ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE CONCENTRATION RÉSEAUX DE TÉLÉCOMMUNICATIONS
M.Ing.

PAR
MUHI-EDDINE, WAFIC

ÉTUDE DES ALGORITHMES DE ROUTAGE ADAPTATIF POUR LE RÉSEAU DÉDIÉ DE SERVICE

MONTRÉAL, LE 2 JUIN 2008

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Zbigniew Dziong, directeur de mémoire
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Jean-Charles Grégoire, membre du jury
Centre Énergie, Matériaux et Télécommunications à l'Institut national de la recherche scientifique

M. Michel Kadoch, président du jury
Département de génie électrique à l'École de technologie supérieure

REMERCIEMENTS

Ce mémoire s'inscrit dans les activités du laboratoire de gestion de réseaux informatiques et de télécommunications (LAGRIT), de l'École de technologie supérieure (ÉTS).

Je souhaite exprimer ma gratitude à mon directeur de mémoire, M. Zbigniew Dziong, pour m'avoir accueilli au sein de l'équipe du laboratoire LAGRIT, pour son support et pour la confiance qu'il m'a accordée en acceptant de diriger ce projet. Je lui suis très reconnaissant pour tous les conseils, toute l'aide, toute la confiance et pour tous les moyens qu'il a mis à ma disposition. Je tiens également à le remercier pour sa grande patience ainsi que ses encouragements dans les moments difficiles.

Ma gratitude va aussi aux membres du jury pour avoir pris le temps de lire attentivement et d'évaluer mon mémoire.

Je veux saluer tous ceux qui ont fait partie de l'équipe de LAGRIT pour la superbe ambiance de travail et l'amitié qui s'est établie entre nous, surtout Mohammad, Achour et Nabila. Un gros merci à mon vrai ami et collègue Zouheir pour son aide et son soutien sans limite.

Enfin, j'aimerais remercier tous ceux qui m'ont soutenu dans le plan moral et affectif pendant ces années : mes parents, mes sœurs, mon frère, mon épouse, mon petit garçon Omar et surtout sans oublier la personne qui a toujours été là pour me soutenir derrière les coulisses, le soldat inconnu de l'informatique, Dr. El-Zammar Chadi.

ÉTUDE DES ALGORITHMES DE ROUTAGE ADAPTATIF POUR LE RÉSEAU DÉDIÉ DE SERVICE

MUHIEDDINE, Wafic

RÉSUMÉ

Le réseau dédié de service fournit une plate-forme de livraison de bout en bout pour les services d'Internet sensibles à la QoS, tels que les applications multimédias. Cette plate-forme de livraison est basée sur des algorithmes de routage adaptatif. Ces algorithmes prennent leurs décisions d'acheminement à la source en fonction des états des liens logiques liant les nœuds constituant le réseau dédié de service. L'état d'un lien logique est représenté par la largeur de bande disponible que ce lien fournit. Ainsi, les nœuds du réseau dédié de service construisent des bases de données des états de tous les liens du réseau en utilisant des mesures actives de la bande passante disponible.

Ce projet fait une étude de performance des algorithmes de routage adaptatif qui emploient des différentes méthodes de mesures de la largeur de bande disponible (SPRUCE, PathLoad et mesures exactes). Les méthodes de mesures citées ainsi que le routage adaptatif ont été modélisés et simulés à l'aide du logiciel NS. L'algorithme qui permet d'obtenir les meilleurs paramètres de QoS (délai de bout en bout, gigue et taux de perte) d'un flux à envoyer, et ceci en utilisant une des méthodes de mesure à la fois, est considéré le plus performant. Les algorithmes ont été comparés afin de déterminer le plus performant.

Les simulations ont montré que les algorithmes utilisant SPRUCE et les mesures exactes ont presque la même performance et que cette performance est mieux que celle de l'algorithme qui utilise PathLoad, spécifiquement quand les conditions du réseau sont mauvaises, en d'autres mots quand le réseau est chargé. Également, ces simulations ont montré les avantages du routage adaptatif du réseau dédié de service par rapport au routage traditionnel.

Mots clés : multimédias, QoS, adaptatif, mesure, simulation, NS

STUDY OF ADAPTIVE ROUTING ALGORITHMS FOR SERVICE OVERLAY NETWORK

MUHIEDDINE, Wafic

ABSTRACT

The Service Overlay Network provides an end-to-end delivering platform for Internet sensitive service of QoS, for example, multimedia applications. The Service Overlay Network counts on adaptive routing algorithms. Adaptive routing algorithms take their routing decisions at the source according to the state of logical links among nodes, which constitute the service overlay network. The available bandwidth represents the state of a logical link. The service overlay network nodes build link-state-database of the network using active measurements of the available bandwidth.

This project is a study of performance of adaptive routing algorithms that employ different measurements methods of the available bandwidth (SPRUCE, PathLoad and accurate measurements). The measurement methods and the adaptive routing are modelled and simulated with the NS simulator. The more the algorithm achieves better QoS parameters (end-to-end delay, jitter, and loss rate) for an overlay stream the more it considers successful. This study shows the comparison of algorithms to determine the most efficient.

The simulations have shown that the algorithms using SPRUCE and accurate measurements have almost the same performance. This performance is better than that of the algorithm that uses PathLoad specifically at bad network conditions otherwise, when the network load is very high. In addition, these simulations have shown the advantages of adaptive routing of service overlay network compared to traditional routing.

Keywords: multimedia, QoS, adaptive, measurement, simulation, NS

TABLE DES MATIÈRES

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AAL5	ATM Adaptation Layer 5
ALMI	Application Level Multicast Infrastructure
AS	Autonomous System
ASN	Autonomous System Number
ATM	Asynchronous Transfer Mode
AT&T	American Telephone & Telegraph Company
BGP	Border Gateway Protocol
CAPP	Cache Array Routing Protocol
CBR	Constant Bit Rate
CDN	Content Delivery Networks
CLVL	Controlled-Loss Virtual Link
DHT	Distributed Hash Table
DiffServ	Differentiated Services
DNS	Domain Name System

DSL	Digital Subscriber
DVMRP	Distance Vector Multicast Routing Proto
EIGRP	Enhanced Interior Gateway Routing Protocol
FIFO	First In, First Out
HTTP	Hyper Text Transfer Protocol
IBGP	Interior Borger Gateway Protocol
ICMP	Internet Control Message Protocol
IntServ	Integrated Services
IP	Internet Protocol
ISP	Internet Service Provider
LSP	Label Switched Path
MBone	Multicast Backbone on the Internet
MCI	Microwave Communications Inc
MPLS	Multiprotocol Label Switching
MTU	Maximum Transmission Unit

NAP	Network Access Point
NS	Network Simulator
OSPF	Open Shortest Path First
OSP	Overlay Service Provider
OTCL	Object oriented extension of TCL
P2P	Peer-to-Peer
PGM	Probe Gap Model
PRM	Probe Rate Model
QoS	Quality of Service
RISQ	Réseau d'informations scientifiques du Québec
RON	Resilient Overlay Networks
RTP	Real-time Transport Protocol
SLA	Service Level Agreement
SLoPS	Self-Loading Periodic Stream
SON	Service Overlay Network

SPRUCE	Spread PaiR Unused Capacity Estimation
TCL	Tool Command Language
TCP	Transmission Control Protocol
TTL	Time to Live
Qwest	Qwest Communications Corporation
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
Yoid	Your Own Internet Distribution
6-Bone	IP version 6 Multicast Backbone on the Internet

INTRODUCTION

Le réseau dédié de service ou SON (*Service Overlay Network*) se pose comme un moyen efficace pour résoudre certains problèmes d'Internet actuel, en particulier la QoS (*Quality of Service*) de bout en bout (*end-to-end QoS*) et pour faciliter la création et le déploiement des services d'Internet sensibles à la QoS tels que les applications multimédias.

Le réseau dédié de service est constitué par un sous-ensemble des nœuds de réseau fondamental Internet. Les nœuds Overlay sont placés de façon stratégique par un tiers ou fournisseur de service Overlay OSP *(Overlay Service Provider)*. L'OSP achète le service d'accès aux nœuds Overlay de différents ISP (*Internet Service Providers*). Ainsi, les liaisons entre les nœuds Overlay sont fournies par les liens Overlay logiques. Un lien logique Overlay est un chemin de la couche IP (*Internet Protocol*) et il peut être composé d'un ou de plusieurs liens physiques. Comme les nœuds Overlay sont reliés par des chemins de la couche IP (des liens Overlay), alors théoriquement, il y a un lien Overlay reliant chaque paire des nœuds Overlay. En d'autres termes, la topologie du réseau dédié de service peut être une pleine maille.

En effet, le réseau dédié de service peut être construit sur deux échelles : sur l'échelle d'un seul domaine ou AS (*Autonomous System*) ou sur l'échelle de plusieurs AS. Le système autonome est un ensemble de réseaux IP indépendant (il est sous le contrôle d'une seule et même entité) typiquement un fournisseur d'accès à l'Internet ISP. Les systèmes autonomes fonctionnent en ensemble pour former l'Internet. La notion de système autonome est donc administrative et non technique et la politique de routage interne dans les systèmes autonomes (routes à choisir en priorité, filtrage des annonces) est cohérente.

Le réseau SON achète donc de la bande passante avec certaines garanties de QoS de domaines d'Internet par le biais d'un accord de niveau de service bilatéral SLA (*Service Level Agreement*) pour construire son infrastructure logique de service de livraison de bout en bout.

Si cette infrastructure est à l'échelle d'un seul domaine, SON peut profiter immédiatement du SLA avec ce domaine, qui lui offre une certaine garantie de QoS. SON peut avoir alors des liens logiques construit par des LSP (*Label Switched Path*) d'une infrastructure MPLS (*Multiprotocol Label Switching*) par exemple.

Également, SON établit son infrastructure de livraison de bout en bout sur l'échelle de plusieurs AS en s'appuyant sur les SLA établies avec chaque AS ou domaine individuellement. À part la QoS de bout en bout, une telle infrastructure offre plusieurs avantages par rapport à l'infrastructure de livraison du réseau IP et elle résout les problèmes essentiels que le routage IP ne le peut pas. D'abord, elle fait face au lent recouvrement des fautes et à la convergence de routage BGP (*Border Gateway Protocol*), les réseaux dédiés de service peuvent contourner les chemins brisés en réacheminant le trafic à travers des nœuds Overlay intermédiaires. La détection des chemins brisés ou congestionnés par les nœuds dédiés peut être effectuée rapidement à l'aide des sondes actives en injectant du trafic spécifique pour savoir l'état des liens logiques. Deuxièmement, le routage IP offre le même chemin indépendamment des exigences de la performance. Au lieu de cela, les réseaux dédies de service peuvent offrir des chemins différents pour la même destination, en fonction de la métrique de performance exigée par le trafic (par exemple, délai, débit, gigue et taux de perte). Troisièmement, le fait que le routage IP inter-domaines est largement déterminé par les politiques commerciales des ISP cause souvent des chemins sous-optimaux. Les réseaux dédiés de service peuvent offrir une meilleure performance de bout en bout grâce à l'acheminement à traves des nœuds Overlay intermédiaires, donc forcer les flux de trafic de suivre des chemins spécifiques de bout en bout qui, autrement, ne seraient pas autorisées par les politiques des ISP.

Un nœud Overlay prend donc la décision de routage en fonction des états des liens logiques dans le réseau dédié de service. Les états de liens sont définis en fonction de paramètres de QoS que ces liens peuvent offrir. Ces paramètres sont : le délai, la gigue, le taux de perte et la bande passante disponible. Les nœuds Overlay peuvent connaître les états des liens logiques de deux façons : la première consiste d'étudier les caractéristiques du trafic

acheminé, en mesurant les délais, la gigue et le taux de perte que ses paquets ont subi. La deuxième recommande l'emploi des mesures actives, quand les nœuds Overlay injectent des trains de paquets spécifiques entre eux pour observer les états actuels des liens logiques indépendamment de trafic Overlay ou d'autres événements dans le réseau.

Les mesures actives permettent l'observation d'un paramètre de QoS très important et critique pour le réseau dédié de service, soit la bande passante disponible. Ce paramètre est essentiel car les problèmes des autres paramètres proviennent par conséquence du manque de bande passante disponible, donc la congestion. En effet, si un lien logique offre suffisamment de la bande passante pour acheminer un flux, alors ce flux aura toujours des délais, des gigues et des taux de pertes acceptables. En plus, une vision globale des états des liens logiques permet une meilleure gestion des ressources dans le réseau dédié de service. Les flux peuvent être acheminés à travers des nœuds Overlay intermédiaires pour contourner les parties saturées du réseau et pour mieux exploiter les ressources (largeur de bande disponible) existants et offerts ailleurs dans SON.

L'utilisation des méthodes de mesure de la bande passante disponible dans le réseau SON peut conduire vers un processus de routage plus efficace et préventif, par conséquence, une meilleure performance pour les applications critiques (multimédias). Ces avantages encouragent l'implémentation et l'utilisation des méthodes de mesure de la bande passante disponible dans les nœuds du réseau SON pour impliquer ce paramètre dans le processus de sélections de chemins.

Le but de ce projet est alors d'étudier les effets de l'utilisation des différentes méthodes de mesures de la bande passante disponible sur la performance des algorithmes de routage adaptatif de réseau dédié de service, car chaque méthode de mesure possède ces propres caractéristiques (ces avantages et ces inconvénients) dans le procédé d'estimation de la largeur de bande disponible. Donc, ces caractéristiques peuvent influer directement sur la performance de routage de SON. En effet, la bande passante disponible sera le paramètre essentiel pour les algorithmes de routage pour prendre les décisions dans le processus de

sélection des chemins. Ainsi, la performance et l'efficacité du routage sont reflétées par les paramètres de qualité de service (délai de bout en bout, la gigue et le taux de perte) de trafic Overlay acheminé.

La première étape du projet fut de choisir les méthodes de mesure de la bande passante disponible. Nous utilisons ces méthodes dans les simulations en nous basant sur des caractéristiques spécifiques : la précision de résultats et le temps de mesure. Les méthodes choisies sont implémentées et testées dans NS2 (*Network Simulator Version 2*).

La seconde étape fut de créer l'environnement de simulation en définissant la topologie, les trafics, l'intégration des méthodes de mesure dans les scénarios de simulations et de modifier le routage dans NS2 de façon qu'il tienne compte de la bande passante disponible sur les liens logiques du réseau dédié de service. Dans cette étape nous implémentons SPRUCE (*Spread PaiR Unused Capacity Estimation*) et PathLoad, comme méthodes de mesure de la bande passante disponible, ainsi que CapProbe pour mesurer la capacité totale des liens.

La troisième étape fut de comparer la performance de routage dédié de service implémenté et qui utilise les sondes, le routage dédié qui utilise des mesures exactes données par NS2 et le routage traditionnel (sans aucune sonde ou mesure). Les métriques utilisées dans la comparaison de la performance sont : le délai de bout en bout, la gigue et le taux de perte.

Le premier chapitre fait une introduction de routage dans l'Internet, la façon dont l'Internet est organisé et ses problèmes de fiabilité. Le deuxième chapitre présente le concept des réseaux dédiés de service et comment nous pouvons construire une infrastructure de routage Overlay complète. Le troisième chapitre fait un survol de différentes technologies de réseaux dédiés. Le quatrième chapitre explique le routage dynamique dans le réseau dédié de service. Le cinquième chapitre présente les techniques d'estimation de la bande passante disponible et propose l'utilisation de deux méthodes différentes dans le processus d'estimation. Le sixième chapitre présente les techniques d'estimation de la capacité totale

des liens et propose l'utilisation d'une. Le septième chapitre présente l'environnement de simulation NS2, la topologie des tests, l'implémentation et des essais des méthodes de mesure. Il présente également des résultats de simulation des méthodes et compare leurs performances. Le huitième chapitre couvrira l'ensemble des expérimentations de routage, des résultats de simulation et l'analyse de ceux-ci.

CHAPITRE 1

MISE EN CONTEXTE

Dans le but de comprendre la différence entre le routage traditionnel du réseau Internet et le routage du réseau dédié de service (qui est le sujet principale de ce mémoire) ce chapitre présente la structure actuelle d'Internet, la façon dont il est organisé et comment se fait le routage dans sa dorsale en soulignant les problèmes de fiabilités envisagés. Ainsi, après une compréhension de ces idées, le problème général de la construction d'une solution de livraison avec qualité de service de bout en bout dans l'Internet, qui est le but des réseaux dédiés de service, sera claire. Ce travail explique comment les réseaux dédiés de service peuvent être cette solution remarquable pour une livraison de bout en bout avec un respect acceptable de la qualité de service qu'exigent les applications multimédias.

1.1 L'organisation d'Internet

L'Internet est un ensemble de réseaux ou systèmes autonomes qui partagent un protocole de communication commun : l'IP. Typiquement, les réseaux sont détenus et gérés par des organisations différentes [Anderson (2001)]. Généralement, les compagnies sont connectées à l'Internet par le biais d'un fournisseur de service Internet (AS de leur région ou de leur ville), par exemple Sympatico ou Bell. Les utilisateurs à la maison au Canada se connectent à l'Internet par le biais des modems (33,6 ou 56 kbit/s), de modems DSL (*Digital Subscriber Lines*) de 128 kbps à 5 Mbps ou par des modems de câble (128 Kbps-7 Mbps). Les universités et les fournisseurs locaux de services sont attachés d'une façon semblable, bien souvent avec une augmentation de la vitesse T1 (1.5 Mbps) ou T3 (45 Mbps). De nombreuses sociétés et des universités peuvent se connecter à plusieurs fournisseurs de services Internet en même temps (*Multi-homing*).

L'Internet possède une collection de nombreux grands fournisseurs de services qui gèrent ensemble une fraction importante de la circulation sur l'Internet (Figure 1.1).

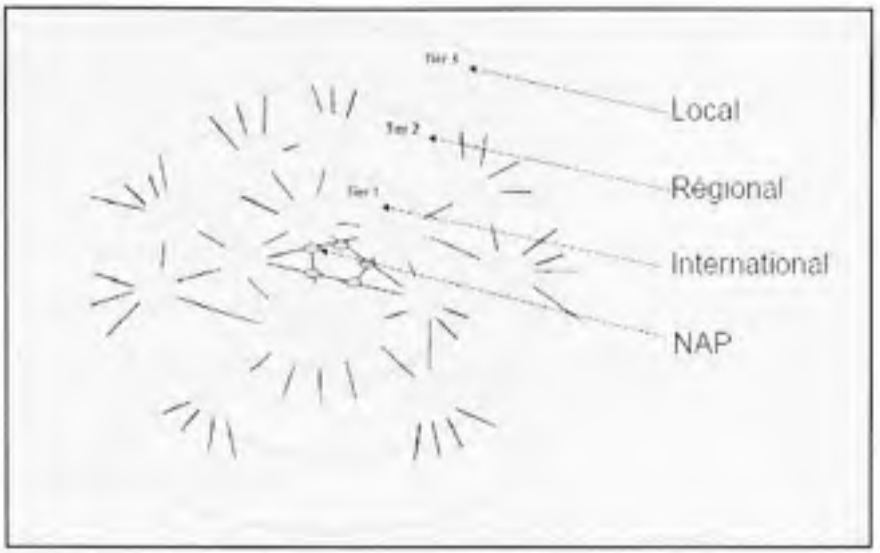

Figure 1.1 ***Structure globale d'Internet.***

(Tiré de Kurose, James et Keith W. Ross, 2007)

Source : Cette figure a été tirée de Kurose, James et Keith W. Ross 2007 « Computer Networking: A top-Down Approach » et correspond à la « Figure 1.11 Interconnection of ISPs » présentée en page 32 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Comme la Figure 1.1 le montre, des réseaux ou systèmes autonomes des fournisseurs de différents grades. Les grands fournisseurs de services Internet sont souvent appelés par des fournisseurs de grade 1 (*Tier1*) par exemple AT&T (*American Telephone & Telegraph Company*) et MCI (*Microwave Communications Inc*) sur l'échelle internationale, BELL Enterprise et Qwest sur l'échelle nationale. Les fournisseurs de *Tier1* sont généralement d'envergure nationale ou internationale, et peuvent fournir une connectivité à l'Internet à haut débit presque partout. Le groupe *Tier1* de fournisseurs n'achète jamais de la connectivité à l'Internet. Au lieu de cela, ils ont tous des accords mutuels pour partager les données entre eux. Ils sont connectés entre eux par des points d'accès NAP (*Network Access Point*).

Les fournisseurs de grade 2 (*Tier2*) servent généralement des régions spécifiques par exemple RISQ (Réseau d'informations scientifiques du Québec) à l'échelle du Québec, et peuvent acheter de la connectivité de certains fournisseurs de *Tier1*. Toutefois, de nombreux

fournisseurs *Tier2* ont également des accords (*Peering*) de partage de données entre eux. Les fournisseurs de *Tier3* ou locaux comme Sympatico : réseau d'accès DSL résidentiel.

Ainsi, l'Internet est composé de nombreux réseaux discrets et les ISP doivent s'échanger des informations de routage (connexion où les deux fournisseurs de services Internet s'échangent d'informations d'accessibilité (*Peering*). Les ISP utilisent le protocole de routage BGP version 4 pour effectuer le routage inter-fournisseurs.

1.2 Le routage d'Internet

L'Internet est constitué de systèmes autonomes AS (*Autonomous System*) indépendants qui fonctionnent en ensemble. Un système autonome est un ensemble de réseaux IP sous le contrôle d'une seule et même entité, typiquement un fournisseur d'accès à l'Internet ou une plus grande organisation qui possède des connexions redondantes avec le reste du réseau Internet. La notion de système autonome est donc administrative et non technique. Ainsi, la politique de routage interne dans les systèmes autonomes (routes à choisir en priorité, filtrage des annonces) est cohérente. Il est évident alors que la notion de système autonome s'oppose à celle de réseau public comme l'Internet, où des différentes entités indépendantes peuvent prendre des décisions contradictoires.

Dans cette architecture, les informations de routage détaillées sont maintenues uniquement à l'intérieur d'un AS unique et dans les réseaux composants. Les informations, qui se partagent avec des autres fournisseurs et systèmes autonomes, sont fortement filtrées et résumées en utilisant le BGP-4 fonctionnant sur les routeurs de bordures entre les systèmes autonomes, ce qui permet à l'internet d'être composé de millions de réseaux. Ce routage, qui est à l'échelle des larges zones, vient au coût de la réduction de la fiabilité de la communication de bout en bout entre les hôtes d'internet. Ainsi que, ce coût est dû de fait que BGP s'élargit en limitant grandement le nombre de liens de réseaux et par conséquence les chemins. Bien que, que la quantité d'informations concernant ces liens soit examinée par le protocole fonctionnel dans chaque routeur de bordure dans l'Internet.

Les mécanismes de recouvrement des fautes de BGP ont besoin parfois de plusieurs minutes avant que les routes convergent uniformément, et il y a des moments où les pannes des chemins conduisent à des perturbations importantes dans la communication. Le résultat est que l'Internet est aujourd'hui facilement vulnérable aux défaillances de liens, routeurs, aux erreurs de configuration et à la congestion.

Ces problèmes sont critiques pour les applications multimédias qui comptent beaucoup sur la qualité de service offert par le réseau, car cette qualité offerte reflète la qualité de la performance de ces applications. Ces problèmes ont poussé vers la recherche d'un moyen pour détecter la qualité des connexions en obtenant par exemple de l'information sur la bande passante disponible ou le délai de bout en bout le long d'un chemin (qui traverse plusieurs systèmes autonomes) afin de pouvoir adapter la qualité et le débit du contenu transmis ou pour choisir un autre chemin alternatif fournissant plus de la bande passante ou un meilleur délai, ce qui signifie une meilleure qualité de service et par conséquence une meilleure performance [Anderson (2001)].

1.2.1 Le protocole BGP

Le protocole BGP version 4 est le protocole de routage de la dorsale d'Internet aujourd'hui. Bien que les fournisseurs peuvent utiliser différents protocoles de routage au sein de leurs propres réseaux, par exemple : OSPF (*Open Shortest Path First*), EIGRP (*Enhanced Interior Gateway Routing Protocol*), IBGP (*Interior Borger Gateway Protocol*), etc. Le routage inter-domaine est géré principalement avec BGP. BGP repose sur l'idée du système autonome (AS), un réseau opérationnel et organisé indépendamment. Chaque AS est identifié par un numéro de système autonome unique. Un grand fournisseur comme MCI peut avoir un ou plusieurs ASN (*Autonomous System Number*), ainsi qu'une institution comme l'École de technologie supérieure.

Les systèmes autonomes envoient des annonces déclarant qu'un réseau peut être contacté par leurs intermédiaires. Ces annonces se propagent dans le réseau Internet, éventuellement,

en recevant ces messages, tous les sites savent comment se rendre à d'autres sites sur le réseau.

1.3 Problème de fiabilité d'Internet

Anderson et *al.* (2001) expliquent que le routage d'Internet (régions étendues), qui est basé sur BGP4, ne s'occupe pas bien des échecs de ce réseau. De point de vue réseau, il existe deux grands types de défaillances : la défaillance des liens qui s'arrivent lors qu'un routeur ou qu'un lien reliant deux routeurs en raison d'une erreur du logiciel, un problème de matériel ou déconnexion de lien et les pannes de chemins qui se produisent pour des raisons diverses, y compris les attaques par déni de service ou d'autres poussées de trafic qui causent un degré élevé de perte de paquets et un délai variable.

Les applications perçoivent toutes les défaillances d'une des deux façons suivantes : panne ou faiblesse de la performance. Les défaillances des liens et les défaillances extrêmes de chemins causent des coupures quand le taux moyen de perte de paquets dans une longue période de plusieurs minutes est élevé (environ 30% ou plus), dégradant la performance de la plupart des protocoles, y compris TCP (*Transmission Control Protocol*). Les faiblesses de performance sont moins extrêmes; ainsi, le débit, la latence et/ou le taux de perte peuvent être dégradés.

Une vaste zone de routage IP souffre de ces deux inconvénients :

1. Faible recouvrement des liens : BGP4 prend beaucoup de temps (de l'ordre de plusieurs minutes) pour converger vers un nouvel chemin valable après la défaillance d'un routeur ou d'un lien causant une panne de chemins.
2. L'incapacité à détecter les échecs de la performance : BGP4 ne peut pas détecter de nombreux problèmes (inondations, congestion persistante, etc.) ce qui peut dégrader considérablement les performances. Tant que le lien est réputé vivant (c'est-à-dire,

que la session BGP est encore en vie), le routage à base des chemins de système autonome de BGP continuera d'acheminer les paquets vers le chemin défectueux. Ainsi, dans une vaste zone, les mécanismes de routage ne peuvent pas traiter convenablement les échecs de performance qui peuvent grandement dégrader la qualité de la communication de bout en bout entre les applications. Considérons, par exemple, un lien qui est excessivement chargé en raison du trafic légitime ou d'une attaque de déni de service. Généralement, BGP n'est pas au courant de ça, et même si une autre voie alternative existe, il ne l'utilisera pas. Ce manque de diversité de chemins complique la manipulation de dégradation de la performance, et conduit à de graves problèmes.

Le routage inter-domaines conventionnel de la couche IP est un routage indépendant de l'application, il ne permet pas généralement une sélection de chemins fondée sur la nature de l'application. Par conséquent, les applications fonctionnant sur l'Internet ne sont pas en mesure d'influer le choix des chemins pour mieux s'adapter à leur délai, taux de perte ou de bande passante.

CHAPITRE 2

TECHNOLOGIE ABORDÉE

Le premier chapitre a présenté l'infrastructure actuelle d'Internet qui supporte essentiellement le service de connectivité *best effort*. Il a expliqué que cet Internet consiste en une collection de domaines (systèmes autonomes gérés par des entités administratives différentes). Ainsi, nous avons vu que le trafic d'un usager à un autre traverse typiquement plusieurs domaines et que ces domaines entrent dans des plusieurs relations d'affaires bilatérales pour l'échange de trafic (par exemple, fournisseur-client, ou accord d'échange de trafic ou *peering*) afin d'achever une connectivité globale.

Dû à la nature de ces relations d'affaires, chaque domaine du réseau s'inquiète seulement de la performance du réseau de son propre domaine et il se trouve responsable de fournir des garanties du service pour ses clients. Du fait qu'il est difficile d'établir des relations d'affaire multilatérales impliquant plusieurs domaines, le déploiement des services de bout en bout au-delà de la connectivité *best effort*, qui exige le support de plusieurs domaines du réseau, demeure encore loin de la réalité. Tels problèmes ont entravé la transformation d'Internet courant dans une infrastructure de réseau qui est vraiment multiservice supportant la QoS (*Quality of Service*) de bout en bout.

Dans ce chapitre, nous présentons le réseau dédié de service SON (*Service Overlay Network*). Également, nous expliquons comment ce réseau peut être un moyen efficace pour fournir une QoS de bout en bout et comment il peut faciliter la création et le déploiement des services d'Internet de valeur ajoutée tels que la voix sur IP VoIP (*Voice over Internet Protocol*) et autres services émergeants et sensibles à la QoS.

2.1 Le réseau dédié de service SON

Afin de fournir une QoS de bout en bout, l'opérateur du réseau dédié de service SON dépend de relations d'affaires bien définies entre lui et les domaines du réseau natif (réseau Internet). Il achète de la bande passante avec certaines garanties de QoS de domaines individuels du réseau par le biais d'un accord de niveau de service bilatéral SLA (*Service Level Agreement*) pour construire une infrastructure logique de service de livraison de bout en bout au-dessus des réseaux de transport de données existant [Duan et al. (2003)].

En effet, SON peut réaliser ces liens logiques dans un même domaine ou système autonome en profitant de la garantie de QoS achetée de ce domaine. Donc, il peut utiliser des LSP d'un domaine MPLS pour avoir des paramètres de QoS de service stables, et ainsi de suite. SON peut également utiliser n'importe quel mécanisme de QoS utilisé dans les domaines auxquels il appartient et selon les SLA avec ces domaines. De cette façon, et avec une bonne stratégie d'implantation des nœuds Overlay dans des différents domaines et leur fournir de service d'accès avec une garantie de QoS selon des SLA, SON peut s'étendre sur une large échelle en construisant une plate-forme de livraison avec garantie de QoS de bout en bout, ainsi en couvrant plusieurs domaines ou systèmes autonomes.

Mais toujours se pose la question de fiabilité d'Internet sur l'échelle globale quand le trafic Overlay doit traverser plusieurs domaines pour arriver à l'autre bout ou sa destination. Parfois le trafic peut être acheminé par des domaines où il n'y aucune garantie de livraison avec de QoS. Ces moments imposent sur SON de prendre des mesures de prévision pour les chemins par lesquels il envoie ses trafics. Cela oblige le réseau SON à étudier l'état des chemins Overlay (constitués par plusieurs liens logiques) entre ses nœuds afin d'être capable de prendre des décisions d'acheminement.

Les usagers payent directement un fournisseur SON, par un contrat du service (un plan de service à prix fixe), pour utiliser les services de valeur ajoutée fournis par SON (ces usagers peuvent de plus payer des frais mensuels pour accéder à l'internet).

Le réseau dédié de service est rassemblé par le biais des passerelles de service (*Service Gateways*) qui effectuent des services spécifiques de transmission de données et des fonctions de contrôle (Figure 2.1).

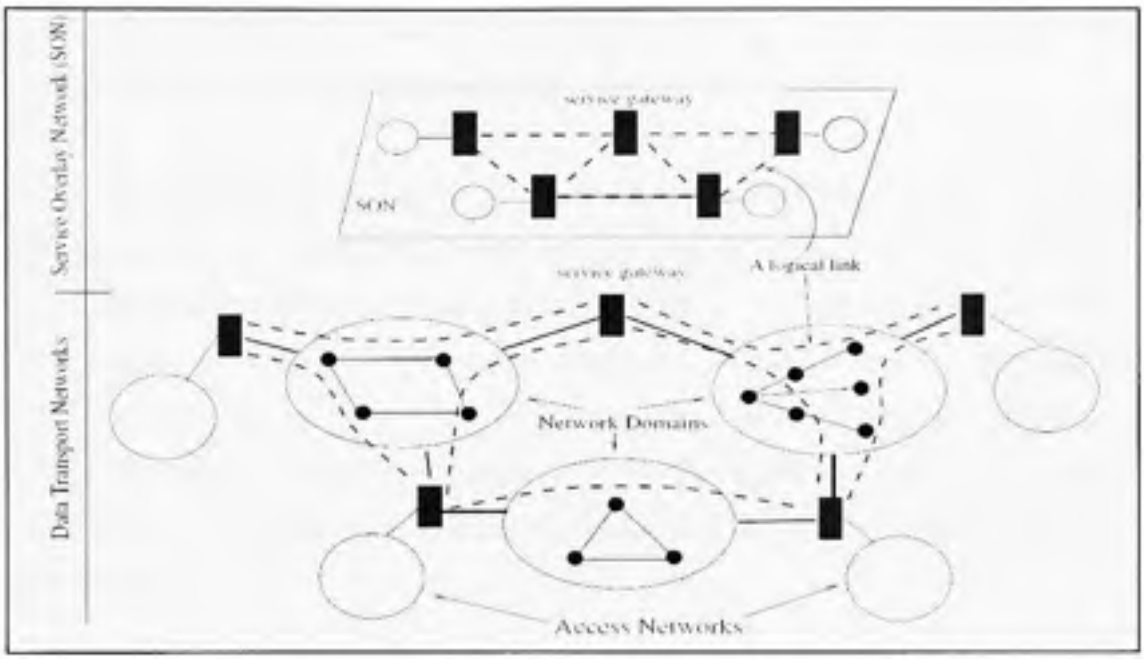

Figure 2.1 ***Réseau dédié de service SON.***

(Tiré de Z. Duan, Z. Zhang, & Y. T. Hou, 2002)

Source : Cette figure a été tirée de Duan, Zhenhai. Zhi-Li Zhang et Yiwei Thomas Hou 2003. « Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning » et correspond à la « Figure 1 An illustration of a service overlay network » présentée en page 2 dans le document original. (La référence complète du document est présentée dans la bibliographie).

La liaison logique entre deux passerelles de service est fournie par le domaine du réseau natif avec une certaine bande passante et des autres garanties de QoS. Ces garanties sont spécifiées dans un SLA entre SON et le domaine de réseau. Cette architecture évite les accords d'échange de trafic entre les domaines du réseau (*Peering*), donc elle évite les problèmes potentiels de performance associés. En se posant sur le SLA bilatéral, SON peut délivrer à ces clients des services sensibles à la QoS de bout en bout par des services spécifiques de gestion de ressource.

SON ne gagne pas son importance seulement de sa capacité à délivrer des services sensibles à la QoS de bout en bout, mais également du fait qu'il dissocie les services de l'application des services du réseau. Cela se fait par la réduction de la complexité de gestion et de contrôle de service du réseau, spécifiquement en termes de gestion et contrôle de la QoS. Les domaines de réseau sont concernés essentiellement dans le provisionnement de service de transport de données avec la gestion de la bande passante associée.

La notion de réseau dédié de service introduit un nouveau niveau d'agrégation de trafic : agrégat de service. Les domaines de réseau natif peuvent agréger le trafic en se basant sur le réseau SON auquel ce trafic appartient, et ils effectuent un contrôle de trafic et de QoS basé conformément sur le SLA correspondant. Avec cette architecture, SON est responsable d'assurer la QoS de bout en bout de ses services. Alors, SON peut déployer des provisionnements de services spécifiques, gestion de ressource et des mécanismes de contrôle de la QoS (sur les passerelles de service) afin d'optimiser ses opérations pour ses propres services.

2.2 Le routage dans le réseau dédié de service

Dans une vue simplifiée, le routage de trafic entre les nœuds dédiés de service se fait sur les passerelles de service. En d'autres termes, ces passerelles de service sont des nœuds étendues par la mise en œuvre des applications de service spécifiques appartenant au réseau dédié de service. Ces applications prennent leurs propres décisions de routage au niveau de réseau SON, entre ses nœuds et à travers les liens logiques du réseau SON. Cela permet la distinction entre l'acheminement des paquets qui se fait au niveau des nœuds IP et le traitement des applications qui signifie le routage de SON. Toutefois, ce type de routage permet aux réseaux dédiés de service de devenir rapidement le mécanisme de choix pour introduire de nouvelles fonctionnalités dans l'Internet.

Comme nous l'avons mentionné ci-dessus, le réseau dédié de service est un réseau logique construit en-dessus du réseau natif par la mise en œuvre de la logique de traitement de SON (Figure 2.2).

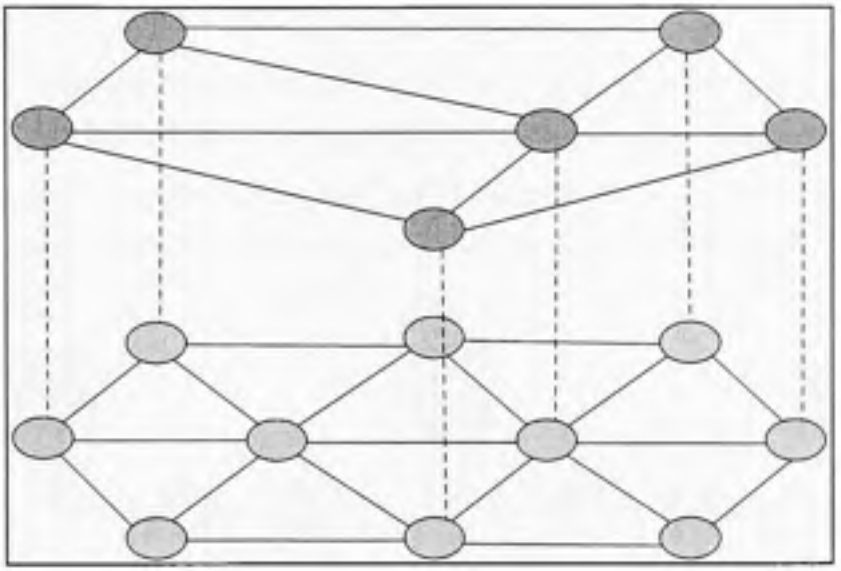

Figure 2.2 ***SON disposé en couche au dessus de réseau natif.***

(Tiré de Peterson, Larry et Bruce S. Davie, 2003)

Source : Cette figure a été tirée de Peterson, Larry et Bruce S. Davie 2003 « Computer Networks: A System Approach » et correspond à la « Figure 9.17 Overlay network layered on top of a physical network » présentée en page 681 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Alors, chaque nœud dans SON existe aussi dans le réseau fondamental ou natif. Il traite et achemine les paquets d'une manière spécifique, donc selon les besoins de flux en QoS. Plusieurs réseaux dédiés de service peuvent exister au dessus du même réseau fondamental. Chaque SON met en œuvre son propre comportement de manière spécifique et convenable aux besoins de son application et aux exigences de QoS de ses clients.

Les liens qui relient les nœuds du réseau SON sont implémentés comme des tunnels logiques à travers le réseau en-dessous. Les nœuds des deux extrémités d'un tunnel traitent

les chemins physiques multi-sauts comme s'ils étaient constitués d'un seul lien (lien logique). Le nœud dédié ajoute son entête au paquet pour le routage à l'intérieur du réseau SON, puis le paquet sera enveloppé dans des paquets IP ordinaires pour le routage dans le réseau natif.

Les nœuds IP constituant le tunnel transfèrent les paquets en se basant sur l'en-tête IP extérieur, jamais conscients que les nœuds des extrémités ont attaché une partie interne dans l'en-tête. La figure 2.3 montre trois nœuds dédiés reliés par une paire de tunnels.

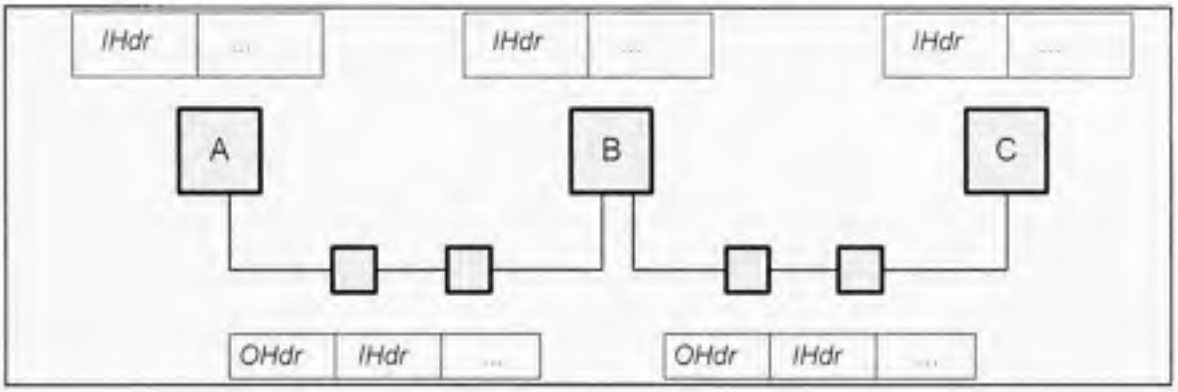

Figure 2.3 ***Tunnel de nœuds dédiés à travers des nœuds physiques.***

(Tiré de Peterson, Larry et Bruce S. Davie, 2003)

Source : Cette figure a été tirée de Peterson, Larry et Bruce S. Davie 2003 « Computer Networks: A System Approach » et correspond à la « Figure 9.18 Overlay nodes tunnel through physical nodes » présentée en page 682 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Ici, le nœud dédié B pourrait prendre une décision pour la transmission des paquets de A à C en se basant sur l'en-tête interne IHdr (*Inner header*) ensuite, il attache un en-tête externe OHdr (*Outer header*) qui identifie C comme destination dans le réseau fondamental. Les nœuds A, B et C sont en mesure d'interpréter à la fois les en-têtes internes IHdr et les en-têtes externes OHdr, alors que les routeurs intermédiaires ne comprennent que les en-têtes externes. De même, A, B et C ont à la fois des adresses dans SON et des adresses dans le réseau fondamental, mais ces adresses ne sont pas nécessairement les mêmes. En fait, les réseaux SON n'ont pas besoin d'utiliser les adresses classiques, mais ils peuvent acheminer

en fonction des URL (*Uniform Resource Locator*), noms de domaine ou même le contenu du paquet.

2.2.1 La sélection des chemins dans le réseau dédié de service

Les nœuds dédiés de service sélectionnent les chemins, par lesquels ils achemineront leurs trafics (envoyés par leurs usagers), en remplissant une seule condition qui est la provision du niveau de la QoS de bout en bout demandé. Ce niveau de QoS est prédéfini entre SON et l'usager. Pour cette raison, les nœuds dédiés de service doivent être au courant de l'état de chaque lien logique reliant chaque paire de nœuds dédiés voisins et à être capables d'estimer la QoS courante.

Donc, la sélection du chemin de passerelle de service d'entrée vers la passerelle de service de sortie se base sur des métriques de QoS données : délai de bout en bout, taux de perte et bande passante disponible. Toujours selon les besoins de trafic à envoyer. Ainsi, les nœuds de réseau SON achève ce but par l'emploi des mesures actives utilisées entre ses nœuds. Ce type de mesures (actives) ne porte pas sur l'accès aux équipements de réseau natif dans lesquels l'accès nécessite des privilèges afin que SON soit capable de collecter les statistiques de performance nécessaires dans les domaines de façon passive.

La mesure active s'agit de l'injection ou de l'envoie de séquences des paquets spécifiques de taille donnée et à un taux donné entre chaque paire des nœuds Overlay voisins. Les deux nœuds Overlay voisins coopèrent pour traiter et analyser le trafic de mesure envoyé afin d'estimer le niveau de QoS offert par le lien actuellement. Également, les autres nœuds du réseau le font, et après quelques temps tous les nœuds Overlay seront au courant des états de tous les liens logiques du réseau et ils seront capables de tracer le meilleur chemin entre n'importe quelle paire d'extrémités du réseau.

SON peut être au courant des états des liens logiques en analysant les paramètres de son propre trafic Overlay acheminé entre ces nœuds. Ce type de mesure n'est pas actif mais,

plutôt passif. Il n'est pas indépendant ni des autres trafics du réseau ni des autres événements qui peuvent se produire sur le réseau. En utilisant ce type de mesure, les nœuds Overlay ne seront capables de prendre des décisions qu'après l'envoi des flux Overlay. Parfois la décision de réacheminement ne sera prise qu'après une perte significative de trafic ou une mauvaise performance des applications génératrices.
Donc, la préférence d'éviter tels problèmes et les avantages qu'offrent les mesures actives (la prévision, l'indépendance, etc.) favorise l'usage de telles mesures.

Les mesures actives sont utilisées pour estimer les paramètres de QoS que le lien offre. Ces paramètres sont : le délai de bout en bout, la gigue, le taux de perte et la bande passante disponible.

Tous ces paramètres sont importants dans le processus d'estimation de l'état d'un lien mais, nous savons que les problèmes de plusieurs de ces paramètres provient la plupart de temps d'une raison bien connue, c'est la congestion. On dit qu'un lien est congestionné quand le taux de trafic à envoyer est proche de taux de service de ce lien. Par conséquence, si un lien est congestionné, la file d'attente grandit, le délai et la gigue s'augmentent et le taux de perte s'élève. Donc, le manque en bande passante cause les autres problèmes dans la plupart de temps (en excluant quand un lien ou un routeur tombe).

Pour SON, l'adoption du paramètre de QoS la bande passante disponible est essentiel. En effet, si les nœuds Overlay étaient capables de trouver le chemin qui offre la bande passante nécessaire pour envoyer un flux, ils pouvaient éviter les longs délais, l'instabilité de la gigue et surtout les pertes. Pour cette raison, SON utilisent les méthodes de mesure de la largeur de bande disponible dans son fonctionnement et ses algorithmes utilisent ce paramètre dans leur processus de décision et d'acheminement.

La mesure de la bande passante disponible, ainsi que le routage basé sur ce paramètre forment la partie majeure de l'objectif de ce projet. Notre but principal est d'étudier les algorithmes de routage adaptatif de réseau SON quand nous allons employons différentes

méthodes de mesure de la bande passante disponible, car chaque méthode aura son propre principe d'estimation et par conséquent sa propre performance. En effet, la performance de l'algorithme de routage adaptatif dépendra bien des caractéristiques des méthodes de mesure (exactitude, délai des mesures et estimation de la largeur de bande disponible). Ce qui nous invite à expérimenter et à étudier la performance des applications qui génèrent les flux de trafic à envoyer à travers le réseau dédié de service, car elle reflète la performance de routage, bien sûr en employant différentes méthodes de mesure (c'est-à-dire différentes performances de l'algorithme de routage) et en les comparant entre elles. Également, nous allons comparer la performance de ces applications sur les réseaux dédiés de service et sur le même réseau IP (sans la logique de réseau dédié de service) pour souligner les avantages offerts par le routage Overlay au niveau de qualité de service.

2.2.2 Les algorithmes de routage de réseau dédié de service

Notre projet se base sur l'étude des algorithmes de routage adaptatif de réseau dédié de service basé sur l'utilisation des mesures de la bande passante disponible (effectuées par les mesures actives entre les nœuds) dans le processus de sélection de chemin. Donc, la métrique principale que l'algorithme de routage prend en considération dans son calcul de meilleur chemin sera la bande passante disponible fournie par ce dernier.

Les deux algorithmes de routage adaptatif essentiels des réseaux dédiés de service SON, dont nous parlerons plus loin dans cette étude, sont : l'algorithme de routage proactif et l'algorithme de routage réactif qui suivent deux approches générales différentes.

Le premier (proactif) essaie toujours d'acheminer le flux par le chemin qui fournit le maximum de la bande passante disponible même si le chemin actuel est satisfaisant, de façon que le flux puisse éviter la congestion transitoire due de trafic traversant (*cross-traffic*) et de sa fluctuation. Tandis que, le routage réactif réachemine le trafic seulement quand le flux ne peut pas satisfaire son besoin en débit sur le chemin actuel et qu'il y a un autre chemin qui pourra fournir un niveau de bande passante plus élevé.

Le but essentiel de ce projet est l'étude de la performance des algorithmes de routage adaptatif de réseau SON employés avec différentes méthodes de mesure de la bande passante disponible, d'observer la sensibilité du comportement de ces algorithmes sur les caractéristiques de ces méthodes de mesure et comment tout ça va refléter la performance des applications. Ainsi, nous concentrons sur l'algorithme de routage adaptatif-réactif qui est plus avantageux que le proactif. En effet, le routage réactif est moins agressif que le proactif et en plus il est plus stable. Il se base sur le même historique que le proactif et utilise les mêmes échelles de temps dans sa logique de fonctionnement. Ainsi, l'instabilité de l'algorithme proactif vient du coût de la recherche permanente des chemins qui offrent le maximum de bande passante disponible pour éviter la fluctuation de trafic traversant et pour avoir une marge de sécurité en bande passante même si la performance de flux est satisfaite.

Nous nous basons dans nos études, sur une plateforme de simulations intensives. Ces simulations sont faites dans l'environnement de modélisation et de simulation de réseaux de télécommunications NS2 [Greis (2006)].

CHAPITRE 3

ÉTAT DE L'ART

Au cours de dernières années, il y a eu de nombreux efforts pour assurer la qualité de service dans l'Internet. Surtout, les architectures de services Intserv (*integrated services*) et Diffserv (*Differentiated Services*) ont été proposées pour offrir un large ensemble de services allant de garanties de flux et de délai jusqu'aux garanties par agrégat et service de priorité. Également, la technologie MPLS présente pour les opérateurs l'énorme avantage de pouvoir créer des réseaux clients complexes; elle permet une commutation rapide des paquets IP et donne la possibilité de créer des réseaux d'entreprises VPN (*Virtual Private Network*) et de gérer les critères de QoS.

Malgré ces efforts, l'Internet continue de fournir un service best-effort et une des principales raisons est l'exigence de ces propositions, c'est-à-dire que tous les éléments du réseau entre une source et une destination mettent en œuvre des mécanismes de QoS. La difficulté inhérente à changer l'infrastructure d'IP associée à l'absence de mesures d'incitation pour les ISP afin de coordonner leur déploiement a rendu cette exigence irréalisable et, en définitive, nuit à l'adoption d'IntServ, DiffServ ou MPLS.

En effet, si nous voulons réaliser des améliorations significatives de QoS qui peuvent être fournis dans l'Internet sans avoir besoin de l'appui des routeurs IP, les efforts vont alors vers les réseaux dédiés ou les réseaux Overlay, car ils présentent une solution alternative pour introduire des nouvelles fonctionnalités qui sont, soit trop lourdes à déployer dans l'infrastructure IP native, ou qui exigent de l'information qui est très difficile à obtenir au niveau IP.

Le genre le plus simple d'Overlay (dédié de service) est celui qui existe pour supporter une stratégie de routage alternative, donc pas de traitement additionnel (au niveau application) et il est fait dans les nœuds Overlay. Un exemple de routage Overlay est un VPN, il ne définit

pas une stratégie alternative ou un algorithme, mais plutôt des entrées dans la table de routage qui seront traitées par l'algorithme d'acheminement IP standard. Dans ce cas, l'Overlay utilise des tunnels IP (l'habilité de l'utilisation de ces VPN est supportée par la plupart des routeurs commerciaux).

Cependant, supposons que nous voulions utiliser un algorithme de routage que les vendeurs des routeurs commerciaux n'auraient pas inclus dans leurs produits. Dans ce cas, nous pouvons l'exécuter sur une collection d'hôtes terminaux et de tunnels via les routeurs d'Internet. Ces hôtes terminaux se comportent alors comme des routeurs dans le réseau Overlay et ils se connectent aux plusieurs voisins par des tunnels.

Puisque que l'Overlay est une façon pour utiliser des nouvelles technologies indépendantes du processus de normalisation, alors nous n'allons pas parler de standards. Nous illustrons plutôt l'idée générale d'Overlay par la description de plusieurs systèmes expérimentaux proposés récemment par les chercheurs dans le domaine des réseaux.

3.1 Resilient Overlay Networks

Anderson *et al.* (2001) ont conçu un réseau Overlay qui gagne de la popularité, c'est le réseau résilient dédié RON (*Resilient Overlay Networks*). RON essaie de déterminer des chemins alternatifs pour les applications unicast traditionnelles. Ce n'est difficile dans l'Internet à trouver trois sites A, B et C où le délai de A à B est plus grand que la somme de deux délais de A vers C et de C vers B (Figure 3.1).

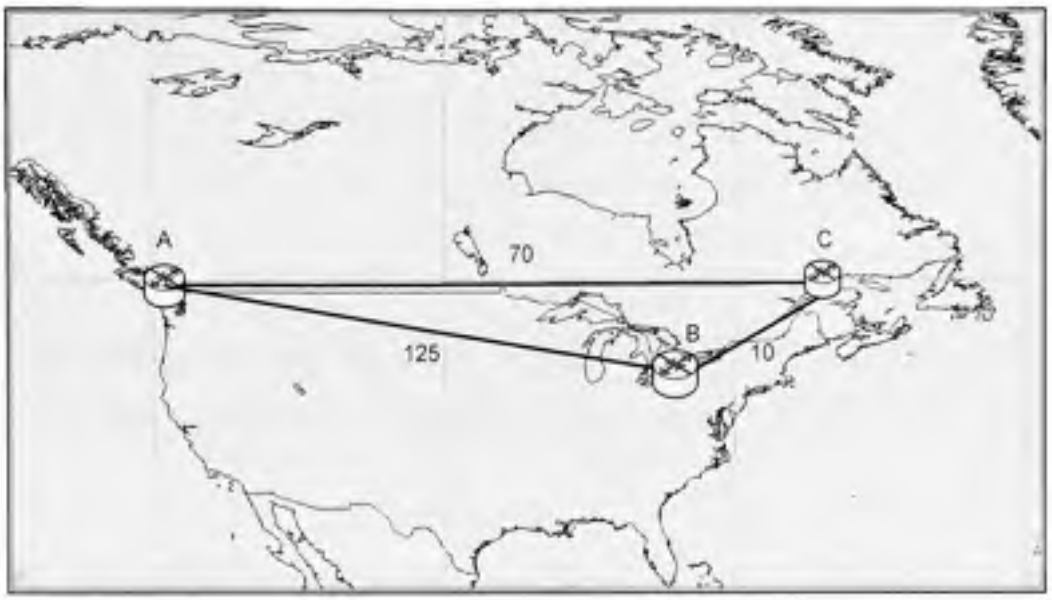

Figure 3.1 ***L'inégalité du triangle n'influe pas nécessairement sur le réseau RON.***

(Tiré de Peterson, Larry et Bruce S. Davie, 2003)

Source : Cette figure a été tirée de Peterson, Larry et Bruce S. Davie 2003 « Computer Networks: A System Approach » et correspond à la « Figure 9.21 The triangle inequality does not necessarily hold in networks » présentée en page 689 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Parfois il est préférable d'envoyer le flux de paquets à travers un nœud intermédiaire et non directement vers la destination. Tandis qu'avec le protocole BGP, les algorithmes cherchent toujours des chemins qui ne sont pas nécessairement les chemins les plus courts entre deux sites de réseau RON.

RON observe de façon continue la qualité des liens entre ses nœuds, ce qui permet de choisir le meilleur chemin et n'importe quand. RON s'étende jusqu'à quelques douzaines de nœuds. Il utilise une stratégie de sonde NxN (entre tous ses nœuds) pour mesurer trois aspects de qualité de liens : le délai de bout en bout, la bande passante disponible et la probabilité de perte entre chaque paire de nœuds. Donc, RON sera capable de choisir le meilleur chemin et de le changer si les conditions évoluent. Par conséquences, RON sera capable d'améliorer la performance des applications et à effectuer des recouvrements de pannes des liens du réseau très rapides.

3.2 OverQoS

OverQoS est une architecture de QoS basé sur le réseau dédié pour améliorer la qualité de service d'Internet. La pièce clé d'OverQoS est l'abstraction du lien virtuel avec contrôle de perte ou CLVL (*Controlled-Loss Virtual Link*). CLVL fournit des garanties de perte statistiques à un agrégat de trafic entre deux nœuds Overlay en face avec des conditions variables du réseau. En outre, elle permet aux nœuds Overlay le contrôle la bande passante et les allocations de perte entre les flux individuels au sein d'un CLVL [Subramania et al. (2003)].

OverQoS ne peut pas fournir la gamme des garanties de service offert par IntServ, mais il peut toujours apporter des améliorations de QoS utiles aux applications. OverQoS peut offrir quelques améliorations au niveau de trafic :

- le lissage des pertes dans le réseau qui a un type de perte *Bursty*,
- la hiérarchisation des paquets,
- des garanties de bande passante basées sur les statistiques,
- des garanties de pertes.

Les nœuds du réseau OverQoS se situent dans des différents systèmes autonomes (Figure 3.2), et ils communiquent entre eux à l'aide de liens virtuels qui utilisent à leur tour les chemins IP dans le réseau natif.

OverQoS souffre de quelques contraintes telles que :

- l'emplacement des nœuds, car ils doivent couvrir des nombreux domaines de routage et ils ne sont pas connectés directement aux liens congestionnés,
- le trafic traversant (*cross traffic*), car la perte est due de trafic traversant qui se varie avec le temps et qu'il est difficile à le prédire.

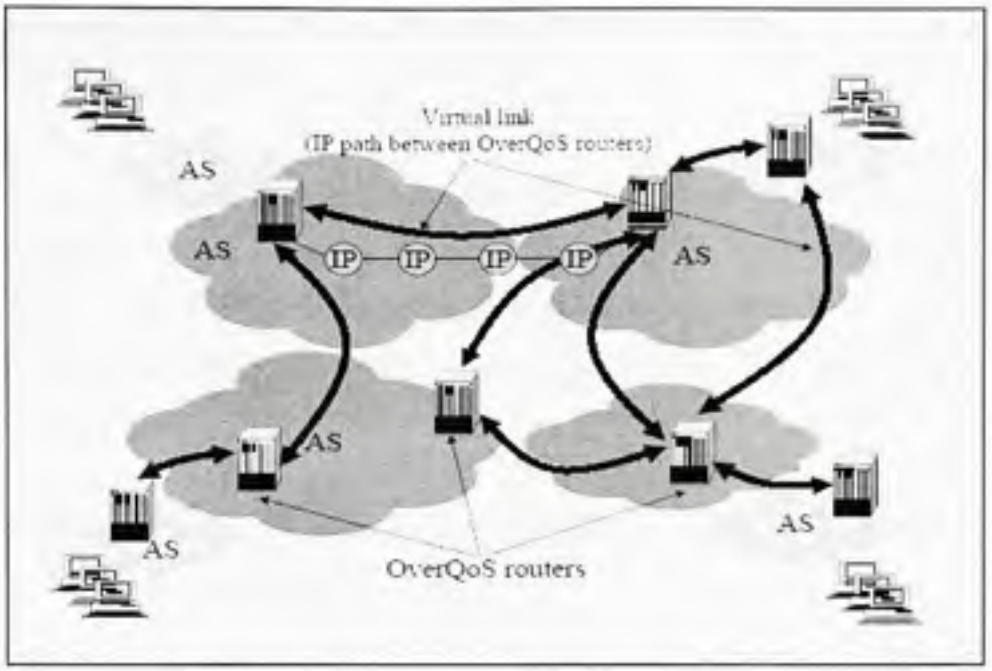

Figure 3.2 ***L'architecture du système OverQoS.***

(Tiré de Subramanian, L., Ion Stoica, Hari Balakrishnan et Randy Katz, 2004)

Source : Cette figure a été tirée de Subramanian, Lakshminarayanan, Ion Stoica, Hari Balakrishnan et Randy Katz 2004 « OverQoS: An Overlay based Architecture for Enhancing Internet QoS » et correspond à la « Figure 1 The OverQoS system architecture. OverQoS nodes in different AS's communicate with each other over virtual links using the underlying IP paths » présentée en page 2 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Le placement des nœuds est pré-spécifié et OverQoS utilise n'importe quelle approche pour déterminer les liens logiques entre eux. Il peut utiliser l'approche que le réseau RON utilise dans son processus de détermination des chemins entre ces terminaux (*End-hosts*). En effet, OverQoS essaie d'améliorer la QoS le long de ces chemins en présence des conditions variables du réseau, par exemple un niveau variable de congestion.

3.3 Les Overlays structurés (*Structured Overlays*)

Les Overlays non-structurés comme Gnutella emploient une construction d'Overlay et des algorithmes d'entretien triviaux et le meilleur qu'ils peuvent offrir sont des recherches qui sont aléatoires et peu fiables. En revanche, les Overlays structurés sont conçus pour se

conformer à une structure graphique qui permet une localisation des objets fiable et efficace (délai probabiliste délimité).

Deux questions principales se posent : comment faire correspondre les objets aux nœuds, et comment peut-on acheminer les demandes pour les nœuds qui sont responsables d'un objet donné? Alors, si nous pouvions contrôler la façon dont les objets sont distribués sur le réseau, nous pourrions être en mesure de faire un meilleur travail de recherche de ces objets à un moment ultérieur.

Une approche fondée sur le hachage vise à répandre de manière uniforme les objets sur l'ensemble de nœuds, par exemple :

Hash (x)

Return x101%

Malheureusement, s'il y a plus de 101 nœuds prêts à accueillir des objets, alors nous ne pouvons pas tirer parti de tous. D'autre part, si nous choisissons un nombre plus grand que le plus grand nombre de nœuds, alors il y aura des valeurs de x qui hache en une adresse pour un nœud qui n'existe pas.

Il y a aussi la question de translation de la valeur retournée par la fonction de hachage en une adresse IP réelle. Pour répondre à ces questions, les réseaux P2P (*Peer-to-Peer*) structurés utilise un algorithme de hachage cohérent, qui hache un ensemble d'objets x uniformément au sein d'un grand espace des *id*. La figure 3.3 visualise un espace d'*id* de 128 bits comme un cercle, où nous utilisons l'algorithme pour placer les objets et les nœuds à la fois dans le cercle :

hash(object_name) -> objid

hash(IP_addr) -> nodeid

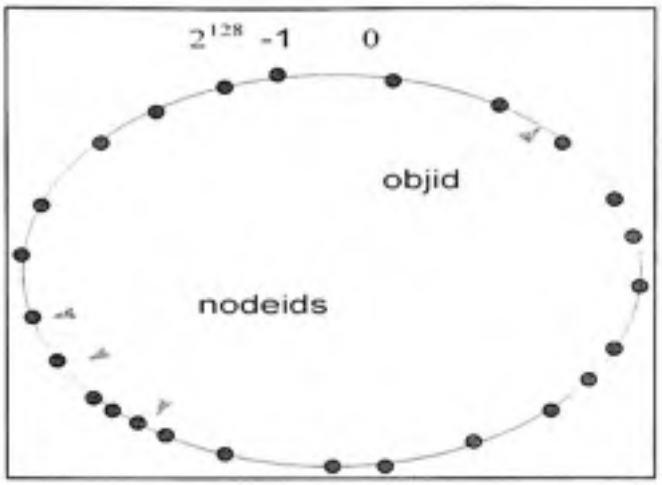

Figure 3.3 ***Nœuds et objets hachent dans un espace d'id de 128 bits.***

(Tiré de Peterson, Larry et Bruce S. Davie, 2003)

Source : Cette figure a été tirée de Peterson, Larry et Bruce S. Davie 2003 « Computer Networks: A System Approach » et correspond à la « Figure 9.23 Both nodes and objects map (hash) onto the id space, where objects are maintained at the nearest node in the space » présentée en page 694 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Du fait que l'espace de 128 bits est énorme, il est un peu probable que l'objet de hachage va exactement dans la même *id* que l'adresse IP haché par une machine. Pour tenir compte de cette improbabilité, chaque objet est maintenu sur le nœud dont l'*id* est le plus proche, dans l'espace de 128 bits, à l'*id* de l'objet.

Si un usager veut accéder un objet *x*, comment peut-il savoir quel est le nœud le plus proche dans l'id de *x* dans cet espace? L'approche utilisée par le réseau pair à pair structurés sera alors le DHT (*Distributed Hach Table*).

La table de hachage sera distribuée à tous les nœuds dans le réseau. Le processus de déplacement proche dans l'espace d'*id* jusqu'à l'arrivée à un nœud qui ne connaît pas de nœud plus proche, ce nœud est, par définition, celui qui héberge l'objet. La figure 3.4 montre que se passe pour un espace d'*id* de 28 bits

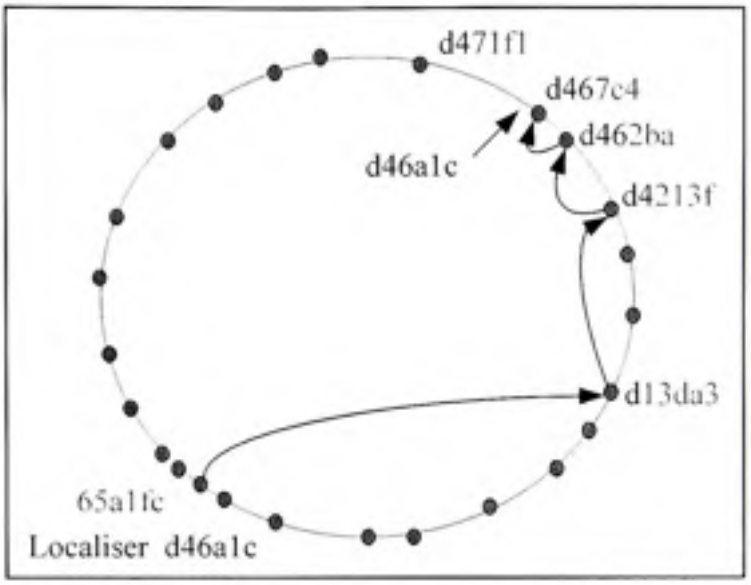

Figure 3.4 ***Localisation des objets par le routage dans le réseau Overlay pair à pair.***

(Tiré de Peterson, Larry et Bruce S. Davie, 2003)

Source : Cette figure a été tirée de Peterson, Larry et Bruce S. Davie 2003 « Computer Networks: A System Approach » et correspond à la « Figure 9.24 Objects are located by routing through the peer-to-peer overlay network » présentée en page 695 dans le document original. (La référence complète du document est présentée dans la bibliographie).

3.3.1 La procédure de routage

Chaque nœud maintient une table de routage avec les adresses IP d'un nombre restreint de grands et petits *id* de nœuds, qui est appelé l'ensemble des feuilles (*Node's Leaf sets*). La pertinence de la feuille qui est fixé une fois qu'un message est routé vers n'importe quel nœud d'un même ensemble de feuilles dans le nœud qui héberge l'objet, ce nœud peut transmettre le message directement à la destination finale. La procédure de routage est définie comme suit :

Route(*D*)

If *D* is within range of my leaf set

forward to numerically closest member in the leaf set

else

let *l* = length of shared prefix

```
    let d = value of lth digit in D's address
  if RouteTable[l, d] exists
    forward to RouteTable[l, d]
  else
    forward to known node with at least as long a prefix
    and is numerically closer than this node
```

3.4 CDN (*Content Delivery Networks*)

Quand nous parlons du téléchargement d'une page web, nous envisageons trois goulots d'étranglement potentiels (*Bottlenecks*) dans le système qui a une dorsal construite de liens OC-192 (Gbps) :

1. le premier km, connexion avec un modem de 56 kbps;
2. le dernier km, le lien qui connecte le serveur à l'Internet (le serveur peut être surchargé par des nombreuses requêtes);
3. les points d'échange (*Peering Points*), si l'usager est connecté à l'ISP A et le serveur est connecté à l'ISP B, alors la page peut être perdue au point d'échange entre A et B à cause de manque de capacité.

Le *Content Delivery Networks* est un système qui utilise la réplication pour traiter les problèmes 2 et 3. L'idée du CDN est de distribuer géographiquement des serveurs de remplacement (statique) qui actualisent (cachent) les pages maintenues dans quelques ensembles de serveurs. Les CDN ont besoin également de fournir un ensemble de redirecteurs qui acheminent les requêtes de clients vers le serveur le plus approprié (voir la figure suivante).

Les CDN utilisent plusieurs éléments pour décider comment distribuer les requêtes du client :

1. pour minimiser le temps de réponse, un redirecteur sélectionne un serveur en se basant sur la approximation du réseau;

2. pour améliorer le débit général du système, il est préférable d'équilibrer la charge entre un ensemble de serveurs;
3. le débit et le temps de réponse seront améliorés si le mécanisme de distribution prend en considération la localité, soit sélectionner un serveur qui a déjà la page dans son antémémoire.

3.4.1 Les mécanismes de redirection

La redirection peut être implémentée en utilisant le DNS pour retourner différentes adresses de serveurs aux clients. Par exemple, le serveur DNS pourrait retourner l'adresse IP d'un serveur d'hébergement de pages Web de CNN, qui est connu pour avoir la charge la plus légère.

L'utilisation de la fonction de redirection http : le client envoie un message de requête à un serveur, qui répond avec un nouveau serveur (meilleure) que le client doit contacter pour la page.

Une alternative est un proxy de Web local, qui est proche du client, peut intercepter le message de requête et rediriger le vers un serveur approprié. Comme un réseau Overlay, un proxy basé sur un redirecteur fait une décision de routage de niveau application. Il envoie les requêtes HTTP basées sur une URL, sa connaissance de localisation et de la charge d'un ensemble de serveurs. La redirection est généralement mise en œuvre indirectement en ayant un retour du redirecteur l'adresse appropriée de destination et le client contacte le serveur lui-même.

3.4.2 Les politiques utilisées par les redirecteurs pour acheminer les requêtes

Les politiques comme le *round-robin* et la sélection d'un serveur disponible au hasard ne font pas un bon travail d'abaissement du temps de la perception client. Ils ignorent la proximité et la localité du réseau. Ainsi, les redirecteurs distribués obligent les requêtes pour

la même page à aller au même serveur par l'utilisation de hachage pour adresser les URL en une petite gamme de valeurs. Dès la réception d'une URL, le redirecteur le hache, ainsi que chacun des serveurs disponibles, et trie les valeurs obtenues.

3.4.3 CARP (*Cache Array Routing Protocol*)

SelectServer(*URL*, *S*)
  for = each server s_i, in server set S
    $weight_i$ = hash(*URL*, address(s_i))
  sort weight
  for each server s_j, in decreasing order of $weight_j$
    if = Load(s_j) < threshold then
      return s_j
  return server with highest weight

L'avantage de cette approche par rapport au hachage plein et cohérent est que l'ordre du serveur est différent pour chaque URL, donc si un serveur tombe en panne, sa charge est répartie équitablement entre les autres machines [Peterson et Davie (2003)].

Enfin, il est possible d'introduire la proximité de réseau dans l'équation au moins dans deux façons différentes :

1. La première consiste à effacer la distinction entre la charge des serveurs et la proximité du réseau en surveillant combien de temps un serveur prend pour répondre aux demandes, et d'utiliser cette mesure "la charge du serveur" comme paramètre dans l'algorithme précédent (c'est-à-dire, préférer tout près / serveurs peu chargé);
2. Une deuxième approche consiste à factoriser la proximité dans la décision à l'avance, en limitant l'ensemble de serveurs candidats considérés par l'algorithme ci-dessus (S), par exemple, de sélectionner uniquement les serveurs qui ont le même

ISP ou qui sont à un certain nombre de sauts du client dans le même système autonome (AS), comme serveur candidat.

3.5 Les versions expérimentales d'IP

Les réseaux dédiés sont idéaux pour le déploiement de versions expérimentales d'IP. Par exemple, le multicast IP est une extension d'IP qui interprète les adresses de classe D comme des adresses de multicast.

3.5.1 MBone (*Multicast Backbone on the* Internet)

MBone est le réseau dédié le plus célèbre déployé sur l'Internet. MBone établit des tunnels virtuels sur l'Internet afin de connecter les réseaux qui supportent multicast de l'IP natif, en permettant une architecture du multicast globale [Anderson (2001)]. Les tunnels MBone sont configurés statiquement par des administrateurs. Le but de la connectivité de MBone est de relier des réseaux entiers par un seul tunnel MBone ou par un petit ensemble de tunnels.

3.5.2 6-Bone (*IP version 6 Multicast Backbone on the Internet*)

Le 6-Bone a été conçu pour faciliter le déploiement et de tester l'IP version 6 [Anderson (2001)]. 6-Bone est un réseau global de sites connectés et supportant l'IPv6. Dans ce réseau, quelques sites se connectent par des liens natifs parlant IPv6, et des autres sites sans liens IPv6 natifs se connectent par des tunnels configurés sur l'IPv4.

3.5.3 X-Bone

X-Bone est un projet d'infrastructure conçue pour accélérer le déploiement de réseaux dédiés qui sont basés sur IP comme MBone [Anderson (2001)]. Il découvre, configure et contrôle les ressources du réseau pour créer des réseaux dédiés sur les réseaux IP existants. X-Bone fournit une interface graphique pour l'utilisateur pour la configuration automatique

des adresses IP des extrémités et pour les noms de domaines DNS (*Domaine Name Service*), configurations dédiées simples, et permet l'entretien à distance de réseaux dédiés par sessions HTTP (*Hyper Text Transfer Protocol*) codées par SSL (*Secure Sockets Layer*).

3.5.4 Yoid (*Your Own Internet Distribution*)

Yoid (encore *Extending the Internet Multicast Architecture*) fournit une "architecture générale pour toute la distribution Internet" [Anderson (2001)]. Il unifie les avantages du multicast d'IP natif avec les avantages de déploiement et de la mise en œuvre du multicast basée sur les systèmes terminaux. Les protocoles cœurs de Yoid génèrent une topologie-arbre, pour une distribution effective de contenu, et une topologie maille, pour une distribution robuste de contenu. La conception de Yoid demande la gestion de congestion intégrée dans les liens du réseau dédié en utilisant TCP ou RTP (*Real Time Protocol*) entre les nœuds. Yoid est conçu pour fournir une "meilleure" (ou peut-être simplement un réalisable), architecture du multicast flexible.

3.5.5 ALMI (*Application Level Multicast Infrastructure*)

ALMI est une infrastructure de multicast de niveau application. Elle fournit un multicast *many-to-many* pour petits groupes (dizaine de nœuds) [Anderson (2001)]. Différemment d'*end-system multicast* et Yoid, ALMI utilise un ordonnanceur centralisé pour calculer ses arbres de distribution du multicast qui réduit la complexité de l'algorithme de la construction de l'arbre, mais elle nécessite un contrôleur central. Dans l'évaluation, l'algorithme de génération de l'arbre d'ALMI utilise dans la construction de son arbre de la distribution une maille pleine de nœuds.

3.5.6 ESM (*End System Multicast*)

Il est important de comprendre qu'ESM n'est pas comme VPN et MBone qui assument que les hôtes d'Internet (opposant aux routeurs d'Internet) participent dans l'Overlay. En plus, ces hôtes échangent des messages entre eux via des tunnels UDP (*User Datagramme*

Protocol) plutôt qu'IP. Cela fait le réseau au-dessous comme s'il est connecté complètement; du fait que chaque hôte dans l'Internet est capable à envoyer un message à tout autre hôte.

L'approche générale procède en deux étapes : d'abord, il faut construire un Overlay maillé simple au dessus d'Internet maillé complètement, puis il faudra sélectionner un arbre de multicast de cette maille (Figure 3.5 où A, B, C et D sont des hôtes). La première étape est toujours la plus critique, car une fois que nous avons choisi un Overlay maillé convenable, un algorithme multicast standard sera fonctionnel au dessus pour construire l'arbre de multicast par exemple DVMRP (*The Distance Vector Multicast Routing Protocol*).

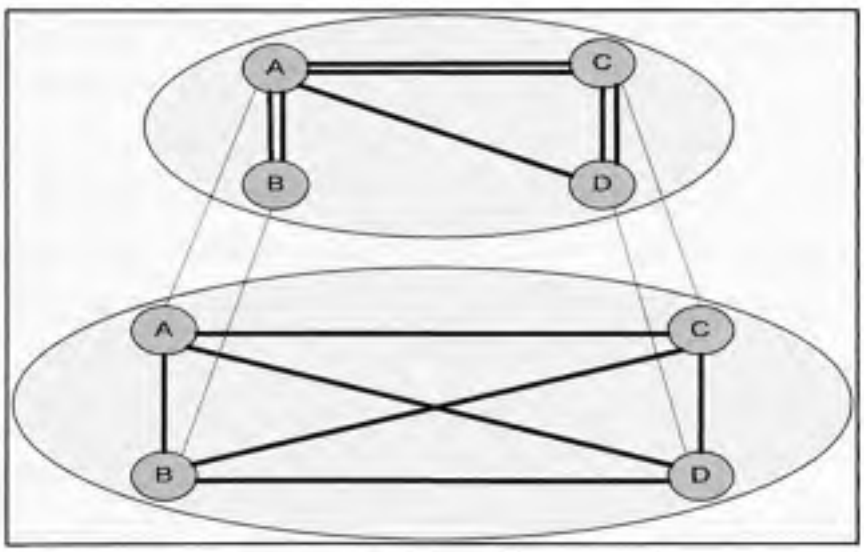

Figure 3.5 ***Arbre de multicast intégré dans une maille Overlay.***
(Tiré de Peterson, Larry et Bruce S. Davie, 2003)

Source : Cette figure a été tirée de Peterson, Larry et Bruce S. Davie 2003 « Computer Networks: A System Approach » et correspond à la « Figure 9.20 Multicast tree embedded in an overlay mesh » présentée en page 686 dans le document original. (La référence complète du document est présentée dans la bibliographie).

La clé de la construction de l'Overlay maillé intermédiaire est de sélectionner la topologie qui correspond approximativement à la topologie physique d'Internet au-dessous. Les hôtes suivent la stratégie de mesure pour construire la maille intermédiaire. Ils mesurent le délai

d'aller-retour aux autres hôtes et décident d'ajouter des liens à la maille seulement si les mesures seront convenables.

En résumé, nous aurons à la fin une maille qui est un sous-graphe d'Internet connecté complètement, et qui peut avoir une performance sous-optimale parce que :

1. la sélection initiale du voisin ajoute des liens aléatoires à la topologie;
2. la réparation de la partition devait ajouter des bords qui sont essentiels pour le moment mais ne le sont pas à long terme;
3. l'adhésion au groupe peut changer à cause des arrivées et des départs dynamiques;
4. Les conditions du réseau au-dessous qui peuvent changer.

Le système évalue continuellement chaque nœud (*Edge*) et conséquemment il en ajoute des nouveaux ou en enlève quelques-uns.
Pour ajouter un nouveau nœud, chaque nœud *i* sonde périodiquement un membre aléatoire *j* qui n'est pas connecté actuellement à la maille, et il mesure la latence d'aller-retour du nœud (*i, j*), puis il évalue l'utilité de l'ajout de ce nœud. Si cette utilité est plus qu'un certain seuil, le lien (*i, j*) sera ajouté à la maille. L'évaluation de l'utilité de cet ajout est défini comme suit :

```
EvaluateUtility(j)
  utility = 0
  for each member m not equal to i
    CL = current latency to node m along route through mesh
    NL = new latency to node m along mesh if edge (i, j) is added
    If (NL < CL) then
      Utility += (CL - NL)/CL
    return utility
```

La décision d'enlever un nœud est similaire, à l'exception que chaque nœud *i* calcule le coût de chaque lien vers son voisin courant *j*, il prend alors le voisin qui a le moindre coût et il le laisse si le coût tombe au-dessous de certain seuil :

EvaluateCost(*j*)

$Cost_{ij}$ = number of members for which *i* uses *j* as next hop

$Cost_{ji}$ = number of members for which *j* uses *i* as next hop

Return max($Cost_{ij}$, $Cost_{ji}$)

Nous avons discuté dans ce chapitre les réseaux Overlay structurés, les CDN et les réseaux liés aux versions expérimentales d'IP. Ces types de réseau n'ont pas une grande relation avec le travail de ce mémoire mais, cela était pour enrichir le contenu de l'état de l'art.

Il y a aussi des autres travaux qui traitent la conception du réseau dédié de service. Parmi eux il y a Clark *et al.* (2006) qui discutent les réseaux Overlay et ses impacts sur le futur Internet. Jinliang et Ammar (2006) traitent la question de reconfiguration d'une topologie dynamique pour SON. Gu *et al.* (2003) s'intéressent dans la gestion de provisionnement générique de QoS pour les applications en se basant sur les SLA. Li et Mohapatra (2004) ont introduit le principe de courtier Overlay et conçu un protocole de routage pour chercher le chemin Overlay qui a une QoS satisfaisante tout en essayant d'équilibrer le trafic Overlay entre dans le réseau Overlay.

La consultation de plusieurs autres travaux ont contribué encore à enrichir des idées très importantes et à éclaircir plusieurs sujets et concepts utilisés dans le domaine du réseau dédié de service. Parmi eux, Li et Mohapatra (2004), Nakao et al. (2003), Seetharaman et Ammar (2005), Mélin (2001) et Tran et Dziong (2008).

CHAPITRE 4

LE ROUTAGE DYNAMIQUE DU RÉSEAU DÉDIÉ DE SERVICE SON

Le chapitre précédant nous a exposé les différentes architectures Overlay avec leurs propres modes de routage et de livraison. En plus, le chapitre deux a expliqué le principe du réseau dédié de service qui est basé spécifiquement sur les algorithmes de routage adaptatif qui sont basés à leur tour sur les méthodes de mesure actives de la largeur de bande disponible. Cela nous fait un point de départ pour commencer nos études de routage dynamique de réseau SON dans ce chapitre.

4.1 Le modèle de routage

Nous considérons donc deux couches d'une infrastructure de réseau : le réseau natif (Internet) et le réseau virtuel dédié de service. Le réseau natif comprend des systèmes finaux, des routeurs, des liens physiques et des fonctionnalités de routage. Il fournit une livraison *best effort* pour délivrer les datagrammes entre ses nœuds. D'autre part, le réseau dédié de service est constitué d'un sous-ensemble de nœuds de la couche native (routeurs et/ou terminaux) interconnectés par des liens dédiés (logiques) pour fournir des services améliorés. Les liens de réseau SON sont des liens virtuels dans le sens où ils sont des tunnels IP au-dessus de réseau natif. C'est-à-dire, les paquets Overlay sont encapsulés dans des datagrammes IP et envoyés d'un nœud Overlay à un autre par le réseau natif.

La figure 4.1 est un exemple d'un réseau Overlay construit au-dessus d'un réseau natif. Notez que, du fait que les liens Overlay sont virtuels, la topologie du réseau Overlay peut être une maille permettant un maximum de souplesse dans le choix des chemins Overlay.

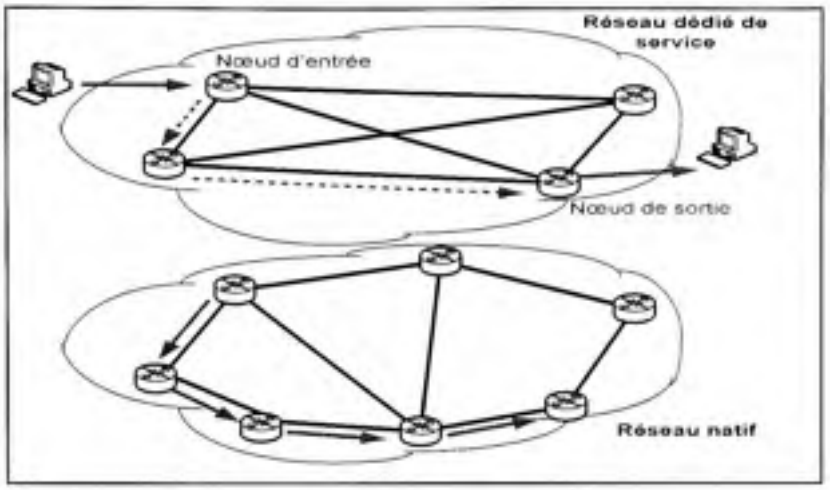

Figure 4.1 ***Les deux couches : SON et réseau natif.***

(Tiré de Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar, 2006)

Source : Cette figure a été tirée de Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar 2006 « Dynamic overlay routing based on available bandwidth estimation: A simulation study » et correspond à la « Figure 1 Overlay and native network layers » présentée en page 5 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Un service important, que les réseaux Overlay fournissent, est la sélection de chemin dynamique fondée sur des objectifs de performance donnés. La performance d'un chemin peut être une fonction du délai, taux de perte et/ou la bande passante disponible du chemin parmi d'autres métriques. En outre, différentes classes de trafic peuvent être associées avec des différentes métriques de performance d'un chemin. Un flux Overlay arrive sur un nœud Overlay d'entrée (*Ingress node*) et est dirigé vers un nœud Overlay de sortie donné (*Egress node*). Lors de l'arrivée de flux Overlay, le nœud d'entrée détermine le meilleur chemin vers le nœud de sortie en se basant, idéalement, sur l'état actuel et sur la performance des liens Overlay (dénommé état de lien Overlay). L'information du chemin Overlay choisi est alors incluse dans l'entête de chaque paquet (routage à la source) et le paquet est transmis à la séquence de nœuds Overlay correspondante.

4.1.1 La robustesse du réseau SON

Le réseau dédié de service fournit une robustesse contre les défaillances et contre les variations de charge sur les chemins en obligeant le nœud d'entrée d'un flux Overlay actif à vérifier continuellement l'existence d'un meilleur chemin à la fin de chaque période P_u (*update Period*) de mise à jour de chemin et pendant la durée de vie du flux (Figure 4.2) :

1. Si l'algorithme employé est réactif, le nœud d'entrée vérifie seulement si la bande passante disponible fournie par le chemin actuel est suffisante pour délivrer le flux actif, et change le chemin actuel autrement;
2. Si l'algorithme employé est proactif, le nœud d'entrée change le chemin directement vers un autre qui peut fournir de plus de la bande passante même si la bande passante fournie actuellement est suffisante.

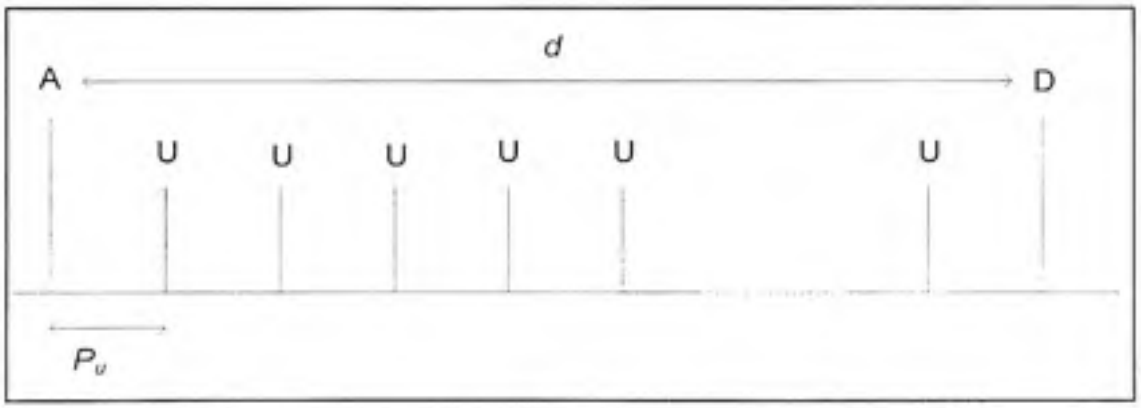

Figure 4.2 ***Événements d'un flux Overlay et leurs échelles de temps.***

(Tiré de Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar, 2006)

Source : Cette figure a été tirée de Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar 2006 « Dynamic overlay routing based on available bandwidth estimation: A simulation study » et correspond à la « Figure 1 Overlay flow events and the related time scales. *A*, *U* and *D* are the flow arrival, path update, and flow departure events, respectively. *d* is the flow duration and P_u is the path update period » présentée en page 6 dans le document original. (La référence complète du document est présentée dans la bibliographie).

A, U et D sont respectivement l'arrivée d'un flux, la mise à jour du chemin et le départ du flux. La durée de flux est représentée par d tandis que la période de mise à jour du chemin est représentée par P_u.

4.1.2 La sélection dynamique du chemin

Pour effectuer une sélection dynamique de chemin, les nœuds Overlay ont besoin d'effectuer des mesures d'état de lien et de diffuser de cet état. L'état de lien Overlay est l'entrée de l'algorithme de routage et il peut être représenté par un ensemble de métriques de performance telles que le délai, le taux de perte et la bande passante disponible. Dans ce travail, nous nous concentrons exclusivement sur la bande passante disponible. Bien sûr, il est possible aussi de restreindre l'algorithme de sélection des chemins par des contraintes supplémentaires comme le délai du chemin ou le taux de perte. La bande passante disponible, aussi connue sous le nom de la capacité résiduelle d'un lien natif, est définie comme la capacité de la liaison moins la charge moyenne de trafic. La bande passante disponible d'un lien Overlay (ou chemin natif), d'autre part, est la bande passante disponible minimale de tous les liens natifs qui composent ce lien Overlay (ou chemin natif).

Contrairement à la bande passante disponible d'un lien natif, qui peut être facilement mesurée passivement par le routeur correspondant, la bande passant disponible des liens Overlay ne peut jamais être estimée passivement par les nœuds Overlay. Au lieu de cela, la bande passante disponible d'un lien Overlay doit être mesurée par des techniques de sonde active de bout en bout et effectuées par les nœuds Overlay. Les développements récents dans l'estimation de la bande passante disponible de bout en bout nous ont fourni des outils et des techniques qui peuvent estimer la disponibilité de la bande passante d'un chemin de réseau. Ces techniques sont basées sur des flux spéciaux de paquets de sonde qui peuvent identifier d'une façon non-intrusive leur taux maximal qui ne provoqueront pas de congestion dans le chemin.

La latence des techniques de mesure actuelles varie de quelques dizaines de millisecondes à des dizaines de secondes. Cela dépend du fait que les outils fonctionnent en permanence ou si elles se lancent dans un mode une fois mesurer et terminer. Leur précision dépend de la nature de trafic qui circule (*burstiness*), de nombre de liens goulots d'étranglement qui ont le minimum de la bande passante le long du chemin et des erreurs relatives de mesure (de 10 à 30% doivent être encore prévues).

Chaque nœud Overlay mesure la bande passante disponible des chemins vers ses voisins (nœuds Overlay). Périodiquement, les informations de l'état de lien qui sont générées par ces mesures seront diffusées à tous les autres nœuds Overlay. La base de données des états de liens d'un nœud Overlay est actualisée lors de la réception de nouvelles informations sur l'état d'un lien donné. À notez que les mesures et la diffusion de l'état de lien sont assurées indépendamment de n'importe quel événement lié à n'importe quel flux.

4.1.3 Les échelles de temps dans le routage Overlay

Il existe trois échelles de temps importantes impliquées dans la mesure de la bande passante disponible et dans la diffusion de l'état de lien : le délai de la mesure (D_m), la période d'actualisation de l'état de lien (P_r) et le délai de diffusion (D_d). Le délai de la mesure est le temps nécessaire pour générer une nouvelle estimation de la bande passante disponible, la période de l'actualisation de l'état de lien est l'intervalle de temps entre les mises à jour consécutives de l'état local de lien. Notez que la période de l'actualisation ne peut être inférieure au délai de la mesure, mais elle pourrait être plus grande pour réduire la surcharge de la diffusion de l'état de lien. La fin de la période de l'actualisation de l'état de lien est déterminée par la fin de la période de la dernière mesure. Le délai de la diffusion est la période de temps nécessaire pour le nouvel état de lien généré par le *i*ième nœud pour atteindre le *j*ième nœud Overlay. Nous supposons que (D_m) et (P_r) sont constants, tandis que D_j^v varie aléatoirement pour chaque paire (i, j) des nœuds Overlay (Figure 4.3).

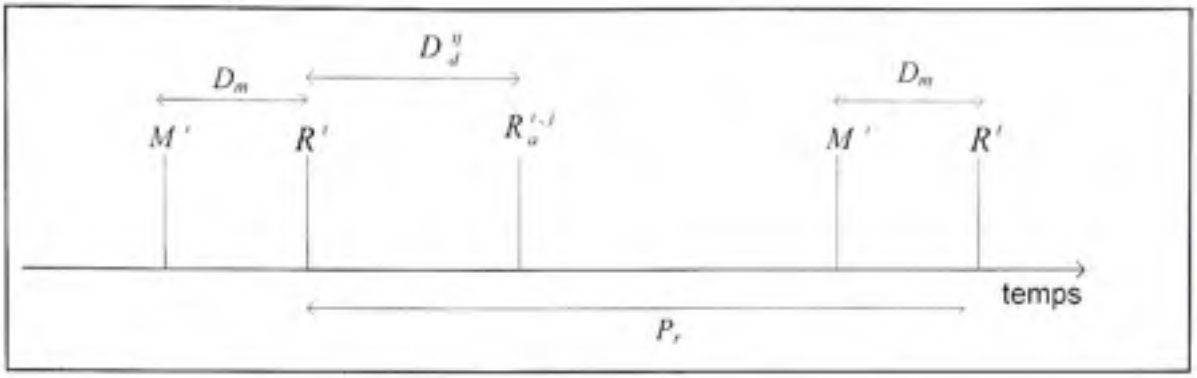

Figure 4.3 ***Les échelles de temps pour mesurer et disséminer l'état de lien du nœud i.***

(Tiré de Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar, 2006)

Source : Cette figure a été tirée de Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar 2006 « Dynamic overlay routing based on available bandwidth estimation: A simulation study » et correspond à la « Figure 3 Time scales for the measurement and dissemination of overlay link-state at the *i*'th overlay node. M^i represents the start of an avail-bw measurement for all the egress overlay links of the *i*'th overlay node. $R_a^{i,j}$ is a link-state refresh event, and it takes place at the end of the last avail-bw measurement. $R_a^{i,j}$ represents the arrival of the new link-state from overlay node *i* to node *j* » présentée en page 7 dans le document original. (La référence complète du document est présentée dans la bibliographie).

M^i représente le début de mesure de la bande passante disponible de tous les liens de sortie pour le nœud Overlay *i*. R^i est l'événement de l'actualisation de l'état de lien et il est déclenché à la fin de la dernière mesure de la bande passante disponible. $R_a^{i,j}$ représente l'arrivée de nouvel état du lien de nœud *i* au nœud *j*.

4.2 Les algorithmes de routage Overlay

La bande passante disponible de chaque lien Overlay $l \in L$ (où L est l'ensemble des liens Overlay), est représentée par $b(l)$. Un chemin Overlay p est une séquence d'un ou de plusieurs liens Overlay et sa bande passante disponible $b(p)$ est définie par : $b(p) = \min_{l \in p} b(l)$

Le flux Overlay représente l'unité de trafic pour le routage Overlay et tous les paquets de ce flux seront acheminés par le même chemin déterminé pour ce flux et il possède quatre

paramètres : $f = (n_i, n_e, d, r)$ où n_i et n_e représentent respectivement les nœuds Overlay d'entrée et de sortie du flux, tandis que d représente la durée du flux. Le débit maximal que le flux peut achever est r. Le débit d'un flux multimédia peut être limité par le taux de la meilleure qualité de l'encodeur. Dû à l'insuffisance de ressources du réseau, le débit réel d'un flux peut être inférieur à son taux maximal r (qu'il peut atteindre). Donc, a représentera la valeur courante atteinte du débit $a \leq r$.

Quand nous comparons un chemin par lequel un flux est actuellement acheminé, avec un autre nous devons prendre en considération la charge que le flux impose déjà sur l'ancien. Pour ce faire, nous introduisons le paramètre m qui représente une marge. Donc, la marge m pour un flux f sur un lien Overlay l est défini par la formule suivante si f est routé sur l.

$$m(f,l) = b(l) + a \tag{4.1}$$

Si f n'est pas routé sur l, nous utilisons alors la formule 4.2 pour calculer la marge.

$$m(f,l) = b(l) \tag{4.2}$$

De façon similaire à celle de la bande passante disponible, la formule 4.3 définie et calcule la marge d'un chemin comme étant la marge minimale de tous les liens le long de ce chemin Overlay.

$$\boxed{m(f,p) = \min_{l \in p} m(f,l)} \tag{4.3}$$

À noter que la marge $m\ (f, p)$ du chemin p est égal à la bande passante disponible $b(p)$ si le flux f n'est pas routé sur p : $\{m = b(p)\}$. Autrement, la marge est plus grande que la bande passante disponible par le débit du flux achevé a : $\{m = b(p) + a\}$.

En ce qui concerne les algorithmes de routage adaptatifs pour les réseaux SON, nous considérons deux plans dynamiques pour la sélection de chemin Overlay : le routage Overlay proactif et le routage Overlay réactif.
La différence majeure entre les deux algorithmes c'est le fait qu'avec l'algorithme proactif le flux sera commuté ou routé vers le chemin qui aura la marge maximale de bande passante après à la fin de chaque période de mise à jour de chemin. Tandis que, avec l'algorithme réactif, le flux reste sur son chemin actuel si son débit maximal r est toujours réalisé (flux satisfait). Si la demande de flux n'est pas satisfaite il sera acheminé par un autre chemin offrant une marge maximale. À noter que, avec les deux algorithmes, le flux sera initialement routé par le chemin qui aura la marge maximale.

L'intuition derrière le routage proactif, c'est que le chemin avec la marge maximale peut fournir un flux avec une limite plus large de sécurité pour éviter la congestion transitoire due à la variation de la charge de trafic circulant et sa fluctuation, les erreurs de mesure et de l'état de lien fatigué.

Tandis que, avec le routage réactif le flux doit demeurer sur son chemin courant si ce chemin est satisfaisant. Ici, les changements des chemins seront moins fréquents ce qui conduit à un routage Overlay plus stable.

4.2.1 L'algorithme de routage réactif

Comme nous avons mentionné ci-dessus, la différence majeure entre les deux algorithmes c'est que l'algorithme proactif achemine le flux Overlay par le chemin qui aura une marge de bande passante plus grande que celle du chemin actuel à la fin de chaque période de mise à jour de chemin. C'est-à-dire, le flux sera acheminé par un autre chemin Overlay qui offre une marge plus grande même si le chemin actuel est satisfaisant. Cela conduit à un changement de chemins fréquent, car l'algorithme cherche toujours la plus grande marge et achemine le flux par le chemin correspondant même si ce flux est satisfait. Ainsi, le nombre de tels chemins alternatifs peut augmenter avec l'augmentation des nœuds Overlay dans le

réseau, ce qui peut conduire à des changements très fréquents et par conséquent un routage non stable.

L'algorithme réactif ne change pas le chemin tant que ses flux sont satisfaits et cela le rend plus stable que le proactif. Il s'occupe toujours à acheminer les flux Overlay en faisant une meilleure performance que le proactif qui est toujours occupé dans les recherches des chemins qui peuvent être très longs et composés de nombreux nœuds Overlay intermédiaires. Donc, l'algorithme proactif utilise les ressources du réseau de plus en diminuant la bande passante disponible même s'il offre une marge plus grande que celle offerte par le réactif.

Zhu *et al.* (2006) montrent que nous pouvons réaliser la marge maximale de la bande passante disponible pour les deux algorithmes en acheminant à travers un seul nœud Overlay intermédiaire, donc si nous allons forcer le routage à acheminer à travers un seul nœud Overlay intermédiaire nous pouvons atteindre la marge maximale même avec l'algorithme réactif. Même chose pour le débit maximale de flux Overlay achevé ou servi, nous pouvons l'atteindre en acheminant à travers un seul nœud intermédiaire pour les deux algorithmes, en mentionnant qu'avec l'algorithme réactif le débit servis sera plus grand qu'avec l'algorithme proactif qui cherche et change toujours les chemins maximisant la bande passante disponible. De même, il a été démontré que si le nombre des nœuds intermédiaires dépasse deux, nous n'allons pas avoir une amélioration de performance de routage ni une augmentation de marge de bande passante disponible ou de débit Overlay servi de plus.

L'algorithme réactif est moins agressif que l'algorithme proactif même si tous les deux algorithmes se basent dans leurs calculs et décisions sur le même historique produit par les mesures actives de la bande passante disponible qui sont toujours fonctionnelles dans l'arrière-plan. Les deux algorithmes produisent presque le même trafic de mesure mais ils se différent seulement par la recherche des chemins qui maximise la marge de la bande passante disponible.

Toutes ces raisons nous conduisent à favoriser et à utiliser l'algorithme réactif dans les simulations de ce mémoire car il est plus stable (changement de chemins moins fréquent), plus efficace (plus de débit Overlay achevé et servi) et il peut offrir une marge de sécurité acceptable en routant à travers un seul nœud intermédiaire. L'algorithme de sélection de chemins dynamique qui sera étudié dans nos simulations sera le réactif et il est illustré par le pseudo-code suivant :

INPUT :

$f = (n_s, n_e, d, r)$: Le flux Overlay considéré;

$P = \{p_i\}$: ensemble de chemins alternatifs de n_s vers n_e ;

a : débit achevé pour de f;

OUTPUT :

Chemin sélectionné p' ;

routage-réactif et $a < r$

Actualiser la marge $m(f, p)$ pour tous $p \in P$;

$p' = \arg\max_{p_i \in P} m(f, p_i)$;

Acheminer f sur chemin p' ;

CHAPITRE 5

LA MESURE DE LA BANDE PASSANTE DISPONIBLE

Le chapitre précédant a expliqué les algorithmes de routage adaptatif de SON, les échelles de temps et comment se fait la sélection dynamique des chemins. Il a également justifié l'utilisation de l'algorithme réactif. Il reste maintenant à décrire et expliquer les techniques de mesure de la bande passante disponible afin de choisir quelques une pour les utiliser avec l'algorithme de routage afin d'observer l'impact de chaque méthode sur la performance de routage. Les méthodes de mesure choisies seront employées pour achever les mesures actives et pour fournir l'algorithme de routage par son paramètre principal de calcul.

5.1 Techniques d'estimation de la bande passante disponible

La diversité des chemins dans les réseaux dédiés de service et le fait que l'algorithme de routage porte essentiellement sur la métrique de la bande passante disponible créent un besoin pour estimer cette bande passante sur ces chemins afin de choisir la meilleure route. Ainsi, il est nécessaire et indispensable de bien l'estimer de bout en bout. En outre, dans un réseau Overlay, on suppose la coopération de l'expéditeur et du récepteur (deux nœuds Overlay qui estiment la bande passante disponible entre eux), qui est nécessaire pour la plupart des techniques de sonde.

La largeur de bande disponible d'un chemin est définie comme la capacité résiduelle moyenne (inutilisée) dans ce chemin. Le lien avec le minimum de la bande passante disponible dans ce chemin s'appelle le lien étroit (*tight link*), tandis que le lien avec la capacité minimale est appelé le lien serré (*narrow link*). La largeur de bande disponible d'un chemin est égale à la bande passante disponible du lien étroit.

Nous montrons dans la figure 5.1 que la capacité du chemin est déterminée par le lien avec le minimum de capacité qui est le lien serré (*Narrow link*). Donc, nous avons $C = C_j$.

Tandis que, la bande passante disponible de ce même chemin est déterminée par le lien avec le minimum de capacité inutilisée qui est le lien étroit (*Tight link*), en d'autres mots le goulot d'étranglement (*the bottleneck*). La bande passante disponible de cet exemple sera égale à A.

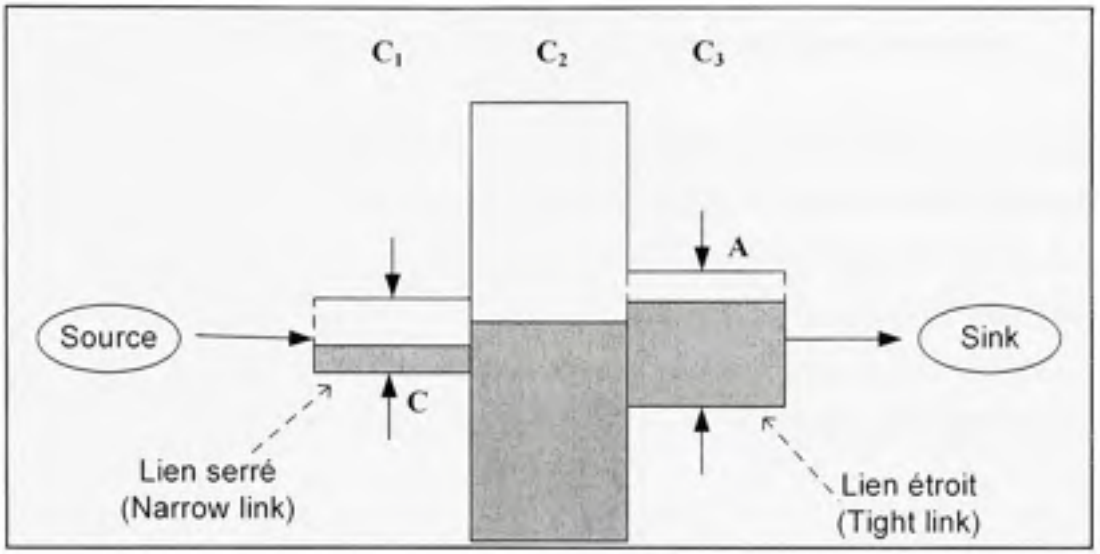

Figure 5.1 ***Le goulot d'étranglement (the bottleneck).***

(Tiré de, Dovrolis Constantine, 2006)

Source : Cette figure a été tirée de Dovrolis Constantine 2006 « Measurement tools for the capacity and load of Internet paths » et correspond à la « Figure 1 The capacity of a path is determined by the link with the minimum capacity (narrow link). The available bandwidth of a path is determined by the link with the minimum unused capacity (tight link) » présentée en page bw.html sur le site web de l'auteur. (La référence complète du document est présentée dans la bibliographie).

En effet, à tout moment un lien est en train de transmettre des paquets à la vitesse ou capacité maximale ou il est inactif; nous pouvons calculer la largeur de bande disponible moyenne d'un lien pendant un intervalle de temps T en appliquant la formule suivante.

$$A_i(t,T) = \frac{1}{T}\int^{t+T} (C_i - \lambda_i(t))dt \qquad (5.1)$$

Où $A_i(t,T)$ est la bande passante disponible sur le lien i et pendant t. C_i est la capacité du lien, λ_i est son trafic instantané. Alors, la largeur de bande disponible, le long d'un chemin composé de plusieurs liens physiques, sera la largeur de bande disponible minimale de tous les liens traversés.

5.2 Classes des techniques de mesure de la largeur de bande disponible

Les techniques de mesure et d'estimation de la largeur de bande disponible se divisent en deux classes principales. Ces deux classes se distinguent en fonction de deux modèles principaux : le modèle de sonde avec écart de temps PGM (*Probe Gap Model*) et le modèle de taux de sonde PRM (*Probe Rate Model*) ou SLoPS (*Self-Loading Periodic Stream*) [Lao et al. (2006)]. Un autre travail pour analyser et évaluer la performance des méthodes d'estimation de la bande passante disponible a été fait par Guerrero et Labrador (2006).

5.2.1 Le modèle de sonde avec écart de temps PGM

Ce modèle est connu aussi sous le nom de sonde directe. Il exploite les informations dans le temps (*Gap*) séparant l'arrivée de deux paquets de sonde successifs au récepteur. Une paire de paquets est envoyée avec un décalage de temps Δ_{in} atteint le récepteur avec un autre décalage Δ_{out}.

En supposant un seul lien étroit (*Bottelneck*) et que la file d'attente de ce lien ne devienne pas vide entre le départ du premier paquet de la paire et l'arrivée du seconde, alors Δ_{out} sera le temps mis par le serveur de ce lien étroit pour transmettre le deuxième paquet de la paire avec le trafic traversant (*cross-traffic*) qui est arrivé pendant Δ_{in} (Figure 5.2).

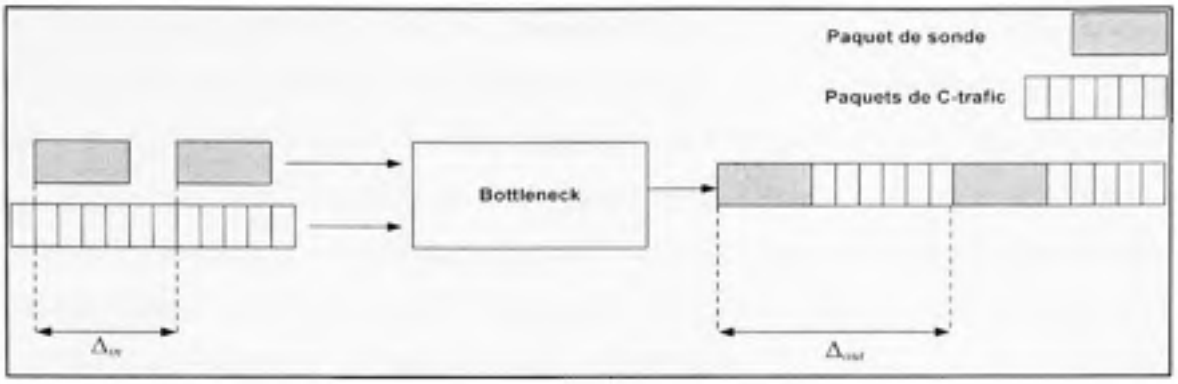

Figure 5.2 ***Le modèle de mesure et d'estimation PGM.***

(Tiré de Strauss, Jacob, Dina Katabi et Frans Kaashoek, 2003)

Source : Cette figure a été tirée de Strauss, Jacob, Dina Katabi et Frans Kaashoek 2003 « A Measurement Study of Available Bandwidth Estimation Tools » et correspond à la « Figure 2 The Probe Gap Model (PGM) for estimating available bandwidth » présentée en page 2 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Donc, le temps nécessaire pour transmettre le trafic traversant est $\Delta_{out} - \Delta_{in}$. Ainsi, la formule 5.2 calcule le taux du trafic traversant.

$$\frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}} \times C \tag{5.2}$$

Où C est la capacité maximale du goulot d'étranglement. Nous pouvons donc calculer la bande passante disponible en utilisant la formule suivante.

$$\boxed{A = C \times \left(1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}}\right)} \tag{5.3}$$

5.2.2 Le modèle PRM ou SLoPS

Le modèle PRM ou le SLoPS se base sur le principe de l'envoi des flux de trafic de sonde pour surcharger le lien afin de détecter la bande passante disponible. Ainsi, si un nœud envoie un trafic de sonde à un taux inférieur à la largeur de bande disponible le long d'un

chemin, le taux d'arrivée de ce trafic de sonde au niveau du récepteur correspondra à (sera le même que) son taux au niveau de l'expéditeur. En revanche, si le trafic de sonde est envoyé à un taux supérieur à la bande passante disponible, alors les files d'attente vont s'accumuler à l'intérieur du réseau et le trafic de sonde sera retardé. En conséquence, le taux d'arrivée des sondes au récepteur sera inférieur à leur taux d'expédition. Ainsi, un nœud peut mesurer la bande passante disponible par la recherche de point de tournage décisif auquel les taux de l'envoie et de la réception commencent à se correspondre.

5.3 Les méthodes de mesure choisies

Il est évident que la performance de l'algorithme de routage dynamique de réseau SON dépend de façon critique de caractéristiques (exactitude, rapidité, implémentation, etc.) des méthodes de mesure utilisées dans le processus de mesure de la bande passante disponible. Donc, la qualité de la mesure se reflète sur la qualité de routage. Pour cette raison, nous avons décidé d'adopter plus d'une méthode de mesure dans notre travail et les intégrer dans l'algorithme de routage pour étudier le comportement de l'algorithme avec chacune d'elles.

Nous avons choisi une méthode pour chaque classe des méthodes de mesure citées ci-haut, car chaque classe a sa logique de fonctionnement et a ses effets sur le réseau et sur l'algorithme lui-même. Cela nous offre un milieu de comparaison de performance de l'algorithme fonctionnel avec chacune de ces méthodes et par conséquent une comparaison des méthodes de mesure et de qualités de leurs résultats.

Les deux méthodes sont SPRUCE (modèle PGM) et PathLoad (modèle PRM) :

1. SPRUCE est une méthode simple qui génère un petit volume de trafic de sonde relativement. D'après les expériences elle se compte parmi les meilleures qui estiment la bande passante disponible [Strauss et al. (2003)];
2. PathLoad est plus compliquée que SPRUCE mais elle offre des bonnes estimations de la bande passante disponible de bout en bout sans aucune affectation des autres

connexions et sans génération d'un volume significatif de trafic [Jane et Dovrolis (2002)].

Les deux approches PGM et PRM assument les hypothèses suivantes (ces hypothèses sont nécessaires pour l'analyse, mais les outils pourraient encore être utiles même si certaines de ces hypothèses ne sont pas respectées.) :

1. Une file d'attente FIFO (*First In, First Out*) dans tous les routeurs sur le chemin;
2. Le trafic traversant suit un modèle fluide (c'est-à-dire que les paquets de trafic traversant ont des tailles très petites);
3. Les taux moyens de trafic traversant changent lentement et ils sont constants pendant la durée d'une seule mesure.

En outre, le modèle PGM suppose un seul goulot d'étranglement qui est à la fois le lien serré (*narrow*) et le lien étroit (*tight*) pour ce chemin.

5.3.1 SPRUCE

SPRUCE est un outil utilisé par les hôtes terminaux pour mesurer la bande passante disponible de bout en bout. Il échantillonne le taux d'arrivée au goulot d'étranglement à l'aide de l'envoi des paires de paquets espacés de façon que le deuxième paquet de sonde arrive à la queue du goulot d'étranglement avant que le premier paquet la quitte. SPRUCE calcule alors le nombre d'octets qui est arrivé à la file d'attente entre les deux paquets de sonde à partir de l'espacement entre eux au récepteur. SPRUCE calcule la bande passante disponible comme étant la différence entre la capacité du chemin et le taux d'arrivée sur le goulot d'étranglement.

5.3.1.1 Le concept de SPRUCE

SPRUCE calcule la bande passante disponible selon l'équation 5.3 qui requiert trois paramètres : C, Δ_{in} et Δ_{out}. SPRUCE suppose que C (la calcule par une méthode spécifique)

est connue, établie Δ_{in} à l'expéditeur et mesures Δ_{out} au récepteur. À l'expéditeur, SPRUCE fixe le temps d'espacement (*Gap*) intra-paire Δ_{in}, comme le temps de transmission d'un paquet de 1500 octets de données sur le goulot d'étranglement. Ce choix assure que la file d'attente ne sera pas vide entre les départs des deux paquets de sonde d'une paire, ce qui est une exigence de l'équation 5.3. Au récepteur, SPRUCE mesure Δ_{out}, le temps de transmission de trafic traversant et d'un paquet de sonde de 1500 octets. Avec cette information et avec une capacité connue du lien de goulot d'étranglement, SPRUCE calcule alors le nombre d'octets qui est arrivé à la file d'attente entre les deux paquets de sonde dans une paire à partir de l'espacement inter-paquets par la formule 5.2, où C est la capacité de goulot d'étranglement. En mettant ces valeurs dans l'équation 5.3, SPRUCE mesure un échantillon de la largeur de bande disponible.

Pour améliorer la précision de l'estimation, SPRUCE effectue une séquence de mesures de sonde et rapporte la moyenne. SPRUCE fixe le temps d'inter-*Gap* (entre deux paires de sonde) à la sortie d'une fonction exponentiellement distribuée, dont la moyenne τ est beaucoup plus important que Δ_{in}, résultant dans un processus d'échantillonnage de Poisson.

Cette décision est prise pour de deux raisons :

1. Premièrement, pour un modèle simple qui suppose un seul goulot d'étranglement et un trafic traversant non-fluide (c'est-à-dire, aucun trafic traversant ou trafic traversant proche de la capacité C), une série de mesures selon un processus d'échantillonnage de Poisson voit le taux moyen du trafic traversant;
2. Deuxièmement, l'échantillonnage de Poisson assure que SPRUCE est non-intrusive. En particulier, l'envoi d'une séquence des paires de paquets au lieu d'un train nous permet de contrôler le *Gap* inter-paire indépendamment de *Gap* intra-paire. Nous utilisons un *Gap* inter-paire τ large pour rendre SPRUCE non-intrusive.

SPRUCE calcule la bande passante disponible au moment t comme la moyenne des K derniers échantillons mesurés. La valeur par défaut de K est 100.

5.3.1.2 Implémentation de SPRUCE

L'expéditeur de SPRUCE envoie une série de paires de paquets UDP de 1500 octets. SPRUCE fixe le *Gap* intra-paire au temps de transmission d'un paquet de 1500 octets sur le lien étroit du chemin. L'expéditeur ajuste le *Gap* inter-paire moyen pour assurer que le taux de sonde sera le minimum de 240 Kb/s ou 5% de la capacité du chemin.

L'expéditeur horodate les paquets des paires envoyés de façon permettant le récepteur de calculer l'écart à la réception de la différence de temps entre les arrivées de deux paquets d'une paire, et il marque encore chaque paquet par un étiquette permettant la classification des paquets de paires en couples afin de calculer l'écart des bons paquets constituant une paire. Puis, le récepteur horodate à son tour les paquets reçus puis les classifie à l'aide de l'étiquette mis par l'expéditeur. Puis, il calcule l'écart de temps à la réception Δ_{out} par la différence des deux horodatages. Enfin, le récepteur utilise ce résultat dans la formule 5.3 pour estimer la bande passante disponible et il moyenne des échantillons individuels en utilisant une fenêtre coulissante de 100 paquets.

5.3.2 La méthode de mesure PathLoad

PathLoad est un outil actif qui estime la bande passante disponible de bout en bout. Son idée principale est que les délais dans une seule direction d'un flux des paquets périodiques montrent une tendance quand le taux de flux est plus grand que la bande passante disponible. Cet algorithme de mesure est itératif et il requiert la coopération des deux bouts : l'expéditeur (SND) et le récepteur (RCV).

5.3.2.1 Le concept de PathLoad

Supposons que SND est en train de transmettre un flux de paquets périodiques à RCV. Le flux est constitué de K paquets (K est la longueur du flux), L en bits sera la taille de chaque paquet et T sera la période de transmission d'un paquet. Donc, le taux de transmission en bits par seconde du flux sera donné par la formule suivante.

$$R = \frac{L}{T} \tag{5.4}$$

SND horodate chaque paquet i par un timbre de temps (horodatage) t_i en priori de leur transmission. Soit a_i le temps d'arrivée d'$i$ème paquet au RCV. RCV calcule le délai dans une direction OWD (One-Way Delay) relatif de chaque paquet comme le suivant.

$$D_i = a_i - t_i \tag{5.5}$$

La méthodologie de mesure ne synchronise pas les horloges, car nous sommes intéressés seulement dans la grandeur relative d'OWD. À la réception complète du flux, RCV inspecte la séquence des OWD relatifs pour vérifier si le taux de transmission R est plus grand que la bande passante disponible A. Quand le taux R d'un flux est plus grand que la bande passante disponible A, le flux causera une surcharge à court terme dans le lien étroit du chemin.

Durant cette période de surcharge, le lien étroit reçoit de trafic plus qu'il pourra transmettre de façon que la queue du lien étroit s'agrandira graduellement. Alors, le temps d'attente de paquet i dans la queue sur le lien étroit sera plus long que le temps correspondant à un autre paquet j où $j < i$. Par conséquent, quand $R > A$, les OWD relatifs $\{D_1, D_2, \ldots, D_k\}$ d'un flux des paquets subiront une tendance ascendante, ce qui s'appelle SLoPS. En outre, si le flux R est moins que la bande passante disponible A, alors le flux ne causera pas de surcharge sur le lien étroit et l'accumulation sur ce lien ne prendra pas une tendance ascendante avec chaque nouveau flux de paquets. Donc, quand $R < A$, les OWD relatifs $\{D_1, D_2, \ldots, D_k\}$ d'un flux de paquets auront une tendance non-ascendante. De cette façon RCV peut conclure si le débit de flux R est plus grand que la bande passante disponible.

Cependant, pour estimer la bande passante disponible sur un chemin, il faut que les deux extrémités coopèrent de façon à ce que le débit R de flux converge itérativement vers A.

Dans la $n$ième pas de cette procédure itérative, RCV vérifie si le taux $R(n)$ du $n$ième flux est supérieur que A en se basant sur la présence d'une tendance croissante dans les OWD de

flux *n*. si $R(n) > A$, SND envoie un flux additionnel périodique avec un taux $R(n+1) < R(n)$. Si $R(n) < A$, le taux de flux périodique suivant sera $R(n+1) > R(n)$. PathLoad utilise UDP pour les flux des paquets périodiques et une connexion TCP entre les deux bouts comme canal de contrôle pour transférer des messages concernant les caractéristiques de chaque flux.

5.3.2.2 La sélection de la période de transmission *T* et de la taille *L* de paquet

Le taux de transmission R d'un flux est donné par la formule 5.4. Étant donné R, PathLoad sélectionne les valeurs de L et T pour satisfaire la relation suivante.

$$\boxed{R = \frac{L}{T}} \tag{5.6}$$

Il y a quelques contraintes dans la sélection de L et T. Nous citons en quelques une :

- L ne peut pas être inférieur à certain nombre d'octets, et elle ne doit pas être plus grande que la MTU (*Maximum Transmission Unit*) du chemin pour éviter la fragmentation. Également, si L est très petite, alors la possibilité de bourrage-nulle dans certains liens de niveau-2 causera un changement significatif dans la taille des paquets, et un changement dans le taux de flux sur ces liens. Cette version de PathLoad utilise L de 96 octets, plus petite que la MTU du chemin et multiple de 48 octets (pour éviter le zéro-bourrage sur les liens d'AAL5 : ATM-AL5 (*Asynchronous Transfer Mode Adaptation Layer 5*);

- T doit être petit autant que possible car ça conduit à un processus de mesure court; et si T augmente, la durée de chaque flux augmente encore. La valeur de T dépend des équipements des hôtes de mesure. Pour les stations de travail ordinaires la période minimale de transmission T_{min} pour des paquets de taille minimale dos à dos (collés ou *back-to-back*) est entre 15-30 μs. Donc, si nous ciblons le taux R, PathLoad choisit $T_{min} = 100\ \mu s$ par exemple puis la valeur L correspondante.

5.3.2.3 La sélection de la longueur *K* de flux

La valeur K est variable car, si K est grande le flux peut déborder la queue du lien étroit. Ce qui cause des pertes dans le trafic traversant et dans le flux. Ces pertes peuvent conduire vers une réduction dans le trafic traversant et par conséquence une réduction dans la bande passante disponible. Et si K est très petite, le flux ne pourra pas fournir le RCV suffisamment des échantillons pour conclure s'il y avait une tendance dans les OWD mesurés. On peut mentionner que Pathrate prend $K = 100$ [Chen et al. (2005)].

5.3.2.4 Les flots de flux

PathLoad ne peut pas déterminer si $R > A$ en se basant sur un seul flux. Par contre, elle envoie N flux. Chaque flux consiste de K paquets de longueur L envoyés périodiquement chaque T secondes. Tous les flux d'un même flot possèdent le même taux de transmission R.

Chaque flux est envoyé seulement lorsque le flux précédant est acquitté. Cela permet au RCV de mesurer les OWD N fois et crée une période *idle* Δ entre les flux. Cette période permet à la queue de s'écouler de trafic (surcharge extra) de mesure. Dans PathLoad $N = 12$ flux. Donc, la durée d'un flot sera connue par la relation suivante.

$$U = N \times (K \times T + \Delta) \tag{5.7}$$

5.3.2.5 La détermination d'une tendance croissante

Il y a un algorithme critique dans PathLoad qui sert à détecter la tendance croissante. En effet, PathLoad divise les K mesures OWD $\{D_1, D_2, ..., D_k\}$ en groupes de $\Gamma = \sqrt{K}$ mesures OWD consécutives. De Γ délais dans chaque groupe i, PathLoad calcule la médiane $\hat{D}_i$ du groupe. Puis, elle utilise la statistique pour vérifier s'il y a une tendance croissante. La métrique PCT (*Pairwise Comparaison Test*) d'un flux est calculée par la formule suivante :

$$S_{PCT} = \frac{\sum_{k=2}^{\Gamma} I(\hat{D}_k > \hat{D}_{k-1})}{\Gamma - 1} \tag{5.8}$$

Où $I(X) = 1$ si X détient et zéro autrement. PCT mesure la fraction des paires d'OWD consécutives qui augmentent, et $0 \le S_{PCT} \le 1$. S'il y a une forte augmentation alors S_{PCT} approche 1. Dans PathLoad, la métrique PCT rapporte une tendance croissante si $S_{PCT} > 0.54$ et une tendance non-croissante si $S_{PCT} < 0.54$. Autrement le résultat est ambigu.

La formule 5.9 nous permet de calculer la métrique PDT (*Pair-Wise Difference Test*) d'un flux.

$$S_{PDT} = \frac{D_\Gamma - D_1}{\sum_{k=2}^{\Gamma} \left| \hat{D}_k > \hat{D}_{k-1} \right|} \tag{5.9}$$

Si les OWD sont indépendantes, alors la valeur prévue de S_{PDT} est zéro. Dans PathLoad, la métrique PDT rapporte une tendance croissante si $S_{PDT} > 0.55$ et non-croissante si $S_{PDT} < 0.45$. Autrement le résultat est ambigu.

Dans PathLoad, la tendance d'un flux est caractérisée comme suit : quand une des métriques PCT et PDT rapporte une tendance croissante, pendant que les autres sont de type croissantes ou ambigues, le flux est considéré de type *I* (*Increasing*). De la même façon, quand une métrique rapporte une tendance non-croissante, tandis que les autres sont non-croissantes ou ambigues, le flux est caractérisé de type *N-I* (*Non-Increasing*). Si les deux métriques sont ambigues ou une des deux est croissante tandis que l'autre est non-croissante, alors le flux sera abandonné.

5.3.2.6 La comparaison de *R* avec *A* après un flot

Après que tous les N flux d'un flot soient reçus, RCV détermine si $R > A$. Si une large fraction f de N flux dans un flot sont de type I, alors le flot entier montre une tendance croissante et nous conclurons que le taux de transmission de flot est plus grand que la bande passante disponible ($R > A$). De la même façon, si une large fraction f de N flux sont de type N-I, le flot ne montre pas une tendance croissante et nous conclurons que le taux du flot est plus petit que la bande passante disponible ($R < A$). Il peut y arrive que nous obtenions un peu de flux $N \times f$ de types I ou un peu de flux $N \times f$ de types N-I, là, ce flot est dans une région grise. Dans PathLoad f est fixé de 70%.

5.3.2.7 Ajustement du taux R

Après la fin d'un flot n, PathLoad utilise un algorithme d'ajustement pour déterminer le taux $R(n+1)$ de prochain flot. L'algorithme est le suivant :

```
{
/* Initialement : G^min = G^max = 0*/
        if (R(n) < A) /*tendance non-croissante*/
                R^min = R(n);
                if (G^min > 0)
                        R(n+1) = (G^min + R^min) / 2;
                else
                        R(n+1) = (G^max + R^min) / 2;
        else
        if(R(n) > A) /* tendance croissante*/
                R^max = R(n);
                if (G^max > 0)
                        R(n+1) = (R^max + G^max) / 2;
                else
                        R(n+1) = (R^max + G^min) / 2;
```

```
        else
                /* Région grise*/
                if (G^min == G^max == 0)
                        G^min = G^max R(n);
                If (G^max ≤ R(n))
                        G^max = R(n);
                        R(n+1) = (G^max + G^max) / 2;
                else if (G^min > R(n))
                        G^min = R(n);
                        R(n+1) = (R^min + G^min) / 2;
/*les conditions de terminaison*/
if (R^max - R^min ≤ ω || (G^min - R^min ≤ x&& R^max - G^max ≤ x))
        return (R^min, R^max);
}
```

Où R^{min} est le plus haut taux mis en valeur pour être inférieur que la bande passante disponible en certain point. Tandis que, R^{max} est le taux le plus bas mis en valeur pour être supérieur que la bande passante disponible jusqu'au ce point. En d'autres mots, $[R^{min}, R^{max}]$ est la plage la plus étroite des limites pour la bande passante disponible de flot.

De la même façon, G^{min} est le taux le plus bas mis en valeur pour être dans la région grise. Tandis que, G^{max} est le plus haut taux mis en valeur pour être dans la région grise après chaque flot. Alors, $[G^{min}, G^{max}]$ est la plage inclusive la plus large pour la région grise après chaque flot.

PathLoad converge vers une estimation de la bande passante disponible quand la plage entre les limites de la bande passante disponible minimale et maximale sera inférieure à une résolution ω spécifiée par l'usager.

L'initialisation de cet algorithme commence avec une phase exponentielle. Cette phase est utilisée pour initialiser R^{min} et R^{max}. Le taux du flot initiale *R(0)* peut être fourni par l'usager ou par une valeur par défaut qui est de 1 Mbps. Si $R(0) < A$, le taux sera augmenté par un facteur 2 après chaque flot jusqu'à $R(n) > A$. En ce point, $R^{max} = R(n)$ et $R^{min} = R(n-1)$. De la même façon, si $R(0) > A$, le taux sera diminué par un facteur de 2 après chaque flot et jusqu'à $R(n) < A$. En ce point, $R^{min} = R(n)$ et $R^{max} = R(n-1)$.

5.3.2.8 Réaction à la perte des paquets

RCV mesure le taux de perte après la réception de chaque flux. Si le taux de perte est moins que 3% (perte moyenne), le flux sera marqué (*lossy*). Si le nombre de flux qui sont marqués *lossy* dépasse une certaine fraction, là le RCV avise le SND pour abandonner les flux de flot actuel. Et si le taux de perte d'un flux dépasse le 10%, alors le flot sera abandonné immédiatement. Dans ce cas, le taux R, avec lequel le flot été abandonné, sera la limite la plus haute R^{max} de l'algorithme d'ajustement de taux.

CHAPITRE 6

LA MESURE DE LA CAPACITÉ MAXIMALE DES LIENS

Le chapitre précédent a décrit les techniques de mesure de la bande passante disponible. Il a également souligné la nécessité de connaître la capacité maximale de goulot d'étranglement des chemins pour l'inclure dans la formule 5.3 de la méthode SPRUCE. Ce chapitre explique ces techniques de mesure de la capacité des liens et décrit la méthode de choix qui est utilisée dans nos études.

6.1 Techniques d'estimation de la capacité

L'estimation de la capacité maximale d'un chemin Overlay est une phase fondamentale et préalable pour la méthode de mesure de la bande passant disponible SPRUCE. Car la capacité C doit être connue en avance de la part de l'algorithme de SPRUCE. Ce qui exige de l'estimer avant de lancer la méthode SPRUCE. Les travaux sur l'estimation de la capacité d'un chemin ont été basés sur la surveillance des délais des paires ou des trains de paquets ou soit par leurs dispersions.

Il y a des techniques d'estimation qui utilisent les réponses ICMP (*Internet Control Message Protocol*) des routeurs pour estimer la capacité des liens en se basant sur la variation de délai d'aller-retour quand la taille des paquets de sonde est augmentée, par exemple la méthode Pathchar.

La dispersion des paquets de sonde est une autre technique reposant sur la sonde d'un chemin avec une série de paires de paquets ou avec des trains de paquets. Puis, elle traite statistiquement la dispersion de ces paquets de sonde induite sur le chemin de préférence, par exemple la méthode Bprobe.

Les techniques mentionnées ci-dessus sont invoquées soit par le délai d'aller-retour uniquement ou soit uniquement par la dispersion de paquets de sonde. Mais, la technique utilisée dans ce travail est tout à fait nouvelle et combine les mesures de la dispersion et les délais des paires de paquets de sonde pour en profiter de ces deux approches en même temps et dans une même méthode, ce qui va conduire vers des résultats plus précis.

Cette technique est la CapProbe [Kapoor (2004)]. Elle est basée sur une observation simple et fondamentale. Une paire de paquets, produisant soit une sous-estimation soit une surestimation de la capacité, devait alors subir de trafic traversant qui induit la queue à un certain lien. CapProbe filtre une telle mesure déformée par la poursuite des délais des paires de paquets. Elle utilise uniquement les paires de paquets qui ont un délai de bout en bout minimal. Le seul scénario où l'outil ne parvient pas toujours à estimer correctement les capacités, c'est quand le trafic traversant est à la fois intensive et non réactifs (comme le trafic UDP taux constant)

6.2 L'algorithme de dispersion d'une paire de paquets de sonde

L'algorithme basique de paire des paquets porte sur le fait que si deux paquets sont envoyés dos à dos, ils sont mis en attente l'un après l'autre au lien étroit, ils vont quitter le lien avec une dispersion T donné par la formule suivante.

$$T = \frac{L}{B} \tag{6.1}$$

Où L est la taille du deuxième paquet et B est la largeur de bande (capacité) du lien étroit (le lien limitant la capacité du chemin).

Si les deux paquets ont la même taille, les délais de transmission seront les mêmes. Cela signifie que, après le lien étroit, une dispersion de T sera maintenue entre les paquets même

si les liens plus rapides sont traversés en aval du lien étroit. Ceci est illustré à la figure 6.1, où S est la source, D est la destination et le lien A-B est le lien étroit.

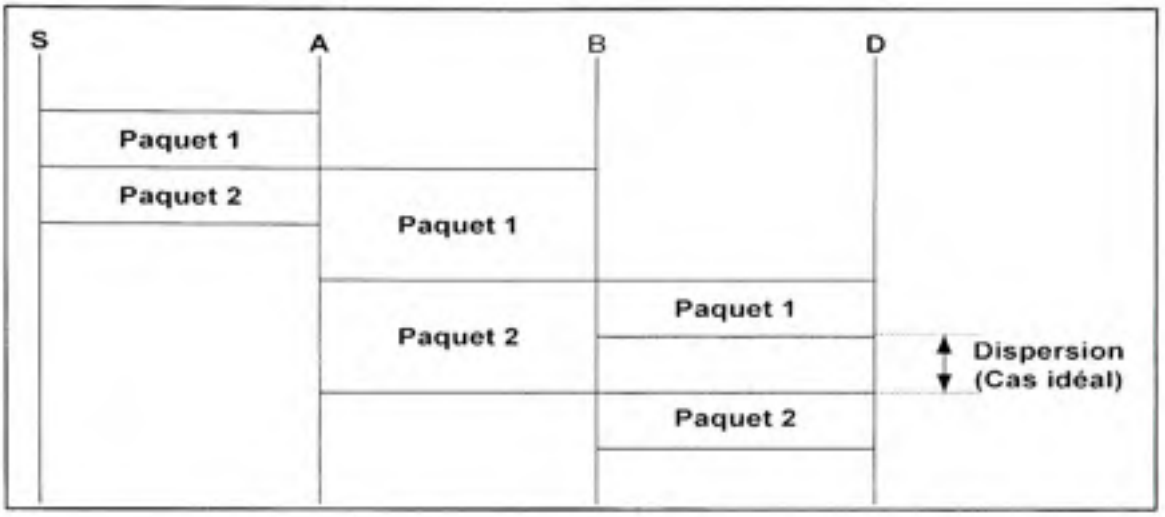

Figure 6.1 ***Dispersion des paquets de sonde (cas idéal).***

(Tiré de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi, 2004)

Source : Cette figure a été tirée de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi 2004 « CapProbe: A Simple and Accurate Capacity Estimation Technique » et correspond à la « Figure 1 Packet Pair dispersion. (a) Ideal case » présentée en page 3 dans le document original. (La référence complète du document est présentée dans la bibliographie).

La capacité du lien étroit peut ensuite être calculée comme suit.

$$\boxed{B = \frac{L}{T}} \tag{6.2}$$

L'algorithme de paire des paquets suppose que les paquets se mettent en attente l'un après l'autre dans la file d'attente sur le lien étroit.

La surestimation de la capacité se produit lorsque la dispersion entre la paire des paquets à la destination est plus petite que ce qui serait présenté par le lien étroit. Cela peut se produire chaque fois si le lien étroit n'est pas le dernier lien sur le chemin. Si le premier paquet d'une paire est mis en attente dans une file d'attente après le lien étroit, tandis que le

deuxième subit une mise en attente pour une durée plus courte que le premier ce qui diminue la dispersion entre les deux paquets.

La figure 6.2 montre comment la dispersion peut diminuer sur un lien après celui qui est étroit. Le premier paquet de la paire de sonde devra être mis en attente en raison de l'interférence du trafic traversant.

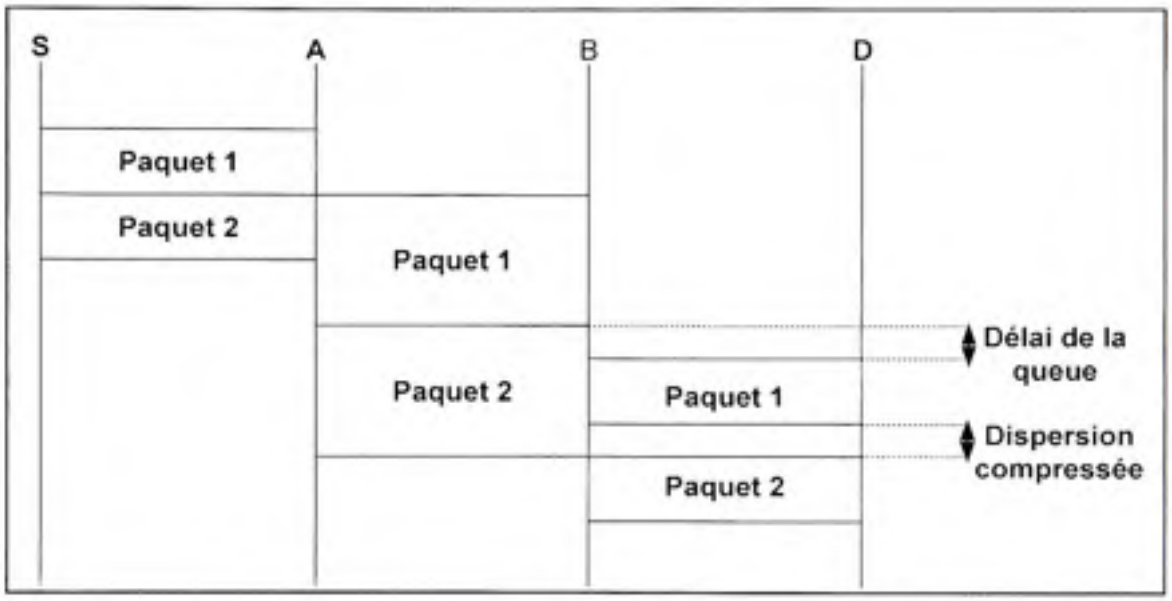

Figure 6.2 ***La surestimation de la capacité du lien étroit.***

(Tiré de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi, 2004)

Source : Cette figure a été tirée de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi 2004 « CapProbe: A Simple and Accurate Capacity Estimation Technique » et correspond à la « Figure 1 Packet Pair dispersion. (b) Over-estimation of capacity » présentée en page 3 dans le document original. (La référence complète du document est présentée dans la bibliographie).

La sous-estimation de la capacité se produit lorsque la dispersion entre les paquets de la paire est plus grande à la destination que ce qui serait présenté par le lien étroit en absence de trafic traversant.

Cette augmentation se produit en raison de paquets de trafic traversant qui ont été servis (transmis) entre les paquets d'une paire de sonde. Une telle augmentation peut se produire

n'importe où sur le chemin, avant, sur ou après le lien étroit. La figure 6.3 montre comment la sous-estimation de la capacité peut se produire.

La méthode de mesure de la capacité de lien subit une telle situation dans le cas où la taille des paquets du trafic traversant est plus grande de celle des paquets de sonde.

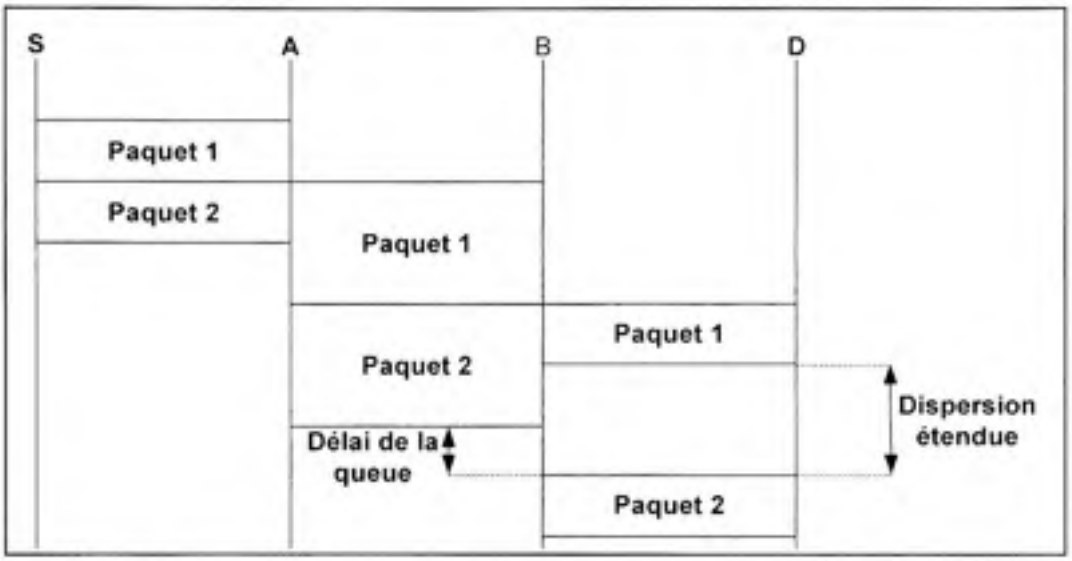

Figure 6.3 ***La sous-estimation de la capacité du lien étroit.***

(Tiré de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi, 2004)

Source : Cette figure a été tirée de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi 2004 « CapProbe: A Simple and Accurate Capacity Estimation Technique » et correspond à la « Figure 1 Packet Pair dispersion. (c) Under-estimation of Capacity » présentée en page 3 dans le document original. (La référence complète du document est présentée dans la bibliographie).

6.3 La méthode de mesure de la capacité CapProbe

L'idée principale de CapProbe est qu'au moins un des deux paquets de sonde a été mis en attente dans une file d'attente si la dispersion obtenue à la destination est déformée de celle correspondante à la capacité du lien étroit.

Cela signifie que, pour les échantillons qui estiment incorrectement la valeur de la capacité, la somme des délais de la paire de paquets, que nous appelons la somme des délais (*delay*

sum), incluant le trafic traversant qui induit les délais de la queue. Cette somme de délais sera supérieure à la somme minimale de délais, qui est la somme de délai d'un échantillon dans lequel aucun des paquets ne souffre de l'interférence du trafic traversant. La dispersion de tel échantillon de paquets ou paire n'est pas déformée par un trafic traversant et elle reflètera la capacité correcte. Donc, CapProbe calcule la somme de délais de tous les échantillons et elle utilise la dispersion de l'échantillon avec la somme minimale de délais à la fois pour estimer la capacité de lien étroit.

6.3.1 Effet de la taille des paquets de sonde sur la précision de mesure

Pour estimer précisément avec CapProbe, il suffit qu'aucun des paquets de la paire ne soufre d'une mise dans une file d'attente. Mais, la taille de paquet a un effet sur la probabilité de la mise du deuxième paquet en attente dans une queue. Supposons que les paquets (chacun a une taille de L bits) d'une paire sont arrivés dos à dos à un lien étroit de capacité C bps. Le second paquet quittera la queue après L / C unité de temps (si le trafic traversant n'interfère pas).

L / C est alors le temps de dispersion T des deux paquets (Figure 6.4).

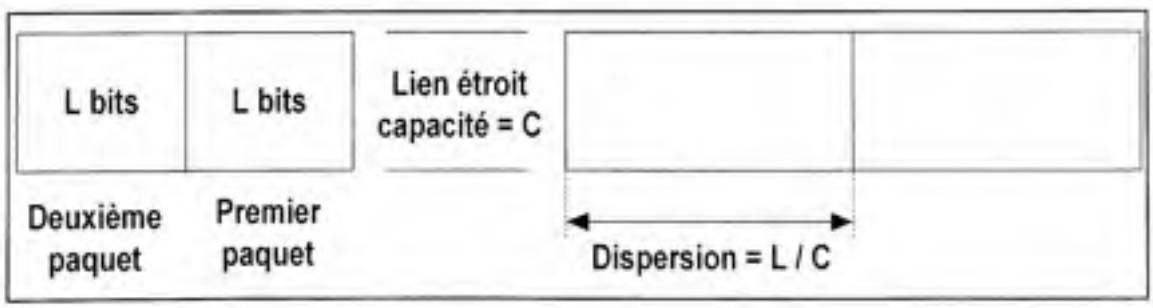

Figure 6.4 ***Une paire de paquet est arrivée sur un lien étroit de capacité C.***
(Tiré de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi, 2004)

Source : Cette figure a été tirée de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi 2004 « CapProbe: A Simple and Accurate Capacity Estimation Technique » et correspond à la « Figure 2 Packet Pair arriving at the narrow link of capacity C » présentée en page 3 dans le document original. (La référence complète du document est présentée dans la bibliographie).

Ce temps T est le temps pendant lequel un trafic traversant peut interférer avec les deux paquets sur un lien situé après le lien étroit. Donc, ce trafic traversant sera servi entre les deux paquets en causant des délais. Ces délais servent à la mise en attente le deuxième paquet, par conséquence une dispersion. Ce temps est connu comme fenêtre de vulnérabilité.

D'autre part, la probabilité de la mise en file d'attente du deuxième paquet décroît quand nous diminuons la taille des paquets (Figure 6.5).

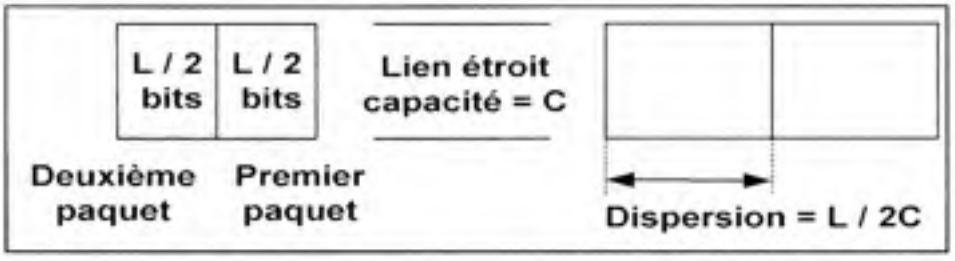

Figure 6.5 ***Dispersion réduite à cause de la petite taille des paquets.***

(Tiré de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi, 2004)

Source : Cette figure a été tirée de Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi 2004 « CapProbe: A Simple and Accurate Capacity Estimation Technique » et correspond à la « Figure 3 Dispersion reduced due to smaller packet size» présentée en page 3 dans le document original. (La référence complète du document est présentée dans la bibliographie).

La taille des paquets est réduite à $L / 2$ bits. Donc, la fenêtre de vulnérabilité T en ce cas, donnée par la formule 6.3, sera plus petit que T obtenue par la formule 6.1. De cette façon, nous pouvons diminuer les cas de sous-estimation de la capacité.

$$T = \frac{L}{2C} \tag{6.3}$$

L'algorithme implémenté dans notre travail utilise les paires des paquets de même taille, tel que les auteurs recommandent; avec la possibilité de l'utilisation de tailles de 700 ou 900 octets dans les expériences. Ces valeurs ont été choisies du fait qu'elles se trouvent parmi

les valeurs qui causent le moins de problèmes dans les mesures au contraire des grandes valeurs (tels que 1500 octets, qui est la valeur typique de la MTU), qui ont une plus grande chance à subir une dispersion.

6.3.2 La technique de convergence

Une question qui se pose lors de l'utilisation de CapProbe est : comment pouvons-nous savoir si la capacité estimée par CapProbe est exacte? En d'autres termes, comment pouvons-nous déterminer si la somme minimale des délais dans un ensemble des échantillons de paires des paquets est égale à la somme des délais minimaux des paquets qui ne sont pas mis en file d'attente. Ainsi, quand le niveau de congestion est élevé le temps pour obtenir un bon échantillon est grand. Alors, nous mesurons les délais minimaux des deux paquets pour obtenir la somme des délais minimaux, puis nous comparons la somme des délais des paires de paquets avec la somme des délais minimaux pour savoir si les paquets de l'échantillon obtenu ont expérimenté de trafic traversant. Ce n'est pas suffisant de calculer la somme des délais des paquets pour filtrer les échantillons, car nous pouvons obtenir une somme minimale mais pas avec les délais minimaux. De cette façon, nous ne pouvons pas être sûr si le résultat était correct ou non.

6.3.3 Algorithme de CapProbe

CapProbe est initialisée pour une période de 40 échantillons au début. Si la condition de la somme minimale des délais n'était pas satisfaisante on augmente le nombre des échantillons jusqu'à 100 échantillons et s'il y avait une large variation dans les échantillons alors il faudra incrémenter la taille des paquets de 20% pour améliorer la précision du système opérationnel. Autrement, il faudra décrémenter la taille des paquets de 20% pour diminuer la probabilité de délais dû au trafic traversant.

Quand CapProbe obtient deux mesures de condition de la somme minimale des délais conformes et séquentiels avec une taille de paquets autour de 700 et 900 octets et quand les

estimations diffèrent de 5% entre elles, alors l'algorithme s'arrête. Autrement, il recommence.

Dans les chapitres précédents nous avons exploré les différents algorithmes et techniques nécessaires pour nos études. La conception ainsi que l'implémentation des simulations de tests seront présentés dans les chapitres suivants. Le chapitre sept décrira l'environnement de simulations, les détails de la topologie du réseau étudié et présentera ainsi les résultats de tests sur les différentes méthodes de mesure.

CHAPITRE 7

SCÉNARIOS ET IMPLÉMENTATIONS

Nous avons exploré dans les chapitres précédents les méthodes de mesure de la bande passante disponible et de la capacité des liens choisies, leurs algorithmes et l'algorithme de routage adaptatif employé du réseau dédié de service.

Ce chapitre est consacré à introduire l'environnement de simulation utilisé dans nos études, la topologie du réseau natif au-dessus duquel le réseau dédié de service étudié a été construit et testé. Également, nous allons voir les différents scénarios de la simulation, l'implémentation des méthodes des mesures et de l'algorithme de routage adaptatif avec leurs intégration dans le milieu des simulations du réseau SON.

Ensuite, nous allons exposer des résultats de quelques tests de mesure effectuées par ces méthodes et nous allons faire une comparaison de performance de ces méthodes dans les différents scénarios des simulations.

7.1 Environnement d'implémentation NS2

NS est un simulateur des événements discrets visé dans les recherches de réseaux. Il est un simulateur orienté objet, écrit en C++ (langage de programmation orienté objet C++), avec un interpréteur OTcl (une extension orientée objet de TCL : *Tool Command Language*) comme interface ou frontière utilisée pour exécuter les scripts des commandes de l'usager [Fall et Varadhan (2006)]. Le simulateur prend en charge deux hiérarchies de classes en C++ et en OTcl. Les deux hiérarchies sont étroitement liées les uns aux autres, et de point de vue utilisateur, il existe une correspondance une-à-une entre une classe dans l'hiérarchie interprétée et une dans l'hiérarchie compilée. La racine de cette hiérarchie est la classe TclObject.

Les utilisateurs créent des nouveaux objets d'un simulateur par le biais de l'interpréteur; ces objets sont instanciés dans l'interpréteur et ils sont étroitement reflétés par un objet dans l'hiérarchie compilée. L'hiérarchie de classe interprétée est automatiquement établie par le biais des méthodes définies dans la classe TclClass. Alors, les objets instanciés par l'utilisateur sont reflétés à travers les méthodes définies dans la classe TclObject.

7.1.1 Le concept de NS2

La question qui se pose est : pourquoi NS2 utilise deux langages?
NS2 utilise deux langues parce le simulateur a deux sortes de choses à faire :

1. D'une part, les simulations détaillées des protocoles exigent un langage de programmation système qui peut manipuler efficacement les octets, les entêtes des paquets et d'implémenter des algorithmes qui fonctionnent avec des larges quantités de données. Pour ces tâches, la vitesse d'exécution est très importante, tandis que le temps de révision (lancement de la simulation, la recherche des problèmes avec leurs réparations, recompilation et le ré-lancement) est moins important;
2. D'autre part, une grande partie des recherches dans le domaine des réseaux implique des paramètres légèrement variés, des configurations ou d'explorer rapidement un certain nombre de scénarios. Dans ces cas, le temps d'itération (changer le modèle et le relancer) est plus important, puisque la configuration s'exécute une fois (au début de la simulation), alors que le temps d'exécution de cette partie de la tâche sera moins important.

Donc, l'utilisation de l'OTcl sera pour la configuration (topologie, trafic, etc.), tandis que le C++ sera utilisé car nous avons besoin de créer des protocoles qui envoient des paquets de sonde entre les nœuds de réseau SON, de manipuler et traiter chacun de ces paquets. De plus, nous devons traiter tous les paquets des autres flux circulant sur le réseau natif et cela exige encore une immense capacité de traitement rapide.

7.2 La topologie et les scénarios du réseau dédié de service étudié

Le réseau dédié de service est composé d'une série de nœuds Overlay spécialisés qui sont placés dans Internet par l'OSP pour construire la plate-forme de communication Overlay. Les nœuds peuvent être placés soit sur le bord d'un domaine (système autonome) ou soit au son cœur. L'OSP provisionne ces nœuds par des connexions de haut débit de dorsale d'Internet.

Le réseau dédié de service utilisé dans les scénarios des simulations de notre projet a une forme circulaire et il est composé de quatre nœuds Overlay placés dans les cœurs de quatre AS différents (Figure 7.1). Les liens logiques liant les nœuds Overlay sont composés par plusieurs liens physiques (des chemins IP), c'est-à-dire ils peuvent être composés de plusieurs nœuds ou sauts du réseau natif. De cette façon, chaque nœud qui veut envoyer un flux vers un autre aura le choix de l'acheminer par l'un des deux chemins Overlay qui le lient avec les autres nœuds.

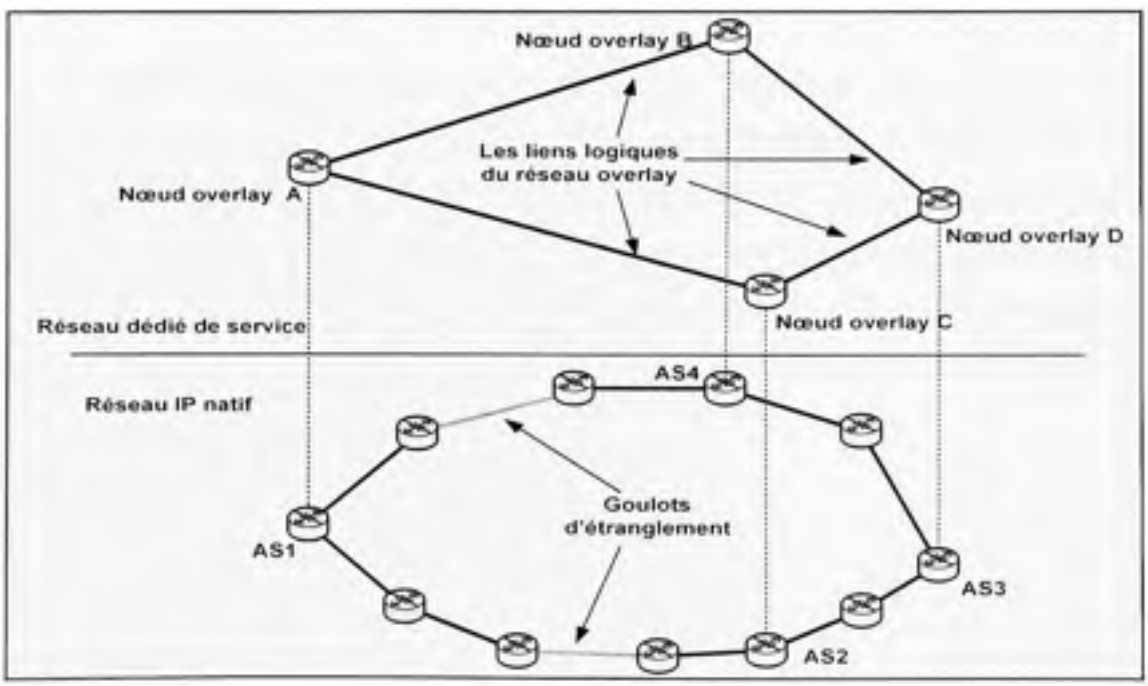

Figure 7.1 *Le réseau dédié de service de simulations.*

Le but de cette étude est de souligner l'influence des méthodes de mesure de la bande passante disponible sur la performance des algorithmes de routage adaptatif de SON. En effet, il était important de choisir une topologie aidant à atteindre ce but, soit une topologie permettant d'analyser juste les effets de mesure sur le routage. L'adoption de telle topologie permet d'éviter les autres problèmes comme l'*Overhead* (trafic de mesure et de dissémination) dans les schémas *Full-Mesh*.

Nous avons construit la topologie du réseau physique de façon à ce que nous puissions envisager des goulots d'étranglement. Nous avons mis des liens physiques ayant une capacité inférieure aux autres. Cette configuration aide à forcer le phénomène de congestion après avoir généré de trafic traversant (*Background traffic*) dans le réseau natif passant à travers les goulots d'étranglement. Les liens entre les routeurs dans le réseau natif ont une capacité de 100 Mbps sauf les liens de goulots d'étranglement qui ont une capacité de 10 Mbps.

7.2.1 Les scénarios Overlay des simulations

Dans le but d'avoir un réseau dans des différentes conditions nous considérons trois scénarios qui se diffèrent entre eux par le débit de trafic circulant (traversant) sur les liens physiques, de façon à avoir un scénario accompagné par un trafic traversant léger (par rapport à la capacité des liens), un autre accompagné par un trafic moyen et un dernier qui doit être chargé. En effet, nous allons déclencher les simulations du routage Overlay avec une des méthodes de mesure (intégrée dans chaque nœud Overlay) à la fois et dans chacun des trois scénarios proposés (Figure 7.2).

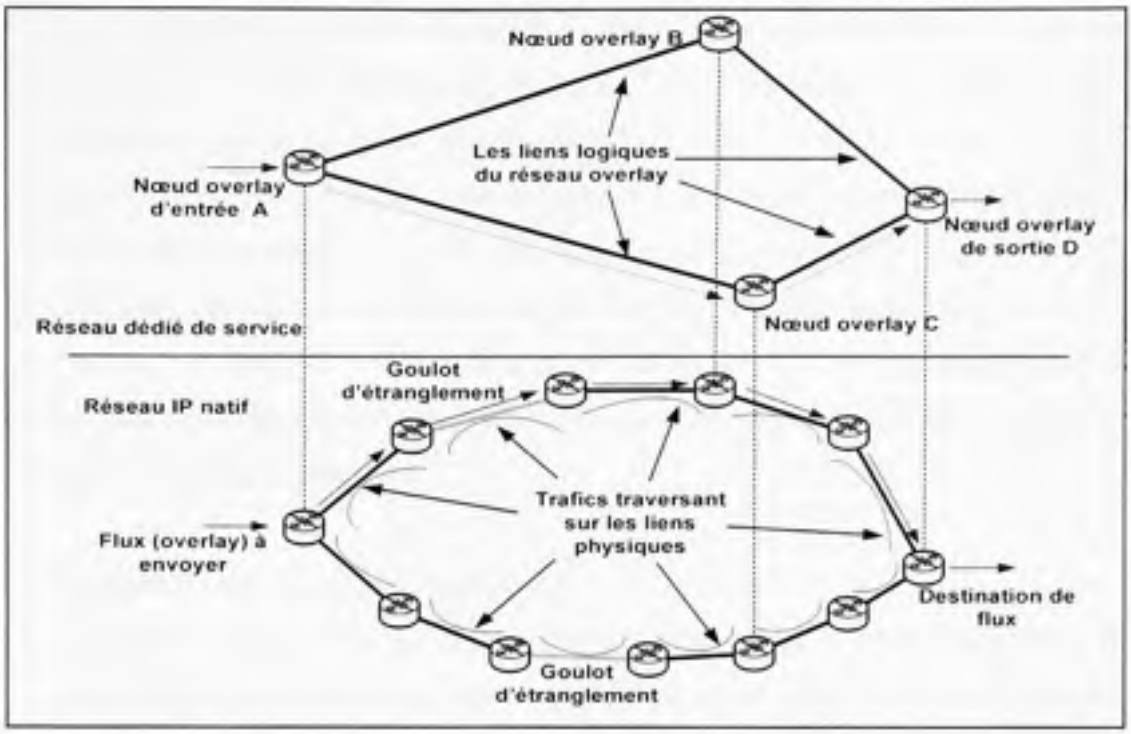

Figure 7.2 ***Le trafic traversant et le trafic Overlay à envoyer.***

Les flux de trafic Overlay qui doivent être acheminés d'un nœud Overlay à un autre, vont expérimenter les conditions actuelles des liens physiques constituant les liens logiques. Ces conditions sont dues au trafic traversant circulant sur les liens physiques. Également, le trafic généré par les méthodes de mesure va expérimenter plusieurs problèmes causés par la congestion.

Ensuite, nous continuons avec des simulations du routage accompagné par des mesures exactes que le simulateur NS2 nous permet d'effectuer. Nous avons cette possibilité de NS2. Il permet de lire le débit sur les interfaces des routeurs et nous allons traiter la situation comme si les mesures étaient faites par une méthode de mesure mais cette fois sans aucune erreur de mesure. Cela nous permet de comparer la performance du routage avec les mesures exactes et le routage avec les méthodes de mesure choisies.

À la fin, nous aurons un routage basé sur trois types de mesures, dont deux sont basés sur des méthodes de mesure implémentées et exécutées et un autre basé sur des mesures exactes. Cela nous permet d'étudier la sensibilité de routage Overlay aux caractéristiques et au type de la méthode de mesure employée, par l'observation des paramètres de qualité de service de trafic des applications qui envoient le trafic Overlay. Ces paramètres seront étudiés dans les trois scénarios de la topologie. De plus, des tests seront faits sur le même trafic Overlay et dans les mêmes conditions mais avec un routage traditionnel pour comparer enfin les performances des applications en employant une fois le routage Overlay et une autre le routage traditionnel.

Les trois scénarios sont configurés comme suit :

1. Le scénario léger : c'est un scénario qui a une charge légère de trafic traversant routé sur les liens physiques du réseau et qui ne peut jamais être saturé. Nous considérons une occupation jusqu'à 25% de la capacité maximale des liens (une occupation légère), car les liens sont capables de servir ce débit facilement par leur capacité et sans aucun problème. Nous avons configuré les nœuds du réseau de façon qu'ils génèrent sur chaque lien ayant une capacité de 100 Mbps des flux de trafic exponentiel (*Bursty*) et de trafic CBR (*Constant Bit Rate*) avec une charge moyenne de 30% de la capacité du lien;
2. Le scénario moyen : La probabilité avec laquelle les liens peuvent être saturés est toujours faible. Ce scénario est configuré pour générer des flux de trafic exponentiel et de trafic CBR ayant en moyen une charge qui peut atteindre 50% de la capacité des liens (toujours les liens qui ont une capacité de 100 Mbps);
3. Le scénario chargé : Dans ce scénario, les nœuds vont générer une charge de trafic qui peut atteindre 95% de la capacité des liens, la probabilité d'être saturé est donc plus grande (toujours les liens qui ont une capacité de 100 Mbps).

En effet, le but essentiel de l'emploi de plus d'une méthode de mesure dans des différents scénarios est d'observer le comportement de ces méthodes et leurs performances quand le réseau n'est pas dans les meilleures conditions. Ainsi, la performance de l'algorithme de

routage dépend directement de la performance ou de la fiabilité de la méthode de mesure en termes d'exactitude et de temps de mesure.

7.2.1.1 Le trafic traversant sur les goulots d'étranglement

Les méthodes de mesure gagnent leur importance de processus d'exploration des liens goulots d'étranglement et de bon traitement de trafic traversant sur ces liens afin de retourner la meilleure information qui peut décrire l'état de la bande passante disponible. Le trafic traversant les goulots d'étranglement est de type exponentiel ayant des périodes de rafales et des périodes fermées (On/Off) ainsi que de trafic CBR.

Les liens de goulots d'étranglement auront des débits variables de trafic traversant de façon que nous ayons toujours un lien goulot d'étranglement offrant une largeur de bande disponible plus grande que celle offerte par l'autre, toujours en respectant les trois niveaux de trafic traversant des trois scénarios (léger, moyen et chargé), car l'état général du réseau est reflété sur les goulots d'étranglement. En effet, le but de l'algorithme de routage adaptatif du réseau dédié de service est d'acheminer le trafic par le chemin qui offre une bande passante disponible suffisante et de protéger les flux Overlay de pertes en cas d'insuffisance de ressource (de la largeur de bande) en diminuant le débit des sources à envoyer.

7.2.1.2 Le trafic Overlay à envoyer

L'algorithme de routage cherche toujours à trouver le chemin Overlay qui peut fournir au nœud-source (génératrice de trafic Overlay) une capacité qui peut atteindre le débit r maximal de son flux. Si r ne peut pas être atteint, il l'acheminera par un autre chemin fournissant une largeur de bande disponible supérieure. Dans nos scénarios, nous acheminons le trafic Overlay du nœud Overlay A vers le nœud Overlay D. Nous allons créer les flux Overlay en générant de trafics UDP qui ont un débit maximal de 2 Mbps. Ce flux

doit traverser les goulots d'étranglement qui ont 10 Mbps de capacité et qui sont à leur tour en train de transmettre ou de servir des autres flux de trafic traversant.

7.3 Le scénario de routage traditionnel

Pour encadrer l'étude de tous les cotés, nous incluons dans nos scénarios trois autres scénarios mais, cette fois nous employons le routage traditionnel d'état de lien pour router le même trafic Overlay. Ce trafic sera ici un trafic IP ordinaire à envoyer dans le réseau IP natif, de même source A vers la même destination B (Figure 7.3). Ainsi, ce trafic va subir les mêmes conditions sur les liens physiques mais sans l'intelligence que le routage Overlay peut fournir. De cette façon, nous serons capables de comparer la performance des applications qui génèrent les flux de trafic à envoyer dans le réseau en employant une fois un routage Overlay et une autre en employant le routage traditionnel, et cela sur la même plateforme et sous les mêmes conditions.

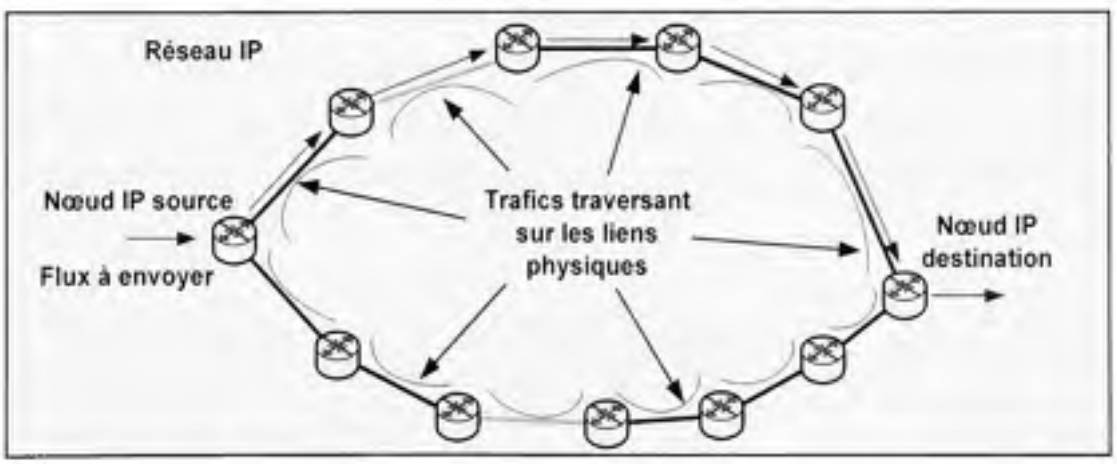

Figure 7.3 ***Réseau IP avec un routage traditionnel.***

Nous utilisons la même configuration de trafic traversant utilisée avec le routage Overlay. Nous aurons donc des moments où les liens physiques étroits seront saturés. Le routage traditionnel ne sera pas au courant de cette situation (la congestion) et il continuera à acheminer le trafic à travers le chemin qu'il considère meilleur selon son algorithme. Cette grave situation va influer la performance des applications. Nous prévoyons une mauvaise

performance de ces application dans le réseau et cela parce que le routage traditionnel achemine toujours vers le même chemin malgré la dégradation des paramètres de qualité de service.

7.4 Implémentations et intégrations des méthodes de mesure de la capacité et de la bande passante disponible dans NS2

Nous implémentons les méthodes des mesures comme des protocoles de communication capables d'envoyer des paquets, de traiter les paquets reçus, d'inclure des étiquettes dans les entêtes des paquets, d'utiliser les protocoles qui sont déjà implémentés dans NS2, de récupérer les données des entêtes de paquets et de faire les calcules nécessaires. Elles sont implémentées en langage C++ et leurs fichiers sont liés au fichier principal de la simulation (qui décrit la topologie et les trafics traversant générés) par le biais des objets créés en C++ qui doit coopérer avec les fichiers C++ et les classes OTCL existants dans le simulateur.

L'intégration des méthodes de mesure de la bande passante disponible ou de la capacité se fait par l'ajout de code de chaque méthode de mesure au code principal de la simulation de façon individuelle. Ainsi, une méthode sera employée à la fois en liant son code en tant qu'application à chaque nœud Overlay dans le réseau afin que ces nœuds Overlay puissent l'utiliser dans leur sonde.

Donc, les nœuds vont être capables d'utiliser la méthode de mesure pour sonder les liens logiques et mesurer la bande passante disponible ou la capacité sur les liens qui le connectent à leurs voisins; une estimation puis une dissémination de ces valeurs prendront lieu par les nœuds, et après quelque temps, tous les nœuds Overlay auront une vision générale de l'état des liens logiques du réseau, en termes de bande passante disponible et de capacité des liens de goulots d'étranglement. Les nœuds vont en profiter alors dans le processus de routage (Figure 7.4).

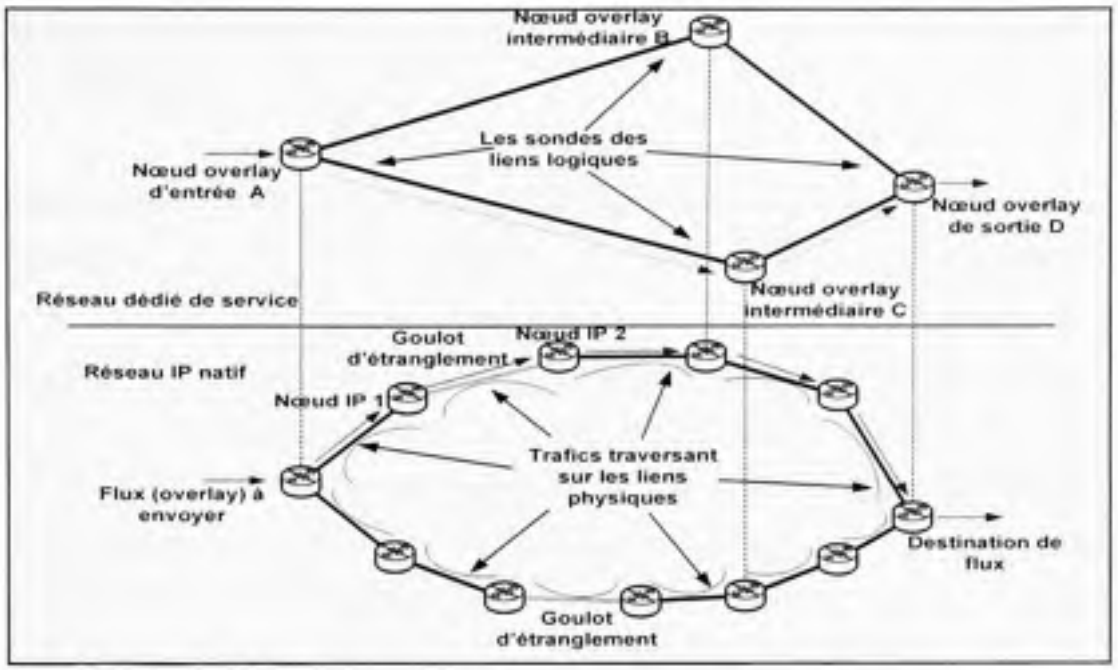

Figure 7.4 ***Trafic de sonde sur les liens logiques.***

7.4.1 Implémentation de CapProbe

CapProbe est employée pour estimer la capacité des goulots d'étranglement sur les liens physiques constituant les liens logiques. Par exemple, pour estimer la capacité du goulot d'étranglement de lien logique liant les deux nœuds Overlay A et B qui se situent entre les nœuds IP 1 et 2 respectivement (Figure 7.4), il faudra que A envoie ces paquets de sonde vers B directement et à travers le nœud 1. Le nœud Overlay B va recevoir ces paquets, faire le calcul et retourner la valeur au nœud A.

À noter que, CapProbe sera exécutée toujours avant l'utilisation de la méthode de mesure de la bande passante disponible SPRUCE, car cette dernière a besoin de connaître en priori la valeur de la capacité du goulot d'étranglement pour l'inclure dans son calcul afin qu'elle puisse estimer la bande passante disponible.

Dans notre cas, les paramètres de CapProbe seront comme le suivant :

1. la taille des paquets de sonde sera 700 octets;

2. le temps séparant deux paquets collés est de l'ordre de $10\,\mu s$ .impossible sur 100mbps

Le nœud Overlay A envoie périodiquement des paires de paquets de sonde UDP séparés par un intervalle de temps de 1,1 ms, tandis que les paquets collés sont séparés par un intervalle de $10\,\mu s$. Le nœud A horodate les paquets envoyés et les étiquettes (*Tagging*) afin que le récepteur (ici c'est le nœud B) puisse les classifier en paires. La classification des paquets en paires sert à calculer les sommes des délais et à déduire le temps T de service du deuxième paquet et calcule la capacité du goulot d'étranglement, et bien sûr en sachant la taille des paquets de sonde et en utilisant la formule 6.2. Cette procédure se répète jusqu'à l'obtention de 100 échantillons (résultats) pour choisir la valeur maximale de la capacité C correspondante à la valeur minimal de la somme des délais des paires des paquets.

7.4.1.1 Tests et résultats

Nous avons exécuté la méthode CapProbe dans les trois scénarios de notre simulation pour observer les caractéristiques de sa performance dans les conditions de chaque scénario. Les caractéristiques les plus importantes sont : la précision et le temps nécessaire pour obtenir une mesure. Les résultats ainsi que les paramètres des simulations sont présentés au tableau 7.1

Les résultats exposés au tableau 7.1 concernent les deux liens logiques qui sont du nœud Overlay A vers le nœud Overlay B et de A vers le nœud Overlay C. Nous avons collecté les résultats d'une simulation qui a une durée de 20 secondes.

CapProbe était capable d'estimer la valeur exacte de la capacité maximale des deux liens logiques (A-B, A-C) qui correspond à la capacité des goulots d'étranglement des liens physiques composant ces liens logiques. CapProbe accepte la valeur de la capacité estimée quand la somme minimale des délais de cette valeur correspond à la somme des délais minimaux.

Tableau 7. 1

Simulation de CapProbe : paramètres et résultats

	Scénario léger	Scénario moyen	Scénario chargé
Charge et type du trafic traversant sur les liens de capacité 100 Mbps	25% de trafic exponentiel et CBR de la capacité	50% de trafic exponentiel et CBR de la capacité	95% de trafic exponentiel et CBR de la capacité
Nombre de paires des paquets envoyées	100	100	100
Taille des paquets (Octets)	700	700	700
Somme des délais minimaux de A à B	7.894 ms	7.894 ms	7.894 ms
Somme des délais minimaux de A à C	10.006 ms	10.006 ms	10.006 ms
Somme minimale des délais (Lien A-B)	7.894 ms	7.894 ms	7.894 ms
Somme minimale des délais (Lien A-C)	10.006 ms	10.006 ms	10.006 ms
Paires nécessaires pour converger (Lien A-B)	5	8	59
Paires nécessaires pour converger (Lien A-C)	5	8	50
Temps de convergence (Lien A-B)	500 ms	807 ms	6.405 secondes
Temps de convergence (Lien A-C)	520 ms	825 ms	5.551 secondes
La capacité estimée pour les deux liens	1.00E+07 Mbps	1.00E+07 Mbps	1.00E+07 Mbps

Plus le taux d'occupation des liens est élevé plus CapProbe aura besoin de temps ou d'échantillons afin de converger. Dans le scénario léger elle a pu estimer la capacité après 500 ms pour A-B (cinq paires) et de 520 ms pour A-C (cinq paires). Dans le scénario moyen, elle a estimé la capacité après un peu plus de temps 807 ms (huit paires) pour A-B et 825 ms pour A-B (huit paires).

Tandis que, dans le scénario ayant une charge lourde, CapProbe avait besoin d'un temps qui est beaucoup plus long que dans les deux autres scénarios. Ce temps est 6.405 secondes pour A-B avec 59 essais et de 5.551 secondes pour A-C avec 50 essais pour A-C. Donc, cette méthode peut estimer la capacité correctement dans des différentes conditions de réseau en ayant besoin de plus de temps quand la charge de trafic traversant est élevée.

CapProbe a réussi à estimer la capacité des goulots d'étranglement dans les trois scénarios. Elle avait besoin d'un temps relativement long dans le scénario chargé par rapport aux temps nécessaires dans les autres scénarios. Ce temps était nécessaire pour CapProbe afin qu'elle puisse confirmer la capacité par une valeur de 10Mbps. Donc, elle avait besoin d'environ 50 paires de paquets de sonde, car les liens ont été congestionnés et les paquets des trafics traversant ont perturbé les paires des paquets de mesure. Cette perturbation a causé des surestimations parfois er des fois des sous-estimations. Ce qui a exigé CapProbe ce consommer ce temps afin de converger.

7.4.2 Implémentations de SPRUCE

Après avoir calculé la capacité C du goulot d'étranglement par CapProbe, SPRUCE doit être exécutée en incluant la valeur de C calculée dans sa formule pour estimer la bande passante disponible sur le lien logique.

Nous avons vu dans le cinquième chapitre que SPRUCE utilise des paquets de sonde de taille 1500 octets et elle maintient le temps séparant deux paquets d'une paire Δ_{in} au temps de transmission de 1500 octets (12000 bits) sur le lien goulot d'étranglement (de capacité 10

Mbps), donc dans notre cas 0,0012 secondes ou 1.2 ms. Également, SPRUCE maintient le débit de trafic de sonde équivalent à 5% de la capacité du goulot d'étranglement et de façon qu'il soit inférieur à 240Kbps. SPRUCE le réalise en ajustant le temps séparant chaque paires consécutives à la sortie d'une fonction de distribution exponentielle ayant un moyen τ beaucoup plus grand que Δ_{in}. La première estimation de la bande passante disponible sort seulement après avoir mesuré 100 échantillons par la formule 5.3. SPRUCE utilise ensuite une fenêtre coulissante des 100 derniers échantillons pour rapporter la bande passante disponible.

Par exemple, si le nœud Overlay A veut savoir la disponibilité de la bande passante sur le lien qui le connecte avec B (Figure 7.4), il attend quelque temps pour connaître la capacité du goulot d'étranglement entre eux à l'aide de CapProbe, puis il va déclencher la procédure de mesure de la bande passante disponible. Le nœud B, qui reçoit les paquets de sonde, va estimer la bande passante disponible de A vers lui après 100 échantillons et retourne ce résultat vers A.

7.4.3 Implémentation de PathLoad

La méthode de mesure PathLoad est réalisé par deux clients principaux : PathLoad_Send et PathLoad_Receive [Dovrolis (2006)]. Les deux clients doivent exister sur chaque nœud d'une paire de nœuds Overlay voulant estimer la bande passante disponible entre eux. PathLoad_Send commence par l'envoi d'un flot de 12 flux constitué chacun de 100 paquets horodatés à l'envoi vers le client PathLoad_Receive qui est toujours à l'écoute et avec un intervalle Δ = 90 ms. Une fois la procédure déclenchée, les deux clients coopèrent en exécutant les algorithmes d'envoi des flots des flux, d'ajustement du taux des flux et de l'estimation jusqu'à la détermination de la bande passante disponible chez le récepteur qui va à son tour va retourner le résultat à celui qui a initialiser l'opération.

7.4.4 Tests de SPRUCE et de PathLoad sur le lien A-B et comparaison de résultats

Les deux méthodes de mesure SPRUCE et PathLoad sont exécutées dans les trois scénarios (léger, moyen et chargé) pour observer leurs performances dans le même environnement de simulation où l'algorithme de routage sera employé (le réseau SON). Cela nous aide à étudier l'influence de la congestion sur leur précision et sur leur temps de calcul de la bande passante disponible.

Les tests des méthodes de mesure sont faits sur le lien A-B dans les trois scénarios. Les niveaux de trafics traversant sur ce lien logique respectent les taux de trafic des trois scénarios et ils ont une fluctuation dans le taux de trafic traversant du goulot d'étranglement, mais toujours dans les limites des taux des trois scénarios. Ces tests, dans ces conditions où le trafic traversant varie brusquement, permettent d'expérimenter la fiabilité des méthodes de mesure.

Nous commençons par le scénario léger, la figure 7.5 montre les estimations des deux méthodes de mesure sur le lien A-B.

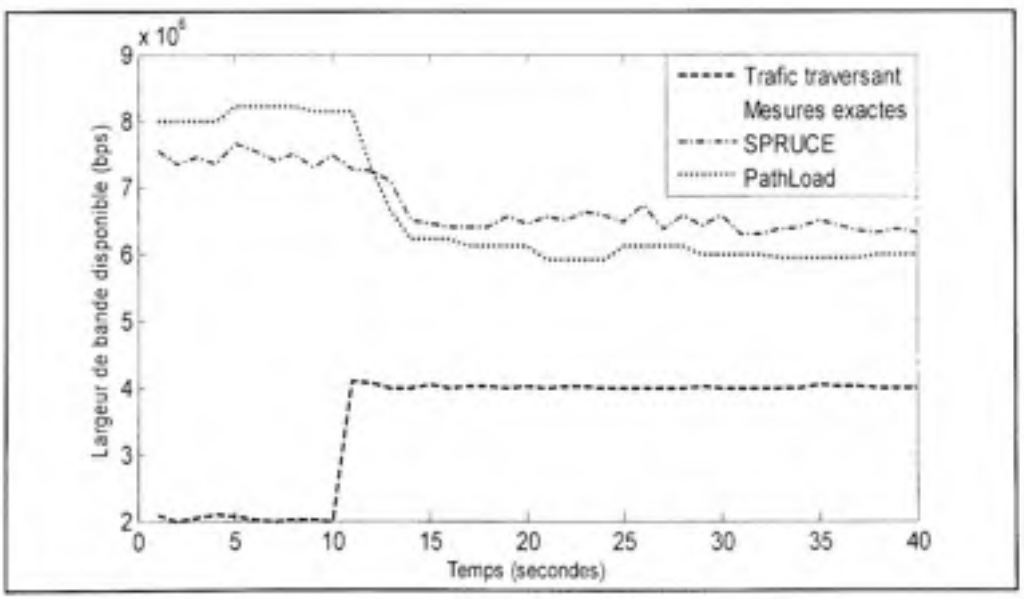

Figure 7.5 ***La bande passante disponible sur le lien A-B dans le scénario léger.***

La figure 7.6 montre les estimations dans le scénario de charge moyenne.

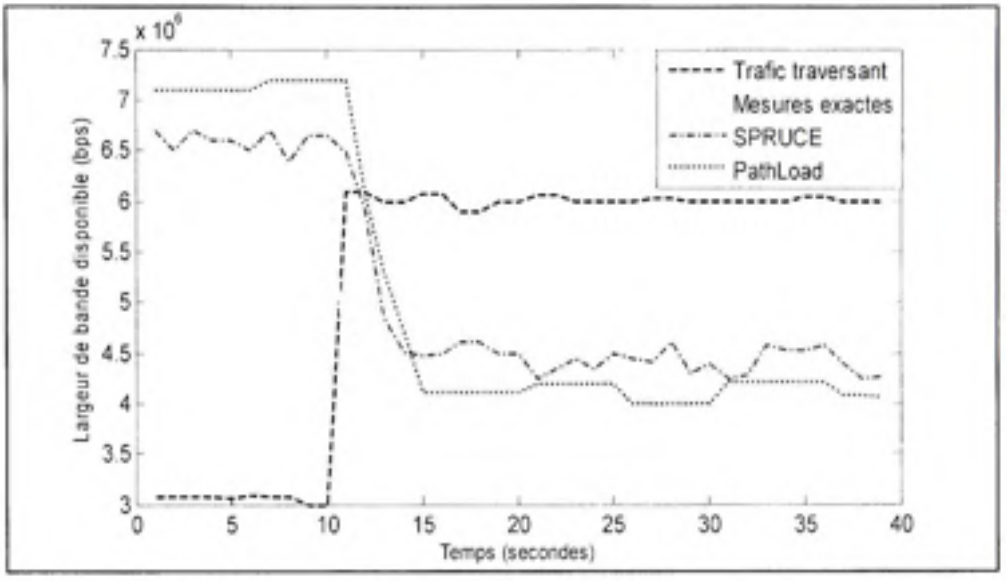

Figure 7.6 ***La bande passante disponible sur le lien A-B dans le scénario moyen.***

Tandis que la figure 7.7 montre les estimations dans le scénario chargé.

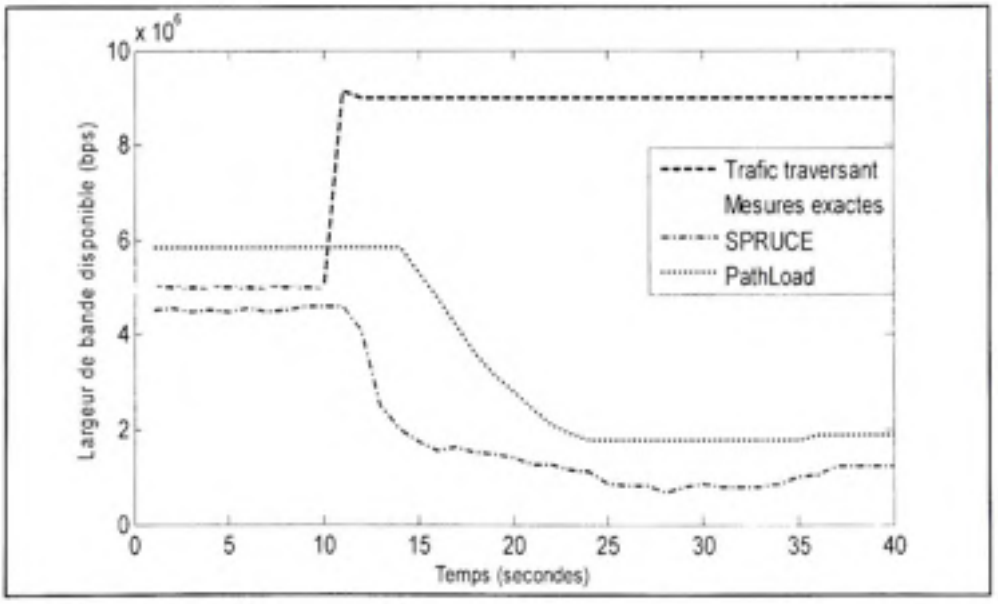

Figure 7.7 ***La bande passante disponible sur le lien A-B dans le scénario chargé.***

Concernant SPRUCE, nous remarquons de ces tests qu'elle a un comportement systématique dans les trois scénarios. Au début elle sous-estime la bande passante disponible puis ses estimations s'élèvent jusqu'à une surestimation dans les scénarios léger et moyen et jusqu'à une estimation presque bonne dans le scénario chargé.

En effet, Δ_{out} est le paramètre le plus important dans l'opération de l'estimation de SPRUCE. Si Δ_{out} augmente, c'est-à-dire la bande passante disponible diminue (trafic traversant dans la queue de plus), et au contraire. Au début, SPRUCE sous-estime la bande passante disponible puis elle la surestime avec une tendance vers les valeurs correctes. Ce comportement est dû de la nature des paquets de trafic traversant (la plus grande portion a une taille plus petite que 1500 octets), l'intensité de ce trafic et sa variation ainsi que l'intensité de trafic traversant après le goulot d'étranglement. Le fait que SPRUCE doit encadrer des paquets de trafic traversant entre les deux paquets de ces paires afin qu'elle calcule la bande passante disponible donne à SPRUCE ce comportement et la fait dépendante de trafic traversant.

Ainsi, quand le taux de trafic traversant augmente brusquement, c'est-à-dire la largeur de bande disponible tombe rapidement, nous remarquons que les estimations de SPRUCE n'ont pas besoin de beaucoup de temps pour réagir. En plus, ses résultats suivent rapidement les valeurs exactes et toujours avec une surestimation. Les valeurs obtenues dans le scénario chargé sont toujours les meilleurs à cause des valeurs de Δ_{out} qui dépendent de l'intensité de trafic traversant et par conséquences de tailles des files d'attente après les goulots d'étranglement.

Concernant PathLoad et d'après les tests faits, nous constatons qu'elle a réussi d'estimer la largeur de bande disponible de bonne façon dans le léger scénario et elle a donné toujours des résultats stables. Par contre, elle l'a surestimé quand le réseau est devenu chargé. Ce comportement est dû à cause de l'effet de l'intensité de trafic traversant qui aidait à pousser et à augmenter les limites de l'intervalle de l'estimation, donc une augmentation de leur moyenne et par conséquence l'augmentation dans les valeurs de la bande passante

disponible estimée. PathLoad avait une réaction relativement lente avec les changements brusques (fluctuations de trafic traversant) dans le scénario chargé et son algorithme essayait toujours de trouver le point de tournage. Ce comportement est dû de la congestion dans le réseau en générale ce qui cause des pertes dans les flux de PathLoad et des résultats ambigus en exigeant des traitements relativement longs. Alors, elle traite difficilement l'état de la largeur de bande disponible pendant les fluctuations de trafic traversant et son temps de mesure va augmenter par conséquence.

La comparaison des méthodes de mesure se fait par rapport aux paramètres importants : la précision de mesure et le temps nécessaire pour avoir une mesure.

Sur la précision et d'après les tests faits, nous constatons que PathLoad performe mieux que SPRUCE quand la charge de réseau est légère ou moyenne (Figures 7.5 et 7.6). D'autre part, SPRUCE performe mieux que PathLoad au début du scénario chargé même si elle sous-estime, car ces estimations sont plus proches des valeurs exactes (Figures 7.7), après un certain temps, la performance de SPRUCE est devenue meilleure qu'au début. Ses estimations sont devenues plus proches de valeurs exactes et presque stables. Nous pouvons dire que la performance de SPRUCE est toujours mieux que la performance de PathLoad en termes de précision quand le réseau est chargé.

Par rapport au deuxième paramètre important des méthodes de mesure qui est le temps d'une mesure (la rapidité d'avoir ou de retourner une valeur nouvelle à l'algorithme de routage), nous procédons par le rappel qu'avant que SPRUCE ne puisse estimer la bande passante disponible, elle a besoin d'attendre quelque temps afin d'obtenir la valeur de la capacité du goulot d'étranglement calculée par CapProbe. De plus, nous avons remarqué que la valeur de ce temps est variable et qu'elle dépend du taux de trafic traversant sur le réseau. Ainsi, SPRUCE attends jusqu'à la réception de 100 premiers échantillons pour qu'elle retourner la première estimation de la bande passante disponible puis elle utilise une fenêtre coulissante pour les 100 derniers échantillons.

Donc, il faut ajouter le temps de calcul de la capacité au temps de calcule de la première estimation pour avoir réellement une estimation primaire de la bande passante disponible.

Le temps des 100 1ers échantillons de SPRUCE dans les trois scénarios ainsi que le temps de la 1ère estimation sont présentés au tableau 7.2

Tableau 7.2

Temps de la 1ère estimation de SPRUCE sur le lien A-B

Scénarios	Temps de CapProbe	Temps des 100 1ers échantillons de SPRUCE	Temps de la première estimation de SPRUCE
Léger	0.500 secondes	11.6 secondes	12.1 secondes
Moyen	0.807 secondes	11.82 secondes	12.627 secondes
Chargé	6.405 secondes	13.9 secondes	20.305 secondes

L'algorithme de routage peut commencer à compter sur SPRUCE après le passage de temps de la 1ère estimation. Après ça, la méthode va posséder toujours une valeur prête pour l'utilisation et qui couvre les 100 derniers frais échantillons.

En effet, SPRUCE a besoin au début de quelque temps afin de recevoir la capacité du lien de CapProbe pour déclencher son processus de mesure de la bande passante disponible qui a besoin d'attendre jusqu'à la réception du centième échantillon afin qu'elle puisse avoir la première estimation. Après, et en utilisant la fenêtre coulissante, SPRUCE possède toujours une valeur prête pour l'utilisation par l'algorithme (la moyenne des 100 dernières mesures).

D'autre part, PathLoad est obligée de lancer le processus de mesure au complet à chaque fois elle aura besoin d'une mesure. C'est-à-dire, elle mesure la bande passante disponible une fois et elle retourne cette valeur à l'algorithme de routage afin qu'il l'utilise. Nous

lançons donc dans nos tests la méthode PathLoad d'une façon continue afin d'être capable toujours d'avoir une valeur à jour de la bande passante disponible.

La figure 7.8 fait une comparaison de temps des mesures des deux méthodes et elle montre que le temps nécessaire pour SPRUCE (20 secondes en moyenne) est relativement long spécialement quand le réseau est chargé. À noter que ce temps est nécessaire juste au début de processus de SPRUCE, après le passage de ce temps SPRUCE possède toujours une valeur prête pour l'utilisation. Cette augmentation est due à la perte des paquets de sonde à cause de la congestion et des délais qui se produisent par conséquences.

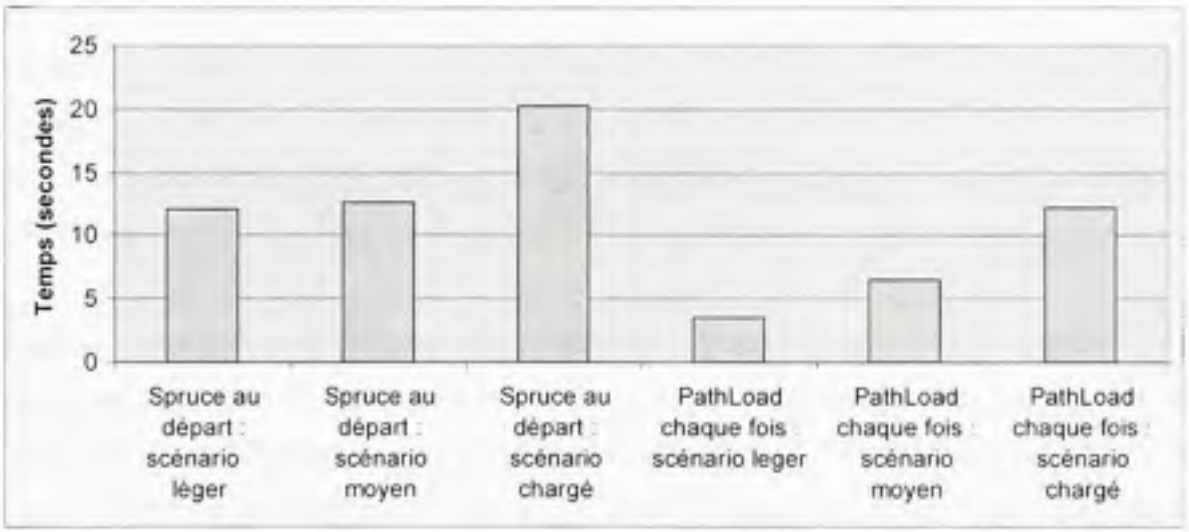

Figure 7.8 ***Temps nécessaire pour une estimation.***

Par contre, PathLoad avait trois valeurs de temps de mesure. Ces valeurs ou ces délais sont nécessaires toujours et à chaque fois elle aura besoin de savoir la disponibilité de la bande passante. Ce temps augmente pour PathLoad avec la charge de réseau, et ceci à cause de trafic traversant les liens physiques et à cause des pertes que peuvent subir les paquets de sonde en causant des résultats ambigus. Cela aussi causer l'abondant des quelques flux ou flots. Ce qui nécessite l'envoi de flots de plus pour estimer la largeur de bande, alors un temps plus long.

CHAPITRE 8

SIMULATIONS DES ALGORITHMES

Le chapitre précédent s'est attardé à l'implémentation et aux tests des méthodes de mesure de la largeur de bande disponible lesquelles seront utilisées par les algorithmes de routage. Les résultats obtenus permettront de mieux comprendre l'impact des différentes méthodes sur le comportement et sur la performance de chacun des algorithmes. Finalement, ce chapitre sera consacré à simuler ces algorithmes de routage. Ils seront comparés afin d'en déterminer le plus efficace.

En effet, à l'arrivée d'un flux Overlay, le nœud d'entrée détermine le meilleur chemin Overlay vers le nœud de sortie en se basant sur l'état des liens Overlay (qui fournit une bande passante disponible maximale). Ainsi, les méthodes de mesure intégrées doivent être prêtes avant qu'un flux Overlay soit routé.

Pendant la durée de vie des flux et pour fournir une robustesse contre les pannes et les variations de charge du réseau, le nœud d'entrée continue à mesurer la bande passante disponible sur le chemin par lequel le flux est actuellement acheminé. S'il trouve que la largeur de bande fournie par le chemin actuel est insuffisante, il réacheminera le flux par un autre chemin qui peut fournir la largeur disponible maximale. Par contre, si la largeur de bande est toujours suffisante le flux reste sur le même chemin.

L'algorithme de routage prends décide par quel chemin il doit acheminer le flux en utilisant l'état de liens Overlay comme entrée. Dans notre étude, cette entrée est représentée par la largeur de bande disponible de liens.

Les paramètres importants dont nous tenons compte dans nos simulations sont :

1. Le délai de mesure D_m : Ce paramètre dépend directement de la méthode de mesure employée et dans quelles conditions de réseau. Ainsi, nous avons vu dans le chapitre

précédant que ce paramètre est variable d'une méthode à une autre et d'un scénario à un autre selon le taux de charge de réseau;

2. La période de mise à jour d'état de lien P_r : ce paramètre dépend de délai de mesure. Une fois un nouvel état de lien obtenu avec PathLoad, la base de données des états de liens sera mise à jour;
3. Le délai de dissémination de l'état de lien D_d : c'est le temps nécessaire pour envoyer l'état d'un lien vers les nœuds Overlay. Il est de l'ordre de [0, 0.2] secondes;
4. La période de mise à jour de chemin : du fait que les délais de mesure de PathLoad sont longs nous avons choisi que; cette période soit égale à la période de mise à jour d'état de lien. D'autre part, cette période est égale à une demi-seconde quand SPRUCE est employée.

En effet, il est nécessaire pour l'algorithme de routage de choisir un chemin parmi les différents chemins alternatifs existants entre deux nœuds Overlay d'entrée et de sortie en se basant sur les paramètres de flux suivants : le nœud d'entrée, le nœud de sortie, le débit maximal de flux (r) et sa durée. À noter que, nous pouvons avoir des cas où le débit achevé a par le flux est moindre que son dédit maximal. Au début, l'algorithme achemine le flux par le chemin qui fournit la bande passante disponible maximale ou qui peut satisfaire le débit maximal de flux. Quand le débit achevé sera moindre que le débit maximal du flux l'algorithme cherche pour un nouvel chemin qui peut fournir une marge maximale de bande passante et re-route le flux par le chemin trouvé.

8.1 Études des algorithmes de routage et comparaison de performance

Le but essentiel des réseaux dédiés de service est d'établir une infrastructure logique qui peut fournir une qualité de service de bout en bout améliorée en satisfaisant les exigences des clients des réseaux SON. La satisfaction des clients se traduit dans le provisionnement des paramètres de qualité de service satisfaisant à leurs applications. C'est-à-dire, le réseau

dédié de service doit fournir une connectivité avec des paramètres acceptables de qualité de service.

Nos études des algorithmes de routage adaptatif pour le réseau dédié de service s'axent sur les tests de performances des applications (des trafics) des clients en nous concentrant sur les paramètres de qualité de service cités ci-dessus. Donc, nous allons étudier dans nos tests et simulations la performance des applications quand nous employons le routage Overlay avec une méthode de mesure différente de la bande passante disponible chaque fois et cela dans les trois scénarios de simulations (léger, moyen et chargé).

En effet, nous allons observer ces paramètres de performance de trafic envoyé encore en appliquant au processus de mesure des valeurs exactes dans les trois scénarios, donc les valeurs de mesure de la bande passante disponible seront sans erreurs. De même, un test de performance sera fait en employant le routage traditionnel, c'est-à-dire un algorithme de routage traditionnel (*Link state*) sera employé pour acheminer notre trafic (Overlay) vers sa destination dans le même réseau et dans les trois scénarios.

Enfin, nous faisons une comparaison de performance de l'application (délai, gigue et taux de perte du trafic envoyé) avec les algorithmes Overlay qui emploient différentes méthodes de mesure de la bande passante disponible (SPRUCE, PathLoad et exacte) et le routage traditionnel pour observer les avantages de routage Overlay par rapport au traditionnel et pour comparer le routage Overlay quand nous employons des méthodes de mesure différentes.

8.2 Simulations et tests

Afin d'effectuer une comparaison efficace de performance des applications et en employant des algorithmes de routage adaptatif utilisant différentes méthodes de mesure, nous avons simulé notre réseau dans les trois scénarios (léger, moyen et chargé) avec les deux méthodes de mesure (SPRUCE et PathLoad) et en employant une à la fois. Puis, nous avons répété la

procédure en appliquant des mesures exactes de la bande passante disponible, cela pour comparer l'effet des erreurs des estimations des méthodes de mesure sur la performance des applications. Enfin, nous avons envoyé le même trafic Overlay mais en employant le routage traditionnel. Cela nous permettra d'observer en général l'avantage de routage Overlay par rapport au routage traditionnel.

8.2.1 Taux de perte

Les résultats obtenus nous permettent de constater que lorsque le réseau n'est pas chargé tous les algorithmes performent de façon similaire même celui de routage traditionnel (Figure 8.1).

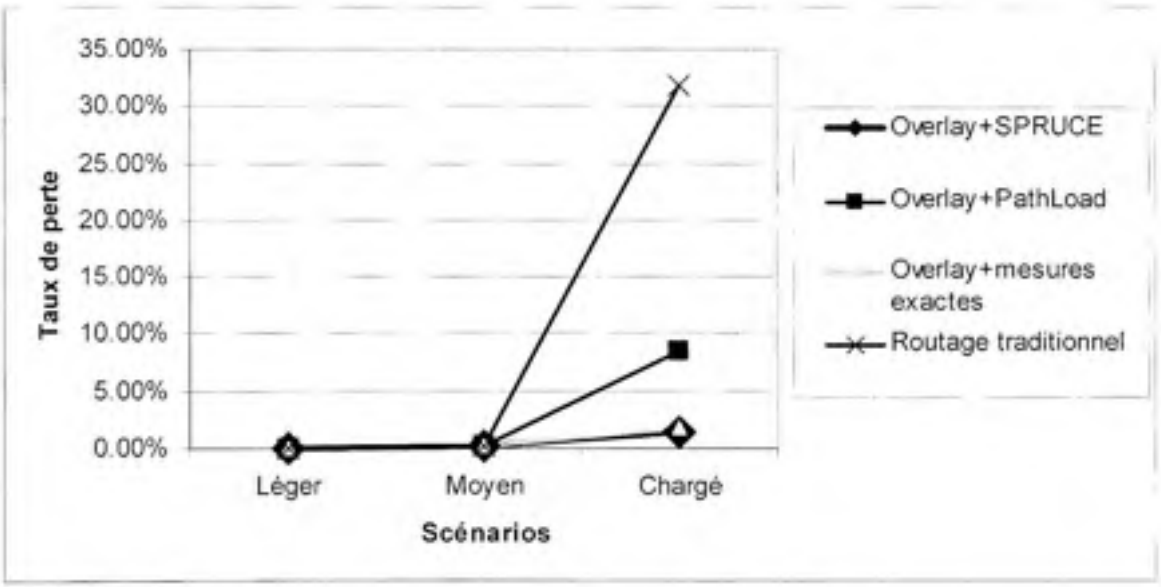

Figure 8.1 ***Taux de perte de flux Overlay avec les quatre algorithmes.***

En effet, dans les scénarios léger et moyen, la largeur de bande disponible sur les liens du réseau (jusqu'au 50%) permet d'acheminer le trafic facilement et avec une bonne performance, rien n'empêche le trafic d'écouler. Mais tous ça se change quand le réseau est devenu chargé. Les résultats obtenus montrent l'avantage du routage Overlay par rapport au

routage traditionnel et ils nous rassurent encore que la performance de routage Overlay est dépendante de la performance de la méthode de mesure.

La performance de routage est meilleure avec SPRUCE qu'avec PathLoad ce qui reflétait un taux d'erreur plus petit avec l'algorithme qui emploie SPRUCE. Ces résultats convergent avec les conclusions des tests des méthodes de mesure où nous avons obtenu une meilleure performance avec SPRUCE dans le scénario chargé. Ainsi, le temps nécessaire pour PathLoad pour calculer une estimation, qui se compte relativement long par rapport au celui de SPRUCE, rend encore le routage avec SPRUCE mieux qu'avec PathLoad car SPRUCE réagit plus rapidement aux changements brusques de trafic traversant, ce qui permet d'économiser des pertes de plus de trafic Overlay et de pendre la décision de changement le chemin.

Le routage Overlay provisionné par des mesures exactes de la largeur de bande disponible performe presque de la même façon que celui qu'avec SPRUCE et il n'a pas montré beaucoup d'avantages. Ces résultats sont expliqués par le fait que l'algorithme de routage adaptatif, dans la plupart des moments, cherche un chemin qui offre de la largeur de bande disponible de plus et non un chemin spécifique. C'est-à-dire que même si l'erreur d'estimation était plus grande, l'algorithme de routage va réacheminer le trafic Overlay de son chemin actuel, qui manque la largeur de bande nécessaire, vers le chemin qui offre la largeur de bande maximale selon la priorisation des chemins qu'il a établie après les mesures, donc l'erreur n'a pas beaucoup d'effet.

Par rapport à la performance avec le routage traditionnel, la large différence entre la performance de routage Overlay en général et la performance de routage traditionnel est évidente. Cela s'explique par le fait le dernier achemine toujours le trafic par le chemin qu'il l'a considère meilleur, et il continuera à le faire même si ce chemin est congestionné et en ne sachant pas des mauvaises conditions de réseau.

8.2.2 Délai de bout en bout

Le deuxième paramètre important étudié était le délai moyen de bout en bout des paquets de flux Overlay. La performance de routage qui reflète la performance des applications est toujours dépendante de la performance de la méthode de mesure employée (Figure 8.2).

Il n'y a pas une grande différence entre tous les algorithmes Overlay dans le scénario léger et dans le scénario moyen. La même explication s'applique toujours, la largeur de bande disponible dans ces deux scénarios permet au trafic Overlay de réaliser une même bonne performance avec tous les algorithmes. Par contre, le routage avec SPRUCE a réussi d'avoir des délais plus petits que ceux réalisés par le routage avec PathLoad. En effet, la rapidité de SPRUCE pour réagir avec les variations de trafic traversant permet d'éviter la file saturée du goulot d'étranglement et d'aller vers un autre chemin offrant plus de la bande passante plus rapidement. C'est-à-dire, de réacheminer le trafic par un chemin qui est moins chargé et par conséquences avoir des meilleurs délais de bout en bout.

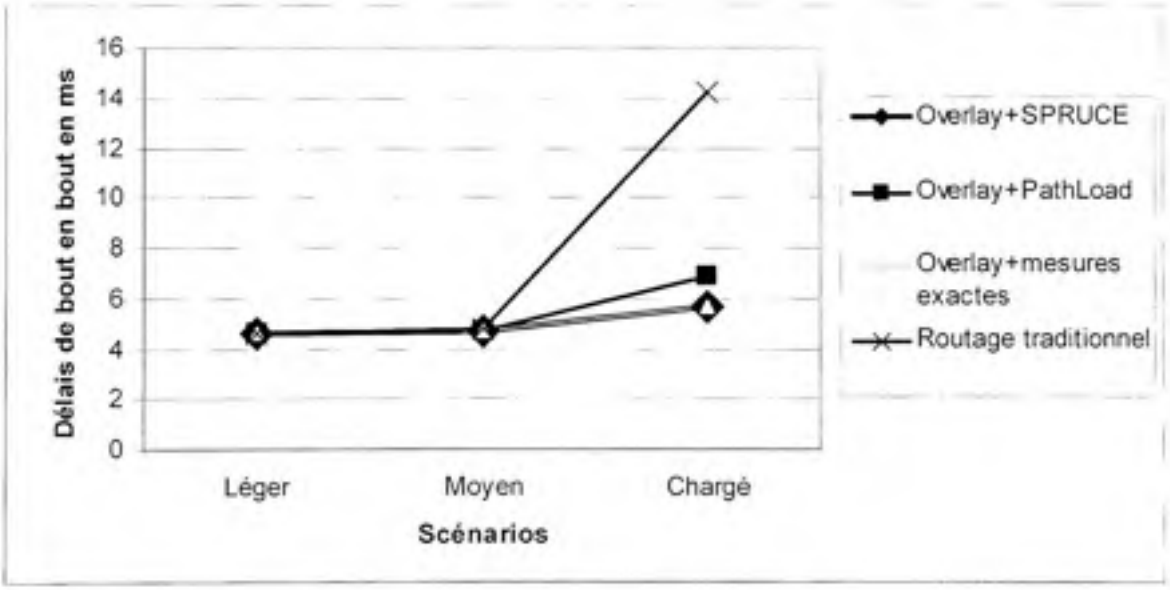

Figure 8.2 ***Délais de bout en bout de flux Overlay avec les quatre algorithmes.***

Par rapport au routage avec mesures exactes, il avait presque la même performance que celui avec SPRUCE. Le routage Overlay cherche toujours le chemin qui a une bande passante disponible plus grande par priorisation et malgré les erreurs d'estimation.

Le routage traditionnel avait des grandes valeurs dans le scénario chargé, cela indique qu'après certain temps la performance de l'application génératrice de trafic va tomber. L'avantage du routage Overlay est évident.

8.2.3 La gigue

Toujours le même comportement avec tous les algorithmes et dans le scénario léger et dans le scénario moyen et bien sûr à cause de la disponibilité de la largeur de bande suffisante pour le trafic envoyé. D'autre part, dans le scénario chargé le trafic acheminé par le routage Overlay employant SPRUCE avait les meilleures valeurs de la gigue, donc une performance plus stable que celle du routage employant PathLoad (Figure 8.3).

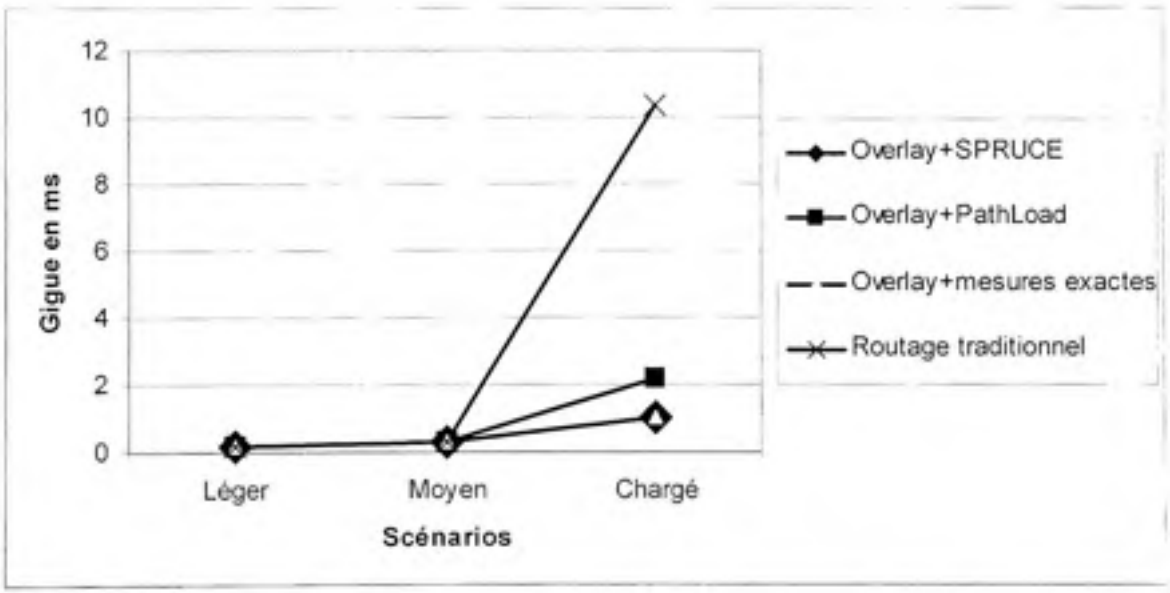

Figure 8.3 ***La gigue de flux Overlay avec les quatre algorithmes.***

Nous avons obtenu des valeurs de la gigue beaucoup plus grandes avec le routage traditionnel et la réaction de la performance de l'application en face de la variation des conditions de réseau n'est pas prévue.

CONCLUSION

Dans le contexte des réseaux Overlay et de routage robuste contre les pannes, ce projet de recherche a présenté une étude des algorithmes de routage adaptatif pour le réseau dédié de service. Le but de ce projet était d'étudier la performance de ces algorithmes en employant des différentes méthodes de mesure de la bande passante disponible. Après avoir choisi, testé et intégré les méthodes de mesure, les algorithmes de routage adaptatif de réseau SON sont comparés en se basant sur les paramètres de qualité de service (délai de bout en bout, la gigue et le taux de perte) de flux Overlay acheminés. Plus ces paramètres obtenus sont meilleurs, plus l'algorithme est considéré comme efficace. Une comparaison de performance entre le routage adaptatif dédié de service et le routage traditionnel a été présentée pour montrer les avantages que les algorithmes de routage adaptatif dans le réseau dédié de service offrent par rapport au routage traditionnel.

Les simulations ont montré que la performance des algorithmes de routage adaptatif de réseau SON dépend de la performance de la méthode de mesure employée par ces derniers et des conditions actuelles du réseau. Tous les algorithmes ont presque la même performance dans un réseau qui a des bonnes conditions, c'est-à-dire dans un réseau qui a une charge légère ou moyenne. Par contre, l'algorithme qui emploie SPRUCE offre toujours des meilleurs paramètres de qualité de service que celui qui emploie PathLoad dans les mauvaises conditions du réseau ou quand ce réseau est chargé. Cela s'explique par le fait que SPRUCE possède toujours une valeur prête à être utilisée par l'algorithme. En plus, SPRUCE plus rapide que la méthode PathLoad dans les cas de variations brusques de la bande passante disponible même si le réseau est chargé. D'autre part, la performance de PathLoad se dégrade quand les conditions de réseau deviennent difficiles. Cela s'explique par le fait que PathLoad présente les deux inconvénients suivants :

1. Incrémentation du temps nécessaire pour une estimation. Ce temps augmente considérablement avec l'augmentation de la charge du réseau et avec la variation brusque de la largeur de bande disponible.
2. Surestimation de la bande passante disponible quand le réseau devient chargé;

L'algorithme utilisant des mesures de la largeur de bande passante disponible exactes performe un peu mieux que celui qui emploie SPRUCE, cependant; parfois ils performent pareillement. En effet, les algorithmes de routage adaptatif de réseau SON, dans la plupart de temps, cherchent à acheminer les flux Overlay par les chemins qui peuvent offrir la largeur de bande disponible la plus grande. C'est-à-dire, ils choisissent le chemin (parmi plusieurs) qui offre la bande passante disponible maximale malgré le taux d'erreur existant dans l'estimation.

L'étude des algorithmes de routage adaptatif de réseau dédié de service qui utilisent la bande passante disponible comme paramètre de routage ont démontré qu'ils fournissent une qualité de service pour les applications multimédias nettement meilleure que celle offerte par les algorithmes de routage traditionnels.

Dans le but de limiter les incertitudes sur les performances de méthodes de mesures et d'algorithme de routage nous avons utilisé dans les simulations une topologie qui n'est pas très compliqué. Dans le futur, il serait intéressant de tester l'algorithme de routage réactif avec les méthodes de mesure dans une topologie plus large et plus compliquée. Il est aussi très intéressant de veiller sur l'avancement des techniques de mesure de la bande passante disponible qui sont actuellement en forte croissance. L'application de nouvelles méthodes de mesure et la comparaison de leur performance formeront une étape très importante et intéressante dans les études de performance des algorithmes de routage adaptatifs pour le réseau dédié de service.

BIBLIOGRAPHIE

Andersen, David. 2001. « Resilient Overlay Networks ». Mémoire de maîtrise en informatique et génie, Massachusetts, Massachusetts Institute of Technology. 86 p.

Andersen, David, Hari Balakrishnan, Frans Kaashoek et Robert Morris. 2001. « Resilient Overlay Networks ». In *18th ACM Symposium on Operating Systems Principles* (Banff, 21-24 octobre 2001). 15 p. New York (NY) : ACM.

Chen, Ling-Jyh, Tony Sun, Guang Yang, M. Y. Sanadidi et Mario Gerla. 2005. « End-to-end asymmetric link capacity estimation ». In *IFIP Networking 2005* (Waterloo, mars 2-6 2005). p. 780-791. USA : Springer.

Clark, Dave, Bill Lehr, Steve Bauer, Peyman Faratin, Rahul Sami et John Wroclawski. 2006. « Overlay Networks and the Future of the Internet ». *COMMUNICATIONS & STRATEGIES*, volume 63, 3eme quart 2006, p. 1.

Constantine Dovrolis. Mai 2006. « *PathLoad source code et PathLoad Manual* ». En ligne. < http://www.cc.gatech.edu/~dovrolis/pathload.html >. Consulté le 30 novembre 2006.

Constantine Dovrolis. 2006. « *Measurement tools for the capacity and load of Internet Paths, Background* ». En ligne. < http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw.html >. Consulté le 9 septembre 2006.

Duan, Zhenhai. Zhi-Li Zhang et Yiwei Thomas Hou. 2003. « Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning ». *IEEE*, volume 11, issue 6, décembre, 870-883 p.

Fall, Kevin et Kannan Varadhan. Décembre 2006. « The ns Manual: formerly ns Notes and Documentation ». En ligne. 415 p. < http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf >. Consulté le 10 décembre 2006.

Fan, Jinliang et Mostafa H. Ammar. 2004. « Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies ». *In INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings.* (Barcelone, avril 2006). p. 1-12. IEEE.

Gu, Xiaohui, Klara Nahrstedt, Rong N. Chang et Christopher Ward. 2003. « QoS-Assured Service Composition in Managed Service Overlay Networks ». In *The 23rd International Conference on Distributed Computing Systems.* (Providence, 19-22 mars 2003). p. 194-201. IEEE.

Guerrero, Cesar D. et Miguel A. Labrador. 2006 « Experimental and Analytical Evaluation of Available Bandwidth Estimation Tools ».In *Proceedings of 31st IEEE Conference on Local Computer Networks.* (novembre 2006). P. 710-717. IEEE.

Jane, Manish et Constantinos Dovrolis. 2002. « Pathload: A measurement tool for end-to-end available Bandwidth ». In *proceedings of Passive and Active Measurements (PAM) 2002 workshop* (Colorado, mars 2002). 12 p. New York (NY) : ACM.

Jean-Louis Mélin. 2001. *Qualité de Service sur IP*. Paris : Eyrolles, 338 p.

Kapoor, Rohit, Ling-Jyh Chen, Li Lao, Mario Gerla et M.Y. Sanadidi. 2004. « CapProbe: A Simple and Accurate Capacity Estimation Technique ». In *SIGCOMM'04* (Portland, août 30–sept. 3, 2004). p. 67-78. New York (NY) : ACM.

Kurose, James et Keith W. Ross. 2007. *Computer Networking: A top-down approach*, 4th edition. Boston (MA) : Pearson Education. 852 p.

Lao, Li, Constantine Dovrolis et M.Y. Sanadidi. 2006. « The Probe Gap Model can Underestimate the Available Bandwidth of Multihop Paths ». *ACM SIGCOMM Computer Communication Review*, volume 36, numéro 5, octobre, p. 29-34.

Leon-Garcia, Alberto et Indra Widjaja. 2004. *Communication Networks: Fundamental concepts and key architectures*, second edition. New York : McGraw-Hill. 848 p.

Li, Zhi et Prasant Mohapatra. 2004. « The Impact of Topology on Overlay Routing Service ». In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*. (Davis, 7-11 mars 2004). 415 p. Californie : IEEE.

Li, Zhi et Prasant Mohapatra. 2004. « QRON: QoS-Aware Routing in Overlay Networks ». *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, volume 22, issue 1, janvier, p. 29-40.

Marc Greis. 2006. « *Tutorial for the Network Simulator NS and Building NS* ». En ligne. < http://www.isi.edu/nsnam/ns/tutorial >. Consulté le 2 décembre 2006.

Nakao, Akihiro, Larry Peterson et Andy Bavier. 2003. « A Routing Underlay for Overlay Networks ». In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. (Karlsruhe, août 25–29 2003). p. 11-18. USA, Ney York : ACM.

Peterson, Larry et Bruce S. Davie. 2003. *Computer Networks: A System Approach*, third edition. San Francisco (CA) : Elsevier Science. 813 p.

Seetharaman, Srinivasan et Mostafa Ammar. 2005. « Overlay-Friendly Native Network: A Contradiction in Terms? ». In *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*. (Maryland, novembre 14-15 2005). 6 p. New York : ACM SIGCOMM.

Strauss, Jacob, Dina Katabi et Frans Kaashoek. 2003. « A Measurement Study of Available Bandwidth Estimation Tools ». In *Internet Measurement Conference : Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*. (Florida, october 27–29 2003). p. 39-44. New York (NY) : ACM.

Subramania, Lakshminarayanan, Ion StoicaHari, Balakrishnan et Randy H. Katz. 2004. « OverQoS: An Overlay based Architecture for Enhancing Internet QoS ». In *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1*. (San Francisco, mars 29–31 2004). 6 p. Berkeley, USA : USENIX Association.

Subramania, Lakshminarayanan, Ion Stoictia, Hari Balakrishnan et Randy H. Katz. 2003. « OverQoS: offering Internet QoS using overlays ». *ACM SIGCOMM Computer Communication Review*, volume 33, issue 1, janvier 2003, p. 11-16.

Tran, Con et Zbigniew Dziong. 2008. « Resource Adaptation for Continuous Profit Optimization in Overlay and Virtual Networks ».In *The 4th EuroNGI Conference on Next Generation Internet Networks (NGI 2008)*. (Krakow, avril 28-30 2008). 8 p. NGI Networks.

Zhu, Yong, Constantinos Dovrolis et Mostafa Ammar. 2006. « Dynamic overlay routing based on available bandwidth estimation: A simulation study ». *Computer Networks: The International Journal of Computer and Telecommunications Networking*, volume 50, issue 6, avril 2006, p. 742-762.