database). En effet, toutes les notifications de défaillances signalées par les GMs sont remontées au GPL et stockées dans cette base de données. Ceci permettra au leader principal d'avoir une vue générale sur l'état de l'équipement et de savoir par la suite si la défaillance est partielle (au niveau d'un port ou d'un lien) ou globale (au niveau de l'équipement). La *qdb* est décrite par la grammaire suivante :

<qdb> :=
<device-id> (GM concerné par la défaillance)
<failed-port>

Vu les fonctions multiples que le GPL assure, nous recommandons dans nos spécifications son déploiement sur un serveur externe du domaine dont nous cherchons à établir la topologie. L'ensemble de ses fonctionnalités ainsi que ses interactions avec les autres entités seront étudiées

avec plus de détails tout au long de ce mémoire.

Le GBL (Group Backup Leader)

Il assure la redondance du GPL. Il détient les mêmes structures de données que le GPL et remplit les mêmes fonctionnalités.

Dans la section suivante, nous étudions en détail le protocole de signalisation STAMP ainsi que la manière avec laquelle il manipule l'ensemble des données de topologie.

3.3 STAMP : Le Protocole de Signalisation

Le protocole STAMP a été bâti sur la base du modèle de données susmentionné. L'idée est de mettre en place un mécanisme de découverte de topologie capable d'assurer les fonctionnalités suivantes :

- Support de plusieurs technologies de niveau 2,
- Prise en compte des technologies ATM et Ethernet, qui sont souvent déployées dans la plupart des réseaux de collecte et de coeur des opérateurs de Télécom,
- Sensibilité aux changements de topologie,
- Déploiement facile sur des réseaux à grande échelle.

Cette section étudie en détail le protocole STAMP, sa structure, le format de paquets et de trames qu'il véhicule ainsi que sa logique de fonctionnement. Une récapitulation des performances et des résultats d'évaluation du protocole sera présentée à la fin du chapitre.

3.3.1 Présentation Générale

STAMP est un protocole de signalisation qui automatise la diffusion des informations de topologie et de QoS dans les réseaux Gigabit Ethernet et ATM. Ce choix cible deux technologies de liaison souvent déployées dans les réseaux des opérateurs de Télécom, plus spécifiquement dans les réseaux de collecte et de coeur.

L'architecture générale de STAMP se base sur une modélisation agents. Trois agents sont conçus pour l'émission et la réception des informations de gestion [Fig.3.4]. Deux agents agissent au niveau de la couche liaison du modèle OSI, à savoir les agents Ethernet et ATM. Le rôle de ces agents se restreint à l'acquisition et la diffusion des informations d'adjacence. En ce qui concerne les fonctions d'authentification et de transfert de données de bout en bout, elles sont accomplies par un troisième agent TCP.

Ainsi, à chaque port physique est associé au moins un agent STAMP. Cet agent peut être de type Ethernet ou ATM selon le type de la technologie déployée au niveau liaison. Un agent TCP peut aussi être associé à l'agent de type liaison si le port en question supporte la pile TCP/IP. Par conséquent, deux agents au maximum peuvent être attribués à un port physique donné.

Une telle architecture agents permet aux équipements STAMP d'interagir facilement avec tous les autres équipements qui implémentent ce protocole. En effet, chaque port physique est capable de transmettre des informations sur ses caractéristiques techniques ainsi que sur ses capacités en termes de QoS. Il peut aussi recevoir des informations du voisinage à partir des ports physiques adjacents auxquels il est connecté. Les informations échangées ainsi que les entités qui les manipulent feront l'objet de la section suivante.

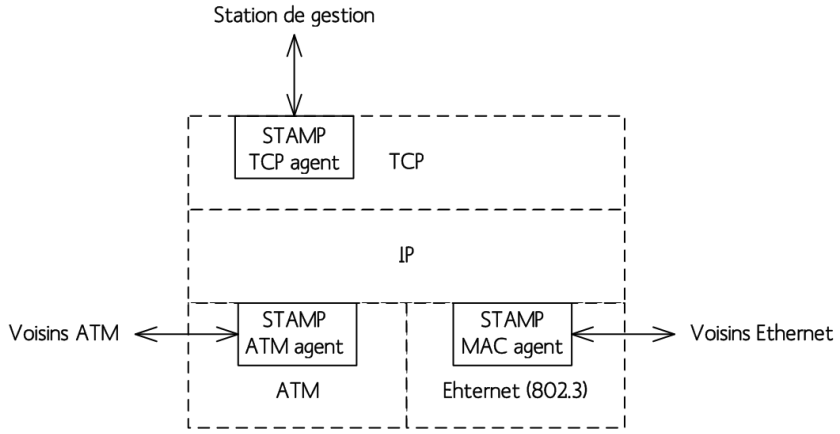

FIG. 3.4: STAMP : Le modèle agents

3.3.2 Structure du Protocole

Contrairement à la plupart des solutions de découverte de topologie qui se basent sur des techniques de probing, STAMP est implémenté sur la base d'un système d'annonces. Tous les GMs dans le groupe échangent des trames hello (au niveau liaison) afin d'annoncer leur présence dans le voisinage direct. L'ensemble des informations collectées sont par la suite transmises au GPL au dessus de la couche TCP. Que la transmission soit accomplie au niveau de la couche liaison ou transport (TCP), tous les messages STAMP ont un entête commun structuré de la façon suivante [Fig.3.5] :

- *v* identifie la version du protocole. Nos contributions ont donné lieu à la version 0x1,
- *M-Type* identifie les dix messages définis dans le cadre de la première version de STAMP : "Leader-Advertise", "Member-Connect", "Accept-Connection", "Ack-Accept-Connection", "Hello-Neighbors", "Leaders-Synchronize", "Ack-Leaders-Synchronize", "Topology-Information", "Leave-Group" et "Failure Notify",
- *Seq* définit le numéro de séquence des messages transmis. Ce champ permet de garantir que les messages reçus sont traités dans l'ordre,
- *Autority-ID* représente l'identité de l'entité administrative à laquelle un groupe STAMP appartient. Nous proposons que l'ensemble des Authority ID soit géré par une organisation centrale comme l'IANA (Internet Assigned Numbers Authority) qui aura la tâche d'attribuer des identifiants uniques ceci afin de pouvoir donner la possibilité de gérer l'acquisition de topologie à l'inter-domaine,
- *Group-ID* identifie le groupe au sein duquel les messages STAMP sont échangés. C'est à l'administrateur réseau d'attribuer des identifiants uniques à ces groupes,
- *Hierarchical Level ID* définit le niveau hiérarchique dans lequel les messages STAMP sont échangés. Dans notre contribution, nous nous sommes contentés d'étudier et développer la recherche de topologie au niveau 0, et d'agréger la topologie de niveau 0 en une topologie de niveau 1. Nous n'étudions pas dans ce mémoire l'échange de messages de topologie aux niveaux agrégés,
- *Length* précise la taille en octets du message STAMP.

C'est au dessus de cet entête que les messages STAMP vont être transmis. Toutes les informations échangées sont injectées dans des unités de données (STAMP Data Unit) sous forme de

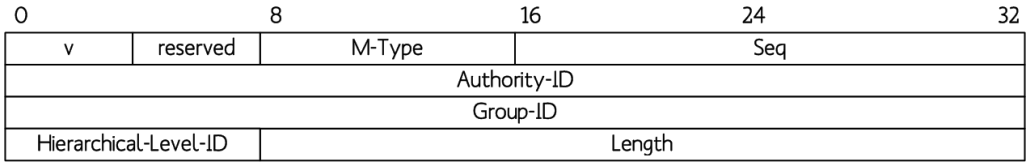

FIG. 3.5: L'entête STAMP

TLVs (Type, Length, Value). Dans la première version, nous avons défini quinze TLVs.

Device-id TLV

Ce TLV identifie un GM émetteur. En effet, il transmet l'identité du GM à l'entité réceptrice. Comme cette identité doit être unique, nous avons utilisé les adresses MAC pour désigner l'ensemble des GMs dans le groupe. Ainsi, nous supposons dans nos spécifications que tous les équipements réseau ont aux moins une adresse de management qui supporte la pile TCP/IP. Ceci permet d'un coté l'identification MAC des GMs et d'un autre coté d'assurer l'échange de données de topologies vers le GPL. Dans le cas des équipements ATM ne possédant pas d'interface Ethernet, nous utiliserons l'adresse ESI du port ATM dévolu à la gestion. Il est à noter que si un GM possède plus d'une interface de gestion, nous choisissons celle ayant l'adresse physique (MAC ou ESI) la plus petite. La structure détaillée de ce TLV est décrite par la Figure 3.6. Il s'étend sur 10 Octets et est identifié par un type égal à 1.

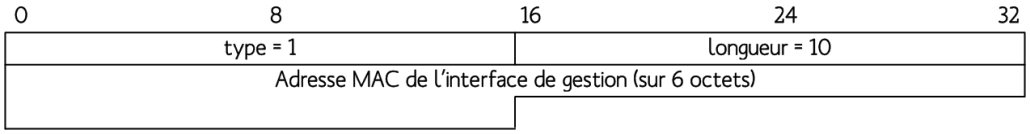

FIG. 3.6: Le TLV "Device-id"

Device-caracteristics TLV

Ce TLV permet d'identifier les caractéristiques propres au GM auprès du GPL. Nous nous sommes restreint aux caractéristiques suivantes :

- Le type de l'équipement identifie la fonction que le GM assure conformément au modèle de donnés identifié précédemment,
- La position dans la hiérarchie identifie si le message provient d'une entité physique (GM situé au niveau 0), ou d'une entité logique (niveau > 0). Les entités logiques sont une abstraction d'un ensemble de GMs et sont construites en utilisant des mécanismes d'agrégation,
- La localisation dans le groupe qui indique si le GM est à la bordure ou dans le coeur,

La structure détaillée de ce TLV est décrite par la Figure 3.7. Il s'étend sur 7 Octets et est identifié par un type égal à 2.

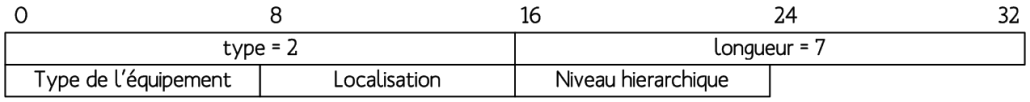

FIG. 3.7: Le TLV "Device-caracteristics"

Port-id TLV

Ce TLV permet d'identifier de façon unique un port physique donné. Conformément à la vue opérationnelle du modèle de donnée [Fig.3.2], l'identification d'un port dépend de la technologie déployée : s'il s'agit d'un port Ethernet, c'est l'adresse MAC qui est transporté dans le TLV, sinon c'est l'adresse ESI (port ATM). Aussi, l'identification du port se fait via l'index de l'interface réseau. La structure détaillée du TLV est décrite par la Figure 3.8. Il s'étend sur 12 Octets et est identifié par un type égal à 3.

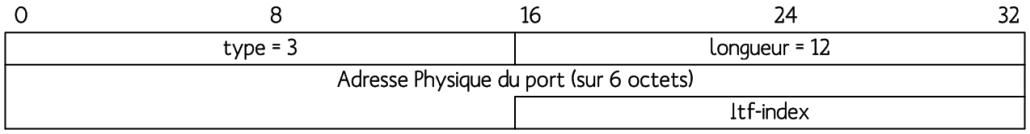

FIG. 3.8: Le TLV "Port-id"

MAC-port TLV

Ce TLV définit l'ensemble des caractéristiques techniques liées à un port Ethernet. En effet, il permet à chaque GM d'annoncer les propriétés de ses ports dans le voisinage direct. Les caractéristiques techniques que nous avons identifiées dans nos spécifications sont les suivantes : la symétrie du port, son débit, le délai de propagation enregistré pour joindre le next-hop ainsi que la gigue et le taux de perte, l'unité maximale de transfert MTU (Maximum Transfert Rate) et le coût administratif associé au port. Il est à noter que l'ensemble des paramètres de délai, gigue et taux de perte est nul dans le cas des ports qui ne supportent pas les fonctionnalités de routage. La structure détaillée du TLV est décrite par la Figure 3.9. Il s'étend sur 17 Octets et est identifié par un type égal à 4.

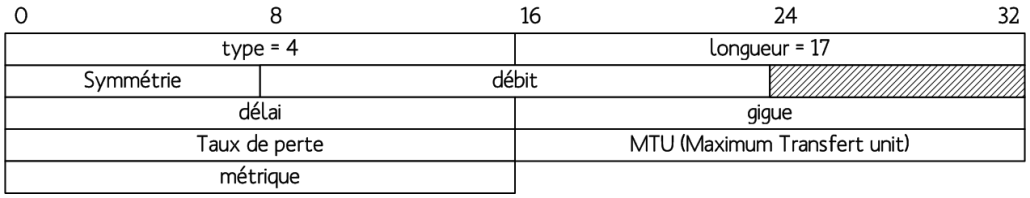

FIG. 3.9: Le TLV "MAC-port"

ATM-port TLV

C'est l'équivalent du TLV MAC-port mais dans le contexte ATM. Il permet ainsi d'identifier les caractéristiques liées à un port ATM. Dans nos spécifications, nous avons identifié les caractéris-

tiques suivantes : le débit de transfert, le délai, la gigue, le taux de perte et l'adresse de service NSAP (Network Service Access Point). La structure détaillée du TLV est décrite par la Figure 3.10. Il s'étend sur 70 Octets et est identifié par un type égal à 5.

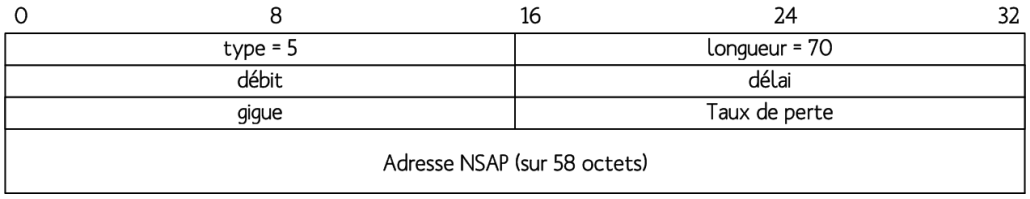

FIG. 3.10: Le TLV "ATM-Port"

QoS TLV

Ce TLV permet de spécifier les classes de services implémentées sur un port physique donné ainsi que leurs caractéristiques de QoS. Ces caractéristiques sont la bande passante, le délai, la gigue et le taux de perte. Le type de service correspond à la valeur du champ DSCP (Differenciated Services Code Point) du paquet IP. La structure détaillée du TLV est décrite par la Figure 3.11. Il s'étend sur 13 Octets et est identifié par un type égal à 6. Il est à noter que ce TLV est optionnel dans le cas où aucune classe de service n'est configurée sur le port.

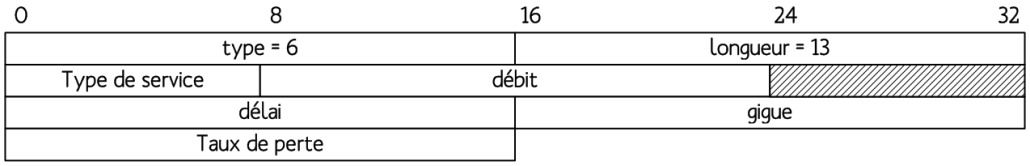

FIG. 3.11: Le TLV "QoS"

Adjacency TLV

Ce TLV fournit des informations sur les propriétés d'adjacence d'un port physique donné. Cette adjacence est identifiée au niveau liaison, c'est-à-dire que les ports voisins seront reconnus par leur adresse MAC (Ethernet) ou par leur adresse ESI (ATM). La structure détaillée du TLV est décrite par la Figure 3.12. Il s'étend sur 16 Octets et est identifié par un type égal à 7.

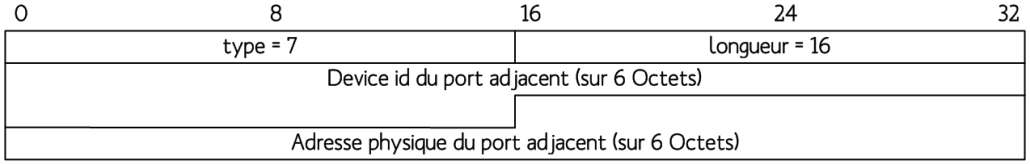

FIG. 3.12: Le TLV "ATM-Port"

Encapsulation TLV

Ce TLV décrit l'ensemble des encapsulations supportées par un port physique donné. Ces encapsulations seront décrites suivant le schéma de dépendances illustré par la figure 3.1. La logique derrière ce schéma de dépendance est d'identifier les différentes technologies montées au dessus de Ethernet et ATM.

Sur la base de ce schéma de dépendance, l'ensemble des encapsulations sera transmis dans les unités de données (STAMP-DU) comme une succession de nombres entiers compris entre 1 et 23. Ainsi, le TLV encapsulation sera composé d'un ensemble de sous-TLVs décrivant les technologies déployées au dessus de la couche MAC ou ATM. Conformément à notre modèle de donnée, cinq sous-TLVs sont définis.

– le *IPv4 sub-TLV* représente l'encapsulation de type IP version 4. Il décrit des informations IP tel que l'alias attribué à l'interface réseau, l'adresse IP, l'adresse de diffusion et le préfixe du masque réseau. Le champ "type d'encapsulation" permet d'identifier, conformément au modèle de dépendance décrit par la figure 3.1, le type d'encapsulation IPv4 (technologie au dessous de IP). Par exemple, dans le cas où IPv4 est encapsulé dans du VLAN (IPoVLAN), la valeur du paramètre "type encapsulation" sera égale à 12. La structure détaillée du sous TLV est décrite par la Figure 3.13. Il s'étend sur 33 Octets et est identifié par un type égal à 9.

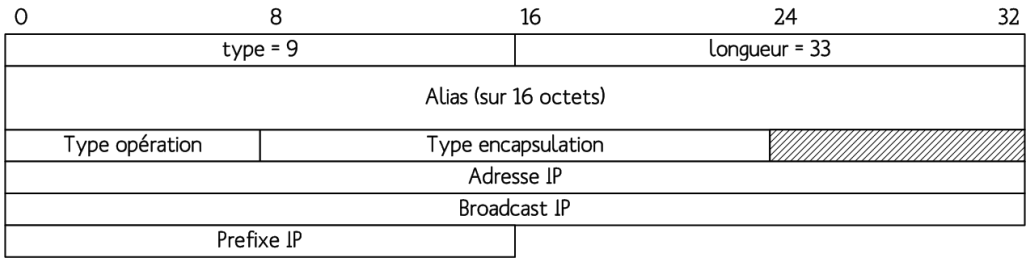

FIG. 3.13: Le sous TLV "IPv4"

– le *IPv6 sub-TLV* représente aussi une encapsulation de type IP, mais version 6. Les types d'informations transportées par ce sous-TLV sont presque les mêmes que dans le cas du sous TLV de type IPv4 à l'exception du champ "broadcast IP". En effet, le broadcast n'est pas défini dans le mode de fonctionnement de IP version 6. La structure détaillée du sous TLV est décrite par la Figure 3.14. Il s'étend sur 41 Octets et est identifié par un type égal à 10.

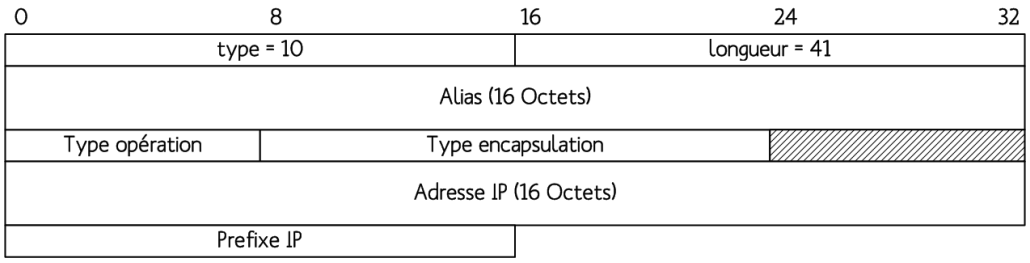

FIG. 3.14: Le sous TLV "IPv6"

– le *VLAN sub-TLV* identifie une encapsulation de type VLAN. Ceci est accompli par le biais des informations tel que l'alias attribué à cette encapsulation, l'identifiant du VLAN ainsi que le type d'encapsulation. Les encapsulations supportées par la technologie 802.1q sont identifiées par les valeurs 14, 15 et 16 [Fig.3.1]. La structure détaillée du sous TLV est décrite par la Figure 3.15. Il s'étend sur 25 Octets et est identifié par un type égal à 11.

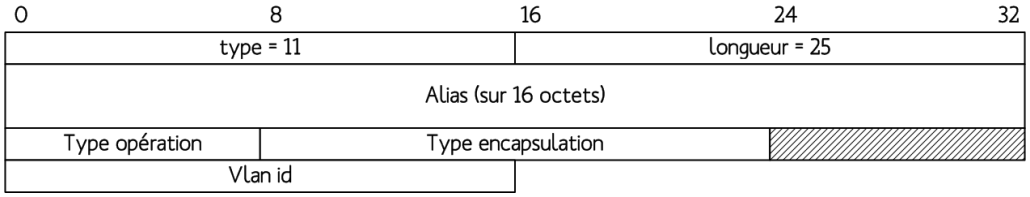

FIG. 3.15: Le sous TLV "VLAN"

– le *MPLS sub-TLV* décrit une encapsulation de type MPLS, c'est-à-dire les propriétés de configuration d'un LSP sur une interface. En effet, au niveau d'une interface, un LSP est identifié essentiellement par un Label-id. Ce label est injecté dans le champ "mpls label". Similairement aux sous-TLVs cités précédemment, le MPLS sous-TLV est décrit par un alias codé sur size octets et un "type encapsulation". La structure détaillée du sous TLV est décrite par la Figure 3.16. Il s'étend sur 25 Octets et est identifié par un type égal à 12.

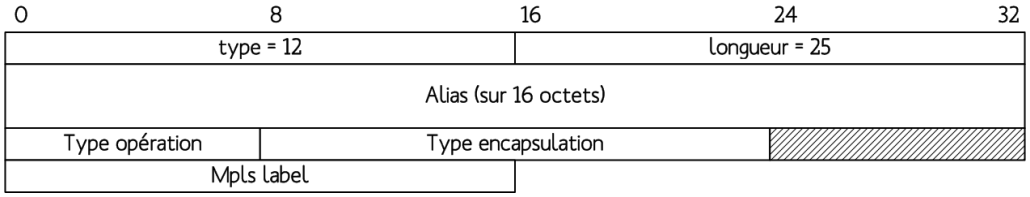

FIG. 3.16: Le sous TLV "MPLS"

– le *VPVC sub-TLV*, dont le nom fais signe au monde ATM, permet de déterminer un chemin logique de bout en bout identifié par un VP-id (Virtual Path) et un VC-id (Virtual Circuit). Il comporte les champs suivants : "alias", "type d'encapsulation", "VP-id" (identifiant du Virtual-Path) et "VC-id" (identifiant du Virtual-Circuit). Il est à noter que dans le cas où le "VC-id" est égal à 0, l'encapsulation est de type VP (l'acheminement de donnée se fait suivant un VP). Dans le cas contraire l'encapsulation est de type VP/VC. La structure détaillée du sous TLV est décrite par la Figure 3.17. Il s'étend sur 27 Octets et est identifié par un type égal à 13.

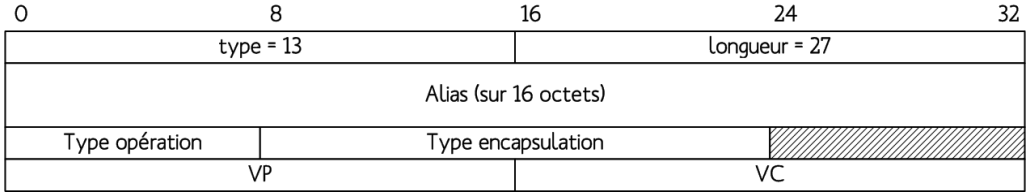

FIG. 3.17: Le sous TLV "VPVC"

Tous ces sous TLVs transportent le champ "type opération". C'est sur la base de la valeur de ce champ que vont dépendre les traitements coté GPL. En effet, il permet de notifier s'il s'agit d'une opération d'ajout, de modification ou de suppression d'une encapsulation dans la *tdb*. En outre, un port physique peut être configuré avec plusieurs technologies de niveau 2 et 3, et par conséquent il peut transmettre plusieurs sous TLVs d'encapsulation. Par exemple, si un port physique possède deux adresses IPv4, deux sous-TLVs IPv4 seront transmis dans le TLV encapsulation.

La grammaire ci-dessous présente l'organisation de l'ensemble de tous les sous-TLVs dans le TLV encapsulation principal.

<Encapsulation TLV> :=
<Encapsulation-Occurence>*
<Encapsulation TLV>*

<Encapsulation-Occurence> :=
<IPv4 sub-TLV>*
<IPv6 sub-TLV>*
<VLAN sub-TLV>*
<MPLS sub-TLV>*
<VPVC sub-TLV>*

Il est à noter que le TLV encapsulation est identifié par un type égal à 8 et une taille variable. En effet, la longueur de ce TLV dépend directement du nombre de sous-TLVs transportés.

Loopback-address4 TLV

Ce TLV est nécessaire pour les équipements de type routeur. En effet, la définition de l'adresse de loopback est très importante pour les tâches d'administration, ainsi que pour quelques fonctions de gestion tel que la configuration du routage. Il transporte les informations concernant les interfaces logiques de loopback de type IPv4 configurées sur les routeurs. La structure détaillée de ce TLV est décrite par la Figure 3.18. Il s'étend sur 10 Octets et est identifié par un type égal à 14.

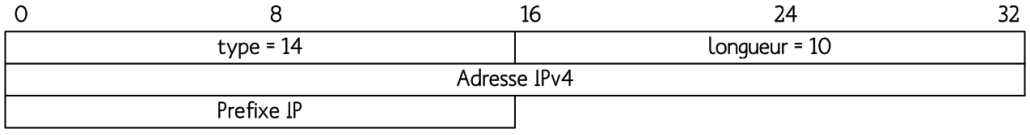

FIG. 3.18: Le TLV "Loopback-address4"

Loopback-address6 TLV

Ce TLV assure la même fonctionnalité que celle définie par le TLV Loopback-address4, mais dans le cas d'une adresse de loopback de type IPv6. La structure détaillée du TLV est décrite par la Figure 3.19. Il s'étend sur 22 Octets et est identifié par un type égal à 15.

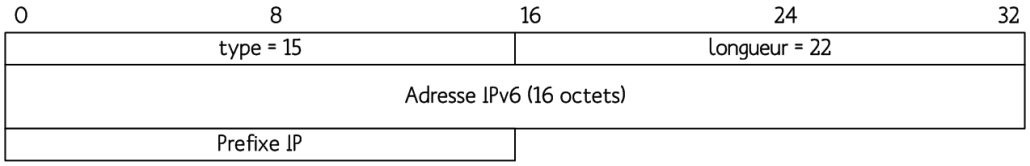

FIG. 3.19: Le TLV "Loopback-address6"

Forwarding-v4 TLV

Ce TLV est construit à partir des données de routage (tables de routage ou MIBs). Il identifie vers quel adresse IP une interface achemine son trafic. En effet, une connaissance des informations de routage est primordiale pour la fonction de contrôle d'admission afin d'identifier le chemin par lequel passe un trafic donnée et par la suite de décider de son acceptation ou pas. La structure détaillée du TLV est décrite par la Figure 3.20. Il s'étend sur 16 Octets et est identifié par un type égal à 16. Il est à noter que le champ "outgoing Label" est utile dans le cas où le prochain saut n'est pas une adresse IP mais plutôt un identifiant d'un tunnel MPLS, sinon il est toujours égal à 0.

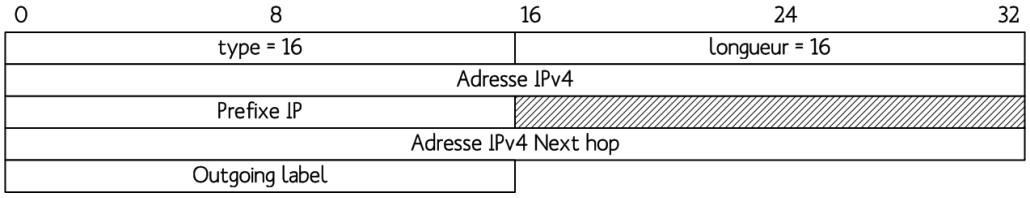

FIG. 3.20: Le TLV "Forwarding-v4"

Forwarding-v6 TLV

Ce TLV assure la même fonctionnalité que le TLV "Forwarding-v4 TLV", mais dans le cas du routage IPv6. La structure détaillée du TLV est décrite par la Figure 3.21. Il s'étend sur 40 Octets et est identifié par un type égal à 17.

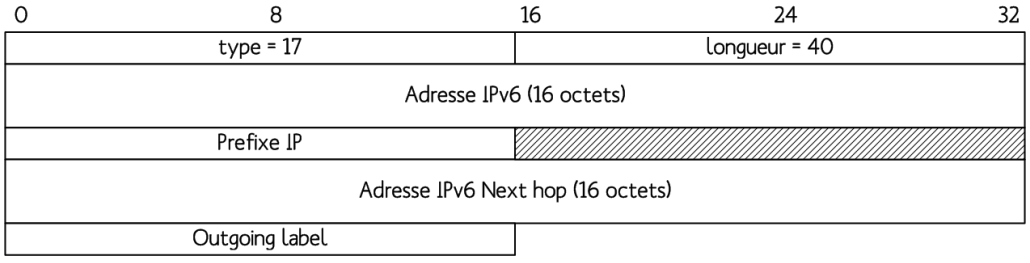

FIG. 3.21: Le TLV "Forwarding-v6"

Forwarding-ATM TLV

Ce TLV est basé sur les informations issues de la table de commutation. Il permet d'identifer le schéma des commutations. Deux types de commutation sont possibles : une au niveau VP/VC et une autre au niveau VP. Dans le deuxième cas, le TLV est envoyé avec les champs VC_{in} et VC_{out} égals à 0. Il est à noter que ce TLV s'étend sur 12 Octets et est identifié par un type égal à 18 [Fig.3.22]. Le champ "Ingoing itf-index" indique le port physique par lequel le trafic arrive.

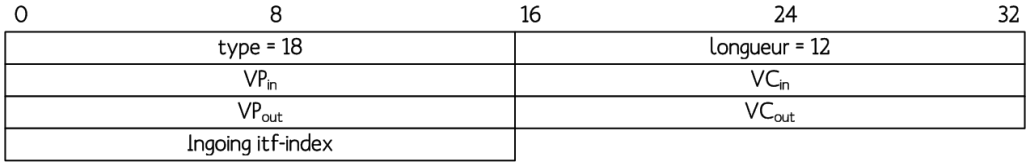

FIG. 3.22: Le TLV "Forwarding-ATM"

Forwarding-MPLS TLV

Ce TLV décrit les informations de la LIB (Label Information Base). Il associe pour chaque label d'entrée un label de sortie. Ceci permettra facilement au GPL d'identifier l'ensemble des LSPs configurés dans son groupe. Ce TLV s'étend sur 8 Octets et est identifié par un type égal à 19 [Fig.3.23]. Le champ "Ingoing itf-index" indique le port physique par lequel le trafic arrive.

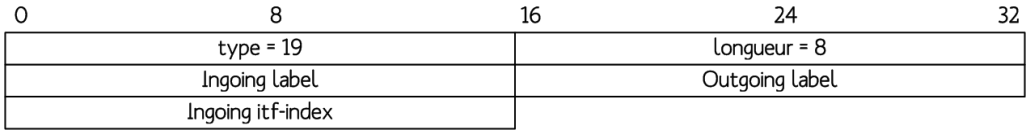

FIG. 3.23: Le TLV "Forwarding-MPLS"

Il est à noter que tous les TLVs forwarding cités ci-dessus (Forwarding-v4, Forwarding-v6, Forwarding-ATM, Forwarding-MPLS) identifient le trafic sortant des ports.

Authentication TLV

Ce TLV est optionnel. Il permet à l'administrateur réseau de choisir entre le mode sécurisé (safe) et le mode non sécurisé (unsafe). En mode sécurisé, ce TLV sert à authentifier le GM auprès du GPL. Dans nos spécifications, nous nous sommes basé sur la technologie TLS (Transport Layer Security) [88, 89, 90, 91]. En effet, nous utilisons le certificat numérique X.509 pour l'authentification et pour le cryptage. Afin d'implémenter un modèle sécurisé souple, nous laissons à l'administrateur la liberté de définir ses exigences en termes de sécurité (longueur de la clé de cryptage, algorithme de cryptage, etc.) selon la politique mise en place au sein du groupe STAMP. Ce TLV est identifié par un type égal à 20.

3.3.3 Les Messages STAMP

Nous consacrons cette section à présenter l'ensemble des messages échangés entre les différentes entités STAMP. Chaque message sera présenté par une règle grammaticale explicitant l'ensemble des TLVs qu'il comporte dans son unité de données (STAMPDU). Étant donné un TLV

de type $\mathcal{T}$, sa cardinalité est déterminée par les signes suivants : (+) indique qu'il s'agit d'au moins un TLV $\mathcal{T}$, (*) indique qu'il peut y avoir zéro ou plusieurs TLVs $\mathcal{T}$ et (?) indique que le TLV $\mathcal{T}$ est optionnel. Sinon, le TLV est obligatoire et doit être présent une seule fois.

Leader-Advertise

Ce message permet au GPL de diffuser son identifiant ainsi que l'identifiant du GBL dans le groupe. Ce type d'informations est utile pour les GMs qui ont besoin de l'identifiant du GPL afin de pouvoir établir une connexion avec lui. Si le multicast est supporté, les identifiants sont diffusés dans le groupe. Sinon, les GMs sont configurés manuellement avec les informations d'interconnexion au GPL. Ainsi, chaque GM est capable d'identifier son GPL et de lui envoyer par la suite ses caractéristiques et les informations d'adjacence qu'il collecte. La grammaire du message est la suivante :

<Leader-Advertise>:=
<Device-id TLV> (GPL)
<@IP GPL>
<Device-id TLV> (GBL)
<@IP GBL>

Member-Connect

Une fois que le GM connaît les informations d'identification de son GPL, il essaye d'établir une connexion TCP avec lui. Cette connexion peut être de type sécurisé (safe) ou non (unsafe) selon les politiques et les stratégies définies dans le groupe. Dans le cas d'une connexion sécurisée, les certificats d'authentification et de cryptage du GM sont envoyés dans le TLV "Authentication TLV". Sinon, seules les informations d'identification du GM sont envoyées au GPL. La grammaire du message est la suivante :

<Member-Connect> :=
<Device-id TLV> (GM authentifié)
<Authentication TLV> ?

Accept-Connection

A la réception du message "Member-Connect", le GPL vérifie les informations d'authentification du GM en question. Si ces informations sont valides, il envoie au GM authentifié un message de confirmation et le rajoute par la suite dans sa base de topologie *tdb*, sinon la demande de connexion est rejetée. Dans le cas d'une communication non sécurisée (mode *unsafe*), le GPL rajoute directement le GM dans sa *tdb* et lui envoie un message "Accept-Connection". Il est à noter que lors de cette phase l'état du GM reste toujours inactif dans la *tdb*. La grammaire du message est la suivante :

<Accept-Connection> :=
<Device-id TLV> (GPL)
<Authentication TLV> ?

Ack-Accept-Connection

Ce message acquitte la réception du message "Accept-Connection". C'est ce message qui change l'état du GM dans la *tdb* en mode actif. En effet, le mode actif permet au GM authentifié

d'entamer la procédure de découverte de topologie de son voisinage. La grammaire du message est la suivante :

<Ack-Accept-Connection> :=
<Device-id TLV> (GM authentifié)

Hello-Neighbors

C'est le seul message échangé au niveau liaison (couche 2 du modèle OSI). Il permet aux GMs d'annoncer leur présence et de lire les informations d'adjacence de leurs voisins directs. Les informations transmises dans ce niveau concernent l'identification des ports transmetteurs ainsi que l'identification des GMs. En effet, chaque port physique est susceptible d'envoyer et de recevoir des messages hello périodiques afin d'annoncer sa présence et de connaître la topologie de son voisinage direct. Le système d'annonce dépend fortement de la technologie déployée. En ce qui concerne la technologie 802.3, nous nous sommes basés dans les premières spécifications du protocole STAMP sur l'adresse de diffusion FF :FF :FF :FF :FF :FF. L'idée est de permettre à chaque GM de propager dans le voisinage direct les informations de topologie de l'ensemble de ses ports physiques. Le problème enregistré avec une telle logique découle directement des propriétés intrinsèques de cette adresse de diffusion. En effet, une telle adresse n'est interprétée que par des stations terminales, ce qui rend impossible l'identification des caractéristiques de voisinage des autres équipements dans le réseau tels que les commutateurs Ethernet par exemple. Pour résoudre ce problème, nous avons eu recours aux adresses de multicast Ethernet. En effet, un intervalle d'adresses (01 :80 :C2 :00 :00 :00 - 01 :80 :C2 :00 :00 :0F) a été définit pour déterminer les adresses de multicast dans Ethernet. Dans nos spécifications, nous avons utilisé l'adresse, réservée pour usage future, 01 :80 :C2 :00 :00 :0C. D'une part, cette adresse sera utile pour identifier les échanges de niveau 802.3 du protocole STAMP, et d'autre part elle permettra au différents GMs (stations terminales ou pas) d'annoncer leurs caractéristiques techniques dans le voisinage. Ainsi, l'unité de donnée (STAMP DU) sera transmise au dessus de la couche Ethernet avec une adresse de destination égale à l'adresse de multicast (01 :80 :C2 :00 :00 :0C). Contrairement à Ethernet, ATM n'implémente pas des mécanismes de multicast. Du coup, il a fallu trouver un autre mécanisme afin d'atteindre les voisins directs. La notion des VPs/VCs standards permet d'établir des chemins logiques de bout en bout transparents à l'ensemble des commutateurs traversés, ce qui ne répond pas à notre problématique. Par conséquent, afin d'atteindre les voisins directs, nous avons opté pour l'utilisation des VPs/VCs réservés. Ainsi, les messages "Hello-Neighbors" sont envoyés au dessus de la couche ALL5 (ATM Adaptation Layer) sur le VP/VC 0/30 (réservé lui aussi pour de futur usage).

Une fois que le message "Hello-Neighbors" est reçu, les informations qu'il transporte sont rajoutées dans la table d'adjacence *adb*. Les ports sont identifiés au sein de l'*adb* par leur adresse MAC ou ESI (End System Identifier) suivant la technologie déployée (Ethernet ou ATM). La grammaire du message est la suivante :

<Hello-Neighbors> :=
<Device-id TLV> (GM)
<Port-id TLV>

Topology-Information

Échangé sur une session TCP, le message "Topology-Information" permet aux GMs d'annoncer leurs données de topologie collectées au GPL. Ces données de topologie sont relatives à

un port donné (<Port-id TLV>). Elles peuvent décrire les caractéristiques techniques et d'adjacence d'un port physique, comme elles peuvent décrire les propriétés techniques d'une interface logique, tel que la loopback. Ce message permet aussi de notifier tous les changements de configuration qui surviennent au niveau des GMs (au niveau de l'*adb* ou de la *cdb*). La grammaire du message est la suivante :

<Topology-Information> :=
<Device-id TLV> (GM)
<Device-caraterisctics TLV >
<Port-id TLV>
<MAC-Port TLV> ? || <ATM-Port TLV> ?
<QoS TLV>*
<Forwarding-v6 TLV>*
<Forwarding-v4 TLV>*
<Forwarding-ATM TLV>*
<Forwarding-MPLS TLV>*
<Adjacency TLV>+
<Encapsulation TLV>*

Il est à noter que, s'il s'agit de la remontée d'information d'une interface de loopback, le TLV <Port-id TLV> est envoyé avec une adresse de port physique égale à 0 (les TLVs <MAC-Port TLV> et <ATM-Port TLV> ne sont pas transmis).

Failure-Notify

Comme les messages "Hello-Neighbors" sont envoyés périodiquement entre GMs adjacents, la non réception de ce message après l'écoulement d'un timer t_{at} autorise un GM à annoncer au GPL la perte de connexion avec son voisin direct. Cette notification est remontée via le message "Failure-Notify". La grammaire du message est la suivante :

<Failure-Notify> :=
<Device-id TLV> (GM émetteur)
<Device-idTLV> (GM concerné par la défaillance)
<Port-id TLV> (Port défaillant)

Leave-Group

Ce message est envoyé par un GM qui sollicite la déconnexion du groupe. Le GM est déconnecté automatiquement après l'envoie de ce message sans attendre une confirmation de la part du GPL. La grammaire du message est la suivante :

<Leave-Group> :=
<Device-id TLV> (GM déconnecté)

Leaders-Synchronize

Ce message est envoyé périodiquement entre le GPL et le GBL afin d'assurer une surveillance mutuelle entre ces deux entités. Il joue le rôle des messages "Heart-Beat" (battements de coeurs). De même, le GBL contrôle le fonctionnement du message GPL via le message d'acquittement Ack-Leaders-Synchronize. La grammaire du message est la suivante :

<Leaders-Synchronize> :=
<Device-id TLV> (GPL)

Ack-Leaders-Synchronize

Ce message acquitte la réception du message "Leaders-Synchronize". Il est envoyé par le GBL et permet au GPL de surveiller le fonctionnement de son backup. La grammaire du message est la suivante :

<Ack-Leaders-Synchronize> :=
<Device-id TLV> (GBL)

Il est à noter que la duplication des données de topologie au niveau du GBL peut s'effectuer soit au niveau du GPL soit au niveau du GM. Dans le premier cas, le GPL sert de relais pour toutes les informations issues des GMs et les retransmets directement au GBL. Dans le deuxième cas, les GMs envoient simultanément leur messages de topologie et de contrôle au GPL et au GBL. Aussi, il est possible d'effectuer la duplication de données au niveau GBL en utilisant un DBMS[21] (DataBase Management System) sophistiqué tel que celui d'Oracle.

3.3.4 Séquencement des Messages STAMP

Si nous abstrayons le fonctionnement du protocole STAMP, nous pouvons identifier trois phases principales dans le processus d'annonce de la topologie [Fig.3.24] : une phase d'initialisation, une phase de découverte du voisinage et une phase d'interaction avec le GPL.

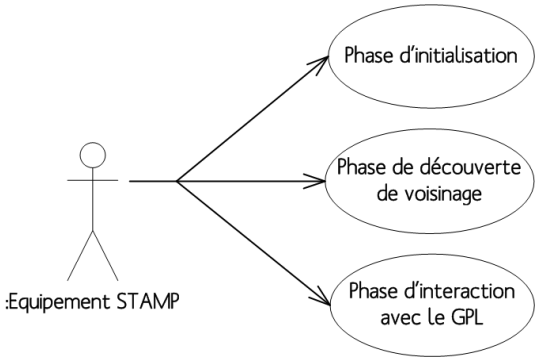

FIG. 3.24: STAMP : Le cas d'utilisation

Lors de la première phase [Fig.3.25], les GMs tentent de se connecter au GPL. En effet, ils envoient des messages "Member-Connect" pour joindre le groupe. Dans le cas du mode sécurisé, le GPL se base sur les données d'authentification (Authentication TLV) afin de décider d'accepter ou non une nouvelle demande de connexion. Dans le cas contraire, la connexion est directement acceptée, le GM authentifié est rajouté dans la base de topologie *tdb* et un message "Accept-Connection" est envoyé à ce GM. Comme mentionné précédemment, un GM n'est mis à l'état actif dans la *tdb* qu'après la réception du message "Ack-Accept-Connection".

[21] Connu dans la littérature française sous le nom de système de gestion de base de données (SGBD), le DBMS est un ensemble de programmes qui permet la gestion et l'accès à une base de données.

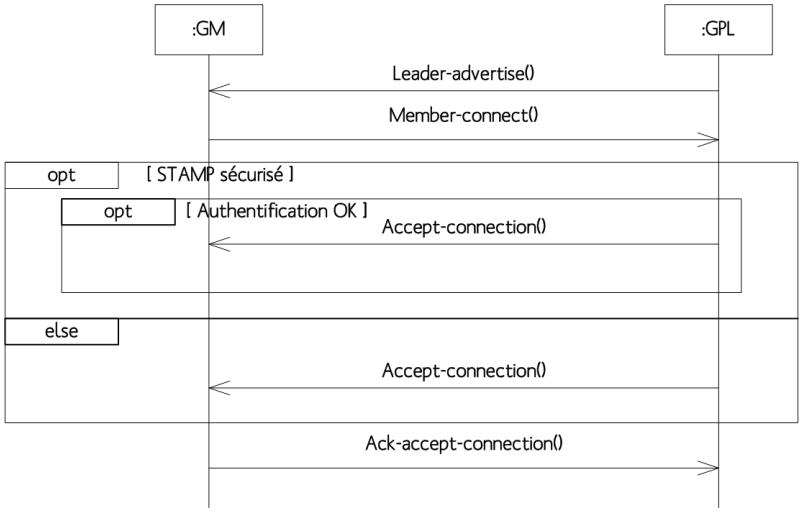

FIG. 3.25: Phase d'établissement de connexion

La deuxième phase constitue l'originalité de STAMP [Fig.3.26]. Malgré sa simplicité, l'utilisation des messages hello permet une interrogation complète et précise du voisinage direct. En effet, quelque soit la technologie déployée (Ethernet ou ATM), ce type de message permet d'annoncer les caractéristiques techniques d'un port physique et de découvrir les voisins immédiats. Chaque GM échange périodiquement (période t_{rt}) avec ses voisins directs des messages "Hello-Neighbors". L'ensemble de ces messages reçus permet ainsi à chaque GM de construire sa table d'adjacence *adb*. A l'expiration du timer t_{at}, si un GM ne reçoit pas de message "Hello-Neighbors" d'un de ses voisins enregistrés dans l'*adb*, il envoi un message "Failure-Notify" au GPL afin de signaler la perte de connectivité avec le voisin en question. Il est à noter que le timer t_{at} est égal à trois fois le timer t_{rt}. Ceci permet d'éviter de signaler de fausses alertes au GPL.

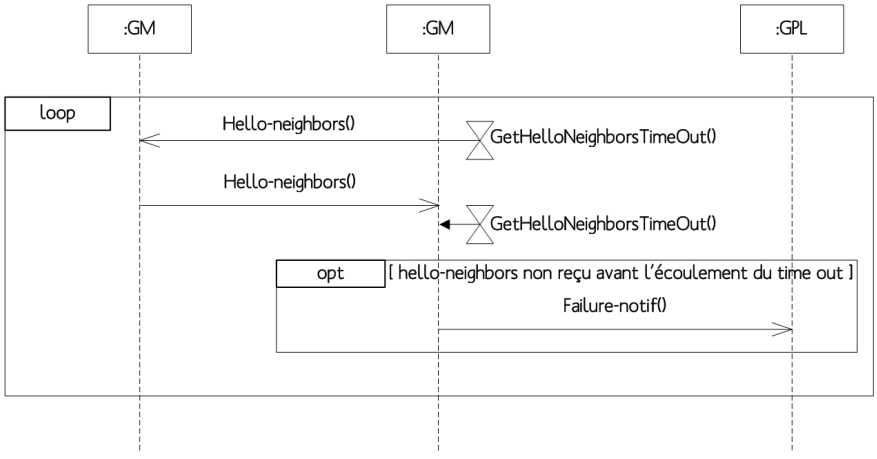

FIG. 3.26: Phase de découverte de voisinage

La dernière phase concerne l'envoie des données de voisinage collectées au GPL [Fig.3.27]. C'est ce dernier qui assure toutes les fonctions de corrélation de données, de construction de topologie et de contrôle d'admission. Ainsi, à la réception d'un message "Topology-Information",

le GPL vérifie s'il s'agit d'un nouveau message de topologie ou juste d'une mise à jour. Pour cela, il vérifie si l'identifiant du port reçu dans le Port-id TLV figure dans sa *tdb* ou pas. Si oui, le message "Topology-Information" est parcouru afin de mettre à jour la base de topologie. Sinon une nouvelle entrée, correspondant au port en question, est créée. Aussi, lors de cette phase, le GM est capable d'envoyer le message de contrôle "Leave-Group" afin de notifier au GPL sa déconnexion du groupe.

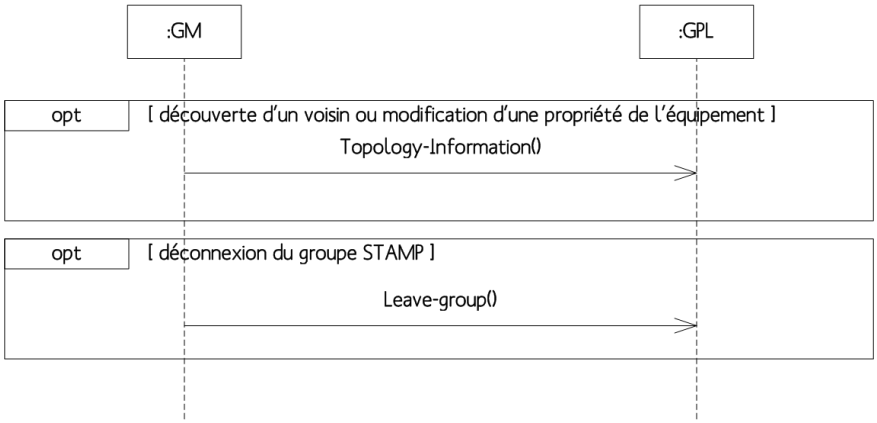

FIG. 3.27: Phase de remonté des informations collectées au GPL

3.3.5 Diagrammes d'États

L'étude approfondie de la logique de fonctionnement du protocole STAMP passe par l'explicitation de toutes les opérations d'envoie et de réception accomplies par les GMs et le GPL. Pour cela, nous proposons les quatre diagrammes d'états suivants :

– Une machine d'état pour la réception des messages coté GPL [Fig.3.28] : ce dernier se met en attente de messages de la part du GBL ainsi que des GMs. Suivant le type de message, un traitement est attribué. La réception d'un message "Member-Connect" permet de rajouter le nouveau GM émetteur dans la base de topologie (*tdb*). En mode sécurisé, le GM n'est rajouté dans la *tdb* que si son certificat est valide. Il ne bascule en état actif, et donc ne fait partie de la topologie effective détenue par le GPL qu'après la réception du message "Ack-Accept-Connection". Le message "Topology-Information" permet d'agir sur la *tdb* par ajout, suppression ou modification. Le message "Failure-Notify" met en quarantaine une interface signalée comme défaillante. Cette interface sera sauvegardée dans la structure *qdb* jusqu'à l'expiration d'un timer t_{qt}. Si avant l'expiration de ce timer, toutes les interfaces d'un GM sont signalisées défaillantes, le membre en question est considéré en panne et est désactivé dans la *tdb*. Sinon, seulement l'interface défaillante est mise en état inactif dans la *tdb*. Contrairement au message "Failure-Notify", la réception du message "Leave-Group" permet de désactiver immédiatement le GM en question dans la *tdb*. Le seul message reçu de la part du GBL, est le message "Ack-Leaders-Synchronize". Sa réception après l'expiration du timer t_{st} permet au GPL de détecter la défaillance du GBL,

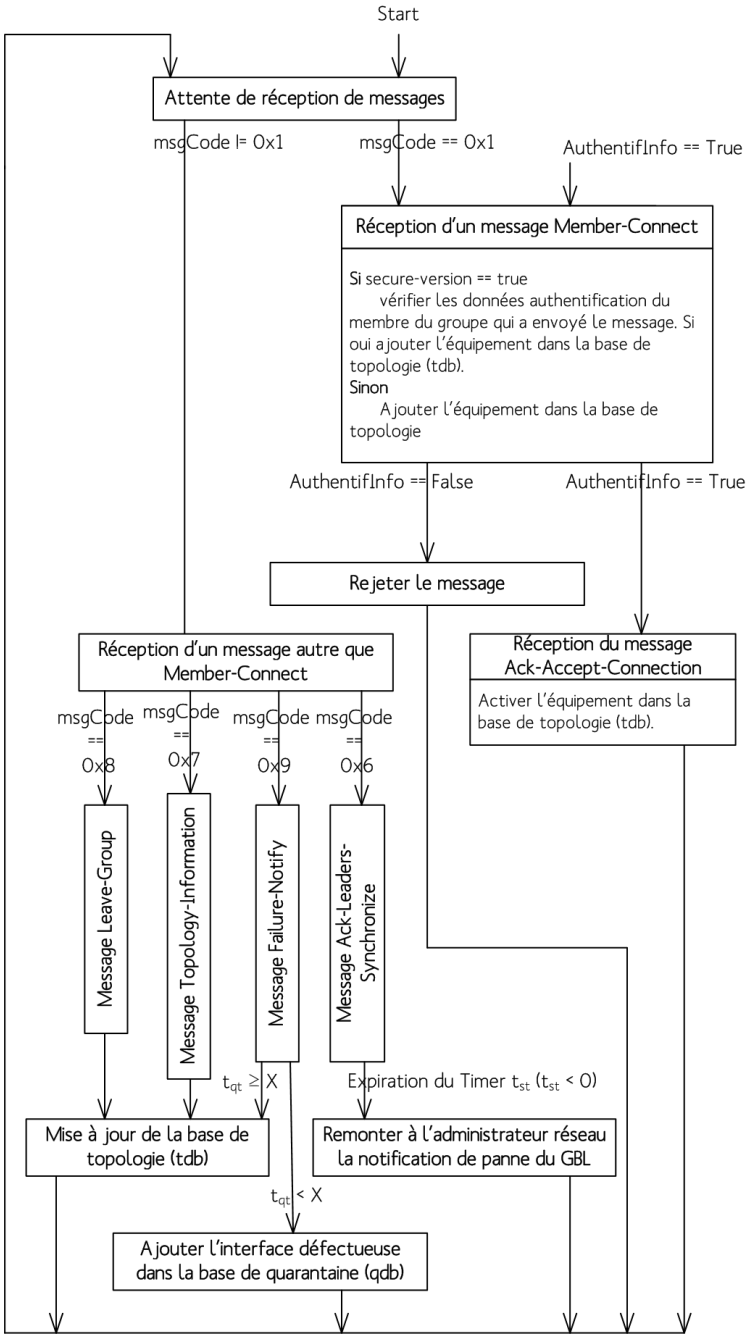

FIG. 3.28: Machine d'état pour la réception des messages coté GPL

– une machine d'état pour la réception des messages coté GM [Fig.3.29] : il s'agit de la machine d'état la plus simple. Si le multicast est configuré dans le groupe, les GMs commencent par recevoir des messages "Leaders-Advertize". Dans le cas où les GMs sont configurés manuellement, ils se mettent en attente d'un message "Accept-Connection" de la part du GPL. Une fois connecté, chaque GM se met à l'écoute des messages "Hello-Neighbors" reçus depuis son voisinage direct. Ce sont ces messages qui lui permettent de mettre à jour sa base d'adjacence (*adb*).

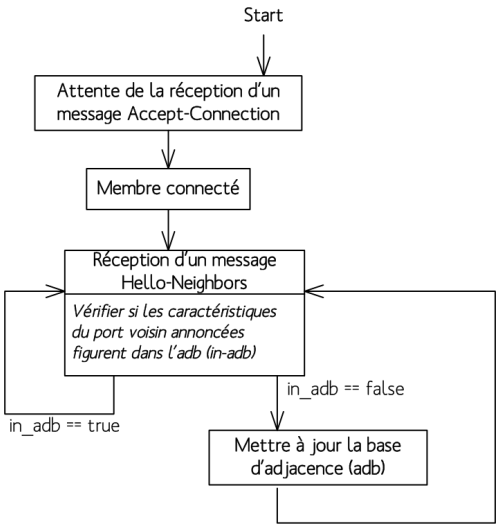

FIG. 3.29: Machine d'état pour la réception des messages coté GM

– Une machine d'état pour la transmission des messages coté GPL [Fig.3.30] : dans le cas où le multicast est configuré dans le groupe, le premier message envoyé par le GPL est le message "Leader-Advertise". Ce message permet de propager l'identité du GPL et du GBL dans le groupe. Dans le cas contraire, le GPL se met directement dans un état inactif de transmission, jusqu'à ce qu'il reçoive une demande de connexion. Si les informations d'authentification sont valides, le GPL répond par le message "Accept-Connection". L'état inactif des transmissions peut aussi être interrompu par l'envoie périodique (t_{st}) des messages de synchronisation "Leaders-Synchronize". A la non transmission de ce message (expiration de t_{st}), le GBL détecte la défaillance du GPL et prend la relève,

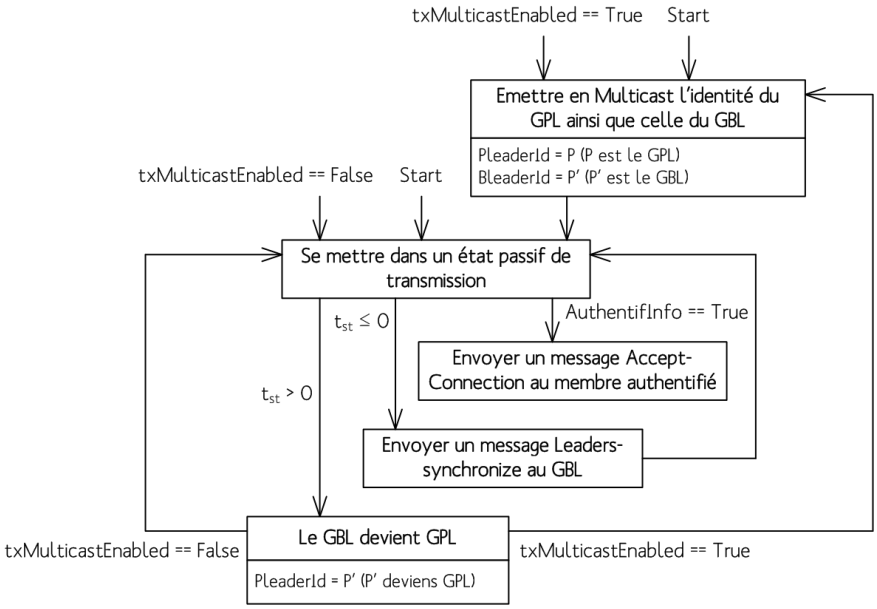

FIG. 3.30: Machine d'état pour la transmission des messages coté GPL

– Une machine d'état des transmissions coté GM [Fig.3.31] : le processus des transmissions commence par l'envoie d'un message "Member-Connect" afin de solliciter une connexion auprès du GPL. Un timer t_{xt} est déclenché. A l'expiration de ce timer, le GM re-transmet un nouveau message de connexion. Si au bout de trois tentatives de connexion le GM ne reçoit pas de notification de la part du GPL, il considère que ce dernier est hors service. Il tente alors de se connecter auprès du GBL. Une fois que la demande est approuvée par le leader principal, un message "Ack-Accept-Connection" est transmis à ce dernier afin d'activer le GM en question dans sa *tdb*. En état actif, le GM peut commencer à interagir avec ses voisins directs par l'envoie mutuel de messages "Hello-Neighbors". La non réception d'un tel message au bout d'un timer t_{at}, permet de signaler, auprès du GPL, la défaillance d'une interface adjacente par le biais du message "Failure-Notify". Toutes les données de topologie et d'adjacence sont transmises dans des messages "Topology-Information". Aussi, chaque modification au niveau de l'*adb* ou de la *cdb* est notifiée via des messages "Topology-Information". La déconnexion du groupe se fait par l'envoi d'un message "Leave-Group",

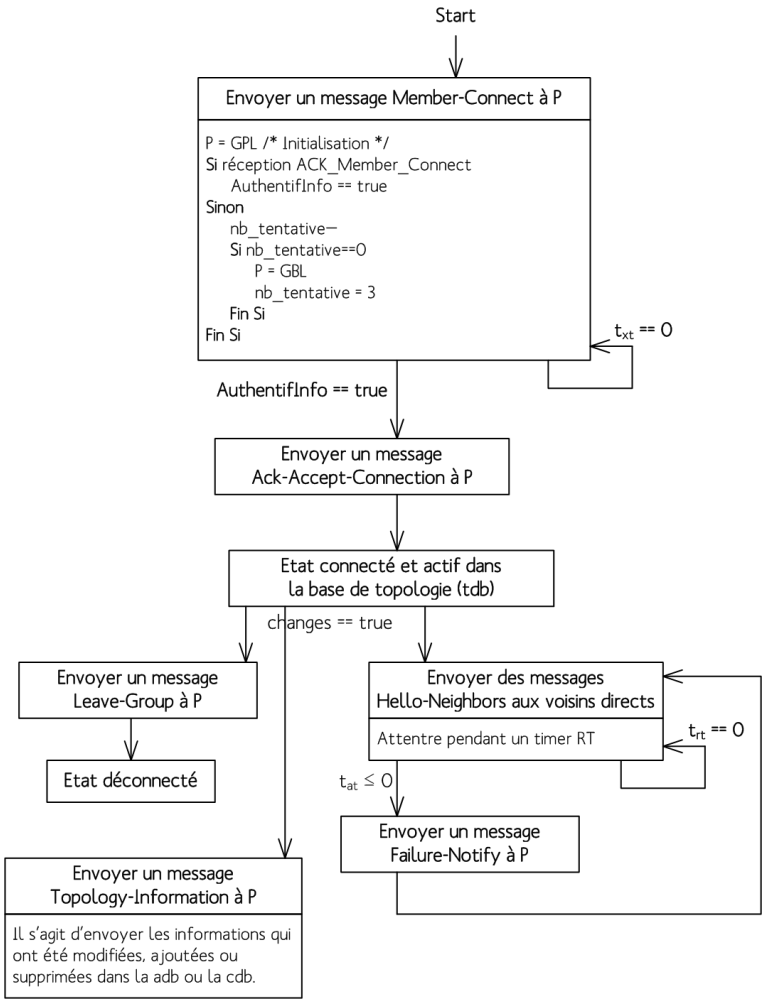

FIG. 3.31: Machine d'état pour la transmission des messages coté GM

3.3.6 Architecture Logicielle

Dans la section précédente nous nous sommes focalisés sur la logique de fonctionnement de STAMP via l'ensemble des messages échangés entre les différentes entités dans le groupe. Nous consacrons ce passage pour expliciter l'architecture de chaque entité et la façon avec laquelle elle gère ses transmissions et réceptions.

Coté GPL

Le GPL dispose de deux modules principaux, à savoir un module d'interface avec le GM/GBL et un module de traitement [Fig.3.32]. Le premier module représente le point d'entrée (resp point de sortie) pour toutes les requêtes issues (resp à destination) des autres entités. Il englobe les fonctions de connexions et les mécanismes de classification dont le but était d'établir un ordre préférentiel dans le traitement des messages reçus par le GPL. Ainsi, selon leur ordre de priorité, tous les messages sont bufférisés dans trois types de files FIFO : une FIFO HP (High Priority), une FIFO MP (Medium Priority) et une FIFO LP (Low Priority). Ces FIFOs assurent le relais entre le module d'interface et le module de traitement.

Derrière l'utilisation des FIFOs, la philosophie est de répartir les messages reçus par le GPL sur des files différentes selon leurs sensibilités au délai et aux critères d'évaluation définis dans le chapitre 1. En effet, la file la plus prioritaire, la file HP, est consacrée aux messages "Ack-Leaders-Synchronize", "Ack-Connection-Accept" et "Leave-Group-Message". Le message "Ack-Leaders-Synchronize" assure le contrôle périodique de l'état de fonctionnement du GBL. Un retard dans le traitement de ce message peut générer de fausses alertes et par la suite perturber le fonctionnement du protocole. Pour cette raison il est traité avec une priorité haute. Le deuxième message de haute priorité est le message "Ack-Connection-Accept". Il permet d'activer un GM dans la *tdb*. Un tel niveau de priorité permet de suivre en temps réel tous les changements de topologie qui peuvent survenir sur un GM ainsi que sur son voisinage. Le message "Leave-Group-Message" bénéficie aussi d'un traitement prioritaire afin de pouvoir détecter en temps réel la déconnexion des GMs du groupe. Il permet aussi d'ignorer le traitement des messages issus du même GM et ayant un numéro de séquence inférieur à celui du message "Leave-Group".

La deuxième file est la file MP. Elle concerne les messages de priorité moyenne. Ces messages sont les messages "Topology-Data" et "Failure-Notify". Comme ils sont les éléments clé du processus de construction de la topologie dans la terminologie STAMP, nous avons accordé à ces deux messages le même ordre de priorité. Vu le nombre très important de messages "Topology-Data" et "Failure-Notify" qui transite dans le réseau comparé au nombre de messages des autres files, la file MP est très souvent sollicitée pour l'approvisionnement de la *tdb*. Ce choix justifie l'ensemble des critères d'évaluation que nous avons fixés au chapitre 1, plus spécifiquement les critères de réactivité et de complétude.

Finalement, la file la moins prioritaire est la file LP. Elle identifie un seul message de faible priorité : "Member-Connect". Par ce choix, nous avons voulu intégrer un mécanisme de régulation des transmissions au sein du groupe. En effet, un GM ne peut commencer la procédure de découverte de son voisinage, et par la suite l'envoie de ses données de topologie au GPL qu'après la réception d'un message "Accept-Connection". Par conséquent, un blocage sur le message "Member-Connect" met le GM en attente d'une confirmation d'acceptation de la part du GPL et permet ainsi à ce dernier de réguler le taux de messages transmis dans le groupe et de bien gérer l'ensemble de ses files. Ce mécanisme s'avère très utile dans la phase d'initialisation du protocole où tous les GMs tentent de se connecter au GPL.

Quels que soient leurs types, toutes les FIFOs sont surveillées par un gestionnaire d'événement permettant l'exécution de traitements appropriés aux types de messages reçus. Ce même gestionnaire est responsable de l'alimentation de la base de topologie *tdb* et de quarantaine.

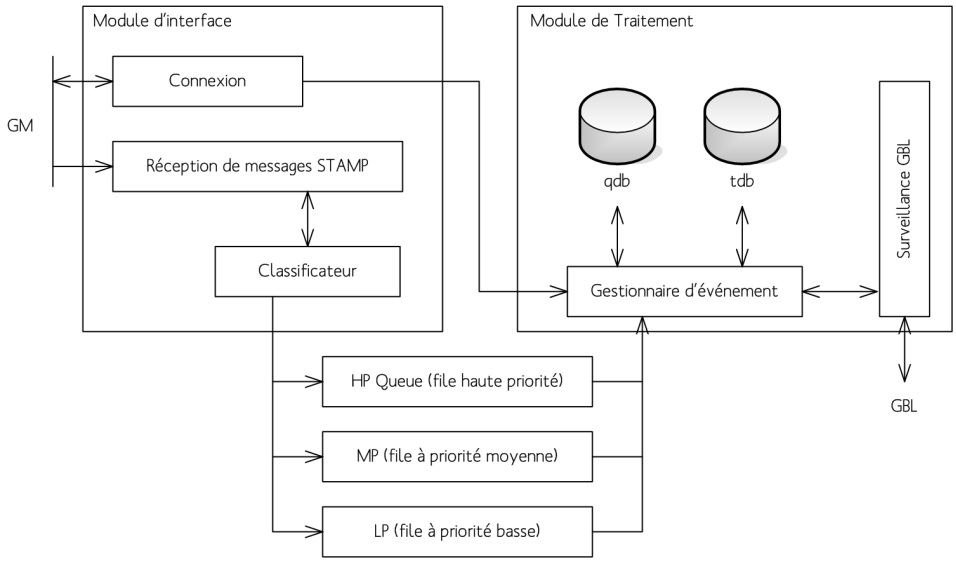

FIG. 3.32: Architecture du GPL

Coté GM

Comparé au GPL, les traitements au niveau du GM sont beaucoup plus simples [Fig.3.33]. En effet, nous distinguons trois modules principaux : un module responsable de l'établissement de connexion avec le GPL. Il interagit avec ce dernier via l'ensemble des messages "Member-Connect" et "ACK-Accept-Connection". Une fois que la session STAMP est établie, deux modules sont exécutés en concurrence. Un module de réception se met à l'écoute de tous les messages "Hello-Neighbors" reçus de la part des GMs voisins. Ce module permet d'alimenter la base d'adjacence *adb*. Le deuxième module concerne les opérations d'envoie de messages. Il permet de diffuser des messages "Hello-Neighbors" dans le voisinage direct et de transmettre des messages "Topology-Information" chaque fois que l'*adb* est mise à jour. Il est aussi responsable de l'envoie des messages "Failure-Notify" et "Leave-Group".

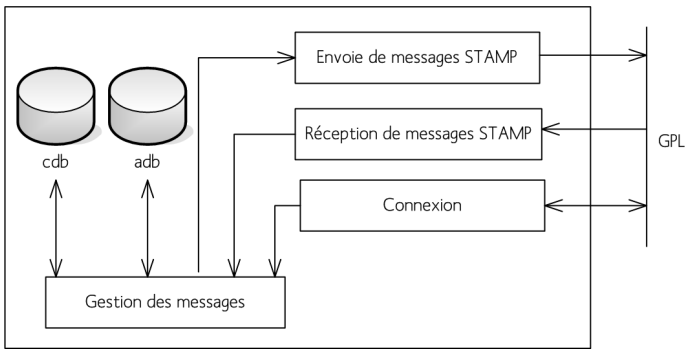

FIG. 3.33: Architecture du GM

3.3.7 La Topologie STAMP

La fonction principale assurée par le GPL, plus spécifiquement le module de traitement, est la construction de la topologie du groupe. En effet, les informations collectées de l'ensemble des GMs sont des informations brutes qui nécessitent un traitement spécifique afin de donner une vision globale de la topologie.

En se référant au modèle de données présenté dans la section 3.2, nous avons distingué deux niveaux de connectivité entre les différentes entités STAMP. Une connectivité de niveau physique et une connectivité de niveau logique. La connectivité de niveau physique est le concept sur lequel se base notre approche pour l'indentification des différents équipements dans le groupe. En effet, un administrateur réseau a besoin de connaître le graphe des connexions physiques qui raccorde tous les éléments du groupe. Un tel graphe est utile pour la détection des éléments défaillants même s'ils sont passifs. La figure 3.34 montre la pertinence d'une vue physique détaillée de la topologie dans le diagnostic des défaillances. Les deux routeurs r_1 et r_2 communiquent entre eux via un commutateur Ethernet c_1. D'un point de vue routage (topologie logique), une coupure de connexion au niveau physique entre ces deux routeurs se traduit par un disfonctionnement au niveau du lien logique [Fig.3.34-d]. La topologie logique n'est pas capable de détecter précisément les raisons de la déconnexion. Au contraire, la topologie physique offre une granularité très fine dans le diagnostic de la panne. La défaillance décrite dans la Figure 3.34-c est identifiée comme étant une rupture au niveau du lien physique qui relie le routeur r_2 au commutateur c_1.

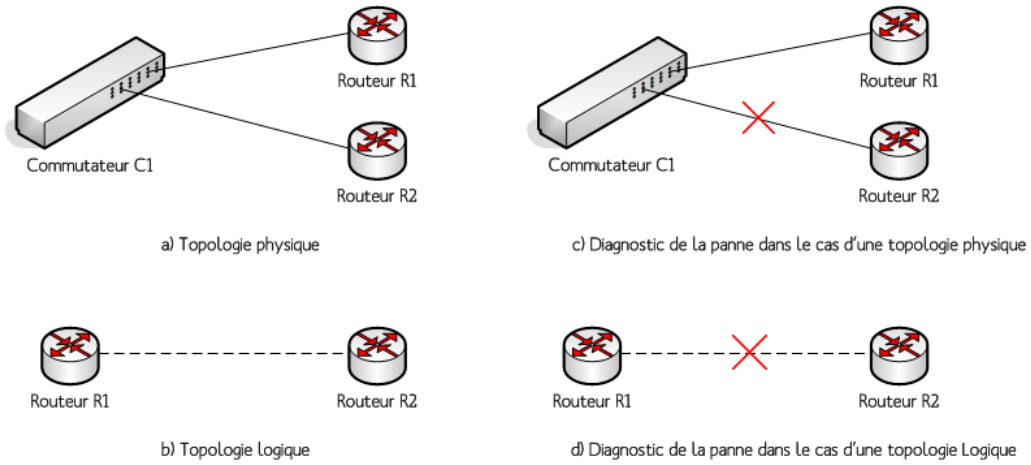

FIG. 3.34: Détection de pannes sur une topologie physique et sur une topologie logique

En outre, un graphe des connexions physiques permet d'identifier de façon unique toutes les interfaces d'un GM dans le groupe et par la suite de faciliter les tâches de gestion. Toutes les informations décrites ci-dessus figurent dans la *tdb*. En effet, chaque GM est identifié par une adresse physique unique. Cette adresse sera l'identifiant de l'équipement réseau dans le groupe. La connectivité d'un GM avec ses voisins est déduite à partir des adresses physiques de ses ports ainsi que des adresses physiques des GMs adjacents (structures <eth-port> et <atm-port> définis dans la section 3.2.2). Ainsi, sur la base des adresses physiques et des informations d'adjacence, le GPL est capable de déduire, à partir de la *tdb*, un graphe de connectivité physique.

Le deuxième type de connectivité que le GPL est capable de fournir, est la connectivité logique. Par le terme logique, nous évoquons un aspect protocolaire mono-niveau, pour un certain niveau de protocoles, où la topologie est déduite vis-à-vis d'un protocole bien défini (Par exemple un protocole de routage comme OSPF). En effet, sur la base du modèle de données présenté dans la section 3.2, nous pouvons identifier plusieurs interconnexions possibles entres les GMs, suivant le type des encapsulations et des technologies déployées. La topologie logique est utile pour identifier le chemin de données qu'il s'agisse d'une technologie de commutation ou de routage. Dans le premier cas, le chemin de données est déduit à partir des structures <mpls>, <vpvc> et <vlan> de la *tdb*. Dans le deuxième cas, les routes sont identifiées à partir des structures <ipv4> et <ipv6>. L'utilité de dresser de telles topologies opérationnelles est de pouvoir identifier les points de défaillance sur les chemins de données. Si nous prenons par exemple le cas de la fonction de contrôle d'admission, l'origine d'une dégradation de performance d'un service réseaux peut être détecté facilement en identifiant le point de congestion sur le chemin de données.

La figure 3.35 montre une portion de la topologie d'un réseau de collecte. La configuration des différents équipements est décrite par les tableaux ci-dessous [Tab.3.1] [Tab.3.3] [Tab.3.4] [Tab.3.2] [Tab.3.5] [Tab.3.6] [Tab.3.7] [Tab.3.8]. Chacun de ces tableaux décrit respectivement l'identifiant du port, ses paramètres de configuration ainsi que le type d'encapsulation correspondant [Fig.3.1].

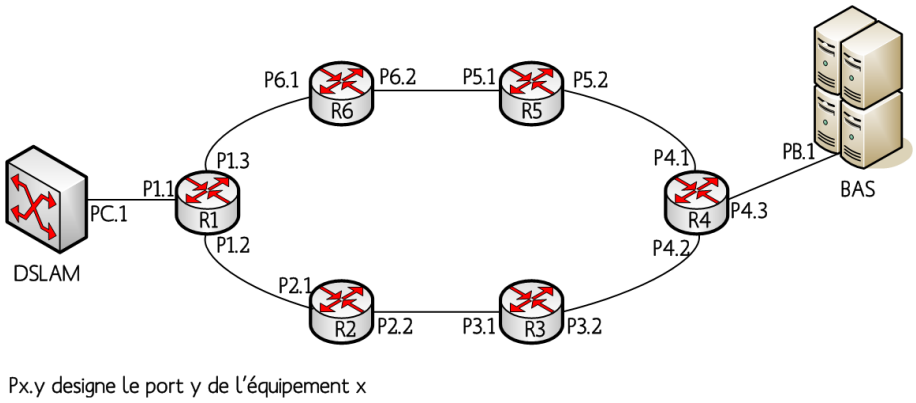

FIG. 3.35: Exemple de topologie de collecte

TAB. 3.1: Encapsulations : routeur r_1

P1.1	vlan-id : 1	16
	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.2	12
	mpls-local-label : null	23
P1.3	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.65	12
P1.2	vlan-id : 1	16
	ipv4-address : 192.168.13.225	12
	mpls-outgoing-label : 21	23

TAB. 3.2: Encapsulations : routeur r_4

P4.2	vlan-id : 1	16
	ipv4-address : 192.168.13.162	12
	mpls-local-label : 31	23
P4.3	vlan-id : 1	16
	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.34	12
	mpls-outgoing-label : null	23
P4.1	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.130	12

TAB. 3.3: Encapsulations : routeur r_2

P2.1	vlan-id : 1	16
	ipv4-address : 192.168.13.226	12
	mpls-local-label : 21	23
P2.2	vlan-id : 1	16
	ipv4-address : 192.168.13.193	12
	mpls-outgoing-label : 25	23

TAB. 3.4: Encapsulations : routeur r_3

P3.1	vlan-id : 1	16
	ipv4-address : 192.168.13.194	12
	mpls-local-label : 25	23
P3.2	vlan-id : 1	16
	ipv4-address : 192.168.13.161	12
	mpls-outgoing-label : 31	23

TAB. 3.5: Encapsulations : routeur r_5

P5.2	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.129	12
P5.1	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.98	12

TAB. 3.6: Encapsulations : routeur r_6

P6.2	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.97	12
P6.1	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.66	12

TAB. 3.7: Encapsulations : DSLAM

PC.1	vlan-id : 1	16
	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.1	12

TAB. 3.8: Encapsulations : BAS

PB.1	vlan-id : 1	16
	vlan-id : 2	16
	vlan-id : 3	16
	ipv4-address : 192.168.13.33	12

Si nous nous intéressons à identifier l'ensemble des LSPs établis, il suffit de se référer aux interfaces où le protocole MPLS est configuré et où les types d'encapsulation propres à MPLS figurent. Dans l'exemple illustré par la figure 3.35, le type d'encapsulation MPLS configuré sur les interfaces est égal à 23, ce qui correspond à une encapsulation MPLS sur Ethernet. L'identification des LSP est alors effectuée en faisant la correspondance entre les labels MPLS des interfaces adjacentes afin de trouver le chemin de données. Ce chemin correspond au LSP passants par les routeurs r_1, r_2, r_3 et r_4 via les interfaces P1.2, P2.1, P2.2, P3.1, P3.2 et P4.2.

3.3.8 STAMP vs SNMP

L'ensemble des solutions de découverte de topologie étudiées dans le chapitre 1 a montré son incomplétude vis-à-vis de nos critères d'évaluation. Parmi ces solutions, SNMP se situe comme étant la solution la plus proche de ce que nous cherchons à établir. En effet, SNMP est aujourd'hui implémenté sur presque tous les équipements réseau. En outre, la multitude de MIBs qui existent rend possible l'identification de toutes les propriétés d'un équipement réseaux. Dans ce passage, nous essayons d'identifier la valeur ajoutée de STAMP par rapport à SNMP et plus particulièrement celle du contrôle d'admission d'appels.

Comme nous l'avons mentionné précédemment, le point principal qui fait la différence entre STAMP et SNMP est le modèle de communication qui existe entre les entités clientes et l'entité serveur. En effet, si les agents ont un comportement passif dans l'architecture SNMP, les GMs sont beaucoup plus autonomes et réactifs aux changements qui se passent dans le groupe dans la philosophie STAMP. Un équipement réseau n'a plus besoin d'attendre une requête ψ afin d'être interroger par son manager (SNMP). Il peut à tout moment envoyer des messages au GPL. Même si les agents SNMP disposent de mécanismes de trap afin de pouvoir notifier des évènements et des alertes prédéfinies à priori par l'administrateur réseau, ces mécanismes ne peuvent pas être

une solution efficace et robuste pour un système où les données de configuration et d'état de liens peuvent changer fréquemment. En effet, pour envoyer un trap il faut établir une session TCP chaque fois où un changement de topologie survient. Ceci présente un défaut majeur d'une telle architecture vu le coût d'établissement de la session TCP. Ce problème peut être résolu en utilisant le protocole UDP, mais malheureusement ce protocole ne répond pas à nos critères de complétude et de précision vu sa non garantie d'acheminement. Dans tous les cas, le manager devra ensuite interroger l'équipement pour connaître la nature du problème, voire explorer plusieurs MIBs afin de mettre à jour la topologie. En outre, SNMP est fortement dépendant de l'utilisation des MIBs. Les informations sollicitées et/ou remontées au manager sont généralement issues des MIBs agents. Ainsi, une incohérence ou une incomplétude au niveau des MIBs peut influer directement sur le fonctionnement du protocole. La diversité et le niveau de détails des informations stockées dans les MIBs, qui divergent parfois du contexte de la recherche de topologie, rendent l'accès à une telle ressource coûteux en termes de CPU. Au contraire, STAMP utilise ses propres structures de données : la *tdb* et l'*adb*. Ces structures se distinguent par leur simplicité ainsi que leur adéquation à notre cas d'utilisation.

3.3.9 Emulation et Résultats

Dans cette section, nous abordons le protocole STAMP d'un point de vue technique. Nous exposons l'environnement de développement, l'architecture du déploiement utilisée, les performances réelles et les résultats issus d'émulation ainsi qu'une analyse et une évaluation de ces résultats.

Environnement Technique

STAMP a été développé dans un environnement Linux. La communication entre les différentes entités dépend fortement du type des deux entités interagissantes. En effet, les GMs interagissent entre eux sur la couche liaison (Ethernet ou ATM), et le GPL communique conjointement aux GMs/GBL sur la couche transport. Ainsi, suivant le type d'interaction, trois types de sockets ont été utilisés pour assurer la communication au sein du groupe.

- Les ROW_SOCKET, utilisés avec le domaine de communication PF_PACKET [92], assurent la communication entre les agents Ethernet. En effet, c'est via ces sockets que s'effectue l'échange des messages Hello-Neighbors entre les ports qui supportent la technologie 802.3.
- Les SOCK_DGRAM [92], utilisés avec le domaine de communication ATMPVC, permettent aux agents ATM de communiquer entre eux sur le VP/VC 0/30 via les messages Hello-Neighbors.
- Les SOCK_STREAM [92] établissent une session TCP entre le GM/GBL et le GPL afin d'assurer l'acheminement des messages de contrôle et de topologie.

Comme la majorité des équipements réseaux sont généralement limités en termes de capacité mémoire et de ressources CPU, nous avons opté pour l'utilisation du langage C pour le codage des GMs. Ce choix se justifie par la souplesse que ce langage offre pour le développement des couches basses et l'accès aux matériels ainsi que par sa rapidité d'exécution. Ceci est primordial pour le développement d'applications sensibles au délai.

Du coté du GPL, comme il s'agit d'une entité centrale englobant plusieurs fonctions de gestion et de contrôle parfois disjointes, nous avons opté pour un langage modulaire qui est le C++. Ainsi, il sera beaucoup plus facile et plus simple de gérer le code du GPL et de l'étendre avec d'autres fonctionnalités si nécessaire.

Déploiement Réel et Résultats

Pour étudier sa faisabilité dans un contexte réel d'utilisation, STAMP a été déployé sur un ensemble de huit machines. Le démonstrateur [Fig.3.36] est réparti sur deux réseaux différents : un réseau pour le test des agents Ethernet et un autre pour le test des agents ATM. Le premier réseau "10.194.77.0" est formé de six machines, un GPL, un GBL ainsi que quatre GMs connectés entre eux deux à deux par un câble 100BaseT croisé. Ceci permettra à ces GMs d'échanger les messages Hello-Neighbors via leurs agents Ethernet. Le deuxième réseau "10.193.11.0" comporte seulement deux machines, le GM5 et le GM6. Ces deux machines sont connectées entre eux via une connexion STM1 afin d'assurer une communication ATM sur le VP/VC 0/30.

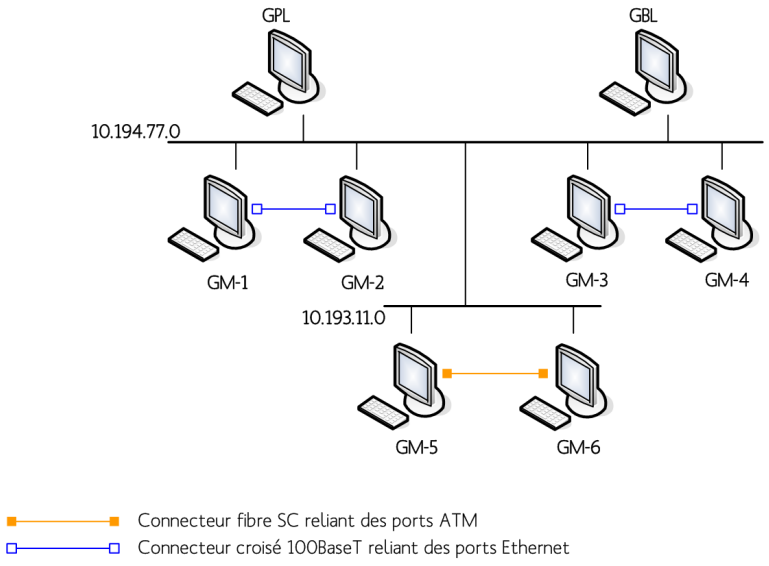

FIG. 3.36: Le testbed des déploiement réels

Les différents GMs commencent par se connecter au GPL avant d'entamer leur phase de découverte de topologie. Chaque fois où une entrée est rajoutée dans l'*adb*, une notification est directement transmise au GPL via un message de topologie. D'un autre coté le GPL et le GBL assurent une supervision mutuelle de leur état de fonctionnement. C'est cette logique de fonctionnement que nous avons pu vérifier avec ce déploiement réel. En effet, nous avons pu enregistrer les observations suivantes :

- La logique d'échange de tous les messages est vérifiée,
- L'utilisation des messages Hello-Neighbors est très efficace pour l'annonce des données d'adjacence. En outre, vu leurs petites tailles, ils n'introduisent pas une surcharge dans le réseau,
- Les données de topologie et les notifications sont envoyées en moins de 2 secondes au GPL, ce qui reflète une réactivité et une sensibilité haute aux changements qui peuvent survenir dans le groupe,
- Le GPL traite en temps réel, via son module de traitement, tous les messages qu'il reçoit de la part de son module d'interface.

La figure 3.37 montre une portion du fichier de log générée par le GPL. Ce fichier renferme toutes les fonctions accomplies par le module de traitements. Nous remarquons que les équipe-

ments et les ports sont identifiés par leurs adresses physiques (ESI ou MAC). Ceci est normal, puisque le point clé du modèle de donnée STAMP est centré sur la notion d'adressage physique.

Bien que le déploiement réel nous ait permis de vérifier la faisabilité et les limites l'implémentation, il ne peut pas prévoir les performances dans le cas d'un déploiement à grand échelle. Pour cela, nous avons monté notre propre scénario d'émulation afin d'évaluer avec beaucoup plus de détails les performances de STAMP lors d'un passage à l'échelle.

```
9/04/2008 14:56:13: The device identified by 00:0d:60:1c:8b:1c is now active on the tdb.
9/04/2008 14:56:14: An accept connection is sent to the device 00:0d:60:1c:8a:b0.
9/04/2008 14:56:14: The device identified by 00:0d:60:1c:8a:b0 is now active on the tdb.
9/04/2008 14:56:14: An accept connection is sent to the device 00:0d:60:1a:c1:3e.
9/04/2008 14:56:14: The device identified by 00:0d:60:1a:c1:3e is now active on the tdb.
9/04/2008 14:56:15: A new port 00:0d:60:1c:8b:1d belonging to the device 00:0d:60:1c:8b:1c is added to the topology db
9/04/2008 14:56:15: An accept connection is sent to the device 00:0e:7f:30:12:93.
9/04/2008 14:56:15: The device identified by 00:0e:7f:30:12:93 is now active on the tdb.
9/04/2008 14:56:16: A new port 00:0d:60:1c:8a:b1 belonging to the device 00:0d:60:1c:8a:b0 is added to the topology db
9/04/2008 14:56:16: A new port 00:0d:60:1a:c1:3f belonging to the device 00:0d:60:1a:c1:3e is added to the topology db
9/04/2008 14:56:17: A new port 00:0e:7f:30:12:89 belonging to the device 00:0e:7f:30:12:93 is added to the topology db
9/04/2008 14:56:27: GBL is alive.
```

FIG. 3.37: Trace du fichier log du module de traitement

Emulation et Résultats

Afin de confirmer les bonnes performances que nous avons obtenues dans un contexte réel, l'émulation du fonctionnement de STAMP sur des topologies de grande échelle s'avère nécessaire. Pour cela, nous avons utilisé une topologie aléatoire générée par l'outil GT-ITM. Cette topologie est formée de 1000 GMs virtuels. Par le terme virtuel, nous définissons des entités logiques ayant des caractéristiques techniques et topologiques identifiées à priori. Chacun de ces GMs virtuels peut avoir entre 8 à 40 ports physiques. Les émulations ont été conduites sur le réseau "10.194.77.0". Chacune des machines GM1, GM2, GM3 et GM4 hébergeant 250 GMs virtuels. Le scénario détaillé des émulations est le suivant :

- Toutes les entités STAMP sont mises sous tension au même temps,
- Les GMs virtuels sont ordonnancés sur chaque machine avec une seconde de différence,
- La logique de transmission des messages "Topology-Information" dans chacun des GMs virtuels suit une loi de poisson avec un taux d'inter-arrivé (inter-arrival rate) égal à 3. Il faut noter que, comme les messages de topologie sont les seuls n'ayant pas une longueur de paquet fixe, nous avons utilisé une fonction aléatoire pour la génération d'unités de données (STAMP-DU) de taille variable allant jusqu'à 1000 Octets,
- Quelques messages "Leave-Group" et "Failure-Notify" sont générés aléatoirement avec une faible fréquence afin de vérifier les fonctionnalités d'ordonnancement et de priorisation implémentées au sein des files HP, MP et LP.

Sur la base de ce scénario, nous avons obtenu de très bons résultats confirmant la robustesse de la conception du GPL ainsi que la logique de fonctionnement du protocole. En effet, les messages bufférisés dans les files HP, MP et LP sont presque tous traités en temps réel. Un retard de quelques millisecondes est parfois enregistré sur le traitement des messages de priorité basse (LP). Ceci n'a aucun inconvénient sur les performances du GPL, car même dans des conditions de charge élevée (nombre très grand de messages à traiter), il n'y a pas de risque de famine ou de débordement des files les moins prioritaires. Comme le montre la figure 3.38, un grand pourcentage des messages STAMP (< 75%) est traité dès son arrivé. Le reste des messages est bufferisé puis traité de manière différée. L'écart entre l'ordre d'arrivée au module d'interface et l'ordre de traitement n'est pas très important. Les figures [Fig.3.39] [Fig.3.40] montrent [Fig.3.41] que

l'ordre de traitement des messages STAMP dépend du type de la file en question. Dans les files HP et MP, les messages sont presque tous traités dès leur arrivée. Un retardement sur le traitement des messages est enregistré dans la file LP. Ceci s'explique par le fait que les messages "Member-Connect" ont été bufferisés et retardés en faveur des messages contenus dans les files HP et MP.

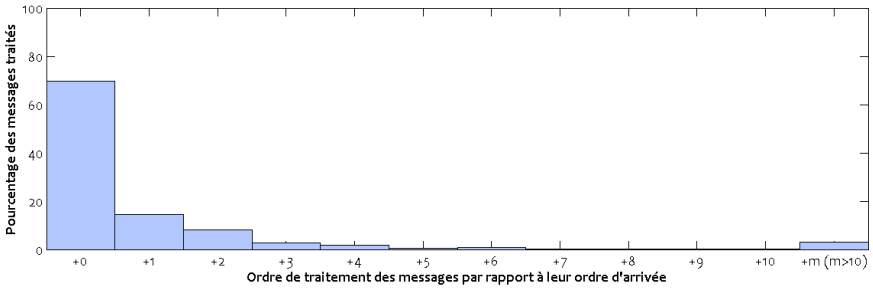

FIG. 3.38: Ordre de traitement global des messages STAMP

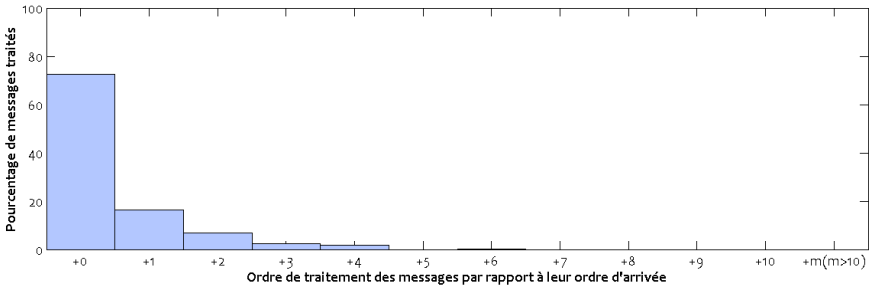

FIG. 3.39: Ordre de traitement des messages STAMP dans la file HP

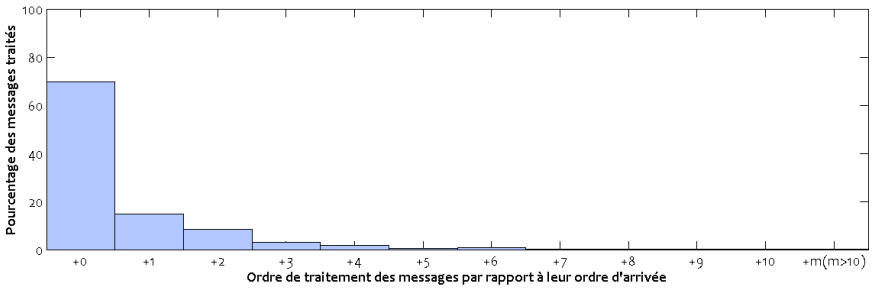

FIG. 3.40: Ordre de traitement des messages STAMP dans la file MP

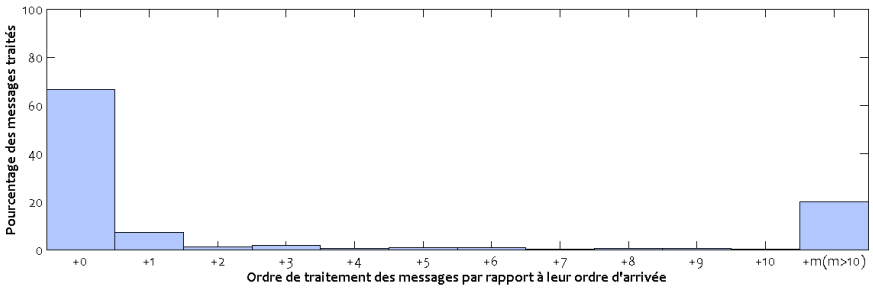

FIG. 3.41: Ordre de traitement des messages STAMP dans la file LP

Comme le message "Topology-Information" est le plus volumineux des messages, et comme c'est le message qui nécessite le plus de traitements (parcours de la *tdb*, et gestion de données de topologie), nous avons étudié le comportement du GPL vis-à-vis de ce type de message. En effet, nous avons mesuré le temps d'occupation du processeur en fonction du nombre de messages "Topology-Information" manipulés par le module de traitement. Nos observations sont les suivantes [Fig.3.42] :

- Le taux d'occupation du processeur est fortement dépendant du nombre et du volume de messages "Topology-Information" traités par le GPL. Plus le nombre de messages est élevé, plus le processeur est occupé,
- Le pic du taux d'utilisation du processeur est enregistré lors de la phase d'initialisation du protocole lorsque tous les GMs annoncent leurs informations d'adjacence au GPL. Ce pic est égal à 20%. Il est enregistré pour un nombre de 190 messages "Topology-Information", ce qui présente un très bon résultat sachant le type de traitements accordé à de tels messages,
- A la fin de la phase d'initialisation, le protocole est plus stable. Le taux d'utilisation du processeur est négligeable.

L'ensemble de ces résultats montre la robustesse du GPL lors d'un passage à l'échelle. Même sous de forte condition de charge, le taux d'utilisation du processeur n'est pas élevé. Ceci est du à la légèreté des fonctions de gestion implémentées par le GPL ainsi qu'à la petite taille des messages STAMP manipulés.

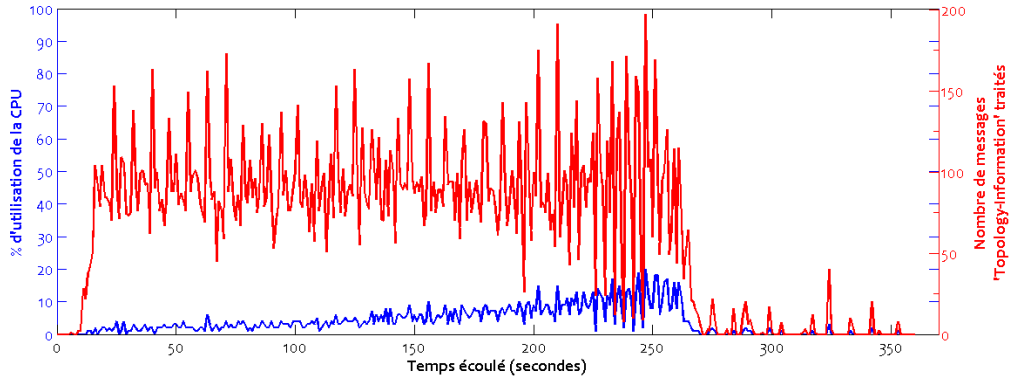

FIG. 3.42: Etude de la performance du GPL

3.4 Bilan

Tout au long de ce chapitre, nous avons décrit le protocole de signalisation STAMP, son modèle de donnée, son architecture, sa structure protocolaire, l'ensemble des messages ainsi que le mode de fonctionnement de ses différentes entités. En effet, STAMP est un protocole permettant la propagation des informations de topologie dans le réseau afin d'offrir une vue détaillée et une mise à jour de la cartographie du domaine à un instant donné. La vue dynamique obtenue est vraie à tous instants, contrairement à la plupart des outils d'inventaire qui remontent une image statique valide à un instant t mais non valide à l'instant $t + \Delta t$.

Deux technologies de liaison sont prises en considération dans les spécifications de STAMP. Il s'agit des technologies Ethernet et ATM, souvent déployées dans les réseaux d'accès ADSL, de collecte et de backbones d'opérateurs de Télécom. L'ensemble des technologies de transport, circuit et paquet, sont considérées comme des encapsulations au dessus de la couche liaison basique. Nous nous sommes restreint dans ce mémoire à étudier IP(v4/v6), MPLS, les VP/VC et les VLANs. D'un point de vue architectural, STAMP se base sur quatre notions de base : a) la notion de *Groupe* qui définit le domaine réseau dans lequel STAMP est déployé, b) le *GPL* qui joue le rôle de l'entité centrale responsable de la collecte et de la gestion des données de topologie, c) le *GM* qui représente le rôle attribué à tous les équipements réseau dans le groupe, et d) la séparation entre découverte des adjacences et remontée des informations.

La communication au sein du groupe est assurée via trois agents : un agent ATM, un agent Ethernet et un agent TCP. Les deux premiers agents permettent la découverte du voisinage direct. En effet, des messages "Hello-Neighbors" sont envoyés aux équipements adjacents afin de leur annoncer les caractéristiques physiques des ports auxquels ils sont connectés. De cette façon, chaque GM détient une liste restreinte d'équipements qui lui sont directement connectés ainsi que de leurs ports physiques d'interconnexion. Le troisième agent est de type TCP. Son rôle est de transmettre les informations de topologie collectées par les GMs au GPL afin de procéder aux fonctions de traitement et de corrélation. Cet agent est aussi responsable de la transmission des messages d'établissement de connexion ainsi que des messages de notification et de contrôle.

Afin de garantir plus d'efficacité au niveau du GPL, nous avons conçu des mécanismes de priorisation dans le traitement des messages reçues. Trois FIFOs ont été mises en place. Une file HP (High Priority) concernant les messages de haute priorité. Elle comporte tous les messages fortement sensibles au délai, là où un retardement de traitement peut fausser la cartographie réelle du groupe. La deuxième file concerne les messages avec une priorité moyenne, il s'agit de la file MP (Medium Priority). La file LP (Low Priority) est celle ayant la plus faible priorité de traitement. Elle concerne les messages de demande de connexion. Un retard dans le traitement de ces messages peut s'avérer utile lors de la phase d'initialisation du protocole. Sur la base de ce système de priorisation, le GPL est en mesure d'établir une image réelle et détaillée de son groupe. Cette image sera plus tard utile pour accomplir des fonctions de gestion du réseau, telle que la fonction de contrôle d'admission dans notre cas d'utilisation.

D'un point de vue technique, STAMP a été implémenté dans un environnement Linux afin d'étudier ses performances dans un contexte de déploiement réel. Les résultats d'évaluation se sont avérés très bons : a) la logique d'échange des messages fonctionne bien, b) les notifications de données de topologie et de changement d'état sont fait en temps réel, et c) la base de topologie *tdb* contient toutes les informations de connectivité logique et physique. Pour confirmer ces résultats dans le contexte d'un déploiement à grande échelle, un autre scénario d'émulation a été conduit afin de tester la charge de traitement du GPL et d'étudier ses limites. Ce scénario a

montré l'efficacité du GPL à traiter divers et nombreux messages concurrents.

Chapitre 4

Schéma d'Admission d'Appels

Sommaire

Dans le Chapitre 3, nous avons présenté le protocole de signalisation STAMP ainsi que sa logique de fonctionnement. Une des motivations derrière le développement de ce protocole est d'assurer la remontée, en temps réel, des informations de topologie et de QoS. Comme l'ensemble de ces informations est stocké coté GPL, ce dernier sera responsable des différentes fonctions de gestion, parmi lesquelles nous nous intéressons spécialement à la fonction de contrôle d'admission. Par conséquent, le GPL devra implémenter les mêmes fonctionnalités de CAC qu'un Bandwidth Broker.

Dans ce chapitre, nous nous intéressons tout d'abord à la façon avec laquelle le GPL exécute la fonction de contrôle d'admission sur un modèle détaillé de la topologie. Ensuite, une étude de passage à l'échelle sera établie afin de fixer les limites d'utilisation du modèle classique. Deux nouveaux schémas d'agrégation de topologie seront alors définis et évalués.

4.1 Problématique de la Fonction de Contrôle d'Admission

4.1.1 GPL : la Fonction de CAC

Suivant nos spécifications, le GPL est le point central dans l'architecture du protocole STAMP qui exécute les fonctions de gestion, en particulier la fonction de contrôle d'admission. Contrairement à un schéma classique de Bandwidth Broker, le GPL bénéficie de la réactivité du protocole STAMP qui l'approvisionne en temps réel en informations détaillées de la topologie du groupe qu'il contrôle. Ainsi, le GPL est capable de trouver, à partir de la *tdb*, le chemin qu'un trafic de données pourra suivre afin de vérifier s'il y a assez de ressources disponibles pour accepter une nouvelle demande de service.

La philosophie suivie par le GPL est la suivante : Il identifie le type de trafic à contrôler, à savoir si la transmission va se faire en mode paquet ou circuit. Le chemin de données est toujours identifié à partir de la *tdb*. Dans le mode paquet, ce sont les informations injectées dans les TLVs Forwarding-v4 et Forwarding-v6 qui seront examinées. Ces données sont obtenues à partir des informations de routage annoncées par les GMs. Dans le mode circuit, l'identification du chemin suivi par le trafic dépend de la technologie déployée qu'ils s'agissent d'un tunnel MPLS, d'un VLAN ou d'un VP/VC. En effet, il suffit de trouver la connectivité entre les différents identifiants de circuits logiques pour déduire le chemin de donnée de bout en bout. Il est à noter que le chemin de données peut être hybride (formé par plusieurs technologies de mode paquet ou circuit). Dans ce cas, le chemin de données est identifié par rapport à la technologie la plus haute dans le schéma des encapsulations du trafic. Si nous prenons l'exemple de la figure 4.1, le trafic est transmis sur le VP/VC 0/128 de la box depuis le port P.B1. Le GPL recherche alors dans la *tdb* le port directement connecté à P.B1, là où un VP/VC égal à 0/128 est configuré. Sachant que le flux de donnée sera transmis sur le VLAN 3, le GPL identifiera dans sa *tdb* le port d'entrée du saut suivant en se référant aux propriétés d'adjacence directe et d'encapsulation VLAN du port PD.2 du DSLAM (Digital Subscriber Line Access Multiplexer). Arrivant au routeur R1, le niveau d'encapsulation supérieur est MPLS. Ainsi, le GPL se basera sur les informations de local-Label et outgoing-Label pour joindre le routeur R3. Enfin, l'identification du BAS est faite par l'identification du voisin direct du port P3.2.

Ainsi, un parsing de la *tdb* permet au GPL de déterminer le chemin qu'un trafic de données devra suivre. Il est ensuite capable par de simples mécanismes d'addition et de suppression d'estimer si les ressources disponibles peuvent satisfaire une nouvelle demande de connexion ou non.

Si ce mécanisme de parcours de la *tdb* n'est pas coûteux dans le cas des réseaux de petites ou

moyennes tailles, il peut présenter une entrave à la qualité de service dans le cas d'un passage à l'échelle, ce qui est le cas généralement dans les réseaux de backbone IP. Par conséquent, il a fallu trouver les mécanismes nécessaires afin de réduire la taille de la *tdb* et par la suite la volumétrie de données induites.

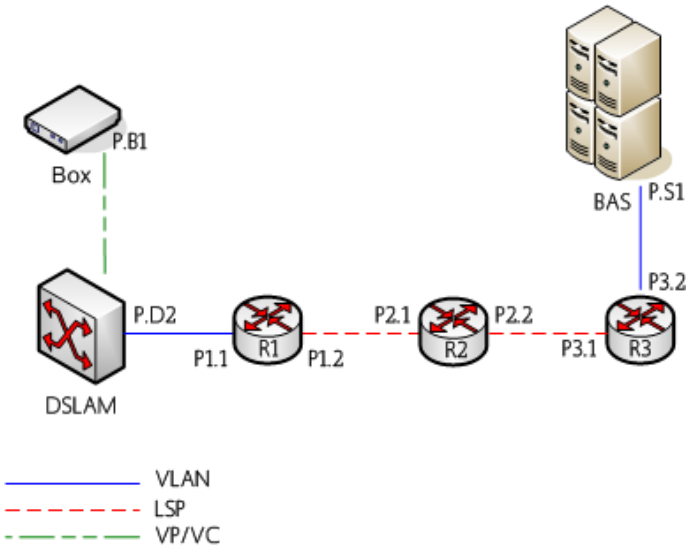

FIG. 4.1: Exemple d'admission d'appel sur la base de la topologie détaillée

4.1.2 Propriété de Passage à l'Échelle

Bien que nous nous sommes restreint dans notre modèle de données à l'ensemble des informations jugées nécessaires pour la fonction de contrôle d'admission, la volumétrie de ces informations devient trop grande lors d'un passage à l'échelle. En effet, plus la topologie est grande, plus le GPL passe du temps pour parcourir la *tdb*. Il doit vérifier la capacité de tous les liens sur le chemin de données pour pouvoir décider de l'acceptation ou non d'une nouvelle demande de connexion. Aussi, le GPL passe par une phase de blockage chaque fois où une mise à jour survient dans le réseau. Il s'agit, lors de cette phase, de refaire le processus d'admission sur tous les liens re-routés. Alors, plus le temps de décision est important, plus la phase de blockage est longue. Pour résoudre ces deux problèmes, nous avons exploré les mécanismes d'agrégation de topologie présentés dans le chapitre 1. Le but est de réduire la taille de la base de topologie *tdb* et de la représenter sous une forme plus compacte. De cette façon, le temps décision sera réduit et par conséquent la phase de blockage aussi. Bien que ces mécanismes d'agrégation puissent apporter une solution aux problèmes de passage à l'échelle, elles introduisent des déformations dans les informations d'états des liens du réseau. Ainsi les paramètres de qualité de service à travers un réseau agrégé vont dévier des valeurs réelles, ce qui peut induire en erreur les fonctions de gestion, plus spécifiquement la fonction de contrôle d'admission.

Notre objectif est de trouver un modèle d'agrégation permettant à la fois de réduire la taille de la *tdb*, et de maximiser la pertinence et la précision des informations agrégées. Il est à noter que, en aucun cas, il est possible d'éviter la déformation introduite par le processus d'agrégation.

Sans trop dévier de la logique des travaux qui ont été présentés dans le chapitre 1, nous

proposons deux schémas d'agrégation de topologie. Le premier schéma est basé sur une représentation *full-mesh* et le deuxième sur une représentation *star*. Comme la technique avec laquelle ces représentations sont construites n'a pas été fixée dans la littérature, nous avons défini nos propres règles d'agrégation. Elles sont triviales dans le cas du schéma *full-mesh* alors qu'elles sont beaucoup plus compliquées dans le cas du schéma *star*. Il est à noter que nous n'abordons pas dans cette partie le problème d'agrégation de liens vu que nous travaillons dans le cas où seule une route est possible d'un noeud source i vers un noeud destination j. Cette hypothèse est souvent vraie, sauf dans le cas où un protocole de routage multi-chemin est activé dans le réseau, tel que ECMP (Equal Cost Multipaths Protocol). Nous nous restreindrons aussi, dans un premier temps, à l'étude de la fonction de CAC dans les réseaux sensibles seulement aux paramètres de délai et de bande passante. La prise en compte de divers paramètres de QoS sera étudiée plus tard. La logique de fonctionnement des mécanismes d'agrégation sera explicitée sur la base de la topologie décrite par la figure 4.2.

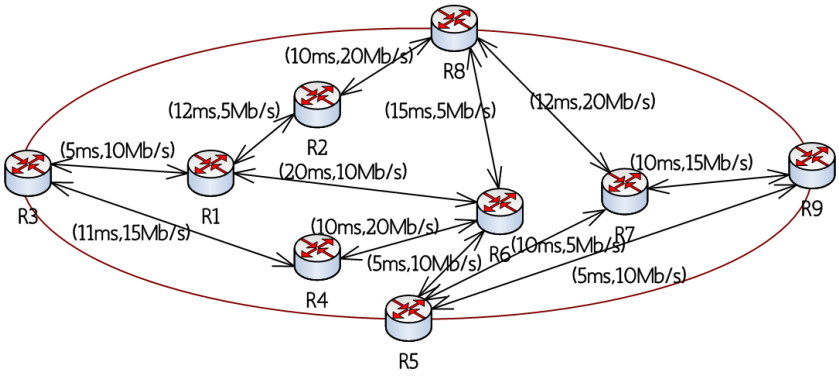

FIG. 4.2: Exemple de topology d'un backbone IP

4.2 Définition des Modèles Agrégés

4.2.1 Modèle Full-mesh

Comme mentionné précédemment, le schéma *full-mesh* est une représentation agrégée fortement connexe. Elle consiste à établir un schéma complètement maillé entre les différents noeuds de bordure d'un domaine via un ensemble de chemins logiques de bout en bout. Dans notre philosophie, nous nous sommes basés sur les informations de routage afin de déterminer la pondération de ces chemins logiques en termes de QoS. En effet, sachant l'ensemble des liens physiques qui forment un chemin logique L, la pondération de L en paramètres de qualité de service est calculée sur la base des définitions 1, 2 et 3 citées dans le chapitre 1. La somme sera appliquée au paramètre de délai et le minimum pour le paramètre de bande passante. Comme le montre la figure ci-dessous [Fig.4.3], la métrique associée au chemin logique $r_3 r_5$ est égale à (26,10), ce qui correspond respectivement à la somme des délais sur la route $r_3 \longmapsto r_4 \longmapsto r_6 \longmapsto r_5$ et au minimum de la bande passante.

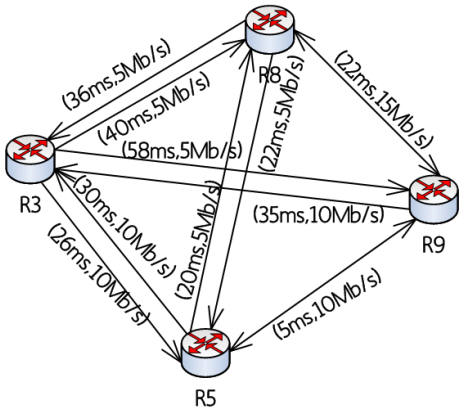

FIG. 4.3: Le schéma abstrait full-mesh

4.2.2 Modèle Star

Contrairement au schéma *full-mesh*, la construction du schéma *star* est beaucoup plus compliquée. La difficulté majeure réside dans le calcul du noyau virtuel (*nucleus*). En effet, ce noyau est la résultante de l'éclatement de chaque chemin logique dans le schéma *full-mesh* en deux morceaux appelés *spokes*. Comme ce noyau doit être unique, il est impossible de le définir de façon exacte. L'idée alors est de trouver une heuristique permettant l'identification de ce noyau de manière efficace. Il est à noter que le calcul du noyau est effectué via la détermination des pondérations de l'ensemble des *spokes*. En outre, l'heuristique est jugée pertinente quand les paramètres de QoS annoncés par la jointure de deux *spokes* (schéma *star*) ne présentent pas une grande déviation par rapport aux paramètres de QoS annoncés par le chemin logique correspondant (schéma *full-mesh*).

L'heuristique de calcul du schéma *star* passe par trois phases : Une première phase consiste à déterminer les *spokes* qui relient les noeuds de bordure au noyau ; Une deuxième phase calcule les *spokes* qui relient le noyau aux noeuds de bordures ; Et une phase finale permet d'établir des courts-circuits appelés *bypasses* entre les noeuds qui présentent une grande déformation par rapport à la représentation *full-mesh* en termes de paramètre de QoS. En effet, il suffit de comparer la pondération d'un lien logique dans le schéma *full-mesh* avec celle des deux *spokes* contigus dans le schéma *star*.

Toutes les équations que nous avons définies pour le calcul des *spokes* découlent des propriétés intrinsèques des paramètres de délai et de bande passante. Afin de pouvoir distinguer ces paramètres dans le cas d'une représentation *full-mesh* ou *star*, nous utilisons respectivement les annotations "m" et "s".

Calcul des Spokes Entrants au Noyau

Comme les *spokes* sont définis à partir de la représentation *full-mesh*, leurs pondérations en paramètres de délai et de bande passante vont forcement dépendre des pondérations de l'ensemble des chemins logiques établis de bout en bout. En effet, les paramètres de QoS associés à un *spoke* reliant un noeud de bordure i au noyau sont déduits à partir des (Card(B)-1) chemins logiques reliant i aux différents autres noeuds de bordure dans la représentation *full-mesh* (B est l'ensemble des noeuds de bordure). L'idée alors, est de trouver un *spoke* médian capable

Algorithm 1 *Calcul du schéma star*

F : Ensemble des chemins logiques du schéma full-mesh
$\mathtt{N}_{input}$: Ensemble des *spokes* entrants au noyau
$\mathtt{N}_{output}$: Ensemble des *spokes* sortants du noyau
/* Phase 1 : calcul des *spokes* entrants au noyau */
Entrées : F
Sorties : $\mathtt{N}_{input}$
/* Phase 2 : calcul des *spokes* sortants du noyau */
Entrées : F et $\mathtt{N}_{input}$
Sorties : $\mathtt{N}_{output}$
/* Phase 3 : calcul des *bypasses* */
Entrées : F, $\mathtt{N}_{input}$ et $\mathtt{N}_{output}$
Sorties : le schéma star

d'agréger les informations de QoS décrites par l'ensemble de ces chemins logiques. La pondération du *spoke* dépend fortement des propriétés intrinsèques des paramètres de délai et de bande passante.

Le paramètre délai, étant un paramètre additif, l'idée était de trouver un critère de comparaison entre le délai d'un *spoke* et le délai d'un chemin logique dans la représentation *full-mesh*. Pour cela, nous sommes parti du fait que le délai d'un *spoke* est toujours inférieur ou égal au délai d'un chemin logique. En effet, comme un *spoke* n'est qu'une partie d'un chemin logique de bout en bout, son délai ne peut pas être supérieur au délai de ce chemin logique. Étant donné $\imath$/$\jmath$ deux noeuds de bordure et n le noyau virtuel du schéma *star*, le paramètre de délai des *spokes* entrant au noyau est calculé de la façon suivante :

$$Pour\ \imath \in \mathtt{B}, \forall\ \jmath \in \mathtt{B},\ avec\ \imath \neq \jmath$$
$$\widehat{dl}(\imath n^s) \leq \widehat{dl}(\imath \jmath^m)$$
$$\widehat{dl}(\imath n^s) \leq \min(\widehat{dl}(\imath \jmath^m))$$

Nous choisissons la borne inférieure pour déterminer le délai d'un *spoke* $\imath n$,

$$\forall \imath \in \mathtt{B},\ \widehat{dl}(\imath n^s) = \min(\widehat{dl}(\imath \jmath^m)),\ \jmath \in \mathtt{B},\ \jmath \neq \imath \tag{4.1}$$

Comme le paramètre de bande passante ne suit pas la même loi additive que le délai, nous nous sommes basés sur d'autres propriétés pour trouver la capacité d'un *spoke*. Contrairement au délai, la bande passante d'un *spoke* est au minimum égale à celle du chemin logique de bout en bout. En effet, la bande passante d'un chemin logique dans la représentation *full-mesh* est égale à la bande passante minimale enregistrée tout au long de ce chemin. Du coup, la valeur de la bande passante d'un *spoke* est au pire égale à la valeur minimale. La façon avec laquelle nous avons identifié la bande passante d'un *spoke* est la suivante :

$$Pour\ \imath \in \mathtt{B}, \forall\ \jmath \in \mathtt{B},\ avec\ \imath \neq \jmath$$
$$\widehat{bw}(\imath n^s) \geq \widehat{bw}(\imath \jmath^m)$$
$$\widehat{bw}(\imath n^s) \geq \max(\widehat{bw}(\imath \jmath^m))$$

Nous choisissons la borne supérieure pour déterminer la valeur de la bande passante du *spoke* $\imath n$,

$$\forall \imath \in \mathtt{B},\ \widehat{bw}(\imath n^s) = \max(\widehat{bw}(\imath \jmath^m)),\ \jmath \in \mathtt{B},\ \jmath \neq \imath \tag{4.2}$$

Compte tenu des deux équations (4.1) et (4.2), nous déduisons la pondération $\widehat{mt}(\imath n^s)$ du *spoke* $\imath n$ en paramètres de délai et de bande passante.

$$\forall \imath \in \mathtt{B},\ \widehat{mt}(\imath n^s) = (\widehat{dl}(\imath n^s), \widehat{bw}(\imath n^s)) \tag{4.3}$$

La figure 4.4 illustre un exemple pratique de calcul des paramètres de qualité de service du *spoke* $r_3 n$. Il faut tout d'abord identifier l'ensemble des chemins logiques issus du noeud de bordure r_3. A partir de la Figure 4.3, trois chemins sont identifiés : $r_3 r_5$, $r_3 r_8$ et $r_3 r_9$ avec les métriques respectives (26,10), (40,5) et (58,5). Ainsi, sur la base des équations que nous avons établies, la métrique du *spoke* $r_3 n$ est calculée à partir des valeurs de délai $\widehat{dl}(r3n^s)$ ($\widehat{dl}(r3n^s)$=$\min(26, 40, 58)$=26) et de bande passante $\widehat{bw}(r3n^s)$ ($\widehat{bw}(r3n^s)$=$\max(10, 5, 5)$=10), ce qui donne un *spoke* avec une métrique égale à (26,10).

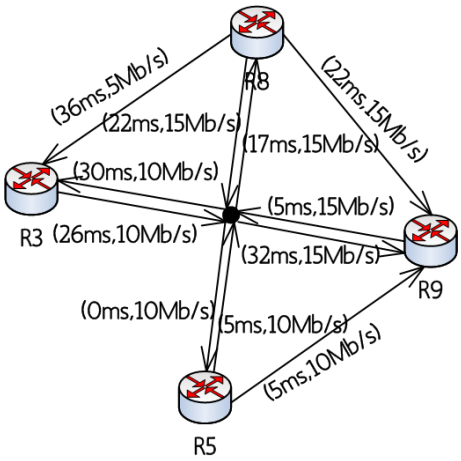

FIG. 4.4: Le schéma abstrait star

Calcul des Spokes Sortant du Noyau

Jusqu'ici nous avons identifié les métriques des chemins logiques dans la représentation *full-mesh* ainsi que les métriques des *spokes* reliant les noeuds de bordure au noyau. Nous nous servirons de ces deux métriques pour trouver la pondération en délai et en bande passante des *spokes* sortant du noyau.

Il est clair que le délai attribué à un chemin logique dans la représentation *full-mesh* représente la valeur minimale que nous pouvons enregistrer sur un schéma agrégé. Ceci est justifié par le fait que les métriques *full-mesh* sont directement déduites à partir de la topologie réelle. Ainsi, afin de garantir l'admissibilité des demandes de connexion dans le schéma *star* vis-à-vis du paramètre délai, il faut que la somme des délais sur deux *spokes* contigus soit au minimum égale au délai du chemin logique de bout en bout. Sur la base de cette observation, nous avons conduit nos calculs pour la déduction de la pondération en délai des *spokes* sortant du noyau :

$$Pour\ \jmath \in \mathtt{B}, \forall\ \imath \in \mathtt{B},\ avec\ \jmath \neq \imath$$
$$\widehat{dl}(\imath n^s) + \widehat{dl}(n\jmath^s) \geq \widehat{dl}(\imath \jmath^m)$$

$$\widehat{dl}(n\jmath^s) \geq \widehat{dl}(\imath\jmath^m) - \widehat{dl}(\imath n^s)$$
$$\widehat{dl}(n\jmath^s) \geq \max(\widehat{dl}(\imath\jmath^m) - \widehat{dl}(\imath n^s))$$

Nous choisissons la borne supérieure pour calculer le délai du *spoke* $n\jmath$,

$$\forall \jmath \in \mathtt{B},\ \widehat{dl}(n\jmath^s) = \max(\widehat{dl}(\imath\jmath^m) - \widehat{dl}(\imath n^s)),\ \imath \in \mathtt{B},\ \imath \neq \jmath \tag{4.4}$$

En ce qui concerne le paramètre de bande passante, nous nous sommes basés sur la même inéquation que celle utilisée pour le calcul des *spokes* entrant au nucleus.

$$Pour\ \jmath \in \mathtt{B}, \forall\ \imath \in \mathtt{B},\ avec\ \jmath \neq \imath$$
$$\widehat{bw}(n\jmath^s) \geq \widehat{bw}(\imath\jmath^m)$$
$$\widehat{bw}(n\jmath^s) \geq \max(\widehat{bw}(\imath\jmath^m))$$

De la même façon que dans l'équation (4.2), nous retenons la borne supérieure de la valeur de la bande passante du *spoke* $n\jmath$,

$$\forall \jmath \in \mathtt{B},\ \widehat{bw}(n\jmath^s) = \max(\widehat{bw}(\imath\jmath^m)),\ \imath \in \mathtt{B},\ \imath \neq \jmath \tag{4.5}$$

Sur la base des deux inéquations (4.4) et (4.5) nous identifions la métrique $\widehat{mt}(n\jmath^s)$ associé au *spoke* $n\jmath$.

$$\forall \jmath \in \mathtt{B},\ \widehat{mt}(n\jmath^s) = (\widehat{dl}(n\jmath^s), \widehat{bw}(n\jmath^s)) \tag{4.6}$$

Comme le montre la figure 4.4, la métrique associée au *spoke* nr_5 est égale à (0,10). Elle est calculée de la façon suivante :

$$\begin{aligned}
\widehat{dl}(nr_5^s) &= \max(\widehat{dl}(r_3r_5^m) - \widehat{dl}(r_3n^s), \widehat{dl}(r_8r_5^m) - \widehat{dl}(r_8n^s), \widehat{dl}(r_9r_5^m) - \widehat{dl}(r_9n^s)) \\
&= \max(0,0,0) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\widehat{bw}(nr_5^s) &= \max(\widehat{bw}(r_3r_5^m), \widehat{bw}(r_8r_5^m), \widehat{bw}(r_9r_5^m)) \\
&= \max(10,5,10) \\
&= 10
\end{aligned}$$

Calcul des Bypasses

En suivant les recommandations PNNI qui imposent la mise en place d'au plus 3*Card(B) liens logiques dans la représentation *star*, nous nous sommes restreint à l'établissement de Card(B) *bypasses* puisque 2*Card(B) *spokes* ont été établis lors des deux phases précédentes. La sélection de ces *bypasses* est faite à partir de la comparaison de la jointure des deux *spokes* $\imath n$ et $n\jmath$ par rapport au chemin logique $\imath\jmath$. Cette comparaison est établie pour chaque couple de noeuds de bordure. Seulement les Card(B) couples présentant une grande déviation par rapport à la représentation *full-mesh* sont retenus. Ils seront interconnectés via des *bypasses*. Il est à noter que la déviation $\hbar$ n'est que la distance qui sépare deux points dans le plan euclidien $\mathcal{P}_{(delay,bandwidth)}$: un point ($(\widehat{dl}, \widehat{bw})$) qui représente la métrique de la jointure des deux *spokes*

in et nj, et un autre point ($\widehat{dl'}$, $\widehat{bw'}$)) qui représente la métrique associée au chemin logique ij dans la représentation *full-mesh* [Fig.4.5].

$$\hbar = \sqrt{\mid \widehat{dl} - \widehat{dl'} \mid^2 + \mid \widehat{bw} - \widehat{bw'} \mid^2}$$

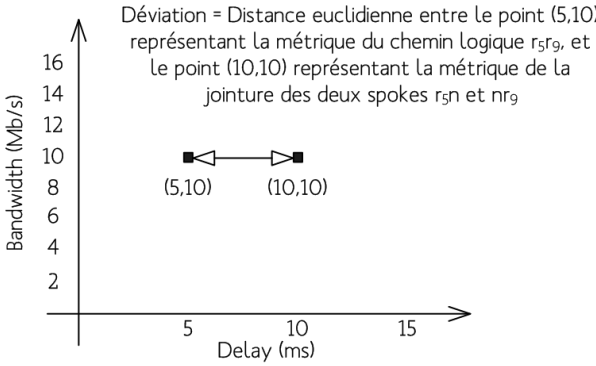

FIG. 4.5: Calcul de la déviation pour l'établissement des Bypasses

Comme le montre la Figure 4.4, trois *bypasses* sont établis entre les noeuds de bordure r_5, r_9, r_8 et r_3, vu la grande déviation qu'ils représentent.

4.3 Fonction de Contrôle d'Admission sur un Modèle Abstrait de Topologie

4.3.1 Définition d'un Modèle Agrégé de la tdb

Le parcours de la *tdb* peut être, dans certains cas, coûteux en termes de ressources CPU. Pour alléger cette tâche, nous proposons en plus du modèle de CAC classique un nouveau modèle basé sur une représentation abstraite de la topologie. L'idée est de minimiser le temps de parcours de la *tdb*. Ce nouveau schéma de CAC se base sur une représentation allégée de la *tdb*, où seules les informations d'états sont décrites. Ces informations sont représentées soit de bout en bout (schéma *full-mesh*) soit en deux sauts (schéma *star*). Nous désignons cette structure par le terme *atdb* (aggregated topology database).

Le but derrière la définition d'une telle structure est d'optimiser la fonction de contrôle d'admission et de la restreindre à un simple test de bout en bout des capacités que le réseau offre. Ainsi, selon les politiques de l'entreprise en matière de qualité de service, c'est à l'administrateur réseau de choisir quelle modélisation de la topologie utiliser afin d'effectuer le contrôle d'admission. En effet, si la topologie est petite, le modèle détaillé de la topologie est plus intéressant car il offre une granularité très fine. Il est alors plus facile de situer les points de congestion. Dans le cas où la topologie est étendue, ce qui est souvent le cas des réseaux de backbone IP, le recours à une topologie abstraite est beaucoup plus intéressant. L'accès à l'*atdb* est plus simple et la fonction de CAC est beaucoup moins compliquée. Avec une telle philosophie, nous estimons apporter une réponse aux problèmes de passage à l'échelle. En effet, l'*atdb* est formée de Card(B)×(Card(B)-1) liens logiques dans le cas d'une représentation *full-mesh*, et de 3×Card(B) *spokes* et *bypasses* dans le cas du schéma *star*. Le calcul des valeurs associées aux paramètres de délai et de bande passante est accompli suivant les équations établies précédemment.

La figure 4.6 montre la localisation de l'atdb et du bloc d'agrégation de topologie dans le module de traitement du GPL.

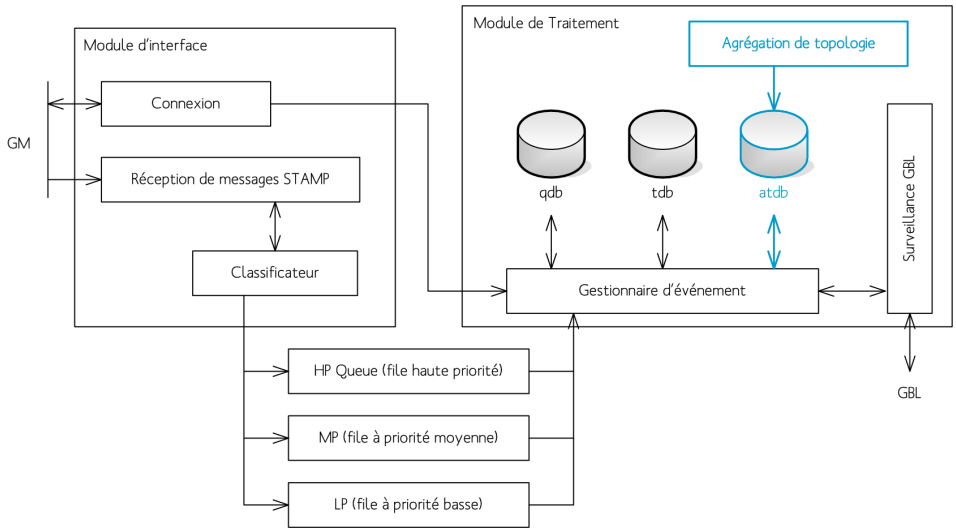

FIG. 4.6: Architecture du GPL avec support de la CAC

Le passage suivant sera consacré à expliciter l'ensemble des règles de contrôle d'admission à appliquer pour gérer les nouvelles demandes de connexions sur le modèle abstrait de topologie.

4.3.2 Les Règles de CAC

Si le contrôle d'admission dans un Bandwidth Broker passe par la vérification de l'admissibilité d'une nouvelle demande de connexion sur la base de la représentation détaillée de la topologie (*tdb*), l'architecture de CAC que nous proposons se base sur une représentation agrégée de bout en bout de la topologie (*atdb*). En effet, le processus d'admission d'appel consiste à vérifier l'état de la demande de connexion vis-à-vis de ce que le modèle abstrait de la topologie peut offrir. Le parcours de l'*atdb* sera plus simple. Une vérification de bout de bout (schéma *full-mesh*) ou en deux passes (schéma *star*) est possible.

Dans le cas du schéma *full-mesh*, l'admissibilité d'une nouvelle connexion ψ, de $\imath$ vers $\jmath$, est vérifiée par rapport aux paramètres de délai et de bande passante. Étant donné ω la demande en bande passante et ϱ la demande en délai, deux tests sont nécessaires afin de décider de l'acceptation ou non de ψ. La première condition consiste à vérifier que la somme de la demande en bande passante ω et des réservations ($\sum_{rsv}$) déjà établies sur le lien logique $\imath\jmath$ est toujours inférieure ou égale à la capacité totale du lien $\widehat{bw}(\imath\jmath^m)$ (4.7). La deuxième condition permet de vérifier l'admissibilité de ψ en termes de délai. Elle est beaucoup plus simple que la condition de bande passante, car il suffit de comparer la valeur de la demande ϱ à ce que peut garantir le lien logique $\imath\jmath^m$ (4.8).

$$\omega + \sum_{rsv(\imath\jmath)} \widehat{bw} \leq \widehat{bw}(\imath\jmath^m) \tag{4.7}$$

$$\varrho \geq \widehat{dl}(\imath\jmath^m) \tag{4.8}$$

La procédure d'admission d'appels dans la représentation *star* n'est pas aussi triviale que celle définie dans la représentation *full-mesh*. En effet, nous distinguons deux types de représentation de chemins logiques, à savoir des *spokes* et des *bypasses*. Ainsi, lors de la phase de prise de décision, le GPL vérifie s'il existe un *bypass* entre les deux noeuds de bordure par lesquels la nouvelle connexion devra passer. Si oui, le test d'admission est calculé par rapport au *bypass* concerné suivant les mêmes règles que celles définies précédemment dans le schéma *full-mesh*. Sinon ce sont les deux *spokes* reliant les deux noeuds de bordure qui seront examinés. Dans ce cas, trois tests sont nécessaires pour pouvoir décider de l'acceptation non pas d'une nouvelle demande de connexion. Les deux premières conditions permettent de vérifier si la somme de la bande passante demandée ω et de l'ensemble des réservations sur le *spoke* entrant $\imath n$ (resp le *spoke* sortant $n\jmath$) est toujours inférieure ou égale à la capacité totale de $\imath n$ (resp $n\jmath$) (4.9) (4.10). La troisième condition examine l'admissibilité de ψ en termes de délai. Elle compare la demande en délai ϱ par rapport à la somme de ce que les deux *spokes* peuvent garantir (4.11).

$$\omega + \sum_{rsv(\imath n)} \widehat{bw} \leq \widehat{bw}(\imath n^s) \tag{4.9}$$

$$\omega + \sum_{rsv(n\jmath)} \widehat{bw} \leq \widehat{bw}(n\jmath^s) \tag{4.10}$$

$$\varrho > \widehat{dl}(\imath n^s) + \widehat{dl}(n\jmath^s) \tag{4.11}$$

4.3.3 Le Scénario d'Emulation

Pour l'évaluation des performances de nos modèles d'agrégation, nous avons conduit nos émulations sur des topologies expérimentales de Backbones commerciaux. Ces topologies sont celles de AT&T, Sprintlink, Abovenet et Exodus [93]. Un tel choix, utilisant des topologies réelles (déjà déployées) et non pas des topologies issues de modèles mathématiques (générateurs de topologie), vise à vérifier l'efficacité de nos mécanismes d'agrégation et leur aptitude à être déployés dans un contexte réel. Les propriétés d'échelle de ces topologies sont représentées dans le tableau ci-dessous [Tab.4.1]. Il est à noter qu'un algorithme SPF (Shortest Path First) est utilisé pour la sélection des routes dans les Backbones étudiés.

TAB. 4.1: Propriétés d'échelle des topologies utilisées

Nom de l'AS	Exodus	Abovenet	AT&T	Sprintlink
Nombre total des routeurs	211	244	729	465
Nombre de routeurs de bordure	32	39	46	56
Degré moyen d'inter-connectivité	5.11	5.05	6.18	6.46

Sur l'ensemble de ces topologies, nous avons utilisé le même scénario de contrôle d'admission afin d'identifier de manière subjective les apports et les limites de nos mécanismes d'agrégation. Ce scénario consiste à émuler un ensemble de 1.000.000 demandes de connexion réparties uniformément entre les différents routeurs de bordure des Backbones. 50% de ces connexions sont permanentes (les réservations sont maintenues jusqu'à la fin de l'émulation). Les 50% restantes identifient les connexions de durée limitée (les ressources réseau réservées sont libérées une fois la connexion arrive à son terme). Les demandes en bande passante sont comprises entre 64Kb/s et 14Mb/s. Ce choix permet de couvrir un ensemble très vaste d'applications réseaux ayant différentes exigences en termes de bande passante. Le délai autorisé est compris

entre 50ms et 1s. Les requêtes de demande de connexion sont envoyées au GPL en spécifiant leurs besoins en délai et en bande passante. A la réception d'une requête, le GPL examine ce que demande l'initiateur de l'appel en termes de paramètres de QoS et vérifie son admissibilité par rapport à ce que le modèle abstrait de topologie offre (*atdb*). L'utilisation de ce modèle abstrait permet de réduire la volumétrie des informations dans le domaine et par la suite de rendre plus rapide le processus de décision, comme le montre son application dans le cas de la topologie Exodus dans les figures 4.7, 4.8 et 4.9. En contre partie, elle introduit une déformation par rapport au modèle détaillé de la topologie.

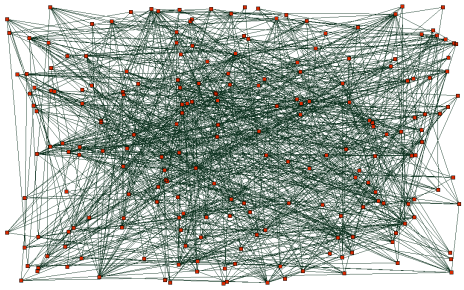

FIG. 4.7: Topologie réelle du Backbone Exodus

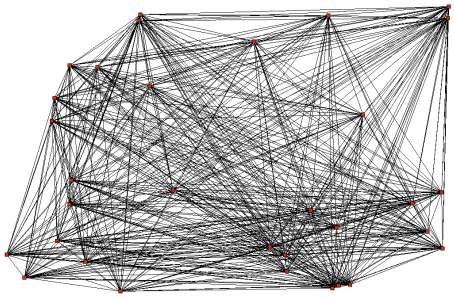

FIG. 4.8: Le Backbone Exodus en full-mesh

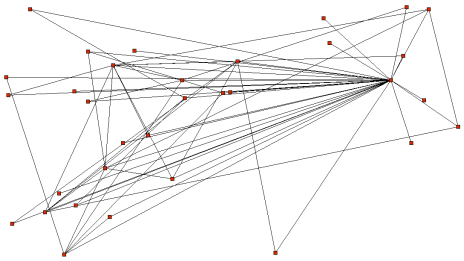

FIG. 4.9: Le Backbone Exodus en star

Dans nos émulations, nous nous sommes basés sur un ensemble de critères d'évaluation. Le premier critère est le coût de construction des modèles abstraits de la topologie. En effet, contrairement à une architecture de Bandwidth Broker classique où la fonction de contrôle d'admission est basée directement sur un modèle détaillé de la topologie, le recours aux mécanismes d'agrégation permet de translater la fonction de CAC sur la modélisation abstraite de la topologie. Cette modélisation abstraite nécessite un temps de calcul τ. Il s'agit du temps pris par le GPL pour la construction des schémas *full-mesh* et *star*. Le but alors, est d'examiner ce temps de calcul afin d'étudier les limites du déploiement de tels schémas. Le deuxième critère identifie le gain apporté sur le temps de prise de décision. En effet, le contrôle d'admission sur les modèles abstraits de la topologie est effectué de bout en bout (schéma *full-mesh*) ou en deux passes (schéma *star*). Il n'y a plus besoin de consulter les informations de routage afin d'identifier le chemin de données et de vérifier par la suite son état vis-à-vis de la demande de service. Dans un modèle abstrait de la topologie, les tables de routage sont consultées seulement lors de la construction ou de la mise à jour des schémas d'agrégation (suite à un re-routage). L'état des ressources est vérifié directement sur les chemins logiques de bout en bout ou sur des *spokes* contigus. Pour la quantification de ce temps de décision, nous avons défini l'indicateur ε.

Il représente le temps moyen en micro-secondes (μs) pour le calcul d'une décision d'admission d'appel. Le troisième critère étudié quantifie le taux de déformation introduit par le processus d'agrégation. En effet, il détermine le nombre de fois où nos modèles d'agrégation ne prennent pas les bonnes décisions d'admission d'appels. Ceci est primordial pour pouvoir juger si la déformation introduite est acceptable ou non. Pour cela, nous avons défini deux indicateurs. Un premier indicateur calcule le pourcentage de décisions correctes prises par le GPL. Il s'agit du cas où les décisions (d'admission ou de refus) prises par le GPL sont similaires à celles prises par un schéma classique de Bandwidth Broker (basé sur un modèle détaillé de la topologie). Cet indicateur est l'indicateur de confiance ϕ. Sur la base de ϕ, il est facile de déduire le pourcentage de fausses décisions. En effet, sur un modèle abstrait de la topologie, le GPL peut rejeter une nouvelle demande de connexion alors qu'il dispose des ressources nécessaires pour l'accepter. Dans ce cas nous parlons d'une sous-estimation des capacités réelles du réseau. Il peut aussi accepter une demande connexion alors qu'il n'a pas les ressources suffisantes pour le faire. Dans ce cas nous parlons d'une sur-estimation des capacités réelles du réseau. Pour des raisons de garantie de service, nous nous focalisons sur le deuxième cas de figure. Nous définissons l'indicateur de sur-estimation φ. Cet indicateur permet de quantifier le pourcentage de demandes de connexion acceptées en excès et d'identifier par la suite le niveau de garantie de service que les schémas d'agrégation sont capables d'offrir. Les deux indicateurs ϕ et φ sont formulés de la façon suivante :

$$\phi = \left(\frac{\texttt{nombre de décisions correctes}}{\texttt{nombre total des demandes de connexion}}\right) \times 100$$

$$\varphi = \left(\frac{\texttt{nombre de décisions surestimées}}{\texttt{nombre total des demandes de connexion}}\right) \times 100$$

4.3.4 Résultats

Sur la base du scénario d'émulation définit précédemment, nous avons étudié les critères d'évaluation τ, ε, ϕ et φ.

Il s'avère que τ est fortement dépendant des propriétés d'échelle de la topologie, en particulier le nombre de routeurs de bordure puisque le degré moyen d'inter-connectivité est presque du même ordre (entre 5 et 6). Les topologies avec le plus petit nombre de routeurs de bordure, sont les plus faciles à construire et inversement [Fig.4.10]. Ceci est évident puisque plus il y a de routeurs de bordure, plus il y a de chemins logiques à construire. Les temps de construction enregistrés sont de l'ordre de quelques dizaines de secondes. De telles valeurs restent acceptables pour des topologies de grandes tailles. Le tableau ci-dessous [Tab.4.2] montre l'ensemble des résultats que nous avons enregistrés sur le modèle *full-mesh* ainsi que sur le modèle *star*. Il s'avère que le GPL passe plus de temps pour la construction du schéma *star*. En effet, la formation de ce schéma passe nécessairement par l'établissement de la représentation *full-mesh*. En outre, les opérations de calcul des *spokes* et des *bypasses* sont compliquées. Le temps de construction le plus élevé est enregistré sur la topologie Sprintlink. Il est égal à 6.40645 secondes dans le cas du schéma *full-mesh*, et à 52.2948 secondes dans le cas du schéma *star*.

Suite à un évènement de re-routage, les modèles abstraits de la topologie doivent être mis à jour afin de garder une vue cohérente de ce qui se passe réellement dans le réseau. Selon le type de schéma d'agrégation, les mises à jour peuvent être plus au moins rapides. Dans le cas du schéma *full-mesh*, le processus de reconstruction du modèle abstrait est très simple. Il suffit de recalculer l'ensemble des chemins logiques affectés par le re-routage. Le temps moyen de

mise à jour enregistré est de l'ordre de quelques millisecondes. En ce qui concerne le schéma *star*, le temps de mise à jour est beaucoup plus élevé. En effet, il faut tout d'abord recalculer le schéma *full-mesh* pour pouvoir par la suite reconstruire le schéma *star*. Le temps moyen de mise à jour enregistré est de l'ordre de quelques dizaines de secondes, ce qui n'apporte pas un gain significatif par rapport au temps de construction initial du schéma *star*.

TAB. 4.2: Etude du temps de formation (secondes) des modèles abstraits

Modèle Abstrait de la topologie	Exodus	Abovenet	AT&T	Sprintlink
full-mesh	0.88609	1.2124	5.32921	6.40645
star	6.8492	11.4947	36.5618	52.2948

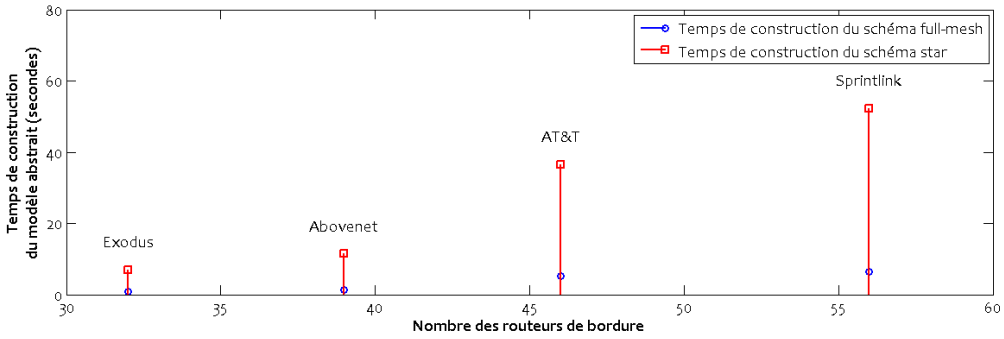

FIG. 4.10: Temps de construction des schémas full-mesh et star

En ce qui concerne le temps de décision ε, les résultats que nous avons obtenus sont très favorables. ε est remarquablement réduit dans le cas du schéma *full-mesh* et beaucoup plus dans le cas du schéma *star*. Le rapport de réduction est au moins égal à 3 dans le cas du schéma *full-mesh*. Il est supérieur dans le cas du schéma *star*, et est au moins égal à 5. Cet écart enregistré entre les rapports de réduction de chacun des deux schémas d'agrégation est dû à leur degré de complexité différent. L'*atdb* d'un schéma *star* est beaucoup plus petite que celle d'un schéma *full-mesh*. En effet, le nombre d'entrée d'un modèle abstrait de topologie *star* est au maximum égal à $3\times$Card(B), alors que dans un modèle *full-mesh*, le nombre d'entrée est égal à Card(B)$\times$(Card(B)-1). Plus Card(B) est grand, plus l'écart est remarquable entre les deux représentations de topologie. Ainsi, le temps d'accès et de parcours de l'*atdb* ajoute une contrainte additionnelle sur le temps global de décision d'appel. Si l'*atdb* est grande, le temps de décision est important et inversement. Ceci justifie les meilleures performances enregistrées avec le schéma *star*. Le tableau ci-dessous [Tab.4.3] illustre en détail les différents résultats calculés sur le schéma *star* et sur le schéma *full-mesh*. Ces résultats montrent que la distribution du temps de décision par rapport à la taille de l'*atdb* suit une loi logarithmique. ε est de l'ordre de o($\log(3\times$Card(B))) dans le cas du schéma star et de o($\log$((B)$\times$(Card(B)-1))) dans le cas du schéma full-mesh.

L'analyse des indicateurs ϕ et φ montre le degré de conformité de nos modèles d'agrégation par rapport aux caractéristiques réelles de la topologie [Tab.4.4] [Fig.4.11] [Fig.4.12]. Il s'avère que le schéma *star* est plus fidèle à la représentation réelle de la topologie. Les indicateurs de confiance et de sur-estimation sont acceptables. Dans le cas du schéma *full-mesh*, la déformation est très grande. Même si dans certains cas, le taux d'exactitude reste acceptable, le taux de

sur-estimation est inadmissible. Il peut atteindre une valeur de 46%, ce qui est inadapté à nos objectifs de garantie de service.

TAB. 4.3: Etude du temps de décision (μs) dans les modèles détaillés et abstraits

Modèles de topologie	Exodus	Abovenet	AT&T	Sprintlink
modèle détaillé	325	495	901	1522
full-mesh	105	117	171	247
star	62	54	69	79

Nous remarquons que le taux de confiance ϕ est aussi dépendant des propriétés d'échelle des topologies étudiées. Plus le nombre des routeurs de bordure est grand, plus la déformation est importante. En effet, un grand nombre de routeurs de bordure engendre un grand nombre de chemins logique à construire. Plus le nombre de ces chemins logiques est important, plus le degré de dépendance entre ces chemins est grand, c'est-à-dire qu'il est fort probable que plusieurs chemins logiques partagent les mêmes liens physiques. Ceci présente la source principale de la déformation enregistrée. Par exemple, si nous prenons les deux topologies Sprintlink et Exodus, l'indicateur ϕ est meilleur sur la topologie d'Exodus du fait qu'elle comporte moins de routeurs de bordure que celle de Sprintlink [Fig.4.11] [Fig.4.12].

TAB. 4.4: Indicateurs de confiance et de sur-estimation des modèles abstraits

Indicateurs	**Modèle abstrait de topologie**	Exodus	Abovenet	AT&T	Sprintlink
ϕ	full-mesh	82.7815%	76.6688%	72.1363%	53.4910%
	star	93.6212%	89.4890%	86.6348%	86.4374%
φ	full-mesh	16.5682%	22.8695%	27.4846%	46.2969%
	star	1.3917%	2.0084%	3.0013%	4.2533%

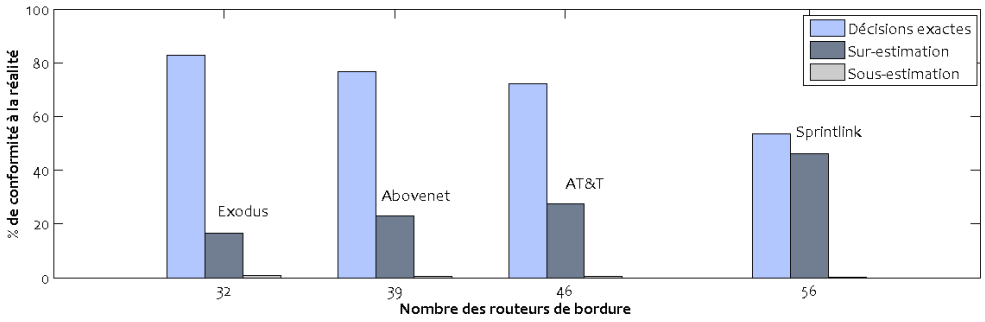

FIG. 4.11: Déformation introduite par le schéma full-mesh

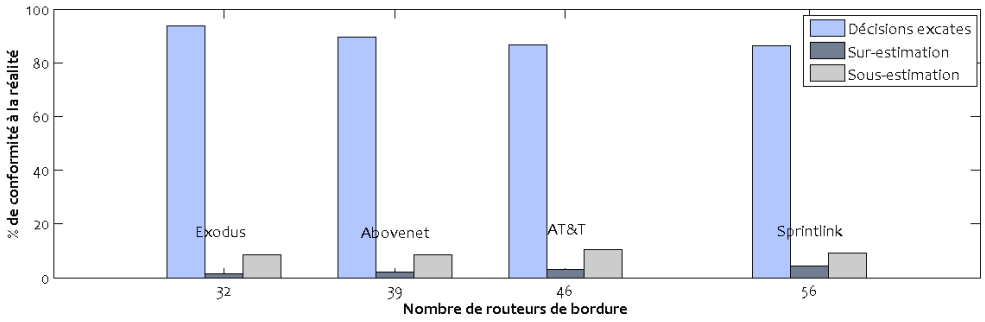

FIG. 4.12: Déformation introduite par le schéma star

4.4 Amélioration du Schéma Full-mesh

Sur la base des résultats enregistrés dans la section précédente, nous pouvons relever deux constatations. En ce qui concerne le schéma *full-mesh*, ses performances sont médiocres. Nous enregistrons parfois l'absence de performances sur certaine topologie telle que celle de Sprintlink (un taux de sur-estimation égal à 46%). Le schéma *star* est meilleur. Ses performances sont acceptables. Un indicateur d'exactitude toujours supérieur à 86% et un indicateur de sur-estimation égal à 3% dans le pire des cas. L'idée alors est d'apporter des améliorations au schéma *full-mesh* afin de réduire essentiellement son indicateur de sur-estimation et de rendre acceptable son indicateur de confiance.

4.4.1 Définition de Règles pour la Gestion de la Bande Passante

En nous focalisant sur la façon dont le schéma *full-mesh* est construit, nous avons remarqué que la cause principale de la grande déviation est la mauvaise gestion de la bande passante au sein du modèle abstrait de la topologie. En effet, il est impossible de gérer, de façon déterministe, la bande passante sur un chemin logique de bout en bout si d'autres chemins logiques l'interceptent en un ou plusieurs liens physiques. Un exemple très simple est illustré par la figure 4.13. x, y et z sont les routeurs de bordure d'une petite topologie formée de quatre noeuds. La modélisation abstraite en *full-mesh* de cette topologie est représentée par une interconnexion entre les différents routeurs de bordure. Supposons qu'un appel est établi entre x et y, avec une réservation de ressources de 4Mb/s. Sur la base de la représentation détaillée de la topologie, une nouvelle demande de connexion de 3Mb/s entre x et z ne peut pas être acceptée à cause de l'épuisement de la capacité du lien physique xw. Au contraire, sur la topologie agrégée, la nouvelle demande est acceptée ce qui n'est pas réalisable.

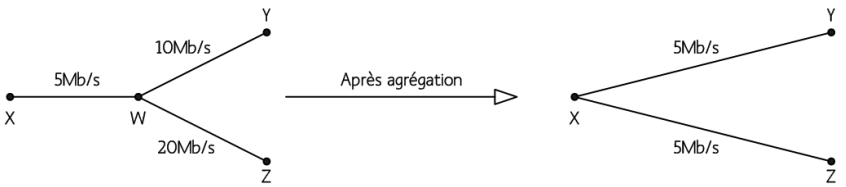

FIG. 4.13: Déformation introduite suite au processus d'agrégation

En effet, la capacité des chemins logiques est calculée indépendamment de l'état des réservations sur les autres chemins logiques. Si des ressources sont réservées sur un chemin logique L, la réservation n'est pas transposée sur les chemins logiques fortement dépendant de L. Nous estimons que plus la dépendance est forte, plus la déformation est grande. Pour résoudre ce problème, il faut dans un premier temps identifier les points de recouvrement des chemins logiques et par la suite ajouter les règles nécessaires pour la gestion de la bande passante. L'identification de ces points de recouvrement passe par l'étude de l'ensemble des liens physiques qui composent chaque chemin logique. Il s'agit de trouver le lien qui a la plus forte probabilité de congestion. La manière dont nous avons défini ce lien de congestion est la suivante :

Définition 6. Un lien physique ℓ est dit lien de congestion d'un chemin logique L si et seulement si il vérifie (dans l'ordre) les trois propriétés suivantes :

- $\ell \subset \mathrm{L}$,
- $\widehat{bw}(\ell) = \min(\widehat{bw}(\ell'))$, $\forall\ \ell' \subset \mathrm{L}/\ell$,
- Card($\theta_{\mathrm{L}(\ell)}$) est maximal. $\theta_{\mathrm{L}(\ell)}$ désigne l'ensemble des chemins logiques qui traversent L en son lien de congestion ℓ.

Ainsi, l'idée est d'identifier le lien physique qui a la bande passante minimale et qui est traversé par le plus grand nombre de chemins logiques [Algorithme.2]. Dans le reste de ce mémoire, nous utilisons la notation *bl* pour désigner un lien de congestion.

Algorithm 2 *Calcul du lien de congestion bl sur un chemin logique L*

1: $\mathcal{M}(\ell)$ /* Ensemble des liens physiques ℓ appartenant à L et ayant la bande passante minimale */
2: **for** $\ell \in \mathcal{M}(\ell)$ **do**
3: indicateur-congestion(ℓ)= (bw_ℓ)/Card($\theta_{\mathrm{L}(\ell)}$)
4: **end for**
5: *bl* = ℓ ayant le indicateur-congestion(ℓ) minimal

En plus des règles de CAC identifiées dans la section 4.3.2, une condition supplémentaire doit être vérifiée. Elle a pour but d'estimer la charge des liens de congestion. En effet, il faut que la somme des réservations sur un chemin logique L ainsi que sur tous les chemins logiques qui le traverse en son lien de congestion *bl*, soit inférieure à la capacité totale du lien de congestion (4.12).

$$\sum_{rsv(\imath\jmath)} \widehat{bw} + \sum_{\ell \in \theta_{\imath\jmath}(bl)} \left(\sum_{rsv(\ell)} \widehat{bw} \right) \leq \widehat{bw}(bl^{\imath\jmath}) \tag{4.12}$$

Donc, l'admissibilité d'une nouvelle connexion ψ, de $\imath$ vers $\jmath$, est vérifiée par rapport aux paramètres de délai et de bande passante. Étant donné ω la demande en bande passante et ϱ la demande en délai, trois tests sont nécessaires pour décider de l'acceptation ou non de ψ. Les deux premières conditions à vérifier sont les mêmes que dans le cas d'un schéma *full-mesh* basique (4.7) (4.8). La troisième porte sur le lien de congestion (4.13). Elle vérifie que la somme de la nouvelle demande en bande passante et de toutes les réservations sur les chemins logiques traversant le lien de congestion de $\imath\jmath$ est inférieure à la capacité totale de ce lien.

$$\omega + \sum_{rsv(\imath\jmath)} \widehat{bw} + \sum_{\ell \in \theta_{\imath\jmath}(bl)} \left(\sum_{rsv(\ell)} \widehat{bw} \right) \leq \widehat{bw}(bl^{\imath\jmath}) \tag{4.13}$$

4.4.2 Emulation et Résultats

Pour évaluer les performances de ce modèle amélioré de la représentation *full-mesh*, nous nous sommes basés sur le même scénario d'émulation que celui défini précédemment. L'idée est de montrer l'apport de la définition des règles de gestion de la bande passante sur la pertinence de la fonction de contrôle d'admission. Les mêmes critères d'évaluation ont été examinés, à savoir τ, ε, ϕ et φ.

En ce qui concerne le temps de construction τ du modèle abstrait de la topologie, la tâche du GPL ne se restreint plus à l'établissement des chemins logiques entre les différents noeuds de bordure mais s'étend aussi à la définition de l'ensemble des liens de congestion. Pour chaque chemin logique, il s'agit d'examiner un ensemble de liens physiques afin d'identifier celui qui peut être l'objet d'une congestion maximale. Cette tâche nécessite un temps supplémentaire de traitement qui augmente suivant le nombre des chemins logiques à établir. Les résultats d'émulation montrent que τ est de l'ordre de quelques dizaines de secondes. En moyenne, la construction du modèle *full-mesh* amélioré dure sept fois plus longtemps que le modèle basique [Fig.4.5]. Aussi, les observations concernant la dépendance entre le temps de construction τ et les propriétés d'échelle des topologies étudiées se sont confirmées [Fig.4.14].

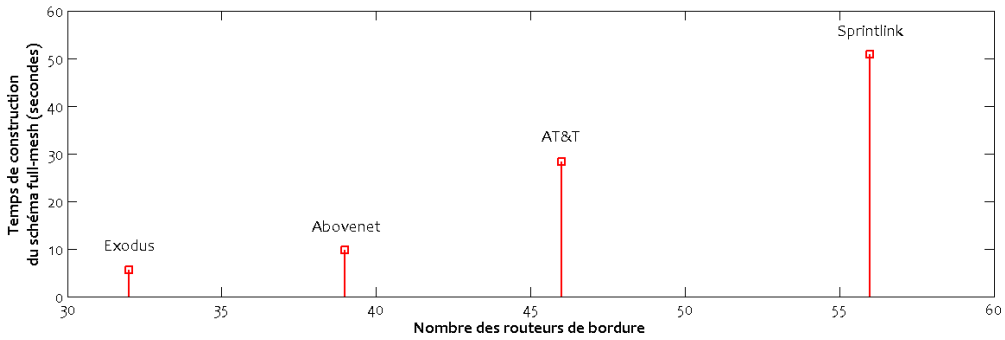

FIG. 4.14: Temps de construction du schéma full-mesh amélioré

Bien que le temps de construction soit élevé, nous estimons que le temps de mise à jour est inférieur au temps de construction. Reste à noter une complexité accrue des mises à jour du schéma amélioré par rapport à celles du schéma full-mesh basique. En effet, l'introduction des liens de congestion impliquent des contraintes supplémentaires pour la reconstruction du modèle abstrait. Il est nécessaire de recalculer à nouveau les liens de congestion pour chaque chemin logique touché par le re-routage. Par définition, un lien de congestion influence l'ensemble des chemins logiques qui le traversent. Ainsi, la mise à jour du schéma *full-mesh* peut conduire à des valeurs différentes à celles obtenues par une reconstruction entière du schéma abstrait. Un chemin logique peut voir son lien de congestion changer sans être touché par les incidents de re-routage. Par exemple, soit L un chemin logique ayant *bl* comme lien de congestion. Par définition, *bl'* est le lien ayant la valeur de bande passante minimale et parcourue par le plus grand nombre de chemins logiques. Supposons que n soit le nombre de chemins logiques qui traversent le lien *bl*. *bl'* est un lien physique appartenant au chemin logique L, ayant la même valeur de bande passante que *bl*, et parcouru par (n-1) autres chemins logiques. Si suite à un incident de re-routage, un nombre m ($2 \leq m \leq n$) de chemins logiques appartenant à $\theta_{L(bl)}$ ne passent plus par *bl*, ce dernier ne peut plus être considéré comme lien de congestion du fait que

bl' a une cardinalité supérieure. Ainsi, nous pouvons constater que le lien de congestion de L a changé malgré que L n'ait pas été touché par l'incident de re-routage. Pour ne pas compliquer la procédure de mise à jour, nous nous sommes restreint à la prise en compte uniquement des chemins logiques affectés par le re-routage. Ainsi, il suffit de recalculer les liens de congestion correspondant à ces chemins pour retrouver le schéma *full-mesh*. Ceci revient à la construction d'une partie du schéma *full-mesh*. Le temps de mise à jour dépend du nombre de chemins logiques affectés par les évènements de re-routage. Le temps moyen enregistré est de l'ordre de quelques secondes. Ceci permettra au GPL de ne pas reconstruire le modèle abstrait chaque fois qu'un re-routage survient dans le réseau.

En ce qui concerne le temps de décision ε, les valeurs enregistrées avec le schéma *full-mesh* amélioré sont légèrement supérieures à celles du schéma basique. La différence est de l'ordre de quelques microsecondes. Contrairement au schéma basique, la fonction de contrôle d'admission ne se base plus seulement sur une simple vérification de ressources sur les chemins logiques de bout en bout, mais aussi sur la vérification des règles de partage de la bande passante sur les liens de congestion. Comme ceci introduit une charge de traitement supplémentaire au niveau du GPL, le temps de décision est légèrement augmenté. L'augmentation est très minime et n'a pas d'impact sur les performances du processus de décision. Par exemple, si nous prenons le cas de la topologie Sprintlink, l'écart enregistré entre les deux représentations (basique et améliorée) est égal à 36 microsecondes. Cette valeur est minime en regard de la valeur du temps de décision [Tab.4.5].

Comme nous l'avons mentionné précédemment, le processus de mise à jour peut conduire à des valeurs différentes à celles obtenues par une reconstruction entière du schéma abstrait. Afin d'assurer un maximum de garantie de service, nous estimons que même si nous reconstruisons en entier le schéma *full-mesh* après chaque événement de re-routage, les performances de la fonction de CAC restent toujours significatives. En effet, de telles performances peuvent être jugées sur la base de deux critères : la fréquence des incidents de re-routage dans le réseau ainsi que le temps pris par le GPL pour exécuter la fonction de CAC. Comme les événements de re-routage (suite à une défaillance dans le réseau) ne sont pas très fréquents, nous nous sommes focalisés sur le critère du temps de décision ε. Les résultats d'émulation montrent qu'il y a un écart remarquable entre le temps de décision au niveau du schéma *full-mesh* amélioré [Tab.4.5] et celui du schéma détaillé de la topologie [Tab.4.3]. Ceci est décisif pour démontrer la pertinence de nos modèles agrégés. Même si la fonction de contrôle d'admission sur un schéma *full-mesh* passe par une phase initiale de construction de la topologie abstraite, elle est plus optimale et rapide dans le processus de décision que celle sur une topologie détaillée. En effet, suite à un évènement de re-routage, la fonction de CAC passe par une phase de blockage. La réception de nouvelles demandes de connexion est bloquée jusqu'à ce que toutes les anciennes demandes de connexion en attente soient traitées. En fonction du temps du convergence du protocole de routage, le nombre de requêtes à traiter peut être élevé. Il peut être de l'ordre de quelques minutes sur des très grands réseaux. Dans le cas d'une représentation agrégée, le modèle abstrait de la topologie doit être recalculé avant d'entamer la phase de blockage [Fig.4.15]. Même si cette phase prend un certain temps pour s'exécuter, nous estimons que le GPL est capable de traiter rapidement les demandes de connexion en attente ainsi que de nouvelles demandes et de surpasser les capacités de calcul d'un Bandwidth Broker classique. En effet, le temps de decision est réduit. La façon de représenter les chemins de données de bout en bout dans la base de topologie agrégée (*atdb*) facilite la tâche de décision du GPL. Ce dernier n'a plus besoin de vérifier l'admissibilité d'une nouvelle demande de connexion sur l'ensemble des liens du chemin de données, une simple vérification de la capacité du chemin logique en question est suffisante.

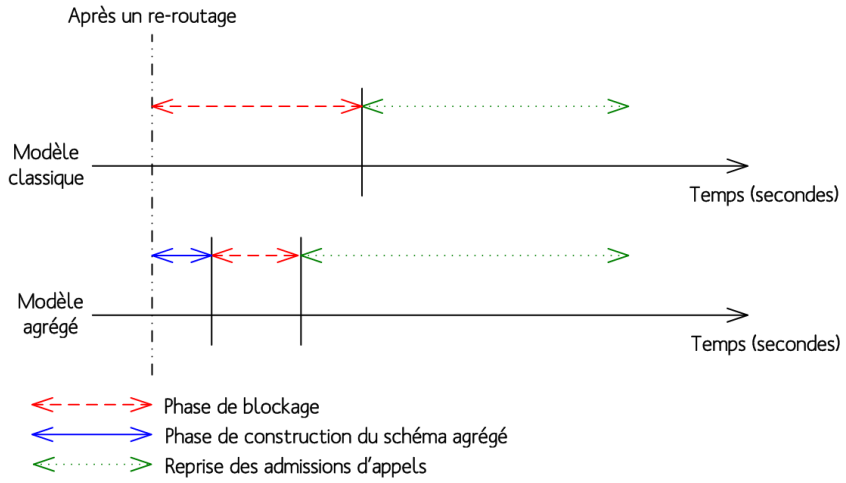

FIG. 4.15: Temps de mise à jour des schémas de topologie

Prenons l'exemple de la topologie Sprintlink. Les résultats d'émulation montrent que le temps de construction du modèle *full-mesh* amélioré est de l'ordre de la minute. Supposons que le GPL soit capable de traiter 5 demandes de connexion par seconde dans la modélisation détaillée de la topologie. Le rapport de réduction du temps de décision entre les représentations détaillées et améliorées est de l'ordre de 5 [Tab.4.5] [Tab.4.3]. Donc, sur la base d'une représentation *full-mesh* améliorée, le GPL est capable de gérer jusqu'à 25 demandes de connexion par seconde. Supposons de plus que suite à un évènement de re-routage, 500 demandes de connexion soient en attente de traitement. Pour un schéma classique de Bandwidth Broker, ces 500 demandes de connexion nécessitent un temps de traitement de 100 secondes. Il s'agit de la durée de la phase de blockage Δk. Dans le cas d'un contrôle d'admission basé sur un schéma *full-mesh* amélioré, Δk est significativement réduite. Les 500 demandes de connexion sont traitées en 20 secondes. Donc, même après une reconstruction entière du modèle abstrait de la topologie, le GPL sera capable de traiter de nouvelles demandes de connexion au bout de 70 secondes (50 secondes de phase de re-construction + 20 secondes de phase de blockage). Nous enregistrons un écart égal à 30 secondes entre la représentation *full-mesh* et la représentation détaillée. Ceci montre bien l'efficacité et la rapidité du principe d'agrégation de topologie, même dans le cas où la représentation abstraite *full-mesh* est reconstruite entièrement après un évènement de re-routage.

TAB. 4.5: Performances du schéma full-mesh amélioré

Indicateurs	**Type de la modélisation full-mesh**	Exodus	Abovenet	AT&T	Sprintlink
τ (s)	full-mesh basique	0.88609	1.2124	5.32921	6.40645
	full-mesh amélioré	5.62461	9.7023	28.3549	50.8401
ε (μs)	full-mesh basique	105	117	171	247
	full-mesh amélioré	119	137	191	283
ϕ	full-mesh basique	82.7815%	76.6688%	72.1363%	53.4910%
	full-mesh amélioré	97.7397%	97.1792%	95.0065%	94.6990%
φ	full-mesh basique	16.5682%	22.8695%	27.4846%	46.2969%
	full-mesh amélioré	1.6414%	2.4067%	4.2977%	4.1698%

Le but de l'amélioration du schéma *full-mesh* basique est de réduire la déformation et de rendre acceptable les indicateurs d'exactitude et de sur-estimation. Les résultats d'émulations [Tab.4.5] montrent de très bonnes performances. La définition des liens de congestion a remarquablement amélioré la précision du processus d'agrégation. Le taux des décisions erronées prises par le GPL est significativement réduit. En effet, l'indicateur de sur-estimation φ est au moins six fois plus petit que dans le cas d'un schéma basique. Le rapport de réduction de la sur-estimation peut atteindre une valeur égale à 11 sur certaines topologies telle que celle de SprintLink. Avec de tels rapports de réduction, la valeur de φ est toujours inférieurs à 5%. Comme conséquence logique de la réduction de φ, l'indicateur de confiance ϕ est augmenté. Cet indicateur est au moins égal 94%, ce qui représente un excellent résultat. Il atteint des valeurs supérieures dans le cas où le nombre des routeurs de bordure n'est pas très grand [Fig.4.16].

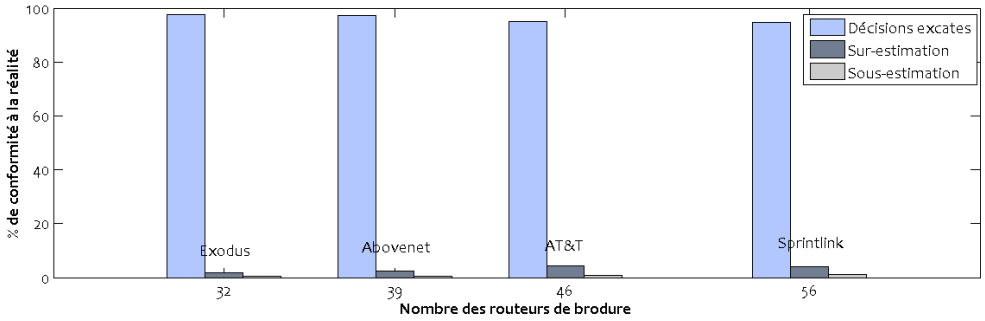

FIG. 4.16: Déformation introduite par le schéma full-mesh amélioré

4.5 Amélioration du Schéma Star

4.5.1 Etude du Schéma Star sans la Définition des Bypasses

En s'appuyant sur l'ensemble des résultats enregistrés dans la section 4.3.4, nous avons voulu pousser plus loin notre façon de construire le schéma *star*. L'idée est de dévier un peu des recommandations PNNI, et de ne pas établir des *bypasses* entre les noeuds de bordure. En effet, nos modèles abstraits de la topologie sont destinés à la fonction de contrôle d'admission. Même en cas de déformation, nous estimons rectifier cette déformation par l'ensemble des règles de CAC que nous définissons.

Les résultats d'émulation précédents [Tab.4.3] [Tab.4.2] ont montré que la présence des *bypasses* alourdit le processus de décision et introduit une charge additionnelle lors de la construction du schéma *star*. En effet, à la réception d'une nouvelle demande de connexion, le GPL doit vérifier dans un premier temps s'il existe un *bypass* entre les deux noeuds de bordure concernés par la nouvelle demande. Selon que ce *bypass* existe ou pas, les règles de contrôle d'admission ne seront pas les mêmes. En outre, le processus de mise à jour des réservations n'est pas optimal. Deux représentations distinctes des liens logiques sont à manipulées à la fois. Si une réservation est accomplie sur un *bypass* ij, elle doit être transposée sur les deux *spokes* in et nj afin de garder un minimum de cohérence entre les deux représentations. Ceci a un impact direct sur l'exactitude de la prise de décision. En outre, la prise en compte des *bypasses* peut poser des problèmes lors de la définition d'éventuelles règles de gestion de la bande passante comme dans le cas du schéma full-mesh amélioré. L'idée alors, est de se restreindre à une modélisation *star*

sans *bypasses* et d'éviter par la suite la manipulation de deux représentations différentes de liens logiques. Ainsi, seul l'ensemble des règles de CAC (4.9), (4.10) et (4.11) sont à vérifier lors du processus de décision.

Nous avons appliqué le même scénario d'émulation à ce nouveau schéma abstrait de la topologie. L'idée est d'étudier l'impact de la suppression des *bypasses* sur la pertinence du schéma *star*. Les résultats enregistrés [Tab.4.6] [Fig.4.18] confirment l'ensemble de nos observations. Le schéma *star* sans les *bypasses* est plus rapide à construire. La différence est de l'ordre de quelques secondes. Le gain en temps de décision est beaucoup plus remarquable. Le processus de décision sur un schéma *star* sans *bypasses* est presque deux fois plus rapide. En effet, les règles de CAC sont vérifiées suivant une même logique (les règles (4.9), (4.10) et (4.11)). En termes de précision, la déformation est encore réduite. Les indicateurs de confiance et de sur-estimation enregistrés sont bien meilleurs dans le cas du schéma *star* sans les *bypasses*. Nous retenons aussi les mêmes observations sur la dépendance du temps de construction τ et de taux de confiance ϕ des propriétés d'échelle des topologies étudiées [Fig.4.17] [Fig.4.18].

TAB. 4.6: Performance du schéma star sans la définition des bypasses

Indicateurs	**Type de la modélisation star**	Exodus	Abovenet	AT&T	Sprintlink
τ (s)	avec bypasses	6.8492	11.4947	36.5618	52.2948
	sans bypasses	5.9825	10.0771	35.1214	50.3851
ε (μs)	avec bypasses	62	54	69	79
	sans bypasses	36	32	39	48
ϕ	avec bypasses	93.6212%	89.4890%	86.6348%	86.4374%
	sans bypasses	93.7005%	89.9550%	87.6810%	86.4560%
φ	avec bypasses	1.3917%	2.0084%	3.0013%	4.2533%
	sans bypasses	0.9868%	1.4824%	2.0956%	3.7759%

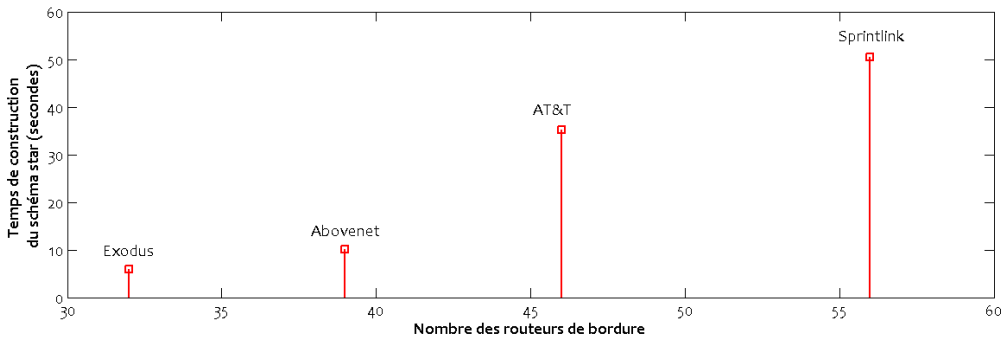

FIG. 4.17: Temps de construction du schéma star sans les Bypasses

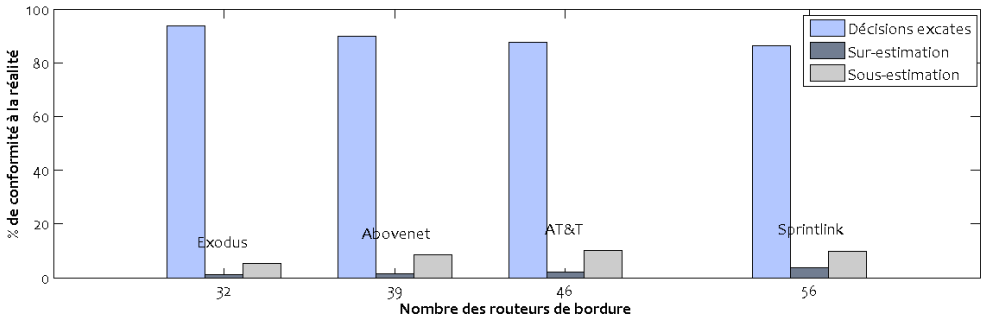

FIG. 4.18: Déformation introduite par le schéma star sans les Bypasses

Sur la base des bonnes performances que la définition des liens de congestion a introduit sur le schéma *full-mesh*, nous avons voulu appliquer les mêmes mécanismes sur le schéma *star* afin de voir s'il est possible d'apporter encore plus d'amélioration à ce schéma.

4.5.2 Mise en Place de Règles de Gestion de la Bande Passante

Si l'identification des liens de congestion est simple dans le schéma *full-mesh*, elle est beaucoup plus compliquée dans le cas de la représentation *star*. En effet, le schéma *star* est déjà construit à partir d'une représentation abstraite qui est le schéma *full-mesh*. Il n'y a pas une relation directe entre la modélisation détaillée de la topologie et la modélisation *star*. Donc, il est impossible d'identifier les liens physiques qui forment un *spoke*. Comme la recherche des liens de congestion passe obligatoirement par une connaissance très fine de la composition des chemins logiques, il nous a fallu trouver les mécanismes nécessaires permettant d'approximer un *spoke* par une succession de liens physiques. Pour cela, nous avons pensé à projeter le schéma *star* sur la topologie réelle et d'identifier par la suite les connexions qui correspondent le mieux à ce schéma *star* en termes de délai et de bande passante. L'idée alors, est de trouver dans la topologie réelle le routeur de coeur qui a les propriétés les plus proches de celles du noyau virtuel. Deux étapes sont nécessaires pour l'accomplissement de cette tâche : la première étape consiste à établir un schéma *star* à partir de chaque routeur de coeur. Il suffit d'établir, sur la base des informations de routage, l'ensemble des *spokes* entrants et sortants depuis ce routeur vers tous les routeurs de bordure. Ce schéma construit à partir de la topologie réelle sera appelé un schéma *star* physique. La deuxième étape permet de comparer le schéma *star* logique au schéma *star* physique. L'idée est de calculer, pour chaque représentation physique, la déviation des *spokes* logiques par rapport aux *spokes* physiques. Cette déviation élémentaire Δ_e est calculée sur les $2\times$Card(B) *spokes* du schéma *star*. La somme de toutes ces déviations donne lieu à une déviation globale Δ_g. Ainsi, c'est sur la base de Δ_g qu'un schéma *star* physique sera sélectionné. Il s'agit du schéma physique ayant une déviation globale minimale, c'est-à-dire le schéma physique le plus proche de la représentation abstraite. Soit $\Upsilon(r) = \Upsilon_{in}(r) \cup \Upsilon_{out}(r)$ l'ensemble de tous les *spokes* physiques, avec $\Upsilon_{in}(r)$ l'ensemble des *spokes* entrants au routeur de coeur *r* et $\Upsilon_{out}(r)$ l'ensemble des *spokes* sortants de *r*. $\mathcal{U}(\ell')$ est le *spoke* logique qui correspond au *spoke* physique

ℓ' dans le schéma *star* logique. La déviation globale Δ_g est calculée de la manière suivante :

$$\forall \ell' \in \Upsilon(r)$$

$$\Delta_e(\ell') = \sqrt{| \widehat{dl_{\ell'}} - \widehat{dl_{\mathcal{U}(\ell')}} |^2 + | \widehat{bw_{\ell'}} - \widehat{bw_{\mathcal{U}(\ell')}} |^2} \quad (4.14)$$

$$\Delta_g(r) = \sum_{b \in \mathtt{B},\ br \in \Upsilon_{in}(r)} (\Delta_e(br)) + \sum_{b \in \mathtt{B},\ rb \in \Upsilon_{out}(r)} (\Delta_e(rb)) \quad (4.15)$$

Il est à noter que la déviation élémentaire Δ_e entre deux *spokes* est calculée comme étant la distance entre deux points dans le plan cartésien $\mathcal{P}_{delay|bandwidth}$. L'algorithme ci-dessous [Algorithm.3] décrit en détail le processus de recherche du schéma *star* physique.

Algorithm 3 *Recherche du schéma star physique optimal*

```
C = /* L'ensemble des routeurs de coeur */
iter = 1 /* compteur d'itérations */
for r ∈ C do
    for b ∈ B do
        nd = nd + Δe(rb) + Δe(br)
    end for
    if iter = 1 then
        min-dev = nd
        computed-nucleus = r
    else
        if min-dev > nd then
            min-dev = nd
            computed-nucleus = r
        end if
        iter ++
    end if
end for
return r
```

Sur la base du schéma *star* physique, il est alors possible d'identifier les liens de congestion de la même façon que dans le cas du schéma *full-mesh*. La logique est de trouver sur chaque *spoke* physique le lien qui peut être l'objet d'une congestion maximale. Si dans le schéma *full-mesh* ce lien était déduit à partir de l'ensemble des chemins logiques, dans le schéma *star* le lien de congestion est identifié à partir de l'ensemble des *spokes* physiques. En effet, il s'agit du lien physique ayant la bande passante minimale et traversé par le plus grand nombre de *spokes* physiques. Il est à noter que le schéma *star* physique n'est utilisé que pour la détermination des liens de congestion. La fonction de contrôle d'admission est toujours appliquée sur le schéma *star* logique. En effet, le *star* logique offre une meilleure approximation de la topologie réelle. Il prend en compte tous les liens qui existent dans le réseau, contrairement au schéma *star* physique qui se restreint à un ensemble de liens.

Comme dans le cas du schéma *full-mesh* amélioré, le processus de contrôle d'admission sur le schéma *star* ne sera plus restreint à la vérification des trois règles de CAC identifiées précédemment (4.9) (4.10) (4.11). Deux nouvelles conditions de gestion de la bande passante sont rajoutées. Elles consistent à vérifier que la somme des réservations sur un *spoke* ainsi que sur l'ensemble des *spokes* qui le traversent en son lien de congestion est toujours inférieure à la

capacité totale du lien de congestion. Donc, pour une nouvelle connexion ψ, de $\imath$ vers $\jmath$, avec une demande en bande passante ω et une demande en délai ϱ, les équations supplémentaires à vérifier sont les suivantes :

$$\omega + \sum_{rsv(\imath n)} \widehat{bw} + \sum_{\ell' \in \theta_{\imath n}(bl)} \left(\sum_{rsv(\ell')} \widehat{bw} \right) \leq \widehat{bw}(bl^{\imath n}) \tag{4.16}$$

$$\omega + \sum_{rsv(n\jmath)} \widehat{bw} + \sum_{\ell' \in \theta_{n\jmath}(bl)} \left(\sum_{rsv(\ell')} \widehat{bw} \right) \leq \widehat{bw}(bl^{n\jmath}) \tag{4.17}$$

4.5.3 Emulation et Résultats

Comme dans le cas du schéma *full-mesh* amélioré, nous nous sommes basés dans nos émulations sur le même scénario de contrôle d'admission ainsi que sur les mêmes indicateurs d'évaluation. Le but est de montrer l'apport et les limites du schéma *star* amélioré par rapport au schéma *star* basique.

Le premier critère étudié est le temps de construction τ. Ce temps est normalement supérieur à celui du schéma *star* basique. En effet, la construction de la représentation améliorée passe, en plus de l'établissement des *spokes*, par la recherche d'un noyau physique représentatif dans le modèle détaillée de la topologie et par la suite par l'identification des liens de congestion. Ceci introduit un temps de traitement supplémentaire de l'ordre de quelques secondes. Ce temps dépend fortement du nombre des routeurs de bordure de la topologie étudiée [Fig.4.19]. Par exemple, dans le cas de la topologie AT&T l'écart du temps de traitement est de l'ordre de 33 secondes alors que dans le cas de la topologie Exodus, il est de l'ordre de 3 secondes [Tab.4.7].

Comme dans le cas du schéma *full-mesh*, la mise à jour du schéma *star* passe par l'identification de l'ensemble des *spokes* affectés par les évènements de re-routage. Deux cas de figure sont possibles : Dans le premier cas, si le routeur de coeur représentatif du noyau virtuel est touché par le re-routage, le schéma *star* doit être reconstruit en globalité. Dans le deuxième cas, les liens de congestion sont recalculés seulement pour les *spokes* affectés par le re-routage. Le temps de mise à jour dépend directement du nombre de *spokes* à recalculer. Le temps moyen enregistré est de l'ordre de quelques secondes.

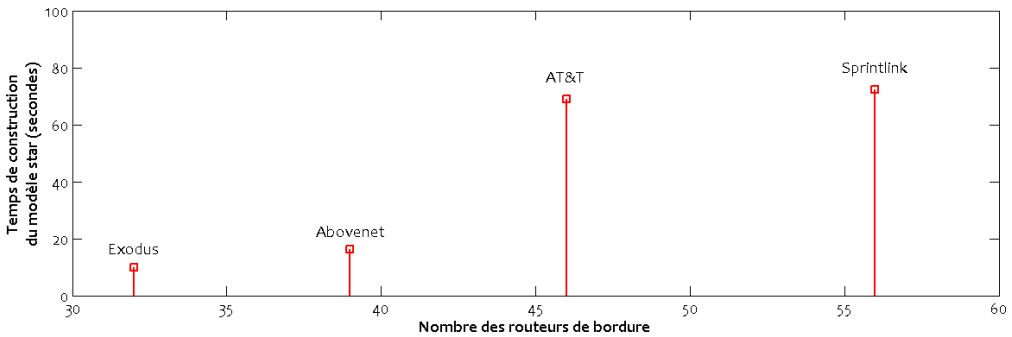

FIG. 4.19: Temps de construction du schéma star amélioré

En termes de temps de décision, le schéma *star* amélioré passe plus de temps lors du processus de décision. Ceci s'explique par le fait qu'il y a deux conditions supplémentaires à vérifier (une par *spoke*). Cette différence de temps de décision est très minime, elle est de l'ordre de quelques microsecondes [Tab.4.7]. Par exemple dans le cas de la topologie Sprintlink, elle est de l'ordre de 2 microsecondes.

TAB. 4.7: Performances du schéma star amélioré

Indicateurs	**Type de la modélisation star**	Exodus	Abovenet	AT&T	Sprintlink
τ (s)	star sans bypasses	6.9825	11.0771	36.1214	52.3851
	star amélioré	9.88609	16.2124	69.0736	72.168
ε (μs)	star sans bypasses	36	32	39	48
	star amélioré	38	33	40	50
ϕ	star sans bypasses	93.7005%	89.9550%	87.6810%	86.4560%
	star amélioré	93.7953%	90.4598%	88.7948%	87.3978%
φ	star sans bypasses	0.9868%	1.4824%	2.0956%	3.7759%
	star amélioré	0.4873%	0.3197%	0.3232%	0.2952%

Sur la base des paramètres τ et ε, nous avons voulu étudier les performances de la reconstruction en entier du schéma *star* sur le processus d'admission d'appel. Même si le temps de construction du schéma *star* est supérieur à celui du schéma *full-mesh*, nous estimons que la rapidité des décisions peut améliorer encore les performances de la fonction de CAC. En effet, si nous reprenons l'exemple de la topologie Sprintlink (section 4.4.2), les résultats d'émulations montrent que le rapport de réduction du temps de décision par rapport à un modèle classique (modèle détaillé de la topologie) est de l'ordre de 30 [Tab.4.7]. Donc, le GPL sera capable de gérer 60 demandes de connexion par seconde sur la base de la représentation *star* améliorée. Supposons que suite à un évènement de re-routage, 500 demandes de connexion soient en attente de traitement. Pour un schéma classique de Bandwidth Broker, ces 500 demandes de connexion nécessitent un temps de traitement de 100 secondes. Il s'agit de la durée de la phase de blockage Δk. Dans le cas d'un contrôle d'admission basé sur un schéma *star* amélioré, Δk est significativement réduite. Les 500 demandes de connexion sont traitées en moins de 9 secondes. Donc, même après une reconstruction entière du modèle abstrait de la topologie, le GPL sera capable de traiter de nouvelles demandes de connexion au bout de 80 secondes (72 secondes de phase de re-construction + 8 secondes de phase de blockage). Nous enregistrons un écart égal à 20 secondes entre la représentation *star* et la représentation détaillée. Ceci montre bien l'efficacité et la rapidité du principe d'agrégation de topologie, même dans le cas où la représentation abstraite *star* est reconstruite entièrement après le re-routage.

Sur la base de très bons résultats enregistrés dans le cas du schéma *full-mesh* amélioré, l'objectif derrière la définition de contraintes de gestion de la bande passante est de réduire au maximum la déformation introduite lors du processus d'agrégation. Ceci a été confirmé dans le cas du schéma *star* amélioré. En effet, nous avons observé une réduction remarquable du taux de la sur-estimation [Fig.4.20]. Le rapport de réduction peut atteindre des valeurs élevées suivant la nature de la topologie étudiée. Par exemple dans le cas de la topologie Sprintlink, le rapport de réduction est égal à 12. Sur toutes les topologies étudiées, φ est toujours inférieur à 0.5% [Tab.4.7], ce qui représente une excellente valeur vue nos objectifs de garantie de service. Nous avons aussi enregistré une augmentation de l'indicateur de confiance. Même si cette augmentation n'est pas très remarquable comme dans le cas du schéma *full-mesh* amélioré, ceci influe directement sur les performances du GPL qui sera capable de minimiser son taux de fausses

décisions. Les observations sur la dépendance du taux de confiance des propriétés d'échelle de la topologie sont retenues [Fig.4.20].

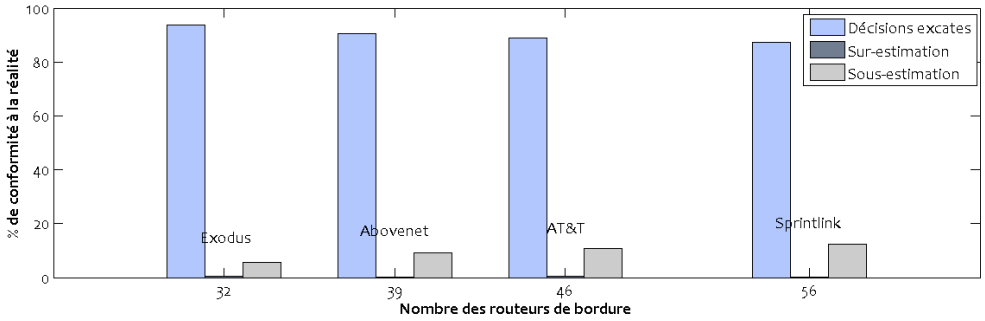

FIG. 4.20: Déformation introduite par le schéma star amélioré

4.6 Intégration des Paramètres de Gigue et de Taux de Perte

Jusqu'à présent, nous avons étudié la fonction de contrôle d'admission dans un contexte de service unique, c'est-à-dire dans le cas où les applications réseaux ne sont sensibles qu'aux paramètres de délai et de bande passante. Dans cette section, nous étendons notre étude au contexte multi-services pour couvrir les paramètres de gigue et de taux de perte. Pour la définition de ce contexte multi-services, nous nous sommes basés sur les spécifications du RFC 4594 [94]. L'idée est d'appliquer la fonction de contrôle d'admission sur une représentation abstraite de la topologie tout en tenant compte des exigences diverses des applications réseaux en termes de paramètres de qualité de service.

Dans les schémas *full-mesh* et *star* décris précédemment, les chemins logiques sont pondérés par le couple $(\widehat{dl}, \widehat{bw})$, où $\widehat{dl}$ et $\widehat{bw}$ représentent respectivement les capacités du chemin logique en termes de délai et de bande passante. Cette représentation est insuffisante, même inadaptée, pour accomplir l'admission d'appel de certaines applications non sensibles seulement aux paramètres de délai et de bande passante. Ainsi, en prenant compte de l'ensemble des paramètres de QoS spécifiés dans le RFC 4594, un chemin logique sera représenté par un quadruplet $(\widehat{dl}, \widehat{lr}, \widehat{jt}, \widehat{bw})$. Ce quadruplet identifie respectivement les capacités du chemin de données en termes de délai, de taux de perte, de gigue et de bande passante. Le calcul du quadruplet passe nécessairement par l'identification des types de paramètres pris en compte. En effet, la connaissance du type d'un paramètre de QoS (additif, concave ou multiplicatif) permet d'estimer sa valeur sur une représentation logique de bout en bout.

Dans les deux sections suivantes, nous décrivons la manière dont les chemins logique et les *spokes* seront calculés.

4.6.1 Construction du Schéma Full-mesh

Même dans un contexte multi-services, le schéma *full-mesh* reste facile à établir. Il suffit d'appliquer les définitions 1, 2 et 3 sur l'ensemble des chemins de données afin de déduire une représentation abstraite de bout en bout de la topologie.

Les deux paramètres intégrés, la gigue et le taux de perte, suivent deux lois différentes. Le paramètre de gigue est additif. Il est calculé de la même manière que le délai (Définition 1).

Il suffit d'appliquer la somme sur les métriques des liens physiques pour retrouver la métrique globale du chemin logique. En ce qui concerne le paramètre de taux de perte, il est multiplicatif. Il est calculé de la façon suivante :

$$\widehat{lr}(\mathtt{L}) = 1 - \prod_{i=1}^{n}(1 - \widehat{lr}(\ell_i)),\ avec\ \mathtt{L} = \{\ell_i / i = 1, \ldots, n\}$$

Ainsi, suivant les types de paramètres de qualité de service, un quadruplet est calculé par chemin logique.

4.6.2 Construction du Schéma Star

La recherche du quadruplet ($\widehat{dl}$, $\widehat{lr}$, $\widehat{jt}$, $\widehat{bw}$) est plus compliquée dans le cas du schéma *star*. En effet, le calcul des *spokes* défini dans la section 4.2.2 est directement issu des propriétés intrinsèques des paramètres de délai et de bande passante. Comme le délai est un paramètre additif, l'intégration du paramètre de gigue ne pose pas un problème particulier puisqu'il est aussi additif. Il sera déduit des mêmes observations que le paramètre de délai. Donc, le calcul de la gigue sur les *spokes* entrants et sortants du noyau sera fait de la façon suivante :

$$\forall \imath \in \mathtt{B},\ \widehat{jt}(\imath n^s) = \min(\widehat{jt}(\imath \jmath^m)),\ \jmath \in \mathtt{B},\ \jmath \neq \imath \tag{4.18}$$

$$\forall \jmath \in \mathtt{B},\ \widehat{jt}(n\jmath^s) = \max(\widehat{jt}(\imath \jmath^m) - \widehat{jt}(\imath n^s)),\ \imath \in \mathtt{B},\ \imath \neq \jmath \tag{4.19}$$

Contrairement à la gigue, le paramètre de taux de perte suit une loi multiplicative. Il est impossible d'utiliser l'ensemble des équations définies précédemment. Comme il n'est pas simple de se baser sur les propriétés intrinsèques de ce paramètre pour le calcul des *spokes*, nous avons opté pour sa transformation en un paramètre additif par l'application du logarithme décimal. De cette façon, il sera possible d'appliquer les formulations définies dans le cas des paramètres additifs. Le paramètre du taux de perte est transformé en paramètre additif de la façon suivante :

$\widehat{lr}(\mathtt{L}) = 1 - \prod_{i=1}^{n}(1 - \widehat{lr}(l_i)),\ avec\ \mathtt{L} = \{\ell_i / i = 1, \ldots, n\}.$

$1 - \widehat{lr}(\mathtt{L}) = \prod_{i=1}^{n}(1 - \widehat{lr}(l_i)).$

Le log décimal est appliqué pour transformer le produit en somme :

$\ln(1 - \widehat{lr}(\mathtt{L})) = \ln(\prod_{i=1}^{n}(1 - \widehat{lr}(l_i))).$

Par analogie à la *Définition 1*, l'expression $-\ln(1 - \widehat{lr})$ suit une loi additive. Cette expression est toujours positive du fait que le logarithme décimal est négatif sur l'intervalle [0,1]. Ainsi, il est possible d'appliquer les règles de recherche des *spokes* définis dans le cas du délai et de la gigue, et de déduire par la suite la valeur associée au taux de perte (4.18) (4.19).

$$
\begin{array}{rcl}
\forall \imath \in \mathtt{B},\ \jmath \in \mathtt{B},\ \jmath \neq \imath & & \\
\ln(1-\widehat{lr}(\imath n^s)) & = & -\min(-\ln(1-\widehat{lr}(\imath \jmath^m))) \\
\exp(\ln(1-\widehat{lr}(\imath n^s))) & = & \exp(-\min(-\ln(1-\widehat{lr}(\imath \jmath^m)))) \\
\widehat{lr}(\imath n^s) & = & 1-\exp(-\min(-\ln(1-\widehat{lr}(\imath \jmath^m)))) \\
\widehat{lr}(\imath n^s) & = & 1-\dfrac{1}{\exp(\min(-\ln(1-\widehat{lr}(\imath \jmath^m))))}
\end{array} \tag{4.20}
$$

$$
\begin{array}{rcl}
\forall \jmath \in \mathtt{B},\ \imath \in \mathtt{B},\ \imath \neq \jmath & & \\
\ln(1-\widehat{lr}(n\jmath^s)) & = & -\max(\ln(1-\widehat{lr}(\imath n^s))-\ln(1-\widehat{lr}(\imath \jmath^m))) \\
\exp(\ln(1-\widehat{lr}(n\jmath^s))) & = & \exp(-\max(\ln(1-\widehat{lr}(\imath n^s))-\ln(1-\widehat{lr}(\imath \jmath^m)))) \\
\widehat{lr}(n\jmath^s) & = & 1-\exp(-\max(\ln(1-\widehat{lr}(\imath n^s))-\ln(1-\widehat{lr}(\imath \jmath^m)))) \\
 & = & 1-\dfrac{1}{\exp(\max(\ln(1-\widehat{lr}(\imath n^s))-\ln(1-\widehat{lr}(\imath \jmath^m))))}
\end{array} \tag{4.21}
$$

Une fois que le schéma *star* est construit, un routeur de coeur est sélectionné [Algorithm 3] afin de pouvoir fixer les liens de congestion sur chaque *spoke*. La même logique, utilisée dans le cas des réseaux mono-service, est appliquée pour la recherche du noyau représentatif. Les déviations élémentaires Δ_e et globales Δ_g sont calculées suivant les quatre paramètres de QoS : le délai, la gigue, le taux de perte et la bande passante. Elles sont formulées de la façon suivante :

$$
\begin{array}{rcl}
\forall \ell' \in \Upsilon(r) & & \\
\Delta_e(\ell') & = & \sqrt{\mid \widehat{dl_{\ell'}} - \widehat{dl_{\mathcal{U}(\ell')}} \mid^2 + \mid \widehat{jt_{\ell'}} - \widehat{jt_{\mathcal{U}(\ell')}} \mid^2 + \mid \widehat{lr_{\ell'}} - \widehat{lr_{\mathcal{U}(\ell')}} \mid^2 + \mid \widehat{bw_{\ell'}} - \widehat{bw_{\mathcal{U}(\ell')}} \mid^2} \\
\Delta_g(r) & = & \displaystyle\sum_{b \in \mathtt{B},\ br \in \Upsilon_{in}(r)} (\Delta_e(br)) + \sum_{b \in \mathtt{B},\ rb \in \Upsilon_{out}(r)} (\Delta_e(rb))
\end{array}
$$

Ainsi, sur la base de la représentation physique du schéma *star*, il est possible d'identifier les liens de congestion avant d'entamer la phase de contrôle d'admission.

4.6.3 Les Règles de CAC

Comme le processus de CAC dépend fortement des exigences des applications en termes de QoS, nous avons identifié sur la base des spécifications du RFC 4594 [Tab.4.8], un ensemble de Méta-Classes de services regroupant les applications ayant droit au même processus décisionnel. Selon le type et le nombre des paramètres de QoS à examiner, un traitement est appliqué par Méta-Classe de service. Dans notre mode de fonctionnement, nous avons identifié trois types de Méta-Classes [Tab.4.9] :

- La Méta-Classe *dljs* regroupe l'ensemble des applications sensibles aux paramètres de délai, de gigue et de taux de perte,
- La Méta-Classe *dls* regroupe l'ensemble des applications sensibles aux paramètres de délai et de taux de perte,
- La Méta-Classe *bs* regroupe l'ensemble des applications élastiques. Aucune condition, sur les paramètres de délai, de taux de perte et de gigue, n'est vérifiée.

TAB. 4.8: RFC 4594 : sensibilité des applications réseaux aux paramètres de QoS

Classe de service	**Sensibilité délai**	**Sensibilité taux de perte**	**Sensibilité gigue**
Fonctions de contrôle	✓	✓	-
Téléphonie	✓	✓	✓
Signalisation	✓	✓	-
Multimedia conferencing	✓	✓	✓
Temps réel intéractive	✓	✓	✓
Streaming multimedia	✓	✓	-
Broadcast vidéo	✓	✓	✓
Données à faible latence	✓	✓	-
OAM	✓	✓	-
Standard	-	-	-
Données à faible priorité	-	-	-

TAB. 4.9: Les trois Méta-Classes de service

Comportement STAMP	**CoSs IETF**
dljs	Téléphonie Multimedia conferencing Temps réel intéractive Broadcast vidéo
dls	Fonctions de contrôle Signalisation Streaming multimedia Données à faible latence OAM Haut débit
bs	Standard Données à faible priorité

Sur la base de cette notion de Méta-Classes, nous étudions dans les deux paragraphes suivants la façon avec laquelle la fonction de contrôle d'admission est exécutée dans le cas des deux schémas *full-mesh* et *star*.

Schéma Full-mesh

Comme dans le contexte mono-service, l'admission d'une nouvelle demande de connexion passe nécessairement par la vérification des exigences de cette demande en termes de paramètres de QoS vis-à-vis de la capacité du chemin logique en question. Ces exigences ne sont pas les mêmes dans toutes les Méta-Classes de service. Elles dépendent du nombre de paramètres de QoS à examiner. Par exemple, la Méta-Classe *dljs* implique beaucoup plus de conditions à vérifier que la Méta-Classe *bs*. Par conséquent, trois scénarios du processus décisionnel sont identifiés.

Le premier scénario concerne la Méta-Classe *dljs*. Il s'agit du scénario le plus exigent en qualité de service. Tous les paramètres spécifiés dans le RFC 4594 doivent être examinés. En effet, sur un chemin logique de bout en bout, l'admissibilité d'une nouvelle connexion est vérifiée par rapport aux paramètres de délai, de gigue, de taux de perte et de bande passante. Il est à noter

que la vérification des contraintes de gestion de la bande passante est faite de la même manière que dans le cas du contexte mono-service.

Donc, l'admissibilité d'une nouvelle connexion ψ, de $\imath$ vers $\jmath$, est vérifiée de la façon suivante. Étant donné ω la demande en bande passante, ϱ la demande en délai, μ la demande en gigue et λ la demande en taux de perte, cinq tests sont nécessaires afin de décider de l'acceptation ou non de ψ. Les quatre premiers tests vérifient l'état de la demande par rapport à ce que le chemin logique $\imath\jmath$ peut offrir en termes de délai (4.22), de gigue (4.24), de taux de perte (4.23) et de bande passante (4.25). La cinquième condition vérifie l'admissibilité de ψ vis-à-vis de la contrainte de gestion de la bande passante (4.26).

$$\varrho \geq \widehat{dl}(\imath\jmath^m) \tag{4.22}$$

$$\lambda \geq \widehat{lr}(\imath\jmath^m) \tag{4.23}$$

$$\mu \geq \widehat{jt}(\imath\jmath^m) \tag{4.24}$$

$$\omega + \sum_{rsv(\imath\jmath)} \widehat{bw} \leq \widehat{bw}(\imath\jmath^m) \tag{4.25}$$

$$\omega + \sum_{rsv(\imath\jmath)} \widehat{bw} + \sum_{\ell' \in \theta_{\imath\jmath}(bl)} \left(\sum_{rsv(\ell')} \widehat{bw} \right) \leq \widehat{bw}(bl^{\imath\jmath}) \tag{4.26}$$

Le deuxième scénario décisionnel est plus simple. Il concerne la Méta-Classe *dls*. Seuls les paramètres de délai et de taux de perte sont examinés. Le paramètre de gigue n'est pas pris en compte. Ainsi, pour décider de l'admissibilité d'une nouvelle demande de connexion, quatre conditions sont à vérifier (4.22), (4.23), (4.25) et (4.26).

Enfin, le scénario le moins contraignant en termes de paramètres de QoS concerne la Méta-Classe *bs*. Deux conditions sont à vérifier : la disponibilité de bande passante sur le chemin logique $\imath\jmath$ (4.25) ainsi que la condition de congestion (4.26). Aucune condition sur les paramètres de délai, de gigue et de taux de perte n'est nécessaire.

Ainsi, c'est la tâche du GPL d'identifier le type d'une nouvelle demande de connexion, et de la traiter par la suite suivant le processus décisionnel de la Méta-Classe correspondante.

Schéma Star

De la même manière que dans le cas du schéma *full-mesh*, le processus décisionnel dans le schéma *star* dépend fortement de la nature de la Méta-Classe sollicitée. Selon le nombre de paramètres de QoS, les *spokes* seront examinés afin de décider de l'acceptation ou non d'une nouvelle demande de connexion. Aussi, trois scénarios du processus décisionnel sont identifiés.

Le premier scénario concerne la Méta-Classe *dljs*. L'admissibilité d'une nouvelle connexion ψ de $\imath$ vers $\jmath$ est vérifiée en deux temps par rapport aux paramètres de délai, de gigue, de taux de perte et de bande passante. Il s'agit de comparer l'état de la demande par rapport aux capacités des deux *spokes* reliant $\imath$ à $\jmath$. Étant donné ω la demande en bande passante, ϱ la demande en délai, μ la demande en gigue et λ la demande en taux de perte, sept tests sont nécessaires afin de décider de l'acceptation ou non de ψ. L'état des réservations sur les deux *spokes* est examiné via les deux équations (4.30) et (4.31). L'admissibilité de ψ en termes de délai, de gigue et de

taux de perte est vérifiée respectivement par les équations (4.27), (4.28) et (4.29). Et enfin, les équations (4.32) et (4.33) assurent la non congestion des *spokes* $\imath n$ et $n\jmath$.

$$\varrho \geq \widehat{dl}(\imath n^s) + \widehat{dl}(n\jmath^s) \tag{4.27}$$

$$\mu \geq \widehat{jt}(\imath n^s) + \widehat{jt}(n\jmath^s) \tag{4.28}$$

$$\ln(1-\lambda) \leq \ln(1-\widehat{lr}(\imath n^s)) + \ln(1-\widehat{lr}(n\jmath^s)) \tag{4.29}$$

$$\omega + \sum_{rsv(\imath n)} \widehat{bw} \leq \widehat{bw}(\imath n^s) \tag{4.30}$$

$$\omega + \sum_{rsv(n\jmath)} \widehat{bw} \leq \widehat{bw}(n\jmath^s) \tag{4.31}$$

$$\omega + \sum_{rsv(\imath n)} \widehat{bw} + \sum_{\ell' \in \theta_{\imath n}(bl)} \left(\sum_{rsv(\ell')} \widehat{bw} \right) \leq \widehat{bw}(bl^{\imath n}) \tag{4.32}$$

$$\omega + \sum_{rsv(n\jmath)} \widehat{bw} + \sum_{\ell' \in \theta_{n\jmath}(bl)} \left(\sum_{rsv(\ell')} \widehat{bw} \right) \leq \widehat{bw}(bl^{n\jmath}) \tag{4.33}$$

Dans le deuxième scénario, celui de la Méta-Classe *dls*, seuls les équations (4.27), (4.29), (4.30), (4.31), (4.32) et (4.33) sont vérifiées. En effet, le paramètre de gigue n'est pas pris en considération lors du processus décisionnel.

Enfin, seul le paramètre de bande passante est examiné dans le cas de la Méta-Classe *bs*. L'état des réservations ainsi que les contraintes de gestion de la bande passante sont vérifiés sur les deux *spokes* via les équations (4.30), (4.31), (4.32) et (4.33).

4.6.4 Emulation et Résultats

Les topologies définies précédemment ont été utilisées pour l'évaluation des performances des deux schémas d'agrégation dans un contexte multi-services. Sur la base des spécifications du RFC 4594, nous nous sommes basés sur les caractéristiques de QoS de cinq types d'applications. Ceci est important pour l'étude du comportement de chacune des Méta-Classes lors de la phase d'admission d'appel. Les applications prises en considération sont les suivantes :

- le peer-to-peer avec une sensibilité *bs*,
- le Web avec une sensibilité *bs*,
- la VoD avec une sensibilité *dls*,
- FTP avec une sensibilité *dls*,
- VoIP avec une sensibilité *dljs*.

Pour se rapprocher de la réalité, nous avons calculé, sur un site de production de France Télécom, le nombre moyen de connexions établies par type d'application pendant une période déterminée. Ceci permet d'avoir une estimation proche de la réalité de la matrice de trafic.

Les résultats enregistrés, avec l'intégration des paramètres de gigue et de taux de perte,

montrent de très bonnes performances dans le cas des deux schémas d'agrégation *full-mesh* et *star*. Ils sont presque du même ordre que ceux enregistrés dans le cas du contexte mono-service.

En ce qui concerne le paramètre τ, deux paramètres additionnels de QoS sont pris en considération lors de la construction des modèles abstraits de la topologie. Par conséquent, le temps de formation des schémas agrégés doit être supérieur à celui calculé dans le cas des réseaux mono-service. Il s'agit de calculer pour chaque chemin logique et/ou *spoke* le taux de perte et la gigue dont il est capable de supporter. Sur la base des résultats de nos émulations, il s'avère que le temps de traitement introduit par ces deux paramètres n'est pas important. Il est de l'ordre de quelques secondes. Le temps global τ enregistré est de l'ordre de quelques dizaines de secondes [Tab.4.10]. Et il est supérieur dans le cas du schéma *star*. Aussi, les mêmes observations concernant la dépendance entre le temps de construction τ et le nombre de routeurs de bordure se sont confirmées [Fig.4.21].

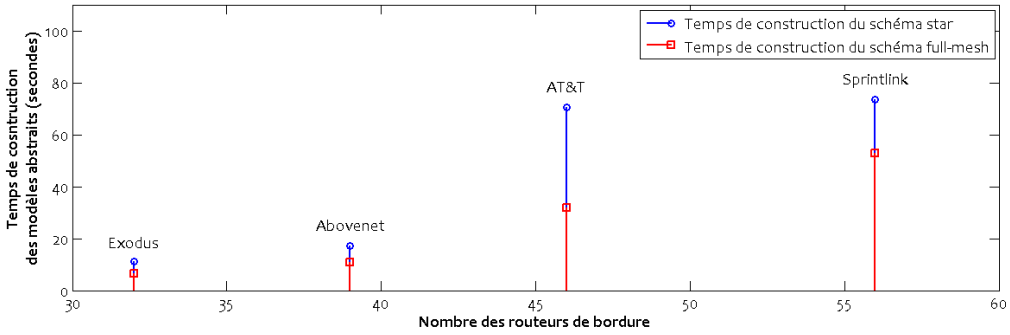

FIG. 4.21: Contexte multi-services : Temps de construction des schémas abstraits

Le temps de décision ε est variable en fonction de la Méta-Classe sollicitée et des paramètres de QoS examinés. Plus l'application réseau est sensible à la QoS, plus le nombre de conditions à vérifier est important et plus le temps de décision est élevé. Par conséquent, les Méta-Classes *dljs*, *dls* et *bs* auront respectivement un temps de décision décroissant. Ces observations ont été confirmées par les résultats d'émulations. Le temps de décision est de l'ordre de dizaines à quelques centaines de microsecondes. La plus grande valeur est enregistrée dans le cas de la Méta-Classe *dljs* et la plus petite dans le cas de la Méta-Classe *bs* [Fig.4.23] [Fig.4.22]. Par exemple, dans la représentation *full-mesh* de la topologie Sprintlink, ε est égal à 2056 μs pour la Méta-Classe *dljs*, à 477 μs pour la Méta-Classe *dls* et à 92 μs pour la Méta-Classe *bs*. La différence avec un schéma classique basé sur une représentation détaillée de la topologie est significative. Le rapport de réduction enregistré est très important. Il dépend des propriétés d'échelle de la topologie étudiée. Par exemple, sur la topologie Exodus, le rapport de réduction est de l'ordre de 2 dans le cas du schéma *full-mesh* et de l'ordre de 9 dans le cas du schéma *star*. Il est à noter que, sur le plan décisionnel, le schéma *star* est irréprochable par la rapidité de ses traitements. Le GPL est capable de traiter jusqu'à 26 fois plus de demandes de connexion qu'un Bandwidth Broker classique.

En termes de performances, nous avons montré jusqu'ici les bons résultats enregistrés suite à l'intégration des paramètres de gigue et de taux de perte. Il reste à vérifier la tolérance du taux de déformation que le processus d'agrégation, à quatre paramètres de QoS, introduit. Pour cela, nous nous sommes basés sur l'indicateur de confiance ϕ et l'indicateur de sur-estimation φ. Les résultats d'émulations montrent de bonnes performances dans le cas des deux schémas d'agrégation. La pire valeur de ϕ sur une représentation *full-mesh* est égale à 90%, et à 84% sur

TAB. 4.10: Etude du temps de décision (μs) en fonction de la Méta-Classe de service

modèle de la topologie	**Méta-Classe**	Exodus	Abovenet	AT&T	Sprintlink
full-mesh	dljs	689	1001	1374	2056
	dls	158	233	318	477
	bs	29	45	61	92
star	dljs	140	170	197	242
	dls	35	43	50	60
	bs	8	10	11	14
détaillée	dljs	1339	2240	3688	6140
	dls	332	559	922	1564
	bs	79	136	227	385

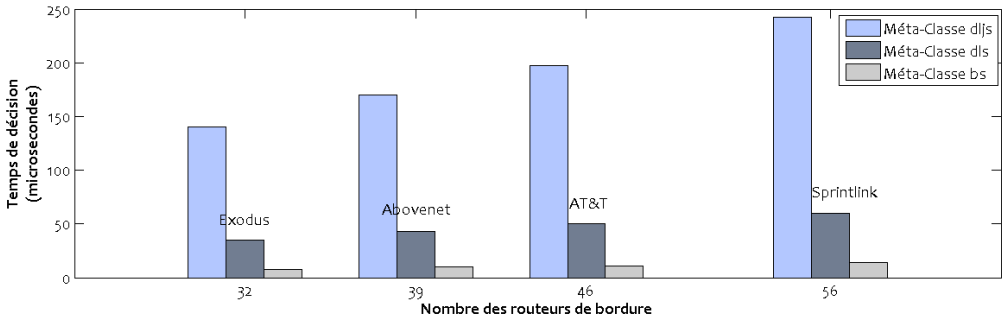

FIG. 4.22: Contexte multi-services : Temps de décision pour le schéma star

une représentation *star*. Ces résultats ne s'éloignent pas trop des performances enregistrées dans le contexte mono-service où seuls les paramètres de délai et de bande passante ont été étudiés. De même pour l'indicateur de sur-estimation φ, les résultats sont bons. Ils sont meilleurs dans le cas du schéma *star* (toujours $< 2\%$) que dans le cas du schéma *full-mesh* (toujours $< 4\%$).

TAB. 4.11: Contexte multi-services : performances des deux schémas d'agrégation

Indicateurs	**modèle de la topologie**	Exodus	Abovenet	AT&T	Sprintlink
τ (s)	full-mesh	6.43068	10.8609	31.6875	52.7781
	star	11.1135	17.0497	70.6338	73.635
ϕ	full-mesh	95.9976%	95.4371%	92.5736%	90.6254%
	star	91.9556%	89.1704%	86.7798%	84.2761%
φ	full-mesh	3.1313%	2.9555%	3.5619%	3.2708%
	star	1.0549%	1.6486%	0.9122%	1.3064%

Ainsi, sur la base de l'ensemble de ces critères d'évaluation, nous remarquons que l'intégration des paramètres de gigue et de taux de perte n'a pas influé sur les performances des deux schémas d'agrégation, les résultats enregistrés étant du même ordre.

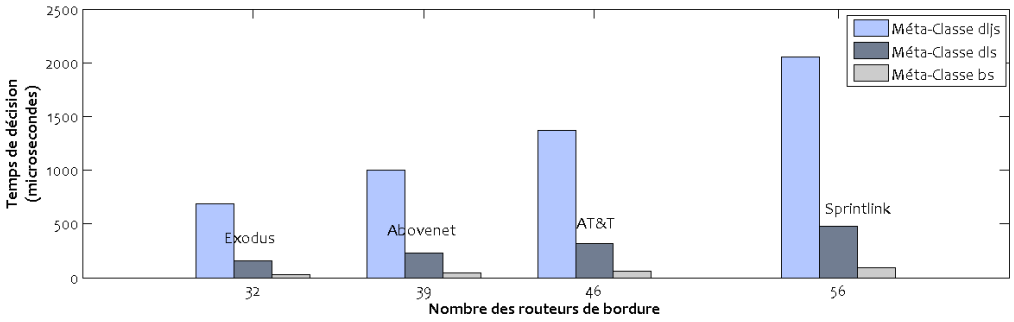

FIG. 4.23: Contexte multi-services : Temps de décision pour le schéma full-mesh

4.7 Prise en Compte du Routage Multi-chemins

Jusqu'à présent, nous avons étudié le problème d'agrégation sous un seul angle de vue : l'agrégation de noeud. En effet, nous supposons qu'il existe toujours un seul chemin de données reliant deux noeuds de bordure dans un domaine réseau. Même si cette supposition est vérifiée dans la majorité des cas, il existe des domaines où le routage multi-chemins[22] est activé. Ainsi, il a fallu prendre en considération ce cas de figure afin de compléter notre modèle conceptuel d'agrégation de topologie.

L'idée est de trouver les mécanismes nécessaires permettant d'identifier un ensemble de routes reliant un couple de noeuds de bordure par une représentation simple et facile à manipuler. L'état de l'art décrit dans le chapitre 1 apporte quelques réponses sur la manière dont il faut aborder un problème d'agrégation de liens. L'évaluation de l'ensemble des solutions proposées a montré les bonnes performances de l'approche *PolyLine*. Cette approche consiste à établir, dans un plan cartésien à deux dimensions (délai/bande passante) $\mathcal{P}_{delay,bandwidth}$, une fonction escalier définie à partir des points représentatifs[23] de l'ensemble des routes reliant un couple de noeuds de bordure [Algorithm.4] [Algorithm.5]. La fonction escalier est par la suite approximée via des segments établis entre une sélection de points représentatifs. C'est sur la base de cette sélection de points qu'un chemin logique sera par la suite pondéré en paramètre de QoS.

[22]Au lieu de ne proposer qu'un seul chemin, le routage multi-chemins propose, pour chaque destination, plusieurs chemins possibles, pondérés selon leur qualité de service. Cela permet de répartir la charge du réseau. Il permet également d'augmenter la tolérance aux pannes, à condition que les différents chemins proposés soient disjoints. ECMP (Equal Cost MultiPaths Protocol) [95] est un exemple de protocole de routage multi-chemins.

[23]Il s'agit de l'ensemble des points sélectionnés suivant les Définitions 4 et 5 du Chapitre 1. Ils sont représentés dans le plan cartésien $\mathcal{P}_{delay,bandwidth}$ par le couple ($\widehat{dl}$,$\widehat{bw}$)

Algorithm 4 *Recherche des points représentatifs*

```
1: /* V : Ensemble de chemins logiques possibles entre deux noeuds de bordure */
2: /* Sort() : fonction de tri */
3: /* handledNodes : Nombre d'itérations pour le parcours de V */
4: /* Trier V suivant les valeurs ascendantes du délai (dl) */
5: Sort(V)
6: for (dl^,bw^) ∈ V do
7:     handledNodes = 0
8:     for (dl'^,bw'^) ∈ V do
9:         if (dl^ < dl'^ ou bw^ > bw'^) et ((dl^,bw^) != (dl'^,bw'^)) then
10:            handledNodes++
11:        end if
12:    end for
13:    if handledNodes == Card(V)-1 then
14:        /* Définition 5 vérifiée pour tous les éléments de V */
15:        R.push((dl^,bw^))
16:    end if
17: end for
```

Algorithm 5 *Construction de la fonction échelle*

```
1: /* W : Ensemble de points définissant la fonction Escalier */
2: for k de 0 à R.size() do
3:     /* chercher les points convexes */
4:     dl^ = R[k].dl^
5:     bw^ = R[k].bw^
6:     W.push ((dl^,bw^))
7:     /* chercher les points concaves */
8:     if (k != (R.size() - 1)) then
9:         dl^ = R[k+1].dl^
10:        bw^ = R[k].bw^
11:        W.push((dl^,bw^))
12:    end if
13: end for
```

D'un point de vue fonctionnel, la logique d'agrégation adoptée par le *PolyLine* est très réaliste. Son seul point faible réside dans la manière avec laquelle les points représentatifs sont sélectionnés pour la construction de la fonction d'approximation. En effet, le *PolyLine* manipule de façon similaire les points concaves et convexes de la fonction escalier. Le point ayant la déformation minime, par rapport à une fonction d'approximation établie dans une étape précédente, est sélectionné pour faire partie de la nouvelle fonction d'approximation. Après un nombre k d'itérations, la fonction d'approximation finale sera représentée par un ensemble de segments reliant les différents points concaves et convexes sélectionnés. Comme les points concaves sont les plus proches de la zone admissible, notre logique était d'accorder un traitement prioritaire à ce type de point par rapport aux points convexes. En effet, plus il y a des points convexes, plus la fonction d'approximation contient des zones non admissibles, ce qui conduit à une surestimation des capacités du réseau. L'inverse est aussi vrai. Plus il y a des points concaves, plus il y a une sous-estimation des capacités du réseau. L'objectif est de couvrir le plus de zones

admissibles et d'écarter le plus de zones non admissibles afin d'améliorer la garantie de service [Fig.4.24]. Pour atteindre cet objectif, Il a fallu définir un ensemble de règles pour la sélection des points représentatifs. Ceci permettra de ne pas se restreindre qu'aux points concaves pour la construction de la fonction d'approximation.

4.7.1 La Méthode WPA

Dans notre mode de fonctionnement, nous avons défini un facteur de garantie *GF* (Guarantee Factor). Son rôle est de refléter l'allure générale de la concavité de la fonction escalier. En effet, cette concavité est calculée par rapport au point ayant les valeurs optimales de paramètres de qualité de service (point BP : Best Point [Fig.4.24]). Dans notre espace de travail $\mathcal{P}_{delay,bandwidth}$, ce point représente la valeur maximale de la bande passante et la valeur minimale de délai. Nous identifions deux points déterminants pour le calcul de la concavité de la fonction escalier : le point convexe le plus proche du point BP et le point concave le plus éloigné du point BP. Sur la base des distances qui séparent ces deux points par rapport au point BP, un indicateur de concavité est calculé. Il s'agit de l'indicateur *GF*. Il représente le rapport entre ces deux distances. Sa formulation est la suivante :

$$GF = \lfloor \frac{\max_{cv|BP}}{\min_{cx|BP}} \rfloor + 1$$

$\max_{cv|BP}$ est la distance maximale qui sépare le point BP des points concaves de la fonction escalier, et $\min_{cx|BP}$ est la distance minimale qui sépare ce point par rapport aux points convexes (A l'exception des deux points d'extrémité).

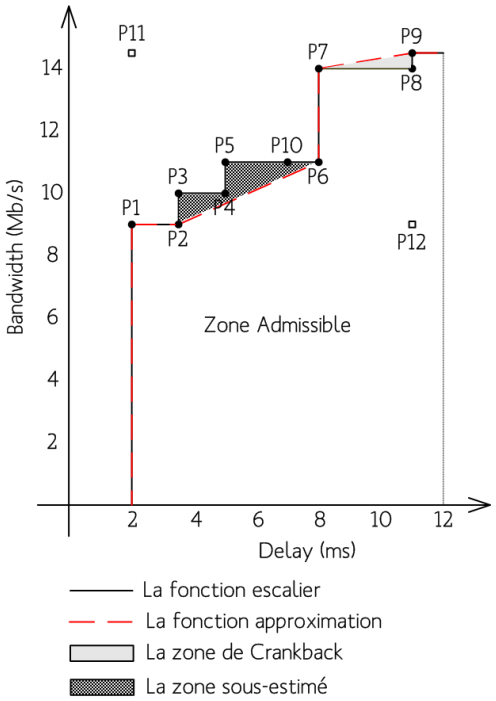

FIG. 4.24: Représentation géométrique des indicateurs d'une fonction d'approximation

Par l'intégration du facteur GF, la démarche d'approximation n'est plus la même. Ce facteur

sera déterminant dans la sélection des points représentatifs candidats à la formation de la fonction d'approximation. En effet, les points convexes et les points concaves ne seront plus traités avec le même ordre de priorité. Il s'agit d'un algorithme d'approximation pondéré par l'indicateur *GF*. En d'autres termes, un PolyLine pondéré ou *WPA* : Weighted Ploy line Approach. Dans cette approche, la construction de la fonction d'approximation passe par les étapes suivantes : Étant donné $\mathtt{W}_{\mathcal{Z}} = \{p_i,\ i = 1, \ldots, n\}$, l'ensemble des points concaves et convexes de la fonction escalier $\mathcal{Z}$, seulement k points sont sélectionnés pour la construction de la fonction d'approximation (k $\leq$ n-2). Le choix de la valeur de k est laissé à l'administrateur réseau. Cette valeur doit être définie soigneusement afin de garantir la pertinence du processus d'agrégation. En effet, si k est petit, un chemin logique sera présenté par un nombre minimal de couples de paramètres de QoS. Par exemple, si k est égal à 1, la fonction d'approximation sera représentée par deux segments reliant le seul point sélectionné aux deux points d'extrémité. Donc, la métrique du lien logique sera identifiée par trois couples de paramètres de QoS : ceux des deux points d'extrémité et du point sélectionné. Le choix d'un k petit accélère les traitements, mais introduit en contre partie une grande déformation par rapport aux valeurs réelles de la topologie. Si k est grand, un chemin logique sera définit par un nombre maximal de couples de paramètres de QoS. En d'autres termes, la fonction d'approximation sera représentée par plusieurs points. Ceci permettra de la rapprocher des caractéristiques de la fonction escalier et par la suite des capacités réelles du chemin de données. Il est à noter que la fonction d'approximation est identique à la fonction escalier dans le cas où k est égal à n-2.

Une fois le paramètre k défini, une première fonction d'approximation est établie. Il s'agit du segment qui relie les deux points d'extrémité p_1 et p_n. Si k > 0, un point est sélectionné pour être le premier point représentatif de la fonction d'approximation. Ce point est choisi sur la base d'une métrique d. Cette métrique n'est pas formulée de la même manière dans le cas des points concaves et des points convexes. Pour les points concaves, d est calculée à partir de la distance $\mathcal{D}$ qui sépare ce point au segment d'approximation le plus proche. Elle est notée d_{cv}. Par exemple, la distance d'un point x_1 sera calculée par rapport au segment d'approximation définit par les points d'extrémités EP_1 et EP_2, où l'abscisse de x_1 est compris entre les deux abscisses de EP_1 et EP_2. En ce qui concerne les points convexes, d est calculée à partir du rapport de la distance qui sépare le point candidat au segment d'approximation le plus proche sur le facteur *GF*. Cette métrique est notée d_{cx}. La formulation des deux métriques d_{cv} et d_{cx} est la suivante :

$$\mathrm{d}_{cx}(point\ candidat) = \frac{\mathcal{D}_{point\ candidat|Segment\ d'approximation}}{GF}$$

$$\mathrm{d}_{cv}(point\ candidat) = \mathcal{D}_{point\ candidat|Segment\ d'approximation}$$

Sur la base de ces métriques, le point ayant la plus grande distance par rapport à la fonction d'approximation est sélectionné pour faire partie de la fonction d'approximation. Comme le montrent les formulations d_{cv} et d_{cx}, les points concaves sont favorisés par rapport aux points convexes puisque leur métrique n'est pas divisée par le facteur GF. De cette façon, la probabilité d'avoir des points concaves dans la fonction d'approximation est plus grande. Les points convexes peuvent aussi être sélectionnés mais avec une probabilité inférieure. Il suffit que ces points soient proches de la concavité de la courbe escalier pour avoir potentiellement une métrique d_{cx} supérieure à d_{cv}. Ceci est très important afin de ne pas trop sous estimer les capacités du réseau. Par exemple, dans la figure 4.24, si le segment p_6-p_9 est sélectionné à la place du segment p_7-p_9, la zone des services admissibles sous-estimés devient très grande.

Une fois le premier point d'approximation sélectionné, k-1 itérations sont nécessaires pour la construction de la fonction d'approximation globale. Si lors de la première itération les mé-

triques d_{cv} et d_{cx} ont été calculées par rapport au segment d'approximation initial. Dans les itérations suivantes, les points candidats doivent vérifier la condition suivante vis-à-vis du segment de calcul [*Définition 7*].

Définition 7. Étant donné $\mathcal{S}$ un segment d'approximation défini par les deux points d'extrémité e_1 et e_2, p_i un point candidat, et la notation x_p désigne l'abscisse d'un point p : $\mathcal{S}$ est dit segment de calcul pour le point p_i, si $x_{p_i} \in [x_{e_1}, x_{e_2}]$.

A chaque fois, un point est choisi sur la base de ses métriques vis-à-vis des segments d'approximation. Ce point est par la suite relié à l'ensemble des points déjà sélectionnés afin de construire une fonction d'approximation formée par (k+1) segments. Donc, un chemin logique sera identifié par (k+2) couples de paramètres de QoS : k couples désignant les points représentatifs de la fonction d'approximation et deux couples pour les deux points d'extrémité. Par exemple dans la figure 4.25, avec un k = 2, le chemin logique L est approximé par trois segments : le segment (3,2)-(4,2), le segment (4,2)-(5,4) et le segment (5,4)-(5,8). Donc, la métrique associée à L est représentée par les quatre couples de paramètres de QoS (3,2)-(4,2)-(5,4)-(5,8). L'algorithme ci-dessous [Algorithm 6] décrit en détails les différentes étapes de l'approche *WPA*.

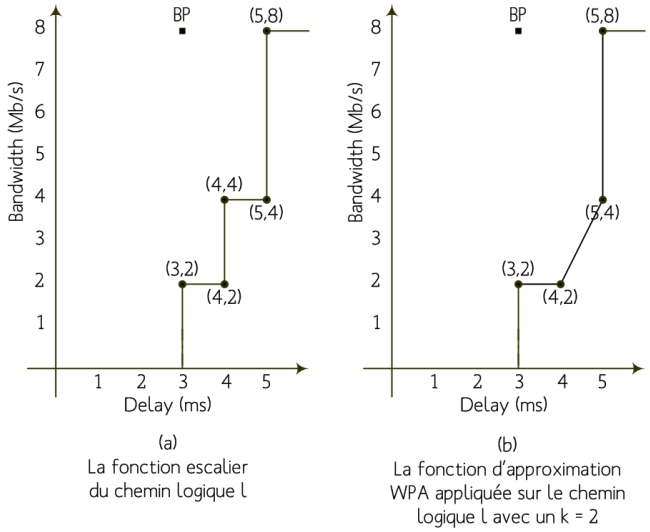

FIG. 4.25: Exemple de construction de la fonction d'approximation WPA

Algorithm 6 *Algorithme WPA*

```
/* E : Ensemble de segments formant la fonction d'approximation WPA */
/* p2p : une fonction qui calcule la distance entre deux points */
for k de 0 à (W.size() - 1)/2 do
    concave-distance = p2p(W[2×k+1],BP)
    if concave-distance > max-distance then
        max-distance = concave-distance
    end if
end for
/* calcul de la distance entre les points convexes et le point BP */
for i de 0 à ((W.size()-1)/2)-1 do
    convexe-distance = p2p(W[2×i],BP)
    if convexe-distance < min-distance then
        min-distance = convexe-distance
    end if
end for
/* Calcul du facteur GF */
GF = max−distance / min−distance
/* Approximation de la fonction escalier */
S.lp = point d'extrémité le plus bas de la fonction escalier
S.up = point d'extrémité le plus haut de la fonction escalier
E.push(S)
/* Appeler la fonction récursive findApproximationPoints */
findApproximationPoints(E, kpoints, GF)

procedure FINDAPPROXIMATIONPOINTS(E,m,GF)
    repeat
        for S ∈ E do
            distance = D((dl^,bw^),S)
            if (dl^,bw^) est convexe then
                distance = distance / GF
            end if
            /* choisir le segment le plus distant */
            if distance > max-distance then
                max-distance = distance
                S' = S
            end if
        end for
        /* Éclater le segment S' en deux segments à partir du point (dl^,bw^) sélectionné */
        /* Insérer les deux segments calculés dans E */
        m–
        findApproximationPoints(E,m,GF)
    until m == 0
end procedure
```

4.7.2 Evaluation du WPA

Afin de comparer les performances de notre fonction d'approximation *WPA* par rapport au *PolyLine*, nous avons conduit quelques émulations sur des petites topologies générées par l'utilitaire GT-ITM [96]. Les topologies en question suivent une loi Waxman [97]. Elles ont une étendue de 20 à 50 noeuds, où 10% seulement des noeuds sont sélectionnés comme bordures. Ce choix de topologies de petite taille a pour but d'étudier finement les performances de la fonction d'approximation *WPA*. En effet, nous avons défini deux indicateurs d'évaluation : un indicateur de cranckback κ et un indicateur d'admissibilité ϑ. L'indicateur de cranckback représente le nombre moyen de requêtes non réalisables. Il s'agit du cas où la fonction d'approximation se met en erreur lors du processus de décision. Dans le plan cartésien $\mathcal{P}_{delay,bandwidth}$, cet indicateur est défini par l'ensemble des points appartenant à la zone admissible de la fonction d'approximation mais pas à celle de la fonction escalier. Ces points représentent d'éventuelles requêtes admissibles par la fonction d'approximation (à l'intérieur de la concavité de la courbe d'approximation), mais non réalisables (à l'extérieur de la concavité de la fonction escalier). Par exemple, dans la figure 4.24, la surface de cranckback est représentée par le triangle p_7-p_8-p_9. Le deuxième indicateur est l'indicateur d'admissibilité. Il représente le taux de requêtes admissibles et réalisables. Contrairement à la zone de cranckback, la zone d'admissibilité est définie par la surface commune entre la fonction escalier et la fonction d'approximation *WPA*. Dans la figure 4.24, cette zone correspond à la surface p_1-p_2-p_6-p_7-p_8-p_9.

Les deux indicateurs κ et ϑ sont formulés de la façon suivante :

$$\kappa = \frac{1}{\mathcal{A}^2_{Card(B)}} \times \sum_{b_i,b_j \in \mathtt{B}} \mathcal{Z}(b_i, bj)$$

$$\vartheta = \frac{\mathcal{G}_{WPA} \times 100}{\mathcal{G}_{Staircase}}$$

avec,

- $\mathcal{G}$ dénote le taux moyen d'admissibilité dans la représentation escalier. Il est égal à $\frac{1}{\mathcal{A}^2_{Card(\mathtt{B})}} \times \sum_{b_i,b_j \in \mathtt{B}} \mathcal{Z}(b_i, bj)$. Son calcul dépend de la valeur de $\mathcal{Z}(b_i, bj)$,
- $\mathcal{Z}(b_i, bj)$ (Crankbacked Areas) représente la surface de la zone non admissible (Escalier ou WPA) de tous les chemins logiques entre b_i et b_j,
- $\mathcal{A}^2_{Card(\mathtt{B})} = \frac{Card(\mathtt{B})!}{(Card(\mathtt{B})-2)!}$ représente le nombre de routes possibles entre les noeuds de bordure dans l'intra-domaine.

Les émulations conduites sur l'ensemble des topologies de test ont montré la supériorité de l'algorithme *WPA* sur le *PolyLine* en termes de garantie de service. La figure 4.26 illustre le taux de cranckback enregistré dans les deux approches. Il s'avère que κ est significativement réduit avec le *WPA*. Pour des petites topologies, Le *WPA* garantit un taux de cranckback presque nul. La croissance de l'indicateur κ en fonction du nombre des noeuds est légère avec l'algorithme *WPA*, alors qu'elle est élevée avec l'algorithme *PolyLine*. En outre, la réduction de la zone de cranckback enregistrée avec le *WPA* n'est pas accompagnée par une importante sous-estimation des capacités du réseau. La figure 4.27 montre que la zone admissible *WPA* est très proche de ce que le réseau peut réellement garantir. Le pourcentage d'admissibilité est toujours supérieur à 90%.

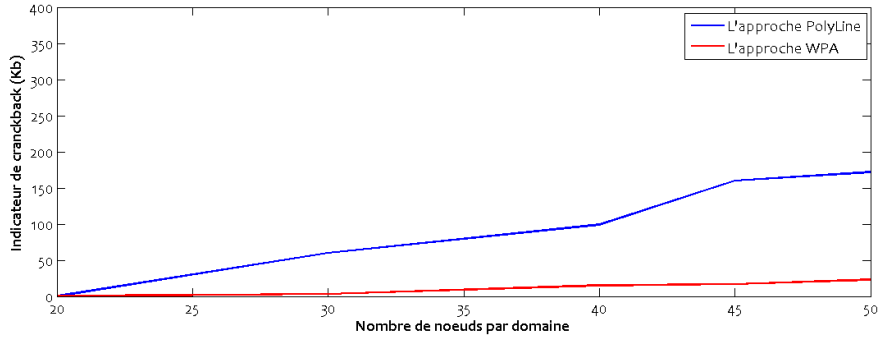

FIG. 4.26: Taux de cranckback introduit par l'approche WPA

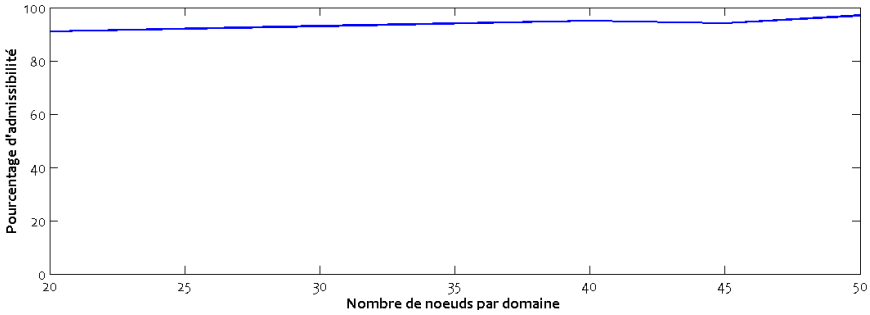

FIG. 4.27: Taux d'admissibilité de l'approche WPA

Sur la base de ces bons résultats, nous avons voulu intégrer les mécanismes d'agrégation de liens dans la spécification de la fonction de contrôle d'admission. Pour des raisons de simplicité, nous nous restreindrons dans notre étude au schéma abstrait *full-mesh*. La construction de cette représentation est très simple. Il suffit d'identifier pour chaque chemin logique la fonction d'approximation qui lui correspond, en d'autres termes, l'ensemble des k couples $(\widehat{dl},\widehat{bw})$.

4.7.3 Contrôle d'Admission et Règles de CAC

La logique de contrôle d'admission dans un contexte de routage multi-chemins n'est pas identique à celle appliquée sur des réseaux à route unique. Les opérations d'addition et de soustraction ne sont plus compatibles avec la nature de la représentation des paramètres de délai et de bande passante. En effet, la fonction d'approximation est une concaténation de segments contigus, définie par un ensemble de couples (délai, bande passante). Donc, la logique de vérification de l'admissibilité d'une requête doit nécessairement se différencier de la méthode classique.

Contrairement à la logique de contrôle d'admission présentée dans les sections précédentes, le routage multi-chemins a sa propre logique. Pour décider de l'acceptation ou non d'une nouvelle demande de connexion, le GPL doit localiser la demande par rapport à la courbe d'approximation du chemin logique sur lequel la demande devra être acheminée. Si la demande (le point $(\widehat{dl},\widehat{bw})$) est à l'intérieur de la concavité de la courbe, la requête est acceptée et les ressources sont réservées. Cette réservation correspond à une translation de la courbe d'approximation vers le bas de la valeur de bande passante demandée. Inversement la libération de ressources cor-

respond à une translation vers le haut de la valeur de bande passante libérée. Dans le cas où la demande est à l'extérieur de la concavité de la courbe, la requête est refusée.

4.7.4 Emulation et Résultats

Pour tester les performances du modèle WPA appliqué à la fonction de contrôle d'admission, nous nous sommes basé sur les mêmes topologies de test que celles citées précédemment (section 4.3.3). Le WPA a été configuré avec un paramètre k égal à 2. Nous avons utilisé le même scénario de CAC (section 4.3.3). Aussi, les mêmes indicateurs ont été mesurés afin d'analyser les performances de ce schéma d'agrégation multi-chemins. Il est à noter que nous avons autorisé jusqu'à quatre routes entre un couple de noeuds de bordure.

Dans un contexte de routage multi-chemins, la construction du schéma *full-mesh* passe par l'identification de la fonction d'approximation pour chacun des chemins logiques. Sur la base des résultats d'émulation, nous remarquons que ce temps de construction τ est de l'ordre de quelques dizaines de secondes. Comme dans un schéma de routage classique, la complexité du schéma *full-mesh* découle directement du nombre de routeurs déployés sur la bordure du domaine. Plus le nombre est grand, plus le temps de calcul de la fonction d'approximation pour chacun des chemins logiques est élevé [Fig.4.28].

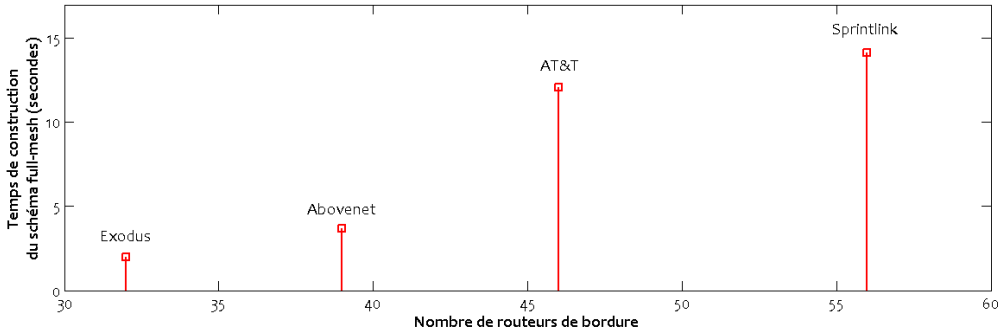

FIG. 4.28: Routage multi-chemins : Temps de construction du schéma full-mesh

En ce qui concerne le temps de décision ε, l'utilisation d'une fonction d'approximation par chemin logique allège le processus de décision. Le GPL n'a plus besoin de manipuler simultanément n routes pour décider de l'admissibilité d'une requête d'un point source i vers un routeur destination j. Il se base sur une représentation agrégée des paramètres de qualité de service qu'un chemin logique peut garantir. Cette observation a été confirmée par les résultats des émulations. Le temps de décision dans la représentation agrégée est au moins quatre fois inférieur à celui de la représentation réelle.

En termes de précision, nous ne nous éloignons pas beaucoup des bonnes performances enregistrées précédemment [Fig.4.29]. L'indicateur de confiance ϕ est acceptable. Il est toujours supérieur à 80%. L'indicateur de sur-estimation φ garde une valeur minime. Il est au plus égal à 0.1652%. Nous confirmons aussi les mêmes observations sur la dépendance qui existe entre le taux de confiance ϕ et le nombre des routeurs de bordure des topologies étudiées [Fig.4.29].

TAB. 4.12: Routage multi-chemins : performances du schéma full-mesh

Indicateurs	**modèle de la topologie**	Exodus	Abovenet	AT&T	Sprintlink
τ (s)	full-mesh multi-chemins	1.98326	3.67279	12.0576	14.1453
ε (μs)	full-mesh multi-chemins	109	162	226	338
	réel	494	1077	2394	3840
ϕ	full-mesh multi-chemins	93.5416%	88.8935%	84.6489%	79.4694%
φ	full-mesh multi-chemins	0.1652%	0.1031%	0.0174%	0.0738%

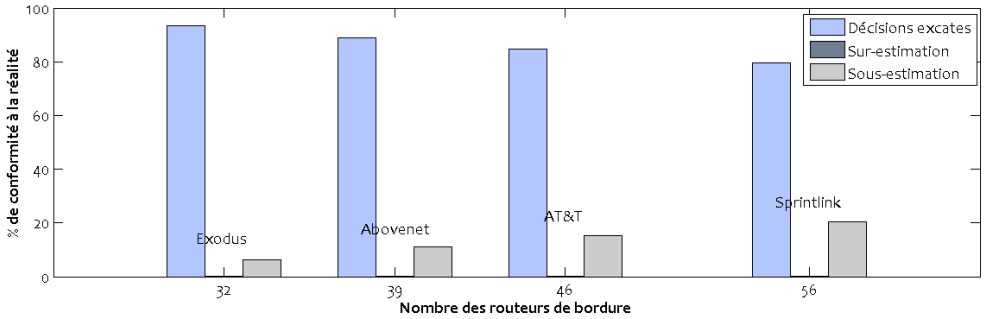

FIG. 4.29: Routage multi-chemins : Déformation introduite par l'approche WPA

4.8 Bilan

Dans ce chapitre nous avons commencé par analyser la façon avec laquelle le GPL accomplit la fonction de contrôle d'admission. En effet, sur la base de la représentation détaillée de la topologie *tdb*, le GPL est capable de décider de l'acceptation ou non d'une nouvelle demande de connexion. Il identifie la disponibilité des ressources tout au long du chemin de données par lequel devra passer le trafic en question. Le parcours de la *tdb* peut être dans certains cas coûteux en ressources CPU. Ceci est souvent vrai avec des topologies connexes de grande taille. Comme solution palliative, nous nous sommes basé sur le concept d'agrégation de topologie. Le principe est très simple : Il s'agit d'abstraire la topologie réelle d'un domaine réseau en un schéma plus simple et plus compacte où seuls les équipements de bordure sont représentés.

Deux schémas ont été proposés : un schéma *full-mesh* et un schéma *star*. Le schéma *full-mesh* consiste à établir des chemins logiques de bout en bout entre les différents noeuds de bordure. La représentation résultante est alors une topologie abstraite complètement maillée. Le deuxième schéma est le schéma *star*. Il a pour but de réduire la complexité spatiale du schéma *full-mesh*. Les éléments de bordure sont connectés à un noyau central virtuel (*nucleus*) jouant le rôle de relais. Ainsi, le nombre de chemins logiques dans la topologie agrégée est significativement diminué. Il n'est plus nécessaire d'établir un chemin logique vers chaque noeud de bordure, il suffit de passer par le noyau virtuel pour joindre n'importe quelle destination.

L'application de la fonction de contrôle d'admission sur ces deux schémas a montré une amélioration du temps de traitement et de décision des requêtes de demande de connexion. En contre parti, une dégradation de la qualité de service est enregistrée. Il s'agit de la déformation causée par le processus d'agrégation. Cette déformation est très importante dans le cas du schéma *full-mesh* et acceptable dans le cas du schéma *star*. Afin d'apporter plus de garantie de service, nous avons ré-étudié la définition des deux schémas *full-mesh* et *star*. Le but était d'introduire des indicateurs de congestion permettant d'estimer le taux de charge des liens dans

le réseau. Il s'agit d'identifier, sur chaque chemin logique, le lien physique qui peut être l'objet d'une congestion maximale. Si l'identification de cet indicateur de congestion est simple dans le cas du schéma *full-mesh*, elle est beaucoup plus complexe dans le cas du schéma *star*. En effet, le schéma *star* n'est pas construit à partir de la topologie réelle. Il est impossible d'identifier l'ensemble des liens physiques qui forment un *spoke* et par la suite de déterminer le lien de congestion. Afin de résoudre ce problème, la logique que nous avons adoptée est la suivante : Nous projetons le schéma *star* sur la topologie réelle pour trouver le routeur de coeur qui vérifie au mieux les caractéristiques du noyau virtuel. Cette projection est jugée efficace suivant un facteur de déviation Δ_g. Le routeur de coeur ayant la plus petite déviation est élu comme nucleus virtuel. Ainsi, le schéma *star* modifié sera représenté par l'ensemble des *spokes* reliant le routeur élu aux différents routeurs de bordure.

L'application de la fonction de contrôle d'admission sur les deux schémas d'agrégation révisés montre de très bonnes performances ainsi qu'une amélioration du niveau de garantie de service. Les taux de confiance et de sur-estimation sont remarquablement améliorés dans le cas du schéma *full-mesh*. De telles performances montrent bien l'efficacité des mécanismes d'agrégation lors d'un passage à l'échelle, à savoir un temps de traitement réduit et une garantie de service très acceptable.

Comme les conditions de déploiement réel ne sont pas restreintes seulement aux paramètres de délai et de bande passante, nous avons intégré les paramètres de gigue et de taux de perte. Sur la base des spécifications du RFC 4594, trois Méta-Classes ont été définies. Chacune de ces Méta-Classes identifie un niveau de sensibilité vis-à-vis des paramètres de qualité de service. Ainsi, suivant la Méta-Classe ciblée, le GPL exécutera des règles de CAC différentes. Les résultats d'émulation ont montré de très bonnes performances. Le gain en temps de décision est comparable à celui du contexte mono-service. Les indicateurs de confiance et de sur-estimation sont aussi dans la même marge des valeurs enregistrées précédemment.

Pour rendre plus générique nos schémas d'agrégation, nous avons aussi étudié le cas où le routage multi-chemins est activé dans le réseau. Ceci revient à utiliser conjointement les mécanismes d'agrégation de liens ainsi que les mécanismes d'agrégation de noeuds. En ce qui concerne l'agrégation de noeud, nous nous sommes basés sur le schéma le plus simple, le schéma *full-mesh*. Pour l'agrégation de liens, il a fallu trouver la manière avec laquelle il est possible de représenter l'ensemble des chemins physiques entre deux routeurs de bordure par un seul chemin logique de bout en bout. Pour cela, nous avons proposé une amélioration de l'approche *PolyLine* proposée dans [87]. Il s'agit de l'algorithme *WPA*. Cet algorithme consiste à agréger l'ensemble des métriques des différentes routes sélectionnées par une fonction d'approximation. Comparé au *PolyLine*, le *WPA* montre de très bonnes performances. Il est comparable en termes du niveau de service fournie, et est supérieur en garantie de service. Sur la base de ce mécanisme d'agrégation de liens, la construction du schéma *full-mesh* se fait par l'identification d'une fonction d'approximation par chemin logique. L'application de la fonction de contrôle d'admission sur ce schéma montre aussi de bonnes performances par rapport à un schéma classique qui ne supporte pas les mécanismes d'agrégation (le Bandwidth Broker doit gérer l'ensemble des routes possibles entre deux noeuds de bordure). Le temps de décision est remarquablement réduit et la déformation introduite est très acceptable.

Chapitre 5

Conclusion & Perspectives

Sommaire

Dans ce mémoire nous avons étudié la problématique de contrôle d'admission dans les réseaux à grande échelle. Ce contrôle d'admission n'est pertinent et efficace que s'il se base sur une connaissance fine, détaillée et à jour de la topologie du réseau. En effet, l'entité responsable de la fonction de CAC doit se référer à une image réelle de la topologie pour pouvoir prendre les bonnes décisions, de façon à ce qu'il n'y ait pas un déphasage entre ce qui existe dans la base de topologie et ce qui se passe réellement dans le réseau. Le but de cette thèse était de trouver les mécanismes nécessaires permettant d'assurer toutes ces fonctionnalités. La section suivante expose nos contributions pour la mise en oeuvre d'une solution efficace d'approvisionnement de la fonction de CAC par les données de topologie.

5.1 STAMP : Une Solution Efficace et Réactive pour la Découverte de Topologie

L'étude de l'ensemble des solutions de découverte de topologie proposées dans la littérature, nous a montré leur insuffisance, voir même leur inadéquation, pour répondre à notre problématique. En effet, l'un des critères les plus déterminants pour pouvoir juger de l'efficacité de la fonction d'acquisition de la topologie n'est pas rempli. Il s'agit du critère de réactivité. La plupart des solutions se basent sur des mécanismes de "probing" : ce n'est pas l'équipement qui annonce ses données de topologie, mais c'est plutôt une station de gestion qui l'interroge afin d'acquérir ces informations. Ceci présente un grand inconvénient sur la précision des données collectées ainsi que sur la notification des changements qui surviennent dans le réseau. Notre philosophie était alors d'inverser les rôles entre station de gestion et équipements et de doter ces derniers de capacités leur permettant d'annoncer de façon autonome leurs données de topologie. Ainsi, ce sont les équipements eux-mêmes qui contacteront la station de gestion afin de l'approvisionner en données de topologie. Pour mettre en oeuvre cette modélisation réactive, nous avons défini une architecture qui s'articule sur trois rôles : un GPL (Group Primary Leader) responsable de la collecte et de la manipulation/corrélation des données de topologie (il est l'équivalent de la station de station dans les modèles à base de probing), un GBL (Group Backup Leader) assurant la redondance du GPL si ce dernier tombe en panne, et un rôle de GM (Group member) attribué à tous les équipements qui appartiennent à la topologie. L'interaction entre toutes ces entités est accomplie via trois agents. Deux agents de niveau liaison permettent aux GMs de s'auto-découvrir : un agent Ethernet pour l'échange des informations d'adjacence au niveau 802.3 et un agent ATM responsable de l'échange des informations au niveau ATM. Le troisième agent de type TCP assure l'interaction entre GM/GBL et le GPL.

La logique de fonctionnement de STAMP peut être abstraite selon les étapes suivantes : a) les GMs adjacents échangent entre eux des messages Hello-Neighbors afin d'annoncer les caractéristiques techniques de leurs ports d'interconnexion physiques actifs, b) les données d'adjacence collectées par les GMs sont stockées en local dans une table d'adjacence (*adb*), c) les informations d'adjacence sont par la suite envoyées au GPL, pour traitement et corrélation, via les messages "Topology-Information", d) chaque fois que des changements surviennent dans le voisinage direct d'un GM ou sur ses caractéristiques techniques, des messages de mise à jour de la topologie sont envoyés au GPL.

L'ensemble des données reçues par le GPL est stocké au niveau d'une table de topologie *tdb*. C'est par le biais des informations contenues dans cette table que le GPL est capable de construire la cartographie de son domaine. Cette cartographie peut être de niveau 2 (couche liaison) ou de niveau 3 (couche réseau). En outre, cette cartographie fournie des informations

sur le plan de transfert. Elle décrit l'ensemble des VLAN, LSPs et routes configurés dans le domaine réseau afin de permettre à la fonction de CAC d'identifier le chemin par lequel passe le trafic.

Vu les fonctionnalités diverses que le GPL assure, des mécanismes de priorisation ont été mis en place afin d'alléger et rendre plus efficace le processus de construction de la topologie. En effet, les messages issus du GBL et des GMs sont bufferisés dans trois files FIFO (First In First Out) : une file de Haute priorité HP, une file de priorité moyenne MP et une file de faible priorité LP. Le choix du niveau de priorité des messages a été fait suivant les critères de sensibilité au délai de traitement et au contrôle de charge. Ce mécanisme de priorisation a montré son efficacité lors de la phase d'émulation. Toutes les files sont servies dans les plus bref délais, ce qui évite le risque de famine pour les files les moins prioritaires. De plus, les émulations ont montré la robustesse du GPL sous de fortes conditions de charge. Il est capable de traiter un très grand nombre de messages STAMP tout en ayant une charge CPU acceptable.

5.2 Vers une Fonction de CAC Scalable

Si l'admission d'appel au niveau du GPL passe par la vérification de la disponibilité de ressources sur le chemin emprunté par le trafic, il a fallu trouver les mécanismes nécessaires permettant d'éviter le parcours de la *tdb* dans le cas de très grandes topologies telles que celles des backbones IP. Pour résoudre ce problème, nous avons examiné la piste des mécanismes d'agrégation de topologie. En effet, notre logique a consisté à transformer une topologie réelle en une structure plus compacte dont la volumétrie d'information est réduite. Pour cela nous avons établi deux schémas d'agrégation, un schéma *full-mesh* et un schéma *star*. Dans le premier type de schéma, la topologie est représentée par un ensemble de chemins logiques interconnectant tous les routeurs de bordure entre eux. Dans le schéma *star*, les routeurs de bordures sont connectés entre eux par le biais d'un noyau virtuel faisant office de relais. Il est à noter que le schéma *full-mesh* est de complexité supérieure à celle du schéma *star* du fait qu'il contient beaucoup plus de chemins logiques.

Pour des raisons de simplicité, nous avons étudié tout d'abord le contexte mono-service QoS, avec comme seuls paramètres le délai et la bande passante. En ce qui concerne le schéma *full-mesh*, sa construction est très simple. Il est déduit à partir des informations de routage. Il suffit d'identifier l'ensemble des routes possibles entre les différents noeuds de bordure et d'appliquer par la suite les formulations adéquates pour le calcul des métriques. Le schéma *star* est plus compliqué à construire. En effet, il est indépendant de la topologie réelle, et est construit à partir du schéma *full-mesh* sur la base d'un ensemble de formules dérivant de propriétés intrinsèques des paramètres de délai et de bande passante.

Le processus de contrôle d'admission, sur chacun de ces schémas d'agrégation, est accompli conformément à un ensemble d'équations à vérifier. Elles permettent de décider si une nouvelle demande de connexion peut être admise ou pas vis-à-vis de la disponibilité des ressources en termes de délais et de bande passante.

Les émulations conduites sur les deux modèles abstraits basiques de topologie ont montré de bonnes performances. Le temps de décision lors du processus d'admission d'appel est significativement réduit. Cependant, sachant qu'un processus d'agrégation est toujours accompagné par une déformation, nous avons calculé le taux de cette déformation dans chacun des deux schémas. Il s'avère que le schéma *full-mesh* introduit une très grande sur-estimation des capacités du réseau. Ceci met souvent en erreur la fonction de CAC. Au contraire, la déformation introduite

par le schéma *star* est acceptable. Le taux de sur-estimation enregistré est réduit.

Pour améliorer les performances des deux schémas basiques d'agrégation, nous nous sommes focalisés sur l'étude de l'origine du taux important de la sur-estimation introduite. En effet, après le processus d'agrégation, la topologie du réseau ressemble à une boite noire. Il n'est plus possible d'identifier les capacités réelles des chemins logiques. Comme ces chemins se chevauchent dans la représentation réelle de la topologie, ceci rend difficile l'estimation de la capacité de chacun des chemins logiques suite à une réservation de ressources sur un autre chemin. Par conséquent, et afin d'apporter plus de réalisme dans notre procédé, nous avons mis en place un ensemble de règles de gestion de la bande passante. L'idée est de limiter au maximum le taux de décisions erronées dans la représentation abstraite de la topologie. Dans le schéma *full-mesh*, ces règles consistent à identifier sur chaque chemin logique le lien physique qui peut être l'objet d'une congestion maximale. Ce lien, nous l'avons défini comme étant le lien ayant la bande passante minimale et ayant parcouru par le plus grand nombre de chemins logiques. Dans le schéma *star*, la recherche des règles de gestion de la bande passante ne sont pas aussi triviales à trouver. En effet, ce schéma n'est pas déduit directement de la topologie réelle ce qui rend difficile l'identification des liens physiques de congestion. Pour se faire, il a fallu trouver une fonction de translation directe entre le schéma *star* et la topologie réelle. Dans notre mode de fonctionnement, nous avons projeté le schéma *star* sur la topologie réelle du réseau afin d'identifier le routeur de coeur qui se rapproche le plus des propriétés du nucleus virtuel. Ceci est accompli via un indicateur de déviation permettant de comparer le schéma *star* virtuel à d'autres schémas *star* construits directement à partir de la topologie réelle. Ainsi, une fois le schéma *star* rapproché de la topologie physique, il est possible de trouver les liens physiques de congestion. Ceci est accompli suivant la même logique que celle du schéma *full-mesh*. La validation des deux schémas d'agrégation améliorés montre de très bonnes performances. Le taux de décision exacte est significativement augmenté et le taux de sur-estimation des capacités du réseau est diminué. Les performances en termes de temps de décision sont toujours vérifiées.

Pour fournir un modèle d'agrégation complet, répondant aux spécifications du RFC 4594, nous avons intégré les paramètres de gigue et de taux de perte. Ceci permet de rajouter plus de contraintes sur la garantie de service pour la fonction de CAC. En effet, le RFC susmentionné a identifié la sensibilité des applications réseau aux paramètres de délai, de gigue et de taux de perte. Sur la base de ces paramètres, nous avons identifié trois Méta-classes de service. Chacune de ces classes représente une sensibilité par rapport à un ou plusieurs paramètres de QoS. Le processus de décision dépendra alors du type de l'application ainsi que de la Méta-classe sollicitée. Ceci a été démontré par l'ensemble des émulations. Le temps de décision est fortement dépendant du type de la Méta-classe de service. Il est élevé dans les Méta-Classes les plus contraignantes en termes de paramètres de QoS, ce qui est normal vu le nombre de règles de CAC à vérifier. En ce qui concerne les indicateurs de précision, il s'avère que les performances enregistrées dans le contexte multi-services sont comparables à celles enregistrées dans le contexte mono-service.

Tous les schémas d'agrégation que nous avons établis précédemment sont basés sur l'hypothèse que seul un chemin est possible entre deux noeuds de bordure. Même si cela est souvent vérifié, il n'est pas vrai dans les domaines réseau où des protocoles de routage multi-chemins sont utilisés. Pour couvrir ce cas de figure, nous avons proposé un mécanisme d'agrégation de liens permettant de représenter l'ensemble des routes possibles entre deux noeuds de bordure par un seul chemin logique. Ceci est très important afin de pouvoir accomplir la fonction de CAC de bout en bout et en une seule passe. Le mécanisme proposé est une amélioration du modèle PolyLine. Il se base sur une approximation des paramètres de QoS des différentes routes

par un ensemble de segments contigus identifiés suivant l'algorithme WPA (Weighted PolyLine Approach). Ce modèle d'agrégation de liens a été appliqué par la suite sur une représentation *full-mesh*. Son application à la fonction de contrôle d'admission a montré de bonnes performances par rapport à un modèle classique de load balancing (toutes les routes sont vérifiées pour décider de l'acceptation ou non d'une demande de connexion). D'un coté, le temps de décision est significativement réduit. De l'autre coté, le taux de déformation par rapport à la représentation réelle est acceptable.

En conclusion, l'architecture générale du modèle STAMP peut être considérée comme étant une architecture à trois niveaux [Fig.5.1]. Un niveau *liaison* permettant la découverte de voisinage dans le groupe. Un niveau *transport* assurant les fonctions d'authentification, de contrôle et de remontée des informations de topologie. Et enfin, un niveau *contrôle* regroupant les modules de construction de topologie, de contrôle d'admission et l'ensemble des algorithmes d'agrégation.

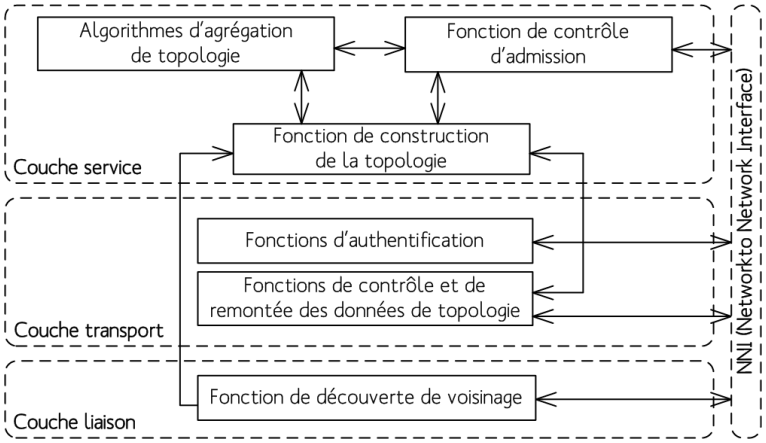

FIG. 5.1: Application de l'agrégation sur plusieurs niveaux hiérarchiques

Dans la section suivante, nous proposons quelques améliorations qui peuvent être apportées au protocole de signalisation STAMP. Aussi, nous exposons d'éventuelles améliorations et possibilités pour l'utilisation des schémas d'agrégation dans un contexte différent que celui du contrôle d'admission.

5.3 Perspectives

En ce qui concerne le protocole STAMP, vu son modèle de donnés extensible, plusieurs améliorations sont envisageables. Nous pouvons citer par exemple l'extension du schéma des encapsulations. Des technologies telles que VPN (Virtual Private Networks) et PPP (Point to Point) peuvent être rajoutées vu leur importance (elles sont souvent déployées par les opérateurs Télécom dans leurs réseaux).

Une autre perspective très intéressante pour le protocole STAMP est la standardisation. En effet, tel qu'il est défini, STAMP ne se base sur aucun standard IETF. L'idée est de garder la logique de fonctionnement STAMP et d'essayer de la mettre en oeuvre via des standards déjà existants. Une piste intéressante qui peut être examinée est l'utilisation du protocole IPfix (IP Flow Information Export) [98] pour la remontée des informations de topologie et de contrôle

entre GM et GPL. En effet, IPfix a été créé par le besoin d'avoir un protocole commun et universel pour l'exportation des informations des flux IP à partir des équipements réseaux (routeurs, sondes, etc.). Ce standard définit la façon avec laquelle les informations des flux sont formatées puis exportées d'un exporter [24] vers un collector [25]. Il est à noter que ce standard se base sur le protocole Netflow Version 9 de cisco. Les flux d'information transportés dans les messages IPfix sont formatés en *Template records* et en *data records*. Le premier type d'enregistrement définit la structure et la grammaire des données contenues dans le *data record* et le deuxième transporte les valeurs des éléments de données (Information Element) définis dans le *template record*. Ces éléments de données correspondent à des types définis dans le RFC 5102 [99]. Il est à noter que l'ensemble des éléments de données est extensible, c'est-à-dire, qu'il est possible de définir ses propres éléments de données. Ainsi, afin d'intégrer les mécanismes d'IPfix dans la logique de fonctionnement de STAMP, nous pouvons tout d'abord considérer que tous les GMs seront des exporters et que le GPL sera le collector. De cette façon, toutes les interactions entre GM et GPL seront formatées suivant le protocole IPfix. Les messages ainsi que le header STAMP doivent transiter dans des paquets IPfix. La grammaire de ces informations doit figurer suivant des éléments de données dans les *Template records*. Certaines informations, qui ne sont pas spécifiées dans le RFC (par exemple Hierarchical-Level-ID), seront définies par de nouveaux éléments de données (Information Elements). Les valeurs correspondantes à ces *Templates records* seront contenues dans les *data-records*. En ce qui concerne la communication entre GMs et entre GM/GPL lors de la phase de connexion, la logique est maintenue. C'est le protocole STAMP qui est responsable de l'authentification des GMs auprès du GPL et de la diffusion des informations de topologie dans le voisinage direct. En effet, IPfix, ne sera déployé que pour l'interaction entre GM et GPL pour l'échange des informations de topologie et de contrôle. De cette façon, une nouvelle modélisation du protocole STAMP est possible. Une phase de connexion et de découverte de voisinage effectuée suivant l'ancienne logique de fonctionnement et une phase d'annonce de données de topologie et de contrôle au GPL accomplie via le protocole IPfix.

Le modèle de données STAMP peut aussi être étendue par la prise en compte de la technologie POS (Packet Over SONET/SDH). Ceci passe par l'identification des différents niveaux d'encapsulation que supporte SONET/SDH par rapport aux technologies identifiées précédemment. Une piste éventuelle pour la découverte du voisinage SONET/SDH est l'utilisation du champ J0 (Section Trace Message) situé dans l'entête section (SOH : Section OverHead) de la trame SONET/SDH. En effet, J0 est analysé par saut d'un équipement SONET/SDH vers un autre. Ceci permet de découvrir le voisinage direct dans un environnement SONET/SDH.

Les mécanismes d'agrégation développés dans ce mémoire sont tous effectués au niveau hiérarchique 1. En effet, la topologie physique détaillée (niveau 0) est agrégée en une structure plus compacte au niveau hiérarchique juste supérieur (niveau 1). Ce modèle d'agrégation de base peut être étendu en un modèle plus général qui opère sur de multiples niveaux hiérarchiques. L'idée est d'agréger des domaines déjà agrégés et appartenant à un même niveau hiérarchique en une représentation plus abstraite. Ceci permet de réduire la volumétrie des informations de topologie et de rendre transparent la communication inter-domaine entre les domaines agrégés. Ce processus "d'agrégation d'agrégation" peut être défini par l'ensemble des étapes suivantes : a) chaque domaine réseau est agrégé en un noeud virtuel. Ce noeud virtuel n'est que la représentation abstraite du domaine réseau en un schéma *full-mesh* ou *star*, b) les noeuds virtuels résultants peuvent aussi être agrégés dans les niveaux hiérarchiques supérieurs suivant les règles d'agrégation définis dans le cadre des schémas *full-mesh* et *star*, c) Le degré

[24] L'équipement qui exporte les informations de flux
[25] L'équipement qui collecte les informations de flux

de granularité du processus d'agrégation est à régler par l'administrateur réseau. En effet, il faut préciser sur combien de niveaux hiérarchiques l'algorithme d'agrégation reste pertinent et n'introduit pas une trop grande déformation. Le choix du type du schéma d'agrégation à appliquer est aussi la tâche de l'administrateur réseau. Il faut noter la possibilité d'utiliser conjointement les représentations *full-mesh* et *star* dans un même niveau hiérarchique ou dans des niveaux hiérarchiques différents. Par exemple, il est possible d'utiliser des représentations *full-mesh* pour le niveau hiérarchique 1 et des représentations *star* pour le niveau 2, comme il est possible d'utiliser les représentations *full-mesh* et *star* pour représenter deux domaines réseau dans le niveau hiérarchique 1. Dans le cas de figure où les modèles abstraits représentent l'agrégat de l'agrégat, la fonction de contrôle d'admission est accomplie sur la représentation abstraite la plus haute dans la hiérarchie. Ceci réduit significativement la complexité de la fonction de contrôle d'admission. Il suffit de vérifier la disponibilité des ressources sur un ensemble de domaines réseau en une seule passe. Comme le montre la figure 5.2, quatre domaines réseaux sont interconnectés entre eux (GS1.0, GS2.0, GS3.0 et GS4.0). Ces domaines sont agrégés dans un premier temps dans le niveau hiérarchique 1, ce qui donne lieu aux quatre noeuds virtuels GS1.1, GS2.1, GS3.1 et GS4.1. Les liens inter-domaines seront maintenus afin d'assurer l'interconnexion des noeuds virtuels. Ces noeuds virtuels sont par la suite agrégés dans le niveau hiérarchique 2 en une seule structure compacte représentée par le noeud virtuel GS12. C'est sur ce noeud que va s'appliquer la fonction de CAC. En effet, le noeud virtuel GS1.2 n'est que la modélisation abstraite de l'agrégation de la topologie représentée par l'interconnexion des noeuds virtuels GS1.1, GS2.1, GS3.1 et GS4.1.

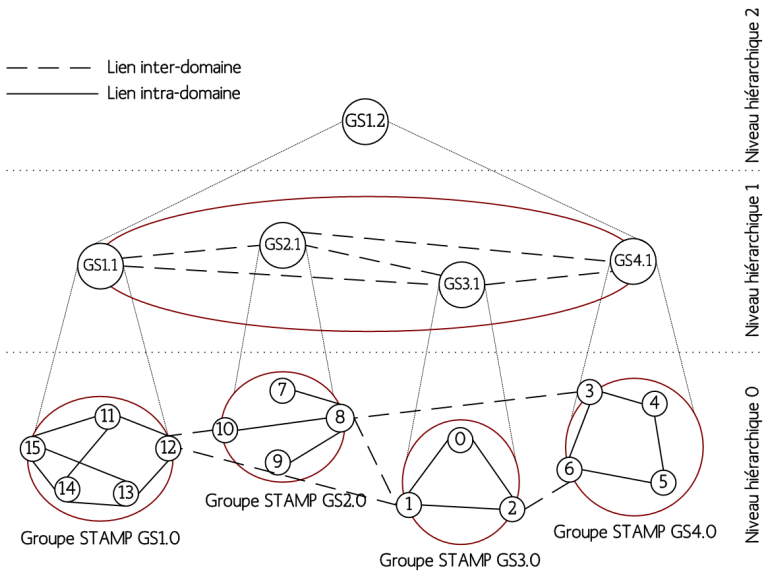

FIG. 5.2: Application de l'agrégation sur plusieurs niveaux hiérarchiques

Il est à noter aussi que chaque noeud virtuel peut être considéré comme étant un routeur complexe dans les niveaux hiérarchiques supérieurs. Ce routeur n'est qu'une représentation abstraite du domaine réseau : les ports sont ceux des routeurs de bordure identifiés dans le niveau juste inférieur de la hiérarchie (device-id est une adresse MAC) et la capacité du routeur représente ce que le modèle abstrait de la topologie garanti en termes de paramètres de QoS [Fig.5.3]. Sur la base de cette modélisation, il est alors possible d'exécuter STAMP dans les niveaux supérieurs de la hiérarchie avec la même logique d'échange de messages.

Sur la base de l'ensemble des émulations conduites sur les schémas d'agrégation *star* et *full-mesh*, nous avons observé que le taux de confiance est fortement dépendant des propriétés d'échelle du domaine réseau. Ce résultat peut être un indicateur important lors de la définition de l'architecture du réseau. En effet, l'administrateur réseau doit bien spécifier et identifier la périphérie de son réseau (les éléments de bordure). De cette façon, nous pouvons garantir qu'une fois agrégé, le réseau n'introduira pas une très grande déformation par rapport à ce qui existe réellement. Toutes ces observations nous mène à conclure que cette approche ne fournit pas seulement un outil performant pour l'optimisation des opérations de calcul et de traitements sur les grandes topologies de réseaux, mais aussi un outil efficace pour la conception d'architectures supportant le passage à l'échelle.

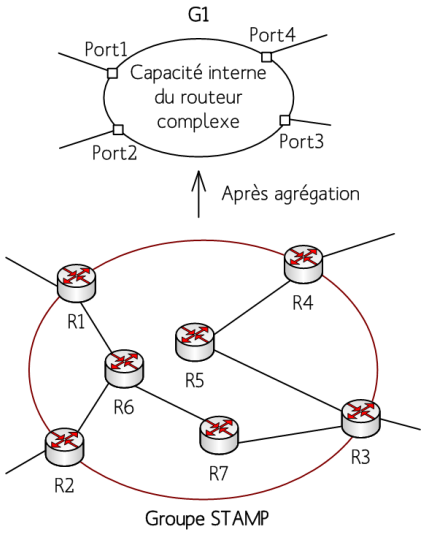

FIG. 5.3: Modélisation du routeur complexe

Aussi, une étude sur l'application des mécanismes d'agrégation sur les réseaux mobiles peut être envisagée. Par réseaux mobiles, nous désignons les réseaux de coeur mobile et non pas les accès radio.

Liste des Publications

W.Htira , O.Dugeon , M.Diaz, *A new approximation model for guaranteed QoS information aggregation*, 12th IEEE Symposium on Symposium on Computers and Communications (ISCC'07), Aveiro, Portugal, Juillet 2007.

Ce papier a été récompensé par le prix du meilleur papier (Phd student best paper award).

W.Htira , O.Dugeon , M.Diaz, *An aggregated delay/bandwith star scheme for admission control*, 12th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'07), Athènes, Grèce, Septembre 2007.

W.Htira , O.Dugeon , M.Diaz, *A mesh aggregation scheme for call admission control*, Australasian Telecommunication Networks and Applications Conference (ATNAC'07), Christchurch, Nouvelle Zélande, Décembre 2007.

W. Htira, O. Dugeon, M. Diaz, *STAMP : Towards A Scalable Topology Announcement and Management Protocol*, The IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA'08), Okinawa, Japan, Mars 2008.

W. Htira, O. Dugeon, M. Diaz, *A Novel Bandwidth Broker Architecture Based on Topology Aggregation in Delay|Bandwidth Sensitive Networks*, IFIP Networking 2008, Singapore, Mai 2008.

Brevet

W.Htira , O.Dugeon, *Admission d'Appel sur Schéma Agrégé de la Topologie*, INPI N° 07 07043.

Glossaire

adb : adjacency database
ADSL : Asymmetric Digital Subscriber Line
AS : Autonomous System
ATM : Asynchronous Transfert Mode
BB : Bandwidth Broker
BGP : Border Gateway Protocol
BE : Best Effort
CAC : Call Admission Control
cdb : characteristics database
CDP : Cisco Discovery Protocol
DNS : Domain Name System
DSLAM : Digital Subscriber Line Access Multiplexer
DPS : Dynamic Packet State
ESI : End System Identifier
FDDI : Fiber Distributed Data Interface
FIFO : First In First Out
FR : Frame Relay
GBL : Group Backup Leader
GE : GigaBit Ethernet
GM : Group Member
GPL : Group Primary leader
HP : High Priority
IANA : Internet Assigned Numbers Authority
ICMP : Internet Control Message Protocol
IETF : Internet Engineering Task Force
IGMP : Internet Group Management Protocol
IP : Internet Protocol
IPfix : IP Flow Information Export
IPoE : IP over Ethernet
IS-IS : Intermediate System to Intermediate System
LAN : Local Area Network
LDP : Label Distribution Protocol
LLC : Logical Layer Control
LLDP : Link Layer Discovery Protocol
LLDP-MED : Link Layer Discovery Protocol for Media Endpoint Devices
LLTD : Link Layer Topology Discovery protocol
LP : Low Priority
LS : Line Segment

MAC : Medium Acces Control
MAN : Metropolitan Area Network
MBAC : Measurement Based Admission Control
MIB : Management Information Base
MP : Medium Priority
MPLS : MultiProtocol Label Switching
MPPBCA : Multiple-Path-Parameters-Best-Case Approach
MPWBCA : Multiple-Path-Parameters-Worst-Case Approach
MTU : Maximum Transfert Rate
NSAP : Network Service Access Point
OSI : Open Systems Interconnection
OSPF : Open Shortest Path First
PBAC : Parameter-Based Admission Control
PCC : Path Computation Client
PCE : Path Computation Element
PDP : Policy Decision Point
PHB : Per Hop Behavior
PNNI : Private Network-to-Network Interface
POP : Point of Presence
POS : Packet Over SONET/SDH
PPP : Point to Point Protocol
qdb : quarantine database
QoS : Quality of Service
SDH : Synchronous Digital Hierarchy
SNAP : SubNetwork Access Protocol
SNMP : Simple Network Management Protocol
SONET : Synchronous Optical Network
SPPA : Single-Path-Parameters Approach
SRP : Scalable resource Reservation Protocol
STAMP : Simple Topology Annoucement and Management Protocol
STAMP-DU : STAMP Data Unit
TCP : Transmission Control Protocol
tdb : topology database
TE : Traffic Engineering
TLS : Transport Layer Security
TLV : Type Length Value
VC : Virtual Channel
VLAN : Virtual LAN
VoD : Video on Demand
VP : Virtual Path
VPN : Virtual Private Networks
WAN : Wide Area Network
WPA : Weighted PolyLine Approach

Bibliographie

[1] *Introducing DWDM*, www.cisco.com

[2] *IST/EuQoS project*, http ://www.euqos.org.

[3] C.P.S. Shenker, R. Guerin, *Specification of Guaranteed Quality of Service*, IETF, Internet RFC 2212, Septembre 1997.

[4] J. Wroclawski, *Specification of the Control-Load Network Element Service*, IETF, Internet RFC 2211, Septembre 1997.

[5] S. Jamin, P.B. Danzig, S.J. Shenker, L. Zhang, *A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks*, IEEE/ACM Transactions on Networking, vol. 5(1), pp. 56-70, Février 1997.

[6] S. Jamin, S. Shenker, *Measurement-based Admission Control Algorithms for Controlled-load Service : A Structural Examination*, University of Michigan, Technical Report CSE-TR-333-97, Avril 1997.

[7] L. Breslau, S. Jamin, S. Shenker, *Comments on the Performance of Measurement based Admission Control Algorithms*, Proceedings of IEEE INFOCOM 2000, Mars 2000.

[8] R. Yavatkar, D. Pendarakis, R. Guerin, *A Framework for Policy-based Admission Control*, IETF, RFC 2753, Janvier 2000.

[9] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, *The COPS (Common Open Policy Service) Protocol*, IETF, Internet RFC 2748, Janvier 2000.

[10] A. Bak, W. Burakowski, F. Ricciato, S. Salsano, H. Tarasiuk, *Traffic Handling in AQUILA QoS IP Networks*, QofIS'01, volume 2156, pages 243-260, Septembre 2001.

[11] L. Breslau, E. Knightly, S. Shenker, I. Stoica, H. Zhang, *Endpoint Admission Control : Architectural Issues and Performance*, In ACM SIGCOMM'00, Aout 2000.

[12] M. Gerla, *A Survey of Admission Control Algorithms*, Technical Report CS215, UCLA, Décembre 1998.

[13] M. Gerla, S. Lee, G. Reali, D. Sorte, *Performance of Different Call Admission Schemes in a QoS Diffserv Domain*, http ://www.cs.ucla.edu/ñrl/hpi/papers/2001-milcom-0.ps.gz, 2001.

[14] E. Knightly, N. Shroff, *Admission Control for Statistical QoS : Theory and Practice*, IEEE Network, 13(2) :20-29, Mars 1999.

[15] D. Tse, M. Grossglauser, *Measurement-based Call Admission Control : Analysis and Simulation*, Proceedings of IEEE INFOCOMM'97, Kobe, Japan, Avril 1997.

[16] S. Jamin, S.J. Shenker, P.B. Danzig, *Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service*, Proceedings of IEEE INFOCOMM'97, Kobe, Japan, Avril 1997.

[17] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, *Resource ReSerVation Protocol (RSVP)*, IETF, Internet RFC 2205 (Updated by RFCs 2750, 3936), Septembre 1997.

[18] R. Braden, D. Clark, S. Shenker, *Integrated Services in the Internet Architecture : an Overview*, IETF, Internet RFC 1633, Juin 1994.

[19] T. Ferrari, W. Almesberger, J.Y. Le Boudec, *SRP : A scalable resource reservation protocol for the Internet*, In Proceedings of the sixth IEEE/IFIP International Workshop on Quality of Service (IWQoS'98), pages 107-116, Napa, CA, USA, Mai 1998.

[20] Y. Bernet, *A Framework for Integrated Services Operation Over DiffServ Networks*, IETF, Internet RFC 2998, Novembre 2000.

[21] I. Stoica, H. Zhang, *Providing guaranteed services without per flow management*, Technical Report CMU-CS-99-133, School of Computer Science, Carnegie Mellon University, Mai 1998.

[22] I. Stoica, H. Zhang, *Providing guaranteed services without per flow management*, In Proceedings of SIGCOMM'99, pages 81-94, Boston, MA, Septembre 1999.

[23] C. Cetinkaya, V. Kanodia, E. Knightly, *Scalable Services via Egress Admission Control*, IEEE Transactions on Multimedia, 3(1) :69-81, Mars 2001.

[24] J. Qiu, E. Knightly, *Measurement-Based Admission Control with Aggregate Traffic Envelopes*, IEEE/ACM Transactions on Networking, 9(2) :199-210, Avril 2001.

[25] J. Qiu, E. Knightly, *Inter-class Resource Sharing using Statistical Service Envelopes*, In IEEE INFOCOM'99, Mars 1999.

[26] J. Schlembach, A. Skoe, P. Yuan, E. Knightly, *Design and Implementation of Scalable Admission Control*, In InternationalWorkshop on QoS in Multiservice IP Networks, Avril 2001.

[27] P. Eardley, *Pre-Congestion Notification Architecture*, draft IETF, Aout 2008.

[28] K. Nichols, V. Jacobson, L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet*, IETF, Internet RFC 2638, Juillet 1999.

[29] *QBone Home Page*, http ://qbone.internet2.edu/.

[30] B. Teitelbaum, S. Hares, L. Dunn, V. Narayan, R. Neilson, F. Reichmeyer, *Internet2 QBone : Building a Testbed for Differentiated Services*, IEEE Network Magazine, Special Issue on Integrated and Differentiated Services for the Internet, Septembre 1999.

[31] *Recommandation G.992.3*, http ://www.itu.int/rec/T-REC-G.992.3-200501-I/fr.

[32] *802.1q - Virtual LANs*, http ://www.ieee802.org/1/pages/802.1q.html.

[33] E. Rosen, A. Viswanathan, R. Callon, *Multiprotocol Label Switching Architecture*, Technical report, IETF, Internet RFC 3031, Janvier 2001.

[34] Y. Rekhter, T. Li, S. Hares, *A Border Gateway Protocol 4 (BGP-4)*, IETF, Internet RFC 4271, Janvier 2006.

[35] D. Meyer, K. Patel, *BGP-4 Protocol Analysis*, IETF, Internet RFC 4274, Janiver 2006.

[36] D. McPherson, K. Patel, *Experience with the BGP-4 Protocol*, IETF, Internet RFC 4277, Janvier 2006.

[37] N. Spring, R. Mahajan, D. Wetherall, *Measuring isp topologies with rocketfuel*, In proceedings of ACM SIGCOMM'02, ittsburgh, PA, USA, Aout 2002.

[38] R. Govindan, H. Tangmunarunkit, *Heuristics for Internet map discovery*, Annals of telecommunications, In Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, Mars 2000.

[39] D. Magoni, M. Hoerdt, *Internet core topology mapping and analysis*, Annals of telecommunications,Computer Communications, 28(5) : 494-506 (2005).

[40] J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin, *Simple network management protocol (snmp)*, IETF, Internet RFC 1157, Mai 1990.

[41] K. McCloghrie, M. Rose, *Management Information Base for Network Management of TCP/IP-based internets : MIB-II*, IETF, Internet RFC 1213, Mars 1991.

[42] S. Routhier, *Management Information Base for the Internet Protocol (IP)*, IETF, Internet RFC 4293, Avril 2006.

[43] K. McCloghrie, *Management Information Base for the Internet Protocol (IP)*, IETF, Internet RFC 2011, Novembre 1996.

[44] K. McCloghrie, *Management Information Base for the Transmission Control Protocol (TCP)*, IETF, Internet RFC 2012, Novembre 1996.

[45] K. McCloghrie, *Management Information Base for the User Datagram Protocol (UDP)*, IETF, Internet RFC 2013, Novembre 1996.

[46] K. Norseth, E. Bell, *Definitions of Managed Objects for Bridges*, IETF, Internet RFC 4188, Septembre 2005.

[47] *Cisco MIBs*, http ://www.cisco.com/public/mibs.

[48] *HP OpenView*, http ://wwww.openview.hp.com.

[49] *IBM Tivoli*, http ://www.tivoli.com.

[50] *Dartmouth Intermapper*, http ://dartware.com/network_monitoring_products/intermapper.

[51] J. Postel, *Internet Control Message Protocol*, IETF, Internet RFC 0792, Septembre 1981.

[52] J.C. Mogul, J. Postel, *Internet Standard Subnetting Procedure*, IETF, Internet RFC 0950, Aout 1985.

[53] J. Schönwälder, H. Langendörfer, *How to keep track of your network configuration*, In LISA, Novembre 1993.

[54] G. Malkin, *STraceroute Using an IP Option*, IETF, Internet RFC 1393, Janvier 1993.

[55] H.C Lin, S.C Lai, P.W Chen, *Automatic topology discovery of IP networks*, In IEICE Transactions, volume E83-D, Janvier 2000.

[56] H.C. Lin, Y.F. Wang, C.H. Wang, *Web-based Distributed Topology Discovery of Ip Networks*, In Proceedings of the 15th International Conference on Information Networking, Janvier 2001.

[57] S. Sharma R. Siamwalla, S. Kashav, *Discovering Internet topology*, Technical report, Cornell University, Ithaca, New York, Juillet 1998. http ://www.cs.cornell.edu/skeshav/papers/discovery.pdf.

[58] *ping*, http ://www.linuxcertif.com/man/8/ping.

[59] J.J. Pansiot, *Local and dynamic analysis of Internet multicast router topology*, Annals of telecommunications, Vol. 62, Num. 3-4, pp 408-425, Mars 2007.

[60] A. Bierman, K. Jones, *Physical topology MIB*, IETF, Internet RFC 2922, Septembre 2000.

[61] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, A. Silberschartz, *Topology discovery in heterogeneous IP networks*, In IEEEE/INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, volume xxvi+1826, 2000.

[62] Y. Breibart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, A. Silberschatz, *Topology Discovery in Heterogeneous IP Networks*, IEEE INFOCOM 2000, pages 265-274, Tel Aviv, Israel, Mars 2000.

[63] Y. Bejerano, Y. Breitbart, M. Garofalakis, R. Rastogi, *Physical Topology Discovery for Large Multi-Subnet Networks*, IEEE INFOCOM 2003, pages 342-35, San Francisco, USA, Avril 2003.

[64] B. Lowekamp, D.R. O'Hallaron, T.R. Gross, *Topology Discovery for large ethernet networks*, In ACM/SIGCOMM proceeding, Aout 2001.

[65] *Cisco Discovery Protocol*, http ://www.cisco.com.

[66] J. Postel, J. Reynolds, *Standard for the transmission of IP datagrams over IEEE 802 networks*, IETF, Internet RFC 1042, Février 1988.

[67] *Link Layer Topology Discovery Protocol Specification*, http ://www.microsoft.com.

[68] *802.1AB - Station and Media Access Control Connectivity Discovery*, http ://ieee802.org/1/pages/802.1ab.html.

[69] J. Moy, *OSPF Version 2*, IETF, Internet RFC 2328, Avril 1998.

[70] R. Coltun, D. Ferguson, J. Moy, *OSPF for IPv6*, IETF, Internet RFC 2740, Décembre 1999.

[71] F. Ruskey, *Combinatorial generation*, Working Version 1j-CSC 425/520, Octobre 2003.

[72] S. Mancoridis, B.S. Mitchell, C. Rorres, Y. Chen, E.R. Gansner, *Using automatic clustering to produce high-level system organizations of source code*, IWPC 98, Ischia, Italie, Juin 1998.

[73] S. van Dongen, *Graph Clustering by Fow Simulation*, PhD thesis, University of Utrecht, Mai 2000.

[74] D. Peleg, *Distributed Computing, a locality-sensitive approach, SIAM Monographs on Discrete Mathematics and Applications*, ISBN 0898714648, Septembre 2000.

[75] T. Eilam, C. Gavoille, D. Peleg, *Compact routing schemes with low stretch factor*, PODC 98, Puerto Vallarta, Mexico, Juin 1998.

[76] F. Brandenburg, J. Edachery, A. Sen, *Graph clustering using distance-k cliques*, Graph Drawing 99, Stirin Castle, Czech Republic, Septembre 1999.

[77] V. Batagelj, A. Mrvar, M. Zaversnik, *Partitioning approach to visualization of large graphs*, Graph Drawing 99, Stirin Castle, Czech Republic, Septembre 1999.

[78] The ATM Forum, *Private Network-Network*, specification version 1.0, Mars 1996.

[79] L. Guo, I. Matta, *On State Aggregation for Scalable QoS Routing*, ATM Workshop 98, Mai 1998.

[80] W. Lee, *Minimum equivalent subspanner algorithms for topology aggregation in ATM networks*, ICATM 99, Colmar, France, Juin 1999.

[81] W.C. Lee, *Topology Aggreagtion for Hierarchical Routing in ATM Networks*, ACM SIGCOMM 95, Cambridge, Massachusetts, USA, Aout 1995.

[82] W. Lee, *Minimum equivalent subspanner algorithms for topology aggregation in ATM networks*, ICATM 99, Colmar, France, Juin 1999.

[83] B. Awaebuch, Y. Shavitt, *Topology aggregation for directed graphs*, ISCC 98, Athens, Greece, Juin 1998.

[84] A. Iwata, H. Suzuki, R. Izmailow, B. Sengupta, *QoS aggregation algorithms In hierarchical ATM networks*, ICC 98, Juin 1998.

[85] T. Korkmaz, M. Krunz, *Source-Oriented topology aggregation with multiple QoS parameters in hierarchical ATM network*, IWQoS 99, London, England, Juin 1999.

[86] K.S Lui, k. Nahrstedt, S. Chen, *Routing with topology aggregation in delay-bandwidth sensitive networks*, IEEE Transactions on Networking, vol. 12, no.1, Février 2004.

[87] Y. Tang, S. Chen, *QoS Information Approximation for aggregated networks*, ICC 2004, Paris, France, Juin 2004.

[88] T. Dierks, E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, IETF, Internet RFC 4346, Avril 2006.

[89] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, T. Wright, *Transport Layer Security (TLS) Extensions*, IETF, Internet RFC 4366, Avril 2006.

[90] S. Santesson, *TLS Handshake Message for Supplemental Data*, IETF, Internet RFC 4680, Octobre 2006.

[91] S. Santesson, A. Medvinsky, J. Ball, *TLS User Mapping Extension*, IETF, Internet RFC 4681, Octobre 2006.

[92] *Socket man*, http ://linux.die.net/man/2/socket.

[93] M. Liljenstam, J. Liu, D. Nicol, *An Internet Topology for Simulation*, http ://www.crhc.uiuc.edu/ jasonliu/projects/topo/.

[94] J. Babiarz, K. Chan, F. Baker, *Configuration Guidelines for DiffServ Service Classes*, IETF, Internet RFC 4594, Aout 2006.

[95] C. Hopps, *Analysis of an Equal-Cost Multi-Path Algorithm*, IETF, Internet RFC 2992, Novembre 2000.

[96] *GT-ITM tool*, http ://www.cc.gatech.edu/projects/gtitm.

[97] B.M. Waxman, *Routing of Multipoint Connections*, IEEE Journal of Selected Areas in Communications, pp. 1617-1622, Decembre 1988.

[98] B. Claise, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*, IETF, Internet RFC 5101, Janvier 2008.

[99] J. Quittek, S. Bryant, B. Claise, P. Aitken, J. Meyer, *Information Model for IP Flow Information Export*, IETF, Internet RFC 5102, Novembre 2000.