Amélioration de la mesure de la Bande Passante dans un réseau basé sur IP

Ahmed Ait Ali

▶ **To cite this version:**

Ahmed Ait Ali. Amélioration de la mesure de la Bande Passante dans un réseau basé sur IP. Automatique / Robotique. Université Henri Poincaré - Nancy I, 2007. Français. NNT : . tel-00203075

HAL Id: tel-00203075
https://theses.hal.science/tel-00203075

Submitted on 8 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale IAEM Lorraine
Département de formation doctorale en Automatique
UFR STMIA

Centre de Recherche en Automatique de Nancy

Amélioration de la Mesure de la Bande Passante dans un Réseau Basé sur IP

THÈSE

Soutenance publique : le 27 Novembre 2007

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité Automatique, Traitement du signal et Génie informatique)

par

Ahmed AIT ALI

Composition du jury

Rapporteurs :	M. Philippe Owezarski	Chargé de recherche (HDR) CNRS au LAAS, Toulouse
	Mme Véronique Vèque	Professeur à l'Université Paris-Sud, Orsay
Examinateurs :	M. Olivier Festor	Directeur de recherche INRIA au LORIA, Nancy
	M. Bernard Thirion	Professeur à l'Université de Haute Alsace, Mulhouse
	M. Francis Lepage	Professeur à l'Université Henri Poincaré, Nancy 1
	M. Thierry Divoux	Professeur à l'Université Henri Poincaré, Nancy 1

Nancy Université, Centre de Recherche en Automatique de Nancy — CNRS — UMR 7039

Faculté des Sciences et Techniques, B.P 239 Vandoeuvre Lès Nancy

Mis en page avec la classe thloria.

À ma grand-mère.

Table des matières

Annexe

Annexe A
La métrologie et notions associées

Table des figures

Introduction générale

Introduction

L'intérêt grandissant que suscitent les réseaux, notamment le réseau Internet et l'avènement des nouvelles technologies de l'information et les nombreuses applications associées telles que les services de téléphonie, la vidéoconférence, le peer-to-peer, la télé robotique, etc, font que le nombre d'utilisateurs (particuliers, entreprises, laboratoires, etc) est sans cesse croissant. Les nouveaux services offerts utilisent des techniques et des infrastructures déjà existantes, ces dernières ne garantissent pas toujours à ces utilisateurs un fonctionnement correct ni une qualité de prestation acceptable, en d'autres mots, ces infrastructures ne répondent plus aux exigences de certaines applications en terme de qualité de service. Par exemple, l'Internet, initialement conçu pour échanger de simples données (mail, FTP, etc), est devenu aujourd'hui un réseau universel qui sert de support à de plus en plus d'applications distribuées plus exigeantes par rapport à des contraintes de temps et sont de plus en plus gourmandes en ressources réseau (particulièrement en bande passante) comme par exemple les applications multimédias (audio et vidéo streaming, etc) [VG96], [Dem02] et les applications de contrôle/commande (la téléopération [Lel00, OF97], etc). La performance de telles applications temps réel sont dépendantes de la qualité de la coopération entre les différents éléments distants. En considérant donc le cas d'une coopération s'appuyant sur un réseau de communication à commutation de paquets, la qualité de service fournie par ce réseau à ces applications s'exprime en terme de délai, de gigue, de débit, de taux de pertes et de distribution des pertes. Pour définir cette qualité ou pour détecter des défaillances, il convient donc de mesurer les paramètres relatifs à cette qualité. Par ailleurs, au fur et à mesure de l'évolution de l'Internet et de ces applications les utilisateurs sont devenus de plus en plus exigeants et ont commencé à demander à leurs fournisseurs d'accès des garanties de services (Service Level Agreement). Les utilisateurs et les opérateurs veulent donc connaître l'état actuel des performances du réseau et de la qualité de service fournie ceci afin d'assurer la bonne supervision du contrat liant les deux parties. Ces nouveaux besoins ont considérablement contribué à l'émergence d'un nouveau domaine de recherche qui est la métrologie réseaux.

La métrologie réseaux est la science des mesures des caractéristiques du réseau et de son trafic ainsi que des paramètres de la qualité de service qu'il fournit. Cette dernière peut être déployée selon trois approches possibles : intégrée, spécifique, ou service complet [Pat03].
L'approche intégrée consiste à inclure directement dans le protocole ou l'application les primitives nécessaires pour la capture des événements pertinents et leur traitement. C'est le cas

par exemple des protocole de routage (BGP, OSPF) ou de TCP Reno [HvB98]. Ces solutions intégrées ont l'avantage d'être parfaitement adaptées aux contraintes du protocole cible. En revanche elles sont souvent inexploitables par les autres applications et ne sont donc pas considérées comme des services de métrologie à proprement parler.
Les outils spécifiques sont dédiés à une seule métrique (ou à un nombre restreint d'entre elles). Ils peuvent effectuer des mesures actives ou passives, les résultats sont récupérés par une applications en vue de les traiter et d'effectuer les actions adéquates. De par leur diversité, ces derniers peuvent être regroupés pour former un véritable service de métrologie capable de prendre en compte un grand nombre de métriques. Ce genre d'outils est utilisé généralement pour mesurer la qualité de service (QoS) d'un réseau considérée d'un point de vue utilisateurs. En effet, chaque application (utilisateur) a des besoins en termes de débit, délai, gigue, fiabilité, etc. Ces besoins sont naturellement différents d'une application à l'autre, et chaque application souhaiterait donc pouvoir bénéficier d'outils de mesure spécifiquement adaptés à ses besoins.
La troisième approche consiste à utiliser des services complets de métrologie. Cette approche est la plus adéquate à la mesure des paramètres de la qualité de service considérée d'un point de vue opérateur. Elle propose plusieurs métriques aux applications qui l'utilisent ainsi que des services annexes qui touchent au traitement des données (synthèse, prédiction, rassemblement, etc). Ces services récupèrent les mesures à partir de plusieurs sources d'informations. Ces dernières peuvent provenir des applications standard (SNMP par exemple) ou de fichiers de traces aux formats variés mais le plus souvent ces mesures proviennent d'un ou de plusieurs outils spécifiques introduits dans la deuxième approche et qui sont directement intégrés au service de métrologie ou bien distribués avec lui. La fiabilité et l'efficacité de ces services dépendent donc directement de la précision de ces outils spécifiques et de leur efficacité. Les premières tentatives de mise en œuvre de services complets de métrologie se sont soldées par des échecs notamment à cause des performances médiocres de ces outils qui sont dans la plupart des cas imprécis ou intrusifs et dans certains cas,utilisant des techniques totalement inadéquates pour la mesure de la métrique considérée. C'est dans ce cadre qu'interviennent mes travaux de thèse. En effet, notre intérêt se focalise sur les outils spécifiques introduits dans la deuxième approche et plus particulièrement sur les outils de mesure de la bande passante qui est un paramètre très important pour le bon fonctionnement de plusieurs applications distribuées et dont la détermination avec précision reste jusqu'à aujourd'hui un défi à relever.

Les travaux de cette thèse sont une contribution à l'amélioration des performances et de l'efficacité des outils de mesure de la bande passante disponible. Ils ont pour objectif :

1. d'étudier les différents modèles et outils de mesure de la bande passante disponible et d'effectuer une analyse comparative de leurs performances,
2. de mettre en évidence leurs faiblesses, de déterminer l'origine et les sources de leurs erreurs et d'évaluer leurs perturbations sur le réseau,
3. de proposer un nouveau modèle de mesure de la bande passante disponible et d'évaluer

ses performances,

4. et finalement d'étudier les paramètres du modèle proposé et d'évaluer leurs degrés d'importance pour la mesure de la bande passante disponible.

La plupart de ces objectifs ont été atteints et les actions menées sont présentées dans les contributions ci-dessous.

Contributions de la thèse

Les contributions de ce travail de thèse concernent quatre points :

1. La première contribution consiste en l'étude des différentes techniques et outils de mesure de la bande passante disponible. En effet, certaines analyses et mesures de cette métrique ont été réalisées mais nous avons remarqué l'absence d'études comparatives significatives des résultats expérimentaux. Nous avons constaté aussi l'absence d'un "outil référence" auquel seraient comparées les performances des outils de mesure de bande passante disponible lors de leur développement. Dans ce premier travail, nous avons donc effectué une analyse et une étude comparative des performances des techniques et des outils de mesure en nous basant essentiellement sur les aspects précision, charge réseau induite et temps de réponse. Les résultats de cette étude montrent que ces outils ne sont pas aussi précis que leurs auteurs le prétendent et que leurs temps de réponse sont trop élevés faisant d'eux des outils inadaptés à la reconfiguration dynamique d'applications temps réel. Dans les conditions de nos tests, les techniques à paires de paquets sont celles qui donnent les résultats les plus proches de la réalité, car elles sont les plus rapides et également les moins intrusives.

2. La deuxième contribution consiste en la proposition d'un nouveau modèle déterministe basé sur la technique de la paire de paquets pour l'estimation de la bande passante disponible dans les chemins réseaux de bout en bout. Le modèle proposé repose sur le " Probe Gap Model " qui exploite dynamiquement les informations temporelles obtenues des paquets sondes pour quantifier les différents paramètres caractérisant un goulet d'étranglement dans un chemin réseau. Les travaux réalisés dans ce domaine ont démontré que la taille des paquets sondes, utilisés pour analyser le chemin étudié, est un paramètre très important qui affecte d'une manière considérable la mesure de la bande passante disponible. Basé sur ces conclusions, nous avons proposé un modèle qui prend en compte ce paramètre et nous l'avons implémenté dans un nouvel outil de mesure appelé IGMPS (Improved Gap Model using Packet Size parameter). Nous avons évalué cet outil sur une plateforme d'expérimentation selon différents scénarios et nous avons constaté que ce dernier permet de mesurer la bande passante disponible avec une très grande précision qui dépasse largement celle offerte par les outils déjà existants. Ensuite nous avons étudié l'effet des différents paramètres du modèle proposé sur la précision de l'outil IGMPS. Nous avons constaté que

la taille des paquets sondes est un facteur décisif dans la mesure de la bande passante disponible et que pour obtenir les meilleurs résultats possibles il faut que la taille des paquets sondes soit égale ou suffisamment proche de la taille des paquets du trafic concurrent.

3. Les principaux résultats obtenus dans la deuxième contribution sont expérimentaux et les solutions proposées pour améliorer la précision des mesures sont intuitives. Une étude beaucoup plus formelle a été nécessaire afin de montrer l'interaction entre les paquets sondes et le trafic concurrent ainsi que de définir la relation entre leurs tailles respectives. La troisième contribution consiste donc en la définition d'un modèle stochastique pour la technique de la paire de paquets qui établit une relation entre les dispersions initiales des paquets et leurs dispersions finales. En se basant sur ce modèle, nous avons démontré en utilisant quelques tests expérimentaux que les propositions faites précédemment sont valides et pourraient être généralisées à tous les outils de mesure de la bande passante disponible utilisant la technique de la paire de paquets.

4. La quatrième contribution consiste en une analyse de sensibilité de la sortie du modèle proposé aux variations des paramètres d'entrée. En effet, le modèle proposé utilise plusieurs paramètres qui peuvent parfois être déterminés avec une forte incertitude. L'importance de ces paramètres et leurs effets sur la sortie du modèle peuvent être déterminés par une analyse de sensibilité globale ou par une étude de propagation d'incertitudes. Il existe plusieurs méthodes pour étudier la propagation d'incertitudes et effectuer une analyse de sensibilité, la méthode de Monte Carlo est l'une des méthodes les plus largement utilisées. Dans notre travail nous avons étudié des approches de propagation d'incertitudes basées sur une utilisation efficace et réduite des nombres de simulations du modèle (Sobol, FAST et McKay ...etc). Parmi ces différentes méthodes nous avons choisi d'utiliser la méthode de Sobol pour estimer les indices de sensibilité de notre modèle aux paramètres d'entrée car comparée aux autres méthodes, cette dernière offre les résultats les plus précis. Les résultats obtenus montrent que les dispersions inter-paquets initiales et finales sont principalement à l'origine des perturbations et des incertitudes sur la sortie du modèle proposé pour l'estimation de la bande passante disponible (avec respectivement 41% et 56% de part de responsabilité). De faibles erreurs de mesure sur ces paramètres engendreront des perturbations considérables sur l'estimation de la bande passante disponible vue le degré de sensibilité élevé de la sortie aux perturbations de ces deux paramètres. Pour améliorer la précision des outils de mesure fondés sur la technique de la paire de paquets, nous avons proposé quelques solutions qui permettrons de réduire les incertitudes sur la mesure des différentes dispersions inter-paquets qui sont dues principalement à l'estampillage des paquets sondes au niveau de l'émetteur et du récepteur.

A l'issue de ces travaux de thèse, nous avons constaté que malgré l'amélioration apportée à la précision des outils de mesure de la bande passante disponible fondés sur la technique de la paire de paquets, les mesures de ces derniers sont toujours sujettes à des erreurs qui sont dues soit aux limites des techniques de programmation utilisées, soit aux difficultés de modélisation du comportement du trafic concurrent qui traverse le réseau. Nous pensons donc que la métrologie réseaux telle qu'elle est aujourd'hui, c'est à dire une métrologie mise en place à l'aide d'outils logiciels n'a pas beaucoup d'avenir et que d'ici quelque années cette métrologie logicielle sera remplacée par d'autres techniques beaucoup plus performantes fondées sur des outils matériels.

Organisation du document

Ce document est composé de 5 chapitres organisés en deux parties.

La première partie de ce manuscrit représente l'état de l'art du domaine, elle est composée du chapitre 1 qui se propose de faire une introduction à la métrologie réseaux. Ce chapitre définit les différents paramètres de la qualité de service et les différentes techniques utilisées pour leur estimation en se focalisant sur l'aspect bande passante. Il se propose également de dresser une liste des principales expériences en métrologie qui ont été conduites de par le monde ainsi qu'en France.

Les chapitres 2, 3, 4 et 5 forment la deuxième partie de ce manuscrit. Ils décrivent les travaux effectués dans cette thèse. Le chapitre 2 décrit les outils de mesure de la bande passante disponible les plus connus et effectue une analyse comparative de leurs performances. Le chapitre 3 présente la contribution principale de cette thèse. Dans ce chapitre nous présentons un nouveau modèle de mesure de la bande passante disponible à base duquel nous développons un nouvel outil de mesure. Nous validons ce dernier à l'aide d'expérimentations sur une plateforme de mesure isolée. Nous étudions ensuite son comportement par rapport à ces différents paramètres et nous le comparons aux autres outils existants. Le chapitre 4 propose un modèle stochastique pour la technique de la paire de paquets qui établit une relation entre les dispersions initiales et les dispersions finales. Les résultats de l'étude de ce modèle confirment les résultats obtenus au chapitre 3. La dernière contribution de ce travail de thèse est donnée au chapitre 5. Dans ce chapitre nous effectuons une analyse de sensibilité et une étude des incertitudes de mesures sur la sortie du modèle proposé au chapitre 3 afin de désigner les paramètres d'entrée les plus importants et dont les perturbations influencent considérablement les résultats des mesures.

La conclusion de ce manuscrit donne un récapitulatif de notre travail, effectue une critique du travail accompli et suggère des travaux futurs.

Première partie

État de l'art

Chapitre 1

Métrologie Réseaux : Outils et Techniques

Résumé

Ce chapitre définit les différents paramètres de la qualité de service et les différentes techniques utilisées pour leur estimation en se focalisant sur l'aspect bande passante. Il dresse également une liste des principales expériences en métrologie qui ont été conduites de par le monde ainsi qu'en France. Sans avoir la prétention d'être exhaustif, il offre un aperçu sur les paramètres QoS jugés importants et dresse un état de l'art des techniques et des outils de mesure qui leur sont associés. Certaines sections de ce chapitre se basent principalement sur les travaux effectués dans [Mic03] et [Mic05].

1.1 Introduction

La métrologie réseau est la science de la mesure des performances du réseau. C'est une discipline récente. Les premiers travaux significatifs et aujourd'hui encore de référence ont été menés par Paxson [Pax97] au milieu des années 90 (des travaux avaient été menés précédemment sur les lignes de télécommunications et les réseaux à commutation de circuits, ils avaient pour objectif de déterminer des critères de performance similaires tels que le délai par exemple. Mais les différences conceptuelles et techniques entre ces réseaux et les réseaux à commutation de paquets impliquaient une nouvelle approche métrologique). Celui-ci a développé une infrastructure de mesure qui lui a permis de capturer puis d'analyser 20.000 traces de connexions TCP entre 35 sites repartis sur 12 pays. L'analyse des traces a été couplée à une analyse de routes de bout en bout (déterminées avec l'utilitaire traceroute). Ces études ont permis d'observer et de mieux comprendre pour la première fois le comportement dynamique du réseau : stabilité du routage, asymétrie des routes, comportement de TCP, livraison désordonnée des paquets, etc. En effet, les travaux de Paxson ont montré que la complexité du réseau est en augmentation constante et que la mesure du comportement de ce dernier est devenue essentielle pour avoir une compré-

hension de son fonctionnement et de l'interaction des protocoles mis successivement en œuvre. Selon le résultat attendu, la nature des mesures et les méthodes employées seront bien différentes.

Ce premier chapitre s'intéresse donc à la métrologie et ces différentes applications dans le domaine des réseaux, il a pour objectif de définir les différents paramètres caractérisant la qualité de service et de décrire les différentes méthodes et techniques de mesure de ces derniers ainsi que leur mise en œuvre. Il présente les paramètres essentiels de la QoS réseau tels que le délai unidirectionnel, la variation du délai, le délai aller-retour, les pertes de paquets et la bande passante, il se focalise particulièrement sur l'aspect bande passante disponible qui est une caractéristique réseau très importante et dont la détermination avec précision reste jusqu'à aujourd'hui un défi à relever.

1.2 La qualité de service

L'Union Internationale des Télécommunications (UIT, ou ITU en anglais) dans sa recommandation E.800 [E800] a défini la qualité de service (QoS pour Quality of Service) comme étant l'effet général de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur du service. Cette définition reflète une perception de la qualité de service observée par un utilisateur. Elle repose principalement sur des critères subjectifs et dépend généralement du type de service offert. Par exemple, pour le transfert de fichiers, le critère principal de jugement pour un utilisateur sera la vitesse de transfert (ou le temps de transfert total y compris le temps de correction des éventuelles erreurs). Par contre, pour une visioconférence, il faudra que l'utilisateur soit audible et reconnaissable. Dans le cas de transactions bancaires, on s'intéressera plutôt à la fiabilité et la sécurité des opérations, etc[Rad04]. La diversité des services offerts et le nombre croissant d'applications qui leur sont inhérentes font que donner une définition globale de la qualité de service est de plus en plus difficile. En effet, celle-ci dépend du point de vue duquel elle est considérée [DS97]. La QoS peut être vue sous deux angles distincts selon que l'on s'y intéresse du point de vue de l'utilisateur (définition précédente) ou du point de vue d'une entité composant l'architecture du système supportant l'application [Tou02]. Ceci nous amène à donner une autre définition de la qualité de service qui est beaucoup plus technique que la précédente. Nous considérons que la qualité de service correspond à tous les mécanismes d'un réseau qui permettent de partager équitablement et selon les besoins requis par un service, toutes les ressources offertes pour lui permettre de répondre adéquatement à des exigences exprimées ou implicites, qui visent à satisfaire autant que possible ses usagers. Ces exigences peuvent être liées à plusieurs aspects d'un service : son accessibilité, sa continuité, sa disponibilité, sa fiabilité et sa maintenabilité, etc.

Aujourd'hui dans les réseaux IP et notamment dans l'Internet, la majorité des services offerts sont à contraintes de bout en bout. Répondre à ces contraintes et garantir les exigences en différentes ressources requises par les services nécessitent la mise au point de mécanismes de

qualité de service déployés à travers plusieurs domaines autonomes, on parle alors de qualité de service de bout en bout [Cam96]. Au départ, l'utilisation d'Internet était essentiellement fondée sur le modèle du " best-effort " qui consiste à traiter tous les paquets de données selon une politique identique et pour lequel le réseau tente de délivrer au mieux les données (d'où l'appellation " best-effort "). Or, ce modèle est rapidement devenu insuffisant pour les données multimédias (flux de type temps-réel) qui requièrent des garanties de délais et une bande passante bien explicite [CL99]. Pour pouvoir mettre en place des garanties de QoS dans les réseaux IP, l'IETF (Internet Engineering Task Force) a proposé deux modèles d'architecture basés sur des politiques de services différentes [Tou02] : IntServ et DiffServ.

Le modèle de réseau à intégration de services (IntServ) [Int] est un modèle orienté flots (ou connexions), dans lesquels chaque flot peut faire une demande de QoS spécifique. La garantie de la QoS de chaque flot est alors effectuée par un contrôle d'admission et une réservation préalable des ressources. Il s'agit d'une gestion préventive de la QoS, effectuée a priori lors de la réservation des ressources. À l'opposé, le modèle de réseau à différenciation de services (DiffServ) [Dif] agrège les flux en quelques grandes classes de trafics qui ont chacune leurs besoins spécifiques de QoS. La QoS est alors assurée au niveau des noeuds du réseau (les routeurs) par des traitements spécifiques à chaque classe de service. Il n'y a plus de signalisation par flot comme dans IntServ (ou ATM). La différenciation de service n'est plus un modèle orienté "flot", mais un modèle orienté "noeud", chaque noeud du réseau pouvant avoir sa propre politique de traitement des paquets [Mah05].

La QoS peut concerner deux situations bien différentes, selon qu'il s'agisse d'une architecture en boucle ouverte [Bao98] ou en boucle fermée [BDS95]. Dans le premier cas, il n'y a pas de retour d'information issue des autres composants de l'architecture ; c'est en général une approche de type préventif qui est adoptée au niveau de la QoS, par exemple lorsqu'on cherche à éviter a priori les congestions et les pertes ; le contexte principal où on la rencontre est celui des applications ayant de fortes contraintes d'interactivité (c'est-à-dire ayant de fortes contraintes de délai) et de certaines situations ne tolérant aucune dégradation [RC00]. En boucle fermée, l'environnement (éventuellement de bout en bout) restitue un signal qui permet d'adapter le comportement du composant aux besoins de l'application ou, réciproquement, d'asservir l'application aux contraintes posées par les composants utilisés. Cette approche est du type réactif, où l'on agit sur la source, de manière adaptative, pour la contraindre par exemple à diminuer son débit en cas de problème ; un exemple où l'on rencontre ce comportement est celui des applications de transmission de données [Sou03], qui n'ont pas, en général, à respecter des contraintes temporelles fortes. Signalons que le délai introduit par les mécanismes de rétrocontrôle (feedback) n'est pas favorable au haut débit [Tou02], [Vic04].

Un utilisateur final peut souhaiter que la qualité de service offerte par le réseau et la satisfaction des exigences requises soient garanties. La qualité de service sera alors l'objet d'un contrat, généralement appelé SLA (Service Level Agreement). La perception qualitative de la qualité de service est dès lors insuffisante. Il va y'avoir besoin de traduire la qualité de service

en paramètres ou quantités mesurables exprimant les performances du réseau. Lorsque de tels paramètres sont soigneusement définis, on leur assigne alors le terme de métriques [Cor04].

1.3 Les métriques

Le but des métriques (RFC 2330) [RFC2330] est d'établir une base commune de connaissance au niveau des performances et de la fiabilité du réseau afin d'en obtenir une connaissance précise pour les utilisateurs et pour les fournisseurs de services Internet. Ces métriques sont définies par la communauté scientifique et les organismes de standardisation au sein de l'IETF et de l'ITU. Une métrique réseau est une quantité soigneusement définie exprimant un niveau de performance directement lié au fonctionnement et la fiabilité de ce réseau (Internet par exemple). Cette quantité doit être déterminée de la manière la plus précise possible et doit être exprimée en unités de mesure universelles (exp : la seconde pour la mesure des délais, etc).

Une métrique de QoS doit être utile à la fois pour les utilisateurs et pour les fournisseurs d'accès en leur permettant de bien quantifier les performances du réseau qu'ils offrent (les fournisseurs) ou qu'ils demandent (les utilisateurs). Une métrique doit offrir la propriété de répétabilité ; en d'autres mots, la mesure de cette métrique doit être une opération répétable : la répétition des mêmes méthodes de mesure sous des conditions identiques doivent reproduire les mêmes résultats pour cette métrique. La quantité exprimant un niveau de performance peut être mesurée une seule fois, il s'agit alors d'une métrique singleton. Cependant, dans certains cas, ces métriques ne sont pas capables d'interpréter d'une manière efficace le phénomène physique qu'elles mesurent. Par exemple, la variation rapide de ce phénomène fait que la quantité mesurée par la métrique donne une information incomplète sur ce dernier. Ce phénomène physique nécessitera donc plusieurs mesures successives échantillonnées sur un intervalle de temps. L'échantillonnage (périodique, aléatoire ou statistique) des mesures de la métrique permet d'introduire des paramètres supplémentaires beaucoup plus pertinents (tels que la moyenne ou l'écart type par exemples) permettant, d'une part, d'analyser le degré de fiabilité de la mesure et d'autre part, de suivre et de représenter l'évolution du phénomène étudié.

A base de ces métriques fondamentales, d'autres métriques plus complexes appelées métriques dérivées peuvent être formées. Ces métriques sont équivalentes à la notion de paramètres introduite dans la recommandation E.800 de l'ITU [E800]. La composition des métriques complexes peut être le résultat d'une combinaison spatiale ou temporelle de plusieurs métriques de base. La combinaison spatiale consiste à définir une nouvelle métrique quantifiant une performance liée à un segment du réseau composé de plusieurs sous-segments à partir de plusieurs métriques correspondants à chaque sous-segment [RFC2330]. Par exemple, nous pouvons définir la métrique mesurant le délai unidirectionnel dans un chemin de bout en bout composé de plusieurs liens comme étant la combinaison (la somme) des métriques mesurant le délai unidirectionnel sur chaque lien. En revanche, la composition temporelle consiste à combiner des métriques mesurées sur des intervalles de temps différents pour définir une métrique quantifiant le niveau de perfor-

mance sur un intervalle de temps englobant les intervalles de temps précédents. Par exemple, il est possible de définir une métrique qui mesure le débit d'une application sur un intervalle de temps T (tel que $T = \cup T_i$) à partir des métriques mesurant le débit sur les intervalles T_i.

Les opérations sur les métriques de base permettent de définir quatre classes de métriques dérivées : les métriques additives, les métriques multiplicatives, les métriques concaves et les métriques convexes. En considérant par exemple un chemin réseau de bout en bout composé de plusieurs liens, la mesure d'une métrique additive sur ce chemin se fait par l'addition des mesures sur les différents liens. Dans l'exemple donné précédemment, le délai unidirectionnel sur un chemin de bout en bout est la somme des délais mesurés sur tous les liens. Les métriques multiplicatives se combinent, comme leur nom l'indique, par le produit des métriques des liens (comme le taux de perte par exemple). Il existe aussi des métriques convexes et concaves qui se combinent respectivement par le maximum et le minimum des métriques des liens (la capacité et la bande passante disponible sont par exemple des métriques concaves)[Cor04].

1.3.1 Les métriques IETF

La mesure de la qualité de service dans l'Internet trouve son origine dans les travaux de Vern Paxon [Pax97] qui ont conduit en 1996 à la création du groupe de travail IPPM de l'IETF. Ce groupe est chargé de définir des recommandations sur les mesures de performances pour différentes technologies. Le groupe composé de chercheurs et de fournisseurs de services a pour objectifs de définir la terminologie associée aux métriques de base, de définir des méthodologies de mesure en vue d'offrir des évaluations de performances standards pouvant être utilisées par les différents ISPs et finalement de développer un ensemble standard de métriques, commun aux utilisateurs et aux ISPs, caractérisant la qualité, la performance et la fiabilité d'un service IP. Le groupe IPPM de l'IETF a défini plusieurs métriques liées à la performance et à la fiabilité de l'Internet. Ces dernières sont des métriques déterministes contrairement aux métriques définies par l'Union Internationale des Télécommunications dans sa recommandation I.380 [I380] qui sont, quant à elles, des métriques probabilistes (les définitions en terme de probabilité peuvent cacher des suppositions quant aux modèles stochastiques des comportements mesurés). Les métriques de base définies par l'IPPM sont :

- La mesure de la connectivité entre deux équipements [RFC2678]
- Le délai unidirectionnel (One-Way Delay) [RFC2679]
- Le taux de pertes unidirectionnelles de paquets (One-Way Loss) [RFC2680]
- Le délai et le taux de pertes de paquets aller-retour (Round-Trip Delay and Loss) [RFC2681]

A partir de ces métriques, sont élaborées plusieurs autres métriques plus complexes telles que :

- La variation de délai (Gigue)
- Les modèles de pertes de paquets (Loss Patterns) [RFC3357]
- Le réordonnancement des paquets (Packet Reordering)

– La bande passante et le débit de transfert d'informations (Bulk Transfer Capacity)

1.4 La mesure des métriques de la QoS

Sur un réseau, les mesures peuvent être réalisées de façon active ou passive. Ces deux techniques diffèrent par leur principe de mise en œuvre, les problématiques qu'elles induisent et l'utilisation potentielle des résultats qu'elles fournissent.

1.4.1 Les mesures passives

Les mesures passives permettent d'étudier les caractéristiques du trafic qui circule sur le réseau. Elles sont réalisées en observant le trafic en un point donné du réseau et n'ajoutent pas de trafic sur ce dernier.
Les mesures passives sont généralement réalisées dans un équipement du réseau (routeur, commutateur, terminal, ...) ou dans des équipements spécialisés comme les carte DAG par exemple (Conçues par l'université de Waikato en Nouvelle Zélande et commercialisées par Endace) qui se chargent d'extraire les entêtes des paquets, de les estampiller suivant une horloge GPS et de les stocker sur un disque dur. Ces mesures fournissent des statistiques par paquet (niveau microscopique : taille des paquets par exemple) ou par flot (niveau macroscopique : nombre de flots par unité de temps, débit par flot, etc.). Leur avantage est qu'elles ne sont pas intrusives et ne perturbent pas l'état du réseau. Par contre, elles posent un sérieux problème qui est celui du volume des données capturées. En effet, la quantité de données capturées peut rapidement devenir colossale sur des liens à haut débit. Un autre inconvénient est lié à leur aspect local. Les mesures sont faites en plusieurs points de mesure du réseau et il est souvent difficile de recouper les résultats issus de points de mesure différents [Owe06b]. Toutefois, SNMP qui est l'un des premier protocole de gestion et de supervision des réseaux à être standardisé, permet de collecter des informations et des mesures de toutes sortes grace à des agents qui interrogent les MIB (Management Information Base) des équipements du réseau sur lesquels ils sont installés.
Les systèmes de métrologie passive peuvent utiliser deux modes d'analyse de traces, une analyse en-ligne ou hors-ligne. Dans le cadre d'une analyse en-ligne, toute l'analyse doit être effectuée dans le laps de temps correspondant au passage du paquet dans la sonde de mesure. Une telle approche, temps-réel, permet de faire des analyses sur de très longues périodes et donc d'avoir des statistiques significatives. Par contre, la complexité maximale pour ces analyses reste très limitée à cause du faible temps de calcul autorisé (d'autant plus faible que la capacité du réseau est importante). A l'inverse, une analyse hors-ligne oblige la sonde à sauvegarder une trace du trafic pour des analyses ultérieures. Une telle approche demande ainsi d'énormes ressources ce qui représente une limitation pour des traces de très longue durée. Par contre, une analyse hors-ligne permet des analyses extrêmement complètes et difficiles, capables d'étudier des propriétés non triviales du trafic. De plus, comme les traces sont sauvegardées, il est possible de faire plusieurs analyses différentes sur les traces, et de corréler les différents résultats obtenus sur la

trace, ou obtenus sur des traces différentes, pour une meilleure compréhension des mécanismes complexes du réseau [Owe01].

Les techniques passives sont particulièrement adaptées à l'ingénierie du trafic puisqu'elles montrent la distribution et le comportement des flux qui circulent sur le réseau. Elles offrent une vision réelle des performances obtenues par les flux applicatifs mais nécessitent en général de lourds investissements pour faire face à l'évolution des réseaux haut débit. C'est l'une des raisons pour lesquelles les mesures actives ont tendance à se développer.

1.4.2 Les mesures actives

Les mesures actives ont pour objectif de déterminer la QoS de bout en bout telle qu'elle est ressentie par l'application. Elles sont réalisées en générant un flot de "paquets-sondes" qui circulent sur le réseau entre une source et une destination (figure 1.1). En effet, en choisissant des configurations particulières pour les différents paramètres des flux sondes à l'envoi (taille des paquets, dispersion inter-paquets,débits, etc), il est possible de mesurer les caractéristiques du réseau en analysant les flux au niveau de la destination (temps de réception, dispersion inter-paquets à l'arrivée ...etc)

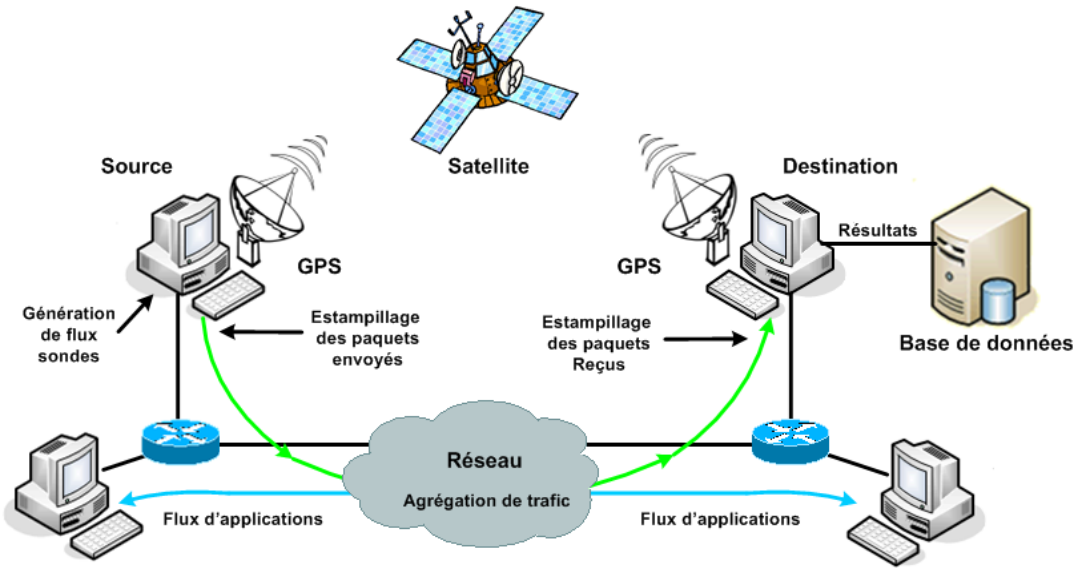


FIG. 1.1 – La mesure active.

Les mesures actives peuvent être classées en deux catégories : les mesures unidirectionnelles et les mesures aller-retour. Les outils de mesures unidirectionnelles sont composés de deux parties distinctes : l'émetteur et le récepteur installés respectivement au niveau de la source et de la destination. Par contre les outils de mesure aller-retour sont composés d'un seul module, installé au niveau de la source. Ce dernier envoie des flux sondes et analyse les réponses reçues pour en extraire les caractéristiques du réseau. Ces outils n'ont pas besoin d'avoir accès à la station

distante, cependant, ils sont généralement basés sur les paquets ICMP qui, pour des raisons de sécurité, sont de plus en plus filtrés par les nœuds du réseau (que ce soit en entrée ou en sortie). Cela rend l'utilisation de cette catégorie d'outils inefficace [Owe03c].
Les mesures actives imposent certaines contraintes temporelles assez fortes. On peut citer par exemple :

- La nécessité d'envoyer des flux sondes avec des dispersion inter-paquets ou des débits très précis
- Une grande précision à l'estampillage des paquets (à l'émission et à la réception)
- Une synchronisation précise des horloges entre la source et la destination.

Le non respect de ces contraintes induira des erreurs dans les mesures et les rendra inexploitables [Mic03].

L'outil de mesures actives le plus couramment utilisé est le Ping [RFC1574]. Celui-ci permet de tester l'accessibilité des stations sur le réseau ainsi que de mesurer certain paramètres comme le délai de transfert aller-retour (RTT) ou le taux de pertes aller-retour des paquets entre deux machines. Un autre outil, Traceroute [Jac] fourni la liste des routeurs traversés par les paquets émis entre deux machines et donne une indication sur le temps de passage à chacun de ces routeurs. Des outils tels que Iperf [Ipe] et Pathload [Dov02a], beaucoup plus intrusifs que Ping et Traceroute, sont utilisés afin d'étudier le comportement du réseau en situations extrême. Ces derniers génèrent des quantités importantes de trafic et stressent le réseau afin d'en extraire ses caractéristiques. Les meures actives fournissent aux utilisateurs et aux applications le meilleur moyen de mesurer les performances de bout en bout offertes par le réseau entre deux points donnés. L'inconvénient majeur de ces dernières est la charge induite par le trafic de mesure ("intrusivité"). En effet, le flot de mesure est susceptible de perturber l'état du réseau et ainsi modifier la grandeur que l'on cherche à mesurer. Le volume des flots de paquets-sonde utilisés doit ainsi être limité . En effet, il est important de s'assurer que le flux de test est suffisamment important pour constituer un ensemble de mesures pertinent pour évaluer les propriétés du réseau. Parallèlement, il est essentiel de vérifier que le flux de test ne perturbe pas ce dernier, ainsi si le réseau est en limite de congestion, il faut absolument éviter que l'introduction du flux de test le fasse entrer en congestion, modifiant de ce fait les propriétés à évaluer [Cor04]. Un autre inconvénient des techniques actives est lié à la vitesse de convergence des mesures. En effet, pour mesurer certains paramètres, il faut mettre en œuvre des algorithmes complexes nécessitant un temps de convergence important. Dans le cas où le paramètre mesuré est très variable alors les résultats de mesures obtenus par ces techniques sont peu fiables.

1.5 Les projets et plates-formes de mesures

Le besoin de surveiller, de mesurer et d'analyser le comportement des équipements et des applications est ancien, de nombreux outils ont été développés et plusieurs projets de mesure

ont été mis en ouvre pour y répondre. Certains de ces projets n'ont pas d'objectif très précis (alimenter des bases de données de mesures, mesurer la croissance d'Internet, etc.). Pour les autres, les travaux sont axés notamment sur la caractérisation et la modélisation du trafic, l'étude des matrices de trafic et la cartographie du réseau : la caractérisation du trafic consiste à déterminer le volume du trafic qui circule sur le réseau, sa variabilité en fonction du temps et sa nature [Lar05a]. La nature du trafic peut être définie par le protocole utilisé (TCP, UDP, etc.) ou par l'application qui le génère (web, mail, multimédia temps-réel) [Lar05c]. On prend aussi en compte la taille des paquets, le nombre de flux simultanés, la taille des flux et leur composition. La modélisation du trafic a pour objet la détermination de modèles des arrivées des flux et/ou des paquets [Owe04b] et des pertes de paquets. Les matrices de trafic sont utilisées pour étudier les routes empruntées par les paquets dans le réseau. Ces études s'intéressent par exemple à la dynamique du routage et à la symétrie des routes. Finalement, certains projets tentent de dresser des cartes de l'Internet en détectant tous les hôtes du réseau [Mic05].

Dans le domaine des mesures actives plusieurs projets ont été mis en œuvre, les plus importants parmi eux sont décrits succinctement dans le paragraphe suivant :

1.5.1 Le projet NIMI

NIMI (National Internet Measurement Infrastructure) est un projet de mesures initié par Vern Paxon [NIM], il fait partie des premières plates-formes de métrologie a avoir vu le jour (figure 1.2). Son objectif est de déployer une infrastructure de mesures actives sur le réseau fédérateur Internet afin de :

- détecter les éventuels dysfonctionnements,
- étudier le comportement de l'Internet et de suivre son évolution,
- mesurer les performances des services offerts par les fournisseurs d'accès et
- faciliter l'accès aux mesures colletées.

L'originalité de ce projet est la séparation des différentes tâches (les demandes de mesures, les mesures, l'analyse des résultats et la configuration des serveurs) qui sont effectuées par des modules distincts qui reposent sur des mesures actives effectuées à partir d'outils standards (Treno, Poip, etc) [RFC2330].

Les mesures effectuées par NIMI sont des mesure point à point, ces dernière comportent des estampilles indiquant la date et la fréquence à laquelle les mesures doivent être faites. Les requêtes sont signées cryptographiquement et un mécanisme de contrôle d'accès sur la machine contactée permet de gérer les politiques de sécurité des différents sites. Ce projet a le grand mérite de poser les bases des services de métrologie qui seront développés par la suite. Cependant, le manque de flexibilité de la plate-forme (métriques réseau point à point uniquement), l'absence de métriques dérivées et la difficulté d'accès aux informations (obligation pour les clients de se connecter aux machines participant aux mesures) restreignent l'emploi de NIMI [PAM00]. L'évolution de cette

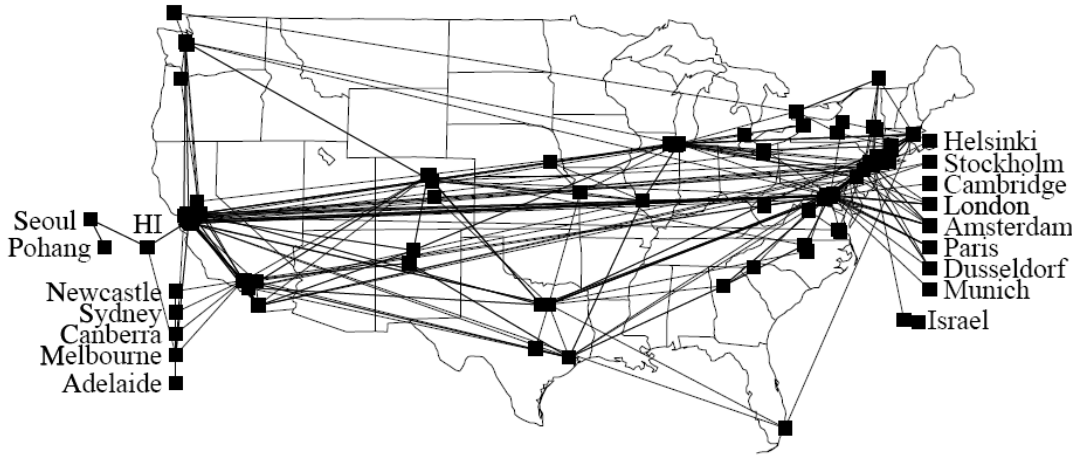


FIG. 1.2 – Déploiement de la plateforme de mesure NIMI.

plate-forme a donné naissance au projet GIMI (Global Internet Measurement Infrastructure) qui a tenté de corriger ces quelques défauts. L'infrastructure NIMI est étendue jusqu'en Europe dans le cadre de plusieurs projets de métrologie tel que le projet METROPOLIS déployé en France [Owe06, Lar05].

1.5.2 Le projet RIPE Test Traffic Measurement

RIPE NCC [RIP] est une organisation collaborative à but non lucratif fondée en 1992. Elle est ouverte aux individus et aux organisations s'intéressant aux déploiement des réseaux IP dans une zone pan-européennes comprenant l'Europe, une partie du Moyen-Orient et de l'Asie. C'est l'un des quatre centres Internet régionaux (Regional Internet Registries : RIR) qui existent dans le monde. Il assure la coordination administrative et technique des fonctions d'adressage et de routage à travers le réseau (attribution d'adresses IP, de numéros AS - Autonomous System, etc).

En 1997 RIPE NCC avait proposé le projet de mesure Test Traffic Measurement (TTM)[RIP], il avait pour objectif de mesurer les métriques unidirectionnelles (OWD, pertes de paquets, gigue, etc) caractérisant les chemins entre deux systèmes autonomes distincts en se basant sur les méthodes de mesures actives. Ce projet déploie une infrastructure de mesure (figure 1.3) entre les différents ISPs en se basant sur les "test-box" qui sont des équipements matériels constitués d'une machine FreeBSD disposant de suffisamment de ressources de stockage et de traitement et équipée d'un GPS pour la synchronisation des horloges. Les "test-box" sont connectés aux routeurs de bordures de chaque système autonome participant aux projet de mesures. Un trafic de mesures est envoyé d'un ISP à un autre en utilisant les "test-box". Ces dernières, génèrent des paquets sondes (paquets UDP de 128 octets) et les estampillent puis les envoient vers une autre "test-box". A l'arrivée, les paquet sondes sont estampillés de nouveau. Comme la destination est synchronisée par le même mécanisme GPS que la source, la différence entre les deux estampilles faites par l'émetteur et le récepteur correspond au délai de transmission (délai unidirectionnel)

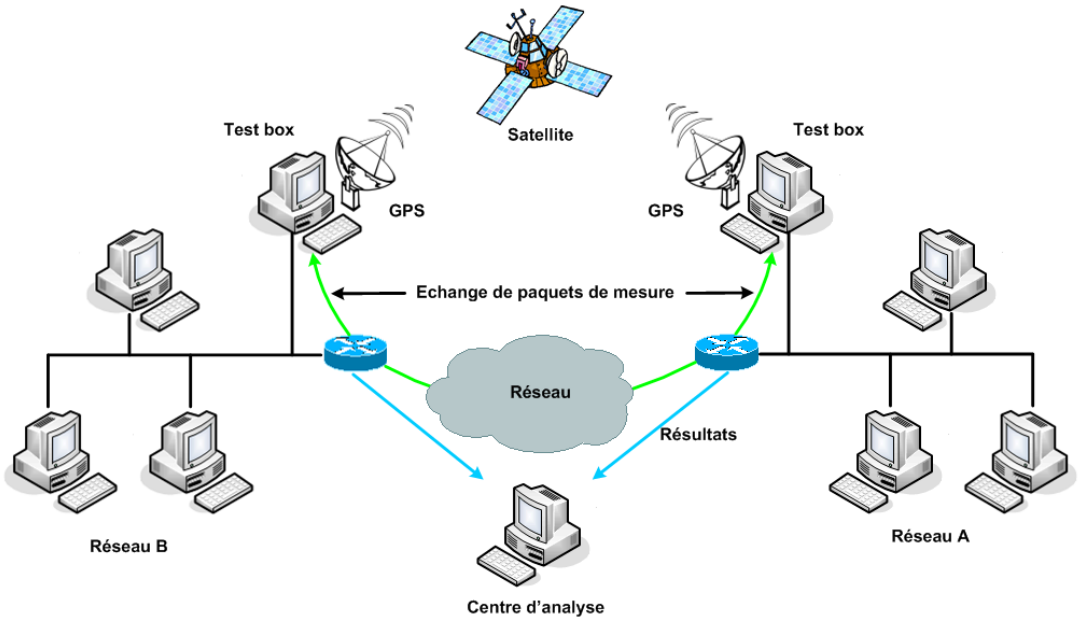


FIG. 1.3 – Architecture de la plateforme de mesure RIPE TTM.

entre les deux "test-box". Les données ainsi prélevées sont transférées vers un centre d'analyse au sein de RIPE NCC, où elles sont traitées et rendues disponibles aux utilisateurs. Vu l'aspect stratégique de ces mesures du point de vue commercial, il est relativement difficile d'avoir accès aux résultats de mesures si l'on n'est pas déjà opérateur et participant au projet.

1.5.3 Le projet MINC

Le projet MINC [Ada00, Cac99] est déployé sur une infrastructure déjà existante du projet NIMI. Il étend les fonctionnalités de ce dernier en diffusant des sondes actives en mode multicast. En exploitant la corrélation entre les performances obtenues au niveau des différents récepteurs multicasts, un client MINC peut extraire des informations concernant la structure interne du réseau et les propriétés (délai, pertes de paquets, etc) sur tous les liens d'interconnexion traversés par les sondes. L'un des inconvénients de ce projet réside dans le fait que ce dernier nécessite des infrastructures réseaux supportant le multicast. Or ce service n'est pas disponible partout. Un autre projet qui constitue une variante de MINC a été réalisé, il s'agit du projet UINC (Unicast INC). Ce dernier a reproduit le travail de MINC mais en mode unicast.

1.5.4 Le projet Netsizer

Le projet Netsizer [Net01] de Telcordia (ex Bellcore) a pour objectifs de mesurer la croissance de l'Internet, les points durs de congestion, les délais, etc. Pour cela, depuis un ensemble de stations situées chez Telcordia, un programme teste la présence sur le réseau de toutes les adresses IP existantes et met à jour suivant les résultats une carte de l'Internet. Un des gros problèmes de ce projet reste ainsi un problème de représentation [Owe06].

D'autres plateformes de mesure ont été développées et d'autres projets qui ont permis de mettre en œuvre la métrologie réseaux dans ces deux principales approches, active et passive, ont été réalisés. Cependant, dans cette section, nous n'avons présenté que les projets de métrologie active que nous avons jugé intéressants. Pour plus de projets et plus de détails concernant ce point nous orientons le lecteur vers les références suivantes : [Owe06, Lar05, Lar04b, Pat03] et [Sur].

1.6 Les projets de métrologie en France

Les premières plateformes de métrologie ont été mises en place au États-unis au milieu des années 90. Cependant, il a fallu attendre l'an 2000 pour que quelques rares projets de mesure voient le jour en France. Les plus intéressants parmi eux (si ce ne sont pas les seuls) sont les projets SATURNE [Cor04] et METROPOLIS [REs01] que nous présentons succinctement dans cette section.

1.6.1 Le projet SATURNE

Lancé en 2000, ce projet s'inscrivait dans le volet recherche et développement du programme MIREHD (Multimedia Interactif et Réseaux Haut Débit) [MIR], du projet RNRT VTHD (Réseau National de Recherche en Télécommunications) [RNR] et du réseau européen TF-NGN (Task Force- New Generation Networks) [NGN]. Il avait pour but de développer et d'exploiter des métriques permettant de mesurer de manière fine le comportement des réseaux en fonction des classes de service et d'appliquer les résultats au dimensionnement et à l'optimisation du réseau [Cor04].

Le but de cette architecture (figure 1.4) est de permettre la mesure du délai unidirectionnel de bout en bout (OWD : One-Way Delay). La mesure est faite d'une façon simple, pour chaque paquet d'un flux prédéfini envoyé d'une machine A vers une machine B, une estampille est insérée. Le délai est calculé en comparant cette estampille contenant le temps de départ avec le temps d'arrivée. Cette plateforme mesure de manière fine les temps de trajet unidirectionnel des paquets et le taux de perte entre deux points d'un réseau IPv4/IPv6. Les mesures sont faites à partir d'un temps global obtenu grâce à des équipements GPS placés dans chacun des points de mesure. L'architecture est composée de modules fonctionnels indépendants (émission, capture et gestion des données) qui interagissent pour effectuer les mesures. Des noeuds de mesure mettant en marche cette architecture ont été installés dans plusieurs établissements français (INRIA, IMAG, France Telecom R&D) et également, dans le cadre des programmes de collaboration internationale, comme le Mexique (ITAM) et la Corée (ETRI). Certains des nœuds de la plateforme Saturne surveillaient les liens du réseau de manière permanente (24h/24), et présentaient en temps réel l'information de mesure sous forme de pages Web [Cor04].

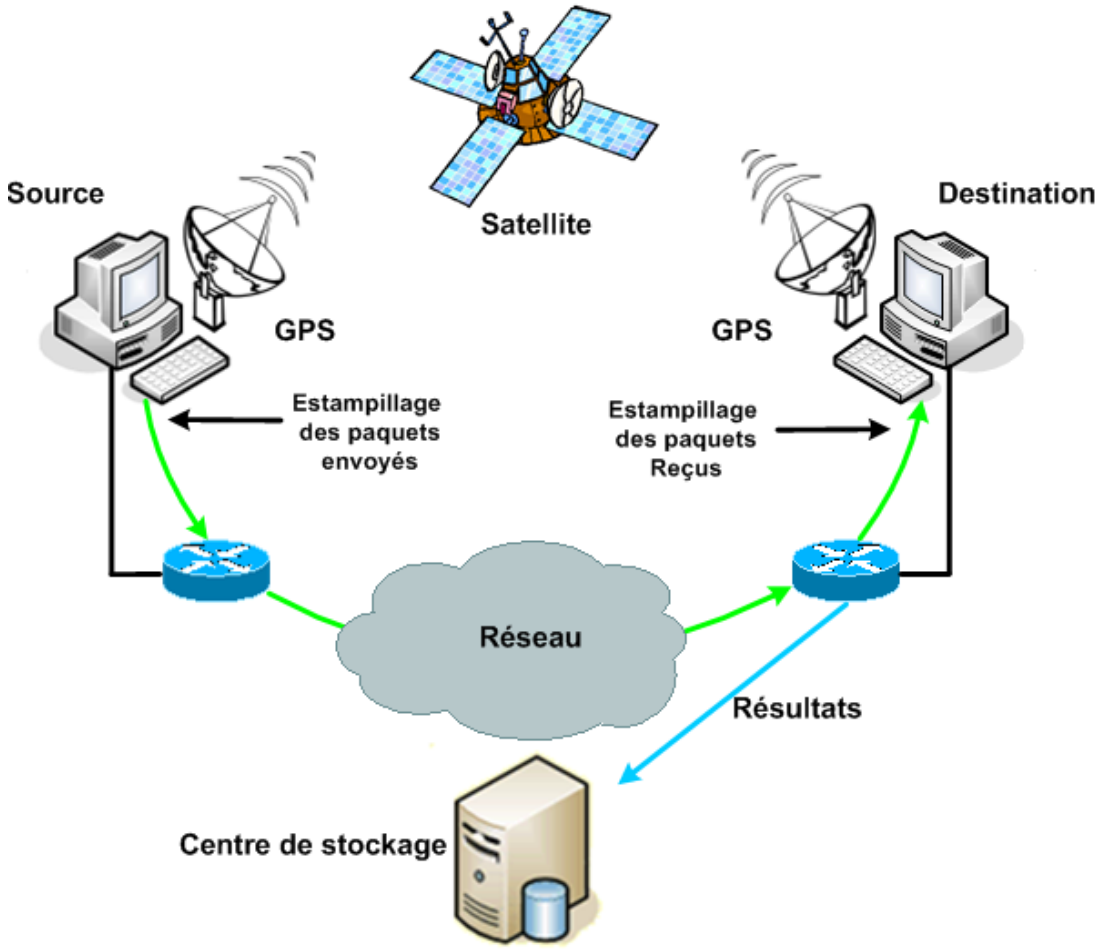


FIG. 1.4 – Architecture de la plateforme de mesure Saturne.

1.6.2 Le projet METROPOLIS

Le projet METROPOLIS [Owe03b] est un projet de métrologie initié par le RNRT (Réseau National de Recherche en Télécommunications) en octobre 2001 et qui s'est achevé en février 2005. C'est un projet fédérant plusieurs équipes de recherche françaises dans le domaine des réseaux (LAAS, LIP6, France Télécom R&D, INRIA, GET, Eurécom et RENATER). Son originalité consistait à combiner à la fois des mesures actives et passives [Lar05]. Les thèmes d'études abordés dans ce projet concernaient :

- La classification du trafic et le dimensionnement du réseau.
- L'analyse du réseau
- La modélisation du trafic et de ses propriétés
- La définition de procédures de tarification et de mise en place de SLA.

Les sondes du projet METROPOLIS ont été déployées sur des réseaux de natures différentes [Owe06]. Ainsi les réseau étudiés sont :

- Un réseau expérimental avec le réseau VTHD ;
- Un réseau public opérationnel avec le réseau Rénater ;
- Un réseau commercial : certaines plaques ADSL du FAI France Télécom.

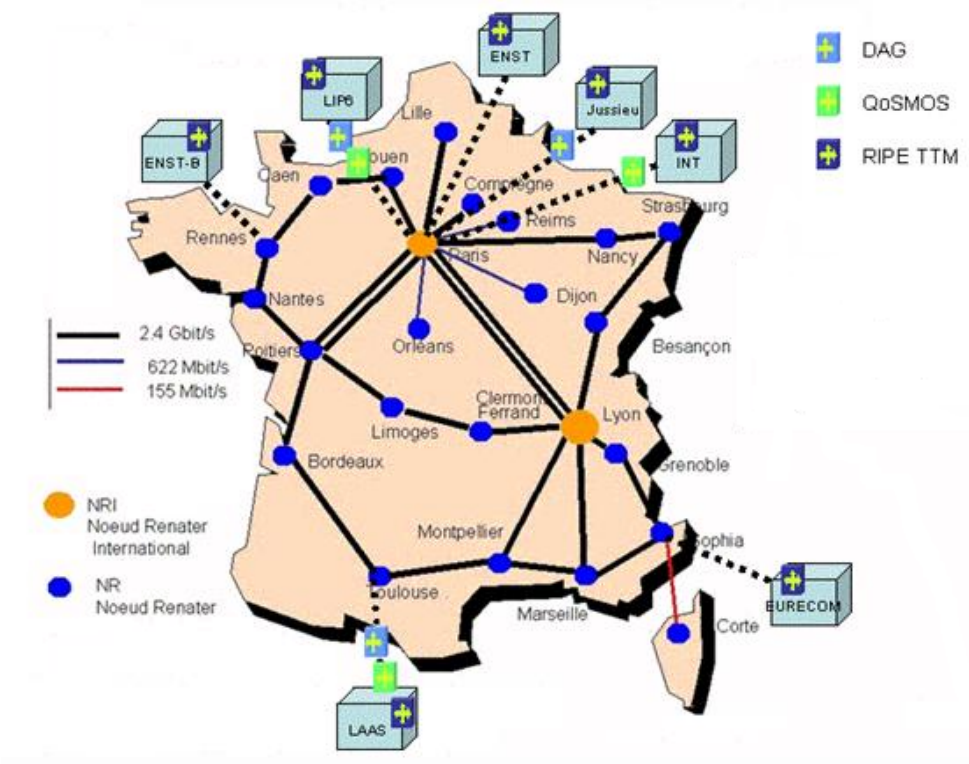


FIG. 1.5 – Architecture de la plateforme de mesure METROPOLIS.

La plate-forme METROPOLIS [REs01] est composée de 3 types de sondes (figure 1.5) :

- Des sondes DAG pour la métrologie passive microscopique déployées à la sortie du LAAS et de LIP6 ainsi qu'a la sortie du réseau de Jussieu.
- Des sondes QoSMOS pour la métrologie passive macroscopique et la classification applicative déployées au niveau du LAAS et du LIP6.
- Des sondes RIPE TTM pour la mesure active. Ces sondes hébergent également des clients NIMI et PANDORA ainsi que MetroMI, un logiciel spécialement développé dans le cadre de ce projet.

Les résultats et les contributions de ce projet sont présentés en détails dans [Owe06, Owe06a] et [Lar05].

1.7 Techniques et outils de mesure des paramètres QoS

1.7.1 Délai unidirectionnel

Le délai unidirectionnel est le temps de transit d'un paquet de la source à la destination. Il inclut les délais de propagation, les délais de transmission ainsi que les délais induits par les mises en file d'attente dans les systèmes intermédiaires.

Soit deux systèmes terminaux SRC et DST. Le délai de SRC à DST pour un paquet P à l'instant t est égal à D si SRC a envoyé le premier bit du paquet à t (heure réelle) et que DST a reçu le dernier bit du paquet à $t + D$ (heure réelle).
Le délai unidirectionnel est souvent calculé en divisant le délai aller-retour par deux. Or, les travaux de Paxson [Pax97] montrent que les routes sont de plus en plus asymétriques. Une tendance similaire concernant les liens (DSL, modem, satellite, etc.) est observée dans [Jia99]. Ce calcul est donc incorrect. Pour mesurer le délai unidirectionnel, une source envoie un paquet estampillé à la destination. A l'arrivée, la destination calcule le délai en soustrayant l'estampille du paquet à l'heure de réception lue sur sa propre horloge. Cette technique implique que les horloges des deux systèmes précédents soient synchronisées. En pratique, la synchronisation peut être réalisée par l'emploi de cartes GPS, de modules de réception d'horloge radiofréquence ou encore en se synchronisant sur des serveurs NTP (Network Time Protocol). Les cartes GPS offrent la meilleure précision mais sont très coûteuses et posent des problèmes de réception en intérieur. Il est nécessaire d'installer des antennes extérieures. Les modules de réception radio sont beaucoup moins chers mais n'offrent pas une exactitude comparable. Finalement, la synchronisation sur un réseau de serveurs NTP est la technique la plus largement utilisée. Elle est capable d'offrir une synchronisation de l'ordre de 10 ms sur l'Internet [JS99].
Dans tous les cas, le protocole NTP est utilisé : il régule la fréquence des horloges et maîtrise leur dérive [Mil91] en fonction des informations fournies par les différentes sources temporelles (serveurs NTP, GPS, etc.). Afin de ne pas briser la continuité temporelle de l'horloge, NTP ne la met pas directement à l'heure, mais la fait dériver volontairement. Cette dérive est réalisée en modifiant la valeur du pas d'incrémentation de l'horloge (*tick*). Des réserves sont à faire concernant l'utilisation de NTP sous Windows : à notre connaissance, les implémentations de NTP sous Windows sont en réalité des implémentations de SNTP (Simple NTP). Ce protocole se contente de remettre l'horloge à l'heure et risque ainsi de créer des discontinuités temporelles.

Les paramètres

Les problèmes habituellement rencontrés lors de la mesure du délai sont détaillés dans [Jia99, Pax97] et [Pax98]. La synchronisation des horloges est une problématique complexe et constitue à elle seule un vaste domaine d'étude. On définit les paramètres suivants [RFC2679] :

– *Erreur de synchronisation (ou offset :* Décalage horaire entre deux horloges, noté T_{SYNCH}. Par exemple, si l'horloge A est en retard de $5,4ms$ sur l'horloge B, alors $T_{SYNCH} = 5,4ms$.

- *Exactitude (accuracy) :* Mesure le décalage d'une horloge par rapport à l'heure UTC absolue.
- *Résolution :* Mesure la précision d'une horloge. C'est le pas avec lequel l'horloge s'incrémente (*tick*). Par exemple, les horloges des vieux systèmes d'exploitation (Linux version inférieure à 2.2.0, etc.) ont une résolution de l'ordre de 10ms. Les versions plus récentes de Linux ont une résolution de l'ordre de $20\mu s$ [LB01].
- *Dérive (skew) :* Variation d'exactitude ou de synchronisation. Par exemple, une horloge peut "gagner" 1.3ms par heure. Dans ce cas, si cette horloge est en retard de 27.1ms par rapport à l'heure UTC à l'instant t, son retard sera de 25.8ms une heure plus tard. On dit que l'horloge dérive de 1.3ms par heure par rapport à l'heure UTC. On peut aussi définir la dérive d'une horloge par rapport à une autre.
- *Variation de la dérive (drift) [Mil91] :* Variation du paramètre précédent. Bien que l'on considère le plus souvent la dérive des horloges constante, celle-ci est en réalité variable. Elle change notamment en fonction de la température.

Les termes *drift* et *skew* sont quelquefois utilisés pour faire référence à la dérive. Dans certains papiers, le terme *skew* fait référence à l'erreur de synchronisation. L'IETF définit les erreurs et incertitudes de mesures suivantes, et propose une méthode de calibration des mesures.

Erreurs et incertitudes relatives aux horloges

On définit T_{SRC} l'heure à laquelle le paquet part de la source. T_{SRC} est mesuré par rapport à l'horloge de la source. De même, T_{DST} est l'heure à laquelle le paquet arrive à la destination. T_{DST} est mesuré par rapport à l'horloge de la destination.
La synchronisation T_{SYNCH} entre les horloges de la source et de la destination entache d'erreur la mesure du délai. Si on connaît T_{SYNCH}, il est possible de corriger le délai mesuré en ajoutant T_{SYNCH} à $T_{DST} - T_{SRC}$. L'exactitude est importante uniquement dans les cas où l'on souhaite dater la mesure (identifier l'heure à laquelle la mesure a été faite). L'exactitude n'a pas d'impact sur la mesure du délai. La résolution des horloges ajoute de l'incertitude à la mesure. On note R_{SRC} la résolution de l'horloge de la source et R_{DST} celle de la destination.
Une partie de la dérive peut quelque fois être estimée par une fonction linéaire. Dans ce cas, il est possible d'utiliser cette fonction pour corriger l'horloge. Malgré cette correction, il reste une incertitude que l'on majore en utilisant une fonction $E_{SYNCH}(t)$. De cette façon, on définit une incertitude relative aux horloges égale à $E_{SYNCH}(t) + R_{SRC} + R_{DST}$.

Erreurs et incertitudes relatives au datage des événements

Le délai est défini comme le temps qui s'écoule entre le départ de la source et l'arrivée à la destination d'un paquet. En pratique, le datage de ces événements est erroné. Le paquet est estampillé juste avant son départ de la source, il existe un temps H_{SRC} entre ce moment et le

moment du départ réel du paquet (figure 1.6). De même, la date d'arrivée est déterminée après l'arrivée du paquet. Ici aussi il y'a donc un délai supplémentaire appelé H_{DST}.

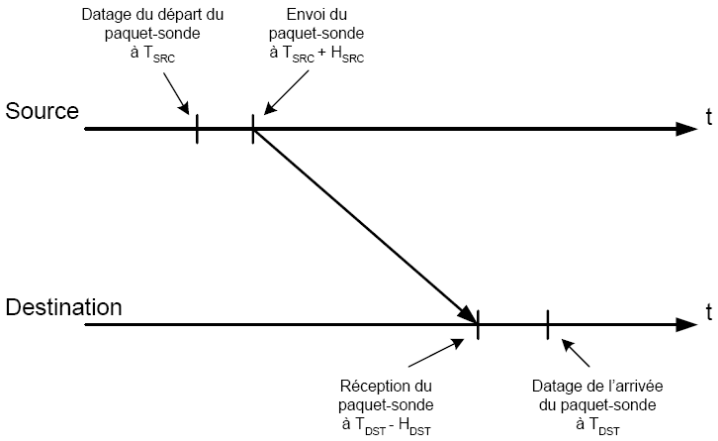


FIG. 1.6 – Erreurs relatives au datage des événements

Calibration

Le rôle de la calibration est de déterminer l'erreur de mesure [Pax98]. D'après les paragraphes précédents, on a $T_{SYNCH} = E_{SYNCH}(t) + R_{SRC} + R_{DST} + H_{SRC} + H_{DST}$. Dans certaines configurations et en utilisant un réseau isolé, il est possible d'évaluer l'erreur :

- Les erreurs relatives aux horloges sont minimisées en utilisant des GPS. La somme $E_{SYNCH}(t)+ R_{SRC} + R_{DST}$ est faible.
- Les erreurs relatives au datage des événements $H_{SRC} + H_{DST}$ peuvent être bornées en connectant les deux machines de mesure directement sur un réseau isolé.

Dans ces conditions, si les paquets-sonde sont petits, on peut faire l'hypothèse que le délai mesuré (lequel dépend du système, du pilote de la carte, etc) est minimal et peut être approximé par 0. Les valeurs mesurées correspondent à l'erreur de mesure. La valeur moyenne des mesures correspond à la part systématique de l'erreur, et les variations à la part aléatoire de l'erreur.

1.7.2 Délai aller-retour

Le délai aller-retour (round trip time ou RTT) [RFC2681] inclut le délai de la source à la destination et le délai de la destination à la source. L'outil ping mesure le délai aller-retour. Soit deux systèmes terminaux SRC et DST . Le délai aller-retour de SRC à DST pour un paquet P à l'instant t est égal à RTT si SRC a envoyé le premier bit du paquet à t (heure réelle), que DST a reçu le paquet et a aussitôt renvoyé un paquet semblable à SRC, et que SRC a reçu le dernier bit du paquet à $t + RTT$ (heure réelle).
Pour effectuer la mesure, la source envoie un paquet à la destination. Dès la réception du paquet, la destination retourne le paquet à la source. Le délai aller-retour est le temps écoulé entre l'envoi du paquet et sa réception par la source. La mesure est plus simple que celle du délai

unidirectionnel car elle ne nécessite pas de synchroniser les horloges des systèmes terminaux. On considère négligeables les erreurs dues à la dérive de l'horloge, la durée de la mesure étant faible. Par contre, il faut s'assurer que rien ne puisse nuire à la continuité de l'horloge (protocole SNTP par exemple). Les erreurs et incertitudes à prendre en compte sont :

- L'erreur relative à la résolution de l'horloge. Cette fois, le datage se fait sur le même système, on a donc $2R_{SRC}$.
- Les erreurs relatives au datage des événements (non simultanéité du datage et des événements départ et arrivée de paquet), $H_{initial} + H_{final}$.
- L'erreur relative au temps de réponse de la destination : temps écoulé entre la réception du paquet par la destination et l'envoi de la réponse correspondante, H_{refl}.

Il est possible d'évaluer l'erreur systématique et l'erreur aléatoire de mesure en utilisant une démarche similaire à celle employée dans le cas du délai unidirectionnel (réseau isolé, paquets de petite taille, ici, on a $2R_{SRC} + H_{initial} + H_{final} + H_{refl}$).

1.7.3 Variation du délai (gigue)

Une variation importante du délai d_i perturbe le transfert de médias continus comme la voix par exemple (figure 1.7). En effet, un "rythme" constant de livraison des données est indispensable à la restitution correcte du signal audio. Plusieurs métriques relatives à la variation du délai sont proposées par l'IETF [DC02] : la métrique "variation de délai" est égale à la différence des délais de deux paquets consécutifs. Bien que l'IETF déconseille l'emploi du terme gigue, une métrique "gigue" (*jitter*) est définie : elle est déterminée en calculant la valeur absolue de la variation du délai et en lui appliquant (ou non) le filtre exponentiel proposé dans [SCFJ96]. On peut aussi calculer la "variation de délai crête à crête" qui correspond à la différence des délais maximum et minimum mesurés pour une séquence de paquets. Certains auteurs définissent la gigue comme l'écart type des valeurs de délai.

Remarque : Calculer la moyenne des différences de délai d'une séquence de paquets consécutifs n'a pas d'intérêt car le résultat est toujours voisin de 0.

La mesure de la gigue ne nécessite pas la synchronisation des horloges des deux systèmes terminaux puisque les erreurs de synchronisation sont annulées lorsqu'on calcule les différences de délai. On considère négligeables les erreurs dues à la dérive des horloges, la durée de la mesure étant faible. Il faut prendre en compte :

- L'erreur relative à la résolution de l'horloge. Cette erreur est deux fois supérieure à celle qui intervient dans le cas du délai. En effet, une valeur de variation de délai est calculée à partir de deux mesures de délai.
- Les erreurs relatives au datage des événements (non simultanéité du datage et des événe-

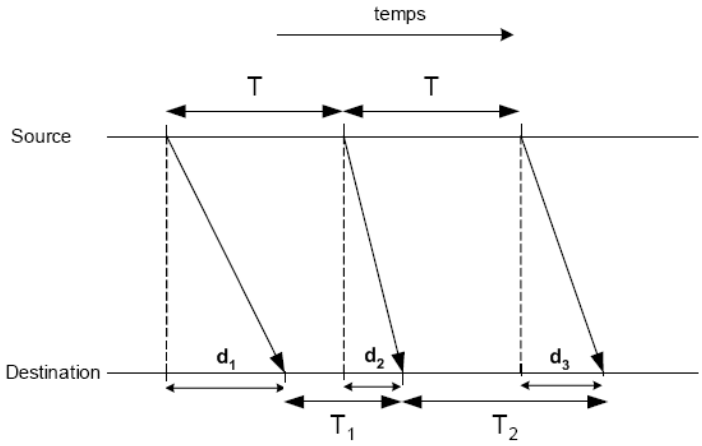


FIG. 1.7 – Variation du délai : impact sur la périodicité des données

ments départ et arrivée de paquet).

1.7.4 Pertes de paquets

La fiabilité d'une liaison est exprimée par le taux de pertes de paquets. Cette métrique est le quotient du nombre de paquets non reçus par le nombre total de paquets envoyés. La détection de paquets non reçus par le destinataire peut se faire en utilisant un temps d'expiration (*time-out*) au delà duquel un paquet est considéré perdu. La valeur du temps d'expiration est libre et dépend des besoins de l'utilisateur [RFC2680]. Il est difficile d'utiliser un temps d'expiration lorsque les horloges de la source et de la destination ne sont pas synchronisées. En pratique, on pourra détecter les pertes en se basant uniquement sur les numéros de séquence des paquets : un paquet est perdu si plusieurs paquets avec un numéro de séquence supérieur sont arrivés mais pas ce paquet.

L'utilitaire le plus connu pour mesurer le taux de pertes de paquets est *ping*. Celui-ci détermine les pertes qui se produisent lors de l'aller-retour de paquets sonde ICMP entre une source et une destination. Il s'agit d'un taux global de pertes car il n'est pas possible de différencier les pertes qui se produisent à l'aller de celles qui se produisent au retour. L'outil *sting* [Sav99] permet de mesurer les pertes de paquets selon les deux directions d'une connexion. La technique utilisée se base sur l'analyse des acquittements du protocole TCP. Dans un premier temps, la source envoie plusieurs paquets à la destination. La source passe ensuite dans une phase d'analyse des acquittements. Si le dernier paquet envoyé est acquitté, tous les paquets ont été transmis, le taux de perte est nul. Dans le cas contraire, l'acquittement reçu permet de déterminer le premier paquet perdu (figure 1.8). La source retransmet alors ce paquet et recommence l'analyse des acquittements jusqu'à ce que tous les paquets soient acquittés. De cette façon, la source détermine le nombre de paquets perdus lors de la phase d'envoi.

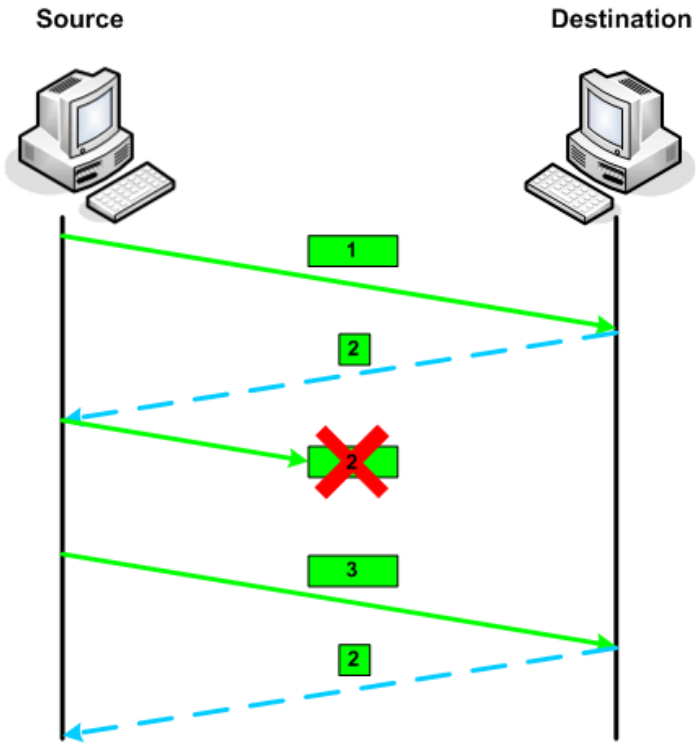


FIG. 1.8 – Sting, détermination des pertes dans les deux directions : aller et retour

Pour déterminer les pertes dans le sens retour, il faut détecter les acquittements perdus. Cela implique de connaître le nombre d'acquittements envoyés. Or celui-ci n'est pas nécessairement égal au nombre de paquets envoyés. En effet, la plupart des implémentations de TCP n'acquittent pas chaque paquet reçu : la destination ne répond pas instantanément lors de la réception d'un paquet, mais attend un certain temps (de 100 à 500ms selon les implémentations) dans l'espoir que le paquet suivant arrive. Cela permet d'acquitter plusieurs paquets à la fois et ainsi de réduire le nombre d'acquittements. Pour forcer la destination à envoyer un acquittement pour chaque paquet qu'il reçoit, *sting* met à profit l'algorithme "*fast-retransmit*" de TCP. Cet algorithme a été initialement conçu pour permettre à la destination de demander explicitement à la source de retransmettre un paquet perdu : en oubliant volontairement d'envoyer le premier paquet, on force la destination à demander sa retransmission pour chaque paquet reçu par la suite (figure 1.9). L'algorithme *fast-retransmit* de TCP intégre un mécanisme qui considère que lorsque l'émetteur TCP reçoit un accusé de réception pour un paquet en trois exemplaires, ce paquet est considéré comme perdu. L'émetteur retransmet ce paquet sans attendre la fin de la temporisation de la retransmission, ce qui permet à TCP de gagner du temps. Ce mécanisme permet à *Sting* de connaître facilement le nombre de paquets reçus par l'application réceptrice ce qui lui permet de calculer le taux de perte de paquets du chemin aller. Connaissant le nombre de paquets reçus par la destination, l'émetteur connaît le nombre d'acquittements renvoyés. En comptant le nombre d'acquittements qu'il reçoit l'émetteur en déduit le nombre d'acquittement perdu et calcule ainsi le taux de pertes de paquets du chemin retour.

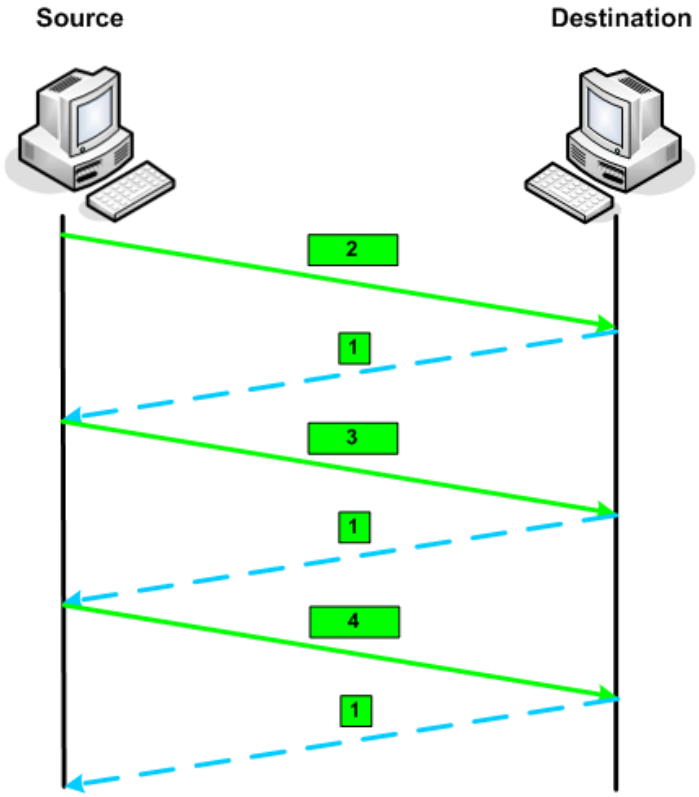


FIG. 1.9 – Mise à profit de l'algorithme "Fast-retransmit" de TCP

Il est couramment admis qu'une perte de paquet est une indication de congestion. Donc le paquet qui suit un paquet perdu a une probabilité à priori plus importante d'être perdu. Cette remarque conduit à prendre en compte la dépendance temporelle entre les pertes. Dans [BS98], il a été montré que cette dépendance est non nulle. La façon dont les pertes de paquets se produisent (en rafales, etc.) est un paramètre important pour certaines applications (telles que les applications audio et vidéo). En effet, à taux de perte équivalents, deux distributions des pertes différentes peuvent impliquer des dégradations différentes des performances de l'application.

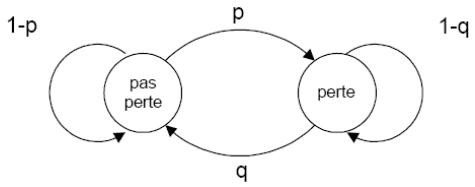


FIG. 1.10 – Modèle de Marlov à 2 états (modèle de Gilbert)

La dépendance temporelle entre les pertes est habituellement modélisée par un modèle de Markov à 2 états [JS99], appelé aussi modèle de Gilbert (figure 1.10). p est la probabilité que

le paquet suivant soit perdu sachant que le précédent est arrivé. q est la probabilité que le paquet suivant arrive sachant que le précédent a été perdu (normalement $p + q < 1$). Ce modèle peut être étendu (figure 1.11) à n états. Cette modélisation prend en considération que les n événements consécutifs de perte ont des conséquences sur les événements à venir. Elle exprime les probabilités p_{ij} qu'un paquet soit reçu ou perdu sachant que les i paquets précédents ont été perdus. De plus, certains auteurs ont proposé des critères pour caractériser la dépendance des pertes de paquets [BS98]. Ces derniers travaux proposent aussi un critère pour caractériser l'asymétrie des pertes.

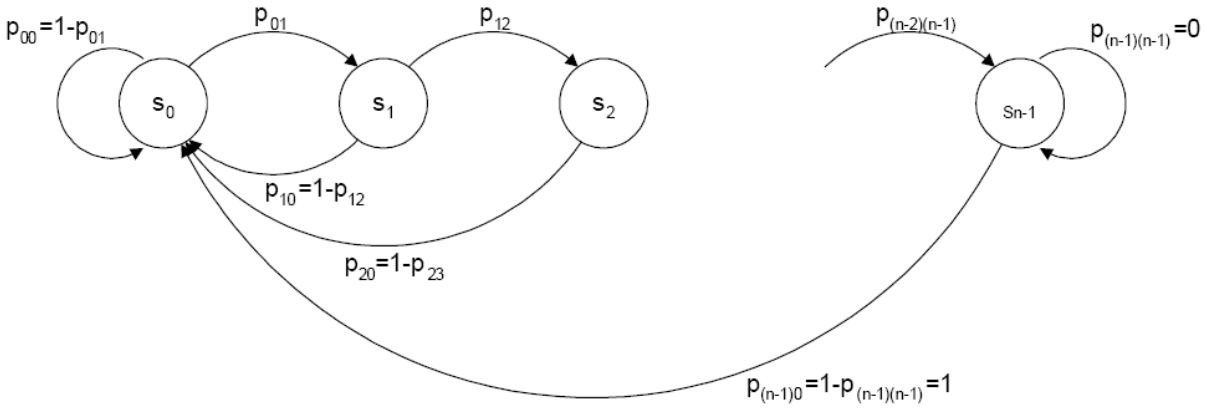


FIG. 1.11 – Modèle de Gilbert étendu pour la modélisation des pertes de paquets

Dans ce contexte, l'IPPM propose un certain nombre de métriques et de statistiques qui tiennent compte de la distribution des pertes [KR02] en introduisant les notions de période de perte et de distance entre pertes : une période de pertes est une séquence de paquets perdus successivement. La distance entre pertes indique le nombre de paquets entre un paquet perdu et le paquet perdu précédent (séparés ou non par des paquets reçus). A partir de ces deux notions, il est simple de calculer des paramètres tels que la longueur des périodes de pertes et la longueur des séquences séparant des périodes de pertes successives (figure 1.12).

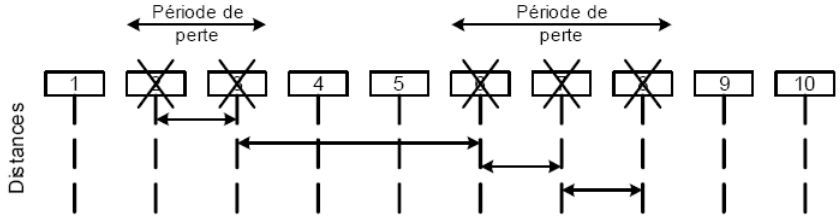


FIG. 1.12 – Périodes et distances de perte

Dans cette section, nous n'avons traité que les pertes de paquets dues à la congestion. Il existe un autre type de pertes causées par les paquets erronés qui sont écartés au niveau de

la couche liaison. TCP ne fait pas de différence entre ces deux types de pertes, il les considère comme étant des pertes dues à la congestion du chemin de bout en bout. Ceci pourrait déclencher les processus *slow start* et *congestion avoidance*, ce qui réduirait considérablement la vitesse de transmission de TCP alors qu'il n'y a pas réellement de congestion sur le chemin [RFC3155].

1.8 OWAMP (One-Way Active Measurement Protocol)

L'IPPM a proposé des métriques standardisées pour mesurer les pertes de paquets et les délais unidirectionnels. Ces dernières années, plusieurs projets de mesure exploitaient ces métriques dans le but d'analyser les performances du réseau. Malheureusement , il n'existait aucune règle ou standard permettant l'interopérabilité entre ces différents projets en terme d'échange de flux de mesure ou de collecte de résultats. Pour résoudre ce problème, l'IPPM a proposé un protocole de mesures unidirectionnelles OWAMP (One-Way Active Measurement Protocol) dans le but de déployer un maillage mondial de points de mesure offrant une interopérabilité maximale et capable de remplacer l'utilisation généralisée des mesures aller-retour basées sur le protocole ICMP.

L'architecture du protocole OWAMP, représentée par la figure 1.13, est basé sur deux protocoles interdépendants : *OWAMP-Control* et *OWAMP-Test*.

Le protocole *OWAMP-Control* s'exécute sur TCP et a pour rôle de négocier, initier et contrôler les sessions de mesures puis récupérer les résultats. Au début de chaque session de mesure ce protocole lance les négociations entre l'émetteur et le récepteur concernant leurs adresses, les numéros de ports UDP d'émission et de réception, l'instant de début et la durée de chaque session, la taille des paquets de test et la répartition poissonienne des dispersions inter-paquets ainsi que de nombreux autres paramètres définis dans la RFC 2330 [RFC2330]. Le protocole *OWAMP-Test* s'exécute sur le protocole UDP. Il gère le transfert des paquets de mesure entre l'émetteur et le récepteur. Les paquets envoyés sont estampillés, ils indiquent au récepteur si l'émetteur utilise ou pas un système de synchronisation des horloges (GPS ou NTP).

Le protocole OWAMP supporte un mode crypté qui rend les paquets illisibles et permet de détecter les erreurs de transmission grâce aux estampilles sur les paquets. La sécurité est renforcée grâce à l'ajout d'un mode d'authentification pour les messages de contrôle et de test. Ceci permet d'éviter les accès non autorisés aux mesures et évite que des pirates génèrent de faux paquets test ou modifient les estampilles des vrais paquets, ce qui fausserait les mesures. OWAMP intègre aussi un mode de différenciation de service au niveau des paquets test et permet l'utilisation de protocoles propriétaires à condition que ces derniers ne compromettent pas l'interopérabilité.

L'architecture d'OWAMP intègre les éléments suivants :

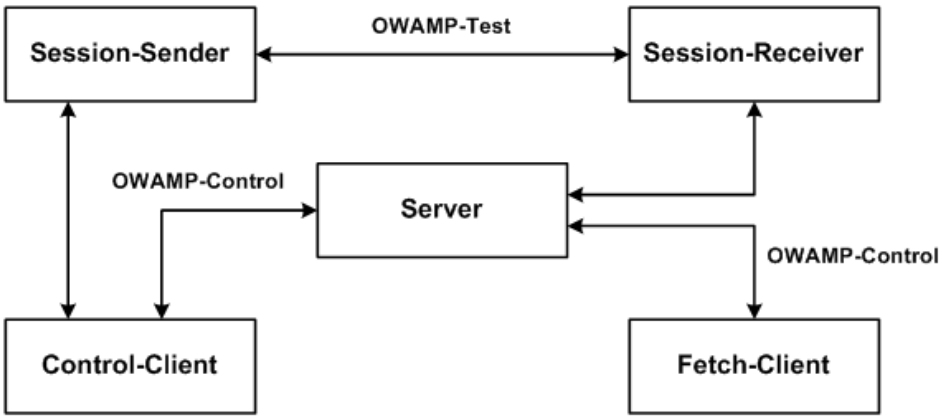


FIG. 1.13 – Architecture du protocole OWAMP.

- *Session-Sender :* émet les paquets test pour une session de mesure
- *Session-Receiver :* reçoit les paquets test
- *Server :* gère plusieurs sessions de tests et récupère les résultats
- *Control-Client :* initie et demande une session de mesure
- *Fetch-Client :* demande le résultats de session de mesure

Ces différents éléments peuvent être implémentés sur des machines distinctes, cependant il est possible de regrouper plusieurs modules sur une même machine comme le montre l'architecture présentée dans la figure 1.14.
Le Contrôleur (*Control-Client*) initie une session de mesure en ouvrant un connexion TCP avec le serveur cible sur un port bien défini (attribué par l'IANA). Le Contrôleur négocie ensuite les paramètres de la session de mesure avec le serveur. Cette connexion reste ouverte durant toute la session de mesure. Les messages de contrôle interviennent seulement avant et après la session de test.
Jusqu'à aujourd'hui, seulement deux implémentations du protocole OWAMP ont été réalisées : *Internet2-OWAMP* et *J-OWAMP*.

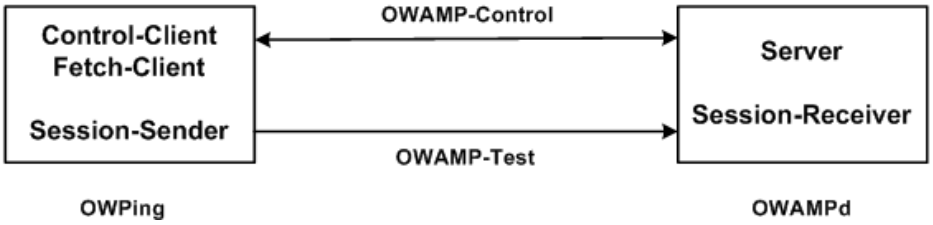


FIG. 1.14 – Architecture de Internet2-OWAMP (One-Way Ping).

Internet2-OWAMP est l'implémentation en langage C du *Ping* selon le protocole OWAMP (One-Way Ping). Cette implémentation correspond à l'architecture de la figure 1.14, elle com-

porte deux partie distinctes *OWPing* et *OWAMPd* englobant les différent modules spécifiés par le protocole OWAMP. *J-OWAMP* est une implémentation de ce protocole en JAVA. Cette implémentation considère ce protocole selon deux niveaux : *Messages* et *Entités*. Ces deux implémentations ont été testées et comparées dans [Vei06].

1.9 La bande passante

L'utilisation du terme bande passante est un abus de langage issu du théorème de Shannon car en théorie, la bande passante désigne la différence en Hertz entre la plus haute et la plus basse des fréquences utilisables sur un support de transmission. Dans la pratique, ce terme désigne le débit d'une ligne de transmission, calculé en quantité de données susceptibles de transiter dans un laps de temps donné (en général exprimé en secondes). Si l'on compare une voie de transmission (par exemple un câble ou satellite) à un tuyau, la bande passante correspond au diamètre de ce tuyau. Plus la bande passante est large, plus le volume d'informations qui transitent est important [Mic03].

Le terme "bande passante" regroupe quatre paramètres différents, à savoir *la bande passante totale d'un lien* (capacité d'un lien), *la bande passante minimale d'un chemin* (capacité d'un chemin), *la bande passante disponible d'un lien* et *la bande passante disponible d'un chemin*. Soit P un chemin entre deux systèmes terminaux, P est constitué de N liens LI_1, LI_2, ,LI_N. La bande passante totale du lien i, ou capacité du lien, notée C_i, définit la capacité totale de transmission du lien i. La capacité C du chemin P est définie par

$$C = \min_{i=1,...,N} C_i$$

La bande passante disponible du lien i est donnée par :

$$A_i = C_i(1 - u_i)$$

où u_i est le taux d'utilisation du lien i avec $0 \leq u_i \leq 1$.

La bande passante disponible A du chemin P durant un intervalle de temps T est le minimum des bandes passantes disponibles de tous les liens LI_i du chemin P :

$$A = \min_{i=1...N} \{C_i(1 - u_i)\} = \min_{i=1...N} A_i$$

1.9.1 La capacité

On distingue deux différents cas, selon qu'il s'agisse d'un lien ou d'un chemin de bout en bout. La capacité d'un lien, appelée aussi bande passante totale, est le débit maximal de transfert de paquets sur ce lien. En revanche, la capacité d'un chemin appelée aussi bande passante

minimale, est la plus petite capacité des liens composant ce chemin de bout en bout, le lien concerné est appelé *lien étroit* (*Narrow link* : lien avec la plus petite capacité, à signaler que ce lien est différent du *lien serré* ou *tight link* en anglais et qui représente le lien avec la plus petite bande passante disponible, cf. figure 1.15). Cette capacité correspond au débit maximal que peut avoir un flux sur un chemin de bout en bout en l'absence de trafic concurrent.

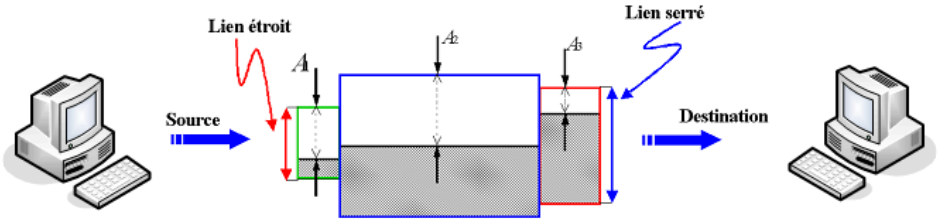


FIG. 1.15 – Chemin de bout en bout constitué de 3 liens.

Dans les deux cas considérés (lien et chemin de bout en bout), la capacité dépend du niveau protocolaire auquel elle est exprimée. En effet, au niveau liaison, la capacité C_i^{N2} correspond au taux de transmission du lien concerné (10 Mb/s dans le cas d'un lien Ethernet 10BaseT par exemple). Au niveau IP, la valeur correspondante C_i^{N3} est différente en raison des entêtes ajoutées aux paquets IP lors de l'encapsulation [Dov03c]. Le temps de transmission T_3 d'un datagramme IP de taille L_3 octets est donné par :

$$T_3 = \frac{8(L_3 + H_2)}{C_i^{N2}}$$

avec H_2 la taille totale des entêtes (en octets). La valeur de la capacité au niveau 3 est ainsi :

$$C_i^{N3} = \frac{8L_3}{T_3} = \frac{8L_3}{\frac{8(L_3+H_2)}{C_i^{N2}}} = C_i^{N2}\frac{1}{1+\frac{H_2}{L_2}}$$

Cette équation met en évidence le fait que la bande passante au niveau 3 dépend de la taille des paquets IP et de la taille des entêtes ajoutés par la couche 2. Comme nous pouvons le constater sur la figure 1.16, la valeur maximale de la bande passante au niveau IP est ainsi obtenue pour une taille de paquet IP égale au MTU (Maximum Transmission Unit)[Dov02b, Dov03c].

Pour mesurer la capacité, on distingue 3 différentes techniques, à savoir :

- La technique VPS (Variable packet size)
- La technique à dispersion de paquets (paire et train de paquets)
- La technique de talonnage (Tailgating).

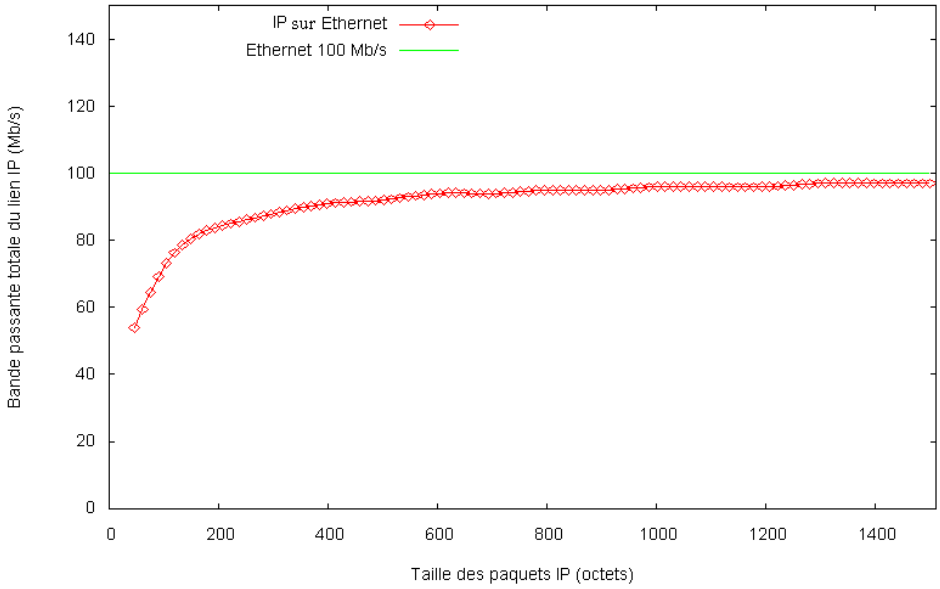


FIG. 1.16 – La capacité d'un lien IP en fonction de la taille des paquets (lien Ethernet 100BaseT).

La technique VPS

Pathchar [Jac97], Bing [Bey95], clink [Dow99] et pchar [Pch] mesurent la capacité d'un lien en utilisant la technique *VPS* appelée aussi "*one-packet technique*"[Dov03a]. Celle-ci se base sur l'hypothèse que le délai d'un paquet varie linéairement en fonction de sa taille. Elle consiste alors à mesurer le délai aller-retour pour atteindre chaque routeur présent entre une source et une destination. Ce dernier est mesuré par la source en envoyant des paquets sondes à chaque routeur. Le routeur à atteindre est déterminé en fixant la valeur du TTL des paquets envoyés.

Dans IPv4, le champ TTL est prévu pour éviter qu'un paquet ne tourne indéfiniment dans le réseau. Il correspond à la durée de vie du paquet exprimée en nombre de routeurs traversés. Il est décrémenté d'une unité lors du passage du paquet dans un routeur. Lorsque la valeur obtenue est égale à zéro, le routeur supprime le paquet et en informe son émetteur en lui envoyant un paquet ICMP "*time exceeded*" (figure 1.17).

La figure 1.17 illustre la relation entre le délai aller-retour et la capacité entre la source et le premier routeur du chemin. Le système source envoie un paquet sonde UDP avec un TTL égal à 1. Arrivé au premier routeur, ce paquet est supprimé et le routeur retourne au système source un message ICMP "*time-exceeded*".

Lorsque le routeur ciblé reçoit un paquet-sonde, le TTL passe à zéro, et en conséquence le routeur retourne un message à la source. Celle-ci détermine alors la valeur de délai aller-retour T_i. Ce mécanisme est répété en faisant varier la taille des paquets sondes. Il est alors possible de

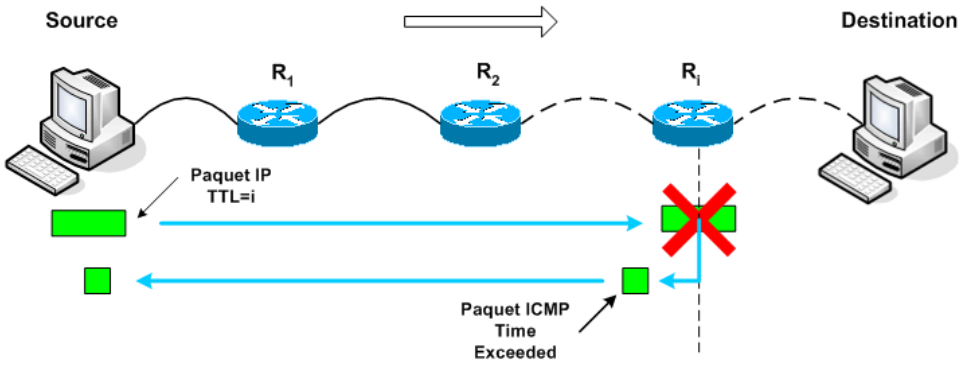


FIG. 1.17 – Le protocole ICMP implémenté dans la technique VPS.

calculer la capacité par régression linéaire à partir des tailles de paquets et des délais aller-retour T_i correspondants.

Cette technique fait les hypothèses suivantes :
- Le temps t qui s'écoule entre la réception du paquet sonde par le routeur et le renvoi du message ICMP est négligeable.
- Le temps de transmission du message ICMP est négligeable en raison de sa petite taille.

Les outils précédents (dits "*pathchar-like*") se distinguent essentiellement par leurs algorithmes de filtrage des résultats. A noter que, contrairement aux autres, l'outil *Bing* considère que le chemin et les liens empruntés entre la source et la destination sont symétriques. Des études et des explications sur le fonctionnement de pathchar et des propositions d'amélioration des algorithmes de mesure et de filtrage des résultats sont disponibles dans [Dow99, Jia99] et [PV02]. [PV02] propose notamment d'utiliser la variation du délai plutôt que le délai pour effectuer les calculs. Cela a pour effet de diminuer le nombre de sondes à envoyer sur le réseau.

La technique à dispersion de paquets

La capacité d'un chemin de bout en bout peut être mesurée par la technique à dispersion de paquets (paires ou train de paquets). Cette technique repose sur le phénomène "*d'écartement*" ou "*dispersion*" subi par deux paquets consécutifs suite à leur passage dans le goulet d'étranglement : l'espace temporel qui les sépare est accru (phénomène mis en évidence par Jacobson [Jac88]). Cette dispersion (Δ_{out}) correspond au temps de traitement du premier paquet au niveau du goulet d'étranglement. En supposant que cette dernière reste inchangée jusqu'à l'arrivée des paquets sondes à la destination, la capacité de bout en bout correspond alors au quotient de la taille S du premier paquet par la dispersion mesurée :

$$C = \frac{S}{\Delta_{out}}$$

Les outils *Bprobe* [CC96], *pathrate* [Dov, DRM01], *sprobe* [Sar01] et *Nettimer* [LB01] utilisent cette technique. *Bprobe* effectue des mesures bidirectionnelles (figure 1.18) sous l'hypothèse que la dispersion finale reste la même si les paquets reviennent à la source. Dans ce cas, la mesure est possible en faisant faire un aller-retour aux paquets sondes (Paquets ICMP echo). La capacité est alors mesurée par la source (cela évite de mettre en place une application côté source et une côté destination). Cette hypothèse est très critiquable car elle suppose que les paquets sondes empruntent le même chemin à l'aller et au retour et que les caractéristiques du réseau soient identiques dans les deux sens (hypothèse inexacte si des liaisons sont asymétriques, ADSL par exemple). De même, les hypothèses et les choix faits dans *sprobe* sont très discutables et ne tiennent pas compte des observations formulées dans [DRM01] relatives à la taille des paquets sondes par exemple. Seul *pathrate* tient compte du trafic concurrent susceptible de s'intercaler entre les paquets et qui bruitent les mesures. De plus, la gamme de mesure est limitée par la dispersion minimale que le système terminal est capable de mesurer. Les expériences menées dans [DRM01] montrent que la dispersion inter-paquet minimum que les stations de travail (PIII, 1Ghz) sont capables de mesurer est de l'ordre de 30 à 40 μs, soit une gamme de mesure limitée à 160 Mb/s.

FIG. 1.18 – Mesure de la capacité de bout en bout par Bprobe.

La technique de talonnage

Une technique de mesure alternative de la capacité de bout en bout est introduite dans *Nettimer* [LB00]. Il s'agit de la technique dite de "*talonnage*" (tailgating). Elle étend la technique du one-packet et celle de la paire de paquets. Dans ce cas, on envoie plusieurs paquets successivement avec un temps inter paquets le plus court possible. Il existe alors des relations entre les temps passés par chaque paquet en file d'attente. Si on fait l'hypothèse qu'il est possible de forcer un paquet à être mis en file d'attente derrière son prédécesseur dans un routeur particulier et pas dans les routeurs suivants, il vient alors une expression relativement simple de la capacité de chaque lien du chemin.

En pratique, cette technique se décompose en deux phases :

– La première phase est similaire à la technique VPS où un ensemble de paquets sondes est envoyé de la source vers la destination avec différentes tailles. Cette phase a pour but de

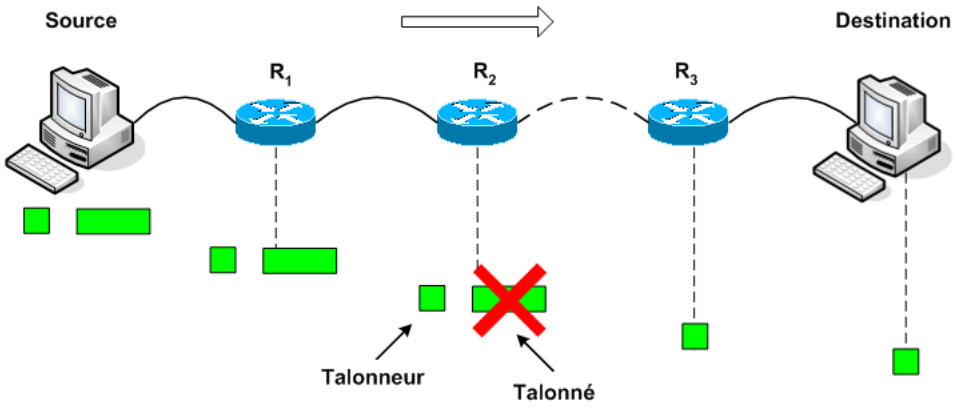


FIG. 1.19 – Technique du "talonnage".

découvrir le goulet d'étranglement (bottleneck) sur le chemin de bout en bout.

– Une fois que le bottleneck est localisé, la source envoie deux paquets sondes avec un temps inter-paquet le plus court possible. Le premier paquet, dit le "*talonné*" (tailgated), a une taille maximale de façon à ce que le deuxième, le "*talonneur*" (tailgater), soit placé en file d'attente sur un routeur bottleneck. Afin que le talonné ne provoque pas la mise en file d'attente du talonneur sur les routeurs suivants, son TTL aura préalablement été fixé de façon à ce qu'il soit détruit sur le routeur concerné (figure 1.19). Le talonneur est un message TCP FIN. De cette façon, la destination est contrainte à retourner un message TCP RESET. Cela permet de mesurer le délai aller-retour. Plusieurs variantes et améliorations de cette technique sont proposées dans [PV02] ("*packet quartets*"). Elles évitent le recours à certains filtrages des mesures, elles suppriment des phases de mesure et optimisent ainsi le nombre de paquets sondes à envoyer sur le réseau.

1.9.2 La bande passante disponible

La bande passante disponible d'un lien peut être définie comme étant la bande passante inutilisée de ce lien pendant un certain temps T. A un instant t, le lien est en train de transmettre les données avec toute sa capacité ou bien il est vide. Donc l'utilisation instantanée u_i de ce lien est soit 0, soit 1. La bande passante disponible $A_i(t,T)$ du lien i à l'instant t est donnée par :

$$A_i(t,T) = \frac{1}{T}\int_t^{t+T} C_i(1-u_i(t))\,dt$$

avec C_i est la capacité du lien i, $u_i(t)$ le taux d'utilisation de ce lien et T la période de mesure.

Les techniques de mesure de la bande passante disponible se scindent en deux grandes catégories : les techniques fondées sur le Probe Gap Model (PGM) et les techniques fondées sur le Probe Rate Model (PRM).

Le Probe Rate Model (PRM)

Le modèle PRM est basé sur l'auto-congestion du chemin de bout en bout, ce qui consiste à injecter du trafic dans le réseau au point de saturer le lien bottlenck afin d'en extraire ses caractéristiques. En émettant les hypothèses que les routeurs utilisent une discipline de service FIFO et que le trafic concurrent change lentement et suit un modèle fluide, on peut représenter le réseau par une file d'attente avec un taux de service égal à la bande passante disponible A (figure 1.20).

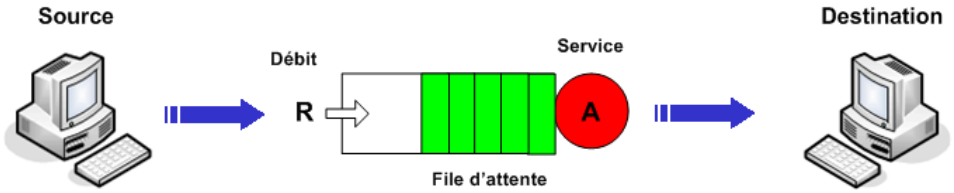


FIG. 1.20 – Illustration du modèle PRM.

Si la source envoie vers la destination des flux de paquets sondes avec un débit R inférieur à A, alors ces derniers présenteront des délais stables. En revanche, si le débit R est supérieur à A alors les flux de paquets sondes présenteront des délais croissants. Le modèle PRM consiste donc à envoyer des flux de paquets sondes vers une destination à des débits différents. Cette dernière détecte le débit à partir duquel les délais commencent à augmenter. Ce débit est égal à la bande passante disponible [Din03].

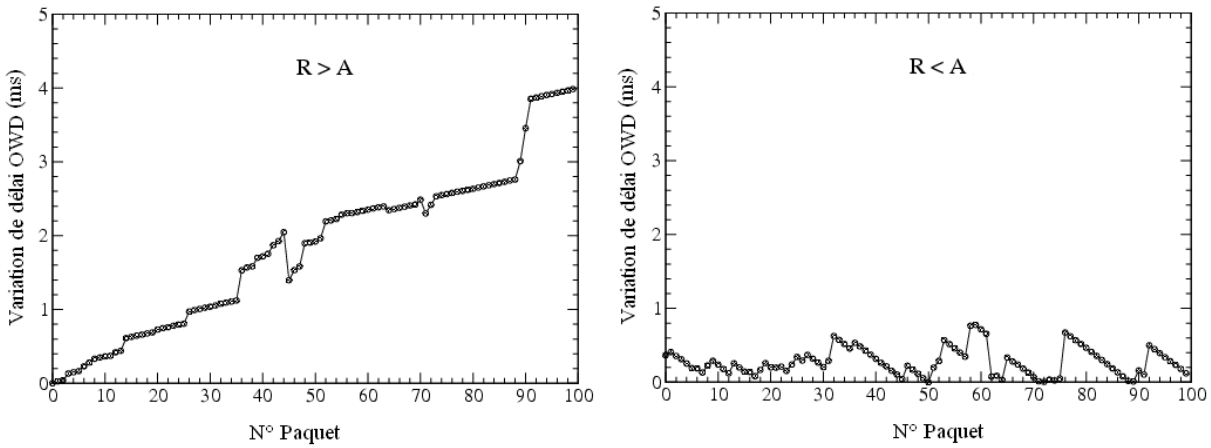


FIG. 1.21 – Tendances des variations de délais dans le modèle PRM.

La méthode d'auto-congestion la plus utilisée est la méthode SLoPS (Self-Loading Periodic Stream) qui consiste à envoyer une rafale de paquets sondes d'une source vers une destination [RFC3432]. Le destinataire mesure le délai de chaque paquet des rafales et analyse sa variation. Si le délai est jugé constant, on en déduit que le débit de la rafale est inférieur à la bande passante disponible. Si le délai est croissant, le débit de la rafale est supérieur à la bande passante disponible. On envoie alors une seconde rafale à un débit supérieur dans le premier cas ou infé-

rieur dans le second cas. Ce mécanisme est répété et on approche par dichotomie la valeur de la bande passante disponible [Dov03a, Dov03b].

Pathload [Dov02a, Dov03a] implémente cette technique telle qu'elle est décrite ici. En revanche, *Pathchirp* [Rib03] implémente une variante de celle-ci, ce dernier propose d'envoyer des sondes selon un processus exponentiel. L'avantage principal de cette approche est de minimiser le volume du trafic de mesure. En effet, un seul train Poissonien de paquets permet à *Pathchirp* de sonder le réseau à des débits différents. L'outil TOPP (*Train Of Paquet Pairs*) [MB00] utilise la même approche que *Pathload* et *Pathchirp*. cependant, comme son nom l'indique, ce dernier utilise un train de paires de paquets au lieu d'un flux simple. Chaque paire est composée de deux paquets de mêmes tailles L séparée initialement d'une dispersion Δ_{in}. Le débit d'envoi du train est ainsi égal à $R_{in} = L/\Delta_{in}$. si le débit R_{in} est supérieur à la bande passante disponible A alors le débit R_{out} mesuré au niveau de la destination sera inférieur à R_{in} (le débit de sortie du lien bottleneck est inférieur au débit d'entrée à cause de la congestion causée par les paires de paquets elles mêmes). L'idée consiste donc à envoyer un train de paires de paquets avec un débit minimal $R_{in} = R_{min}$ et d'incrémenter ce dernier à chaque nouvel envoi jusqu'à obtenir $R_{in} = R_{out}$. C'est à ce moment que le débit R_{in} correspond à la bande passante disponible de bout en bout. Donc au lieu d'utiliser la recherche dichotomique comme le fait *Pathload*, l'algorithme *TOPP* utilise plutôt une recherche par "incrémentation". Cet outil est implémenté sur NS2 seulement et aucun test sur des réseaux réels n'a été effectué, donc les performances réelles de ce dernier ne sont pas encore connues.

Ces dernières années, une multitude d'autres outils basés sur le *Probe Rate Model* ont été développés, on peut en citer par exemple : PTR [Hu03], PATHMON [Kiw04], FEAT [Qia06], Wget [Dov06] ....

Le Probe Gap Model (PGM)

Les techniques et outils de cette catégorie supposent que la capacité C du chemin est connue, et font l'hypothèse que le lien serré et le lien étroit sont confondus en un lien correspondant au goulet d'étranglement (lien bottleneck). La bande passante disponible est alors estimée en deux étapes :

- Estimation du débit du trafic concurrent C_T au goulet d'étranglement.
- Calcul de la bande passante disponible : $A = C - C_T$.

Le Probe Gap Model consiste à exploiter la variation de l'écart temporel (ou dispersion) entre deux paquets consécutifs et le débit du trafic concurrent au goulet d'étranglement du chemin [Hu03, Din03].

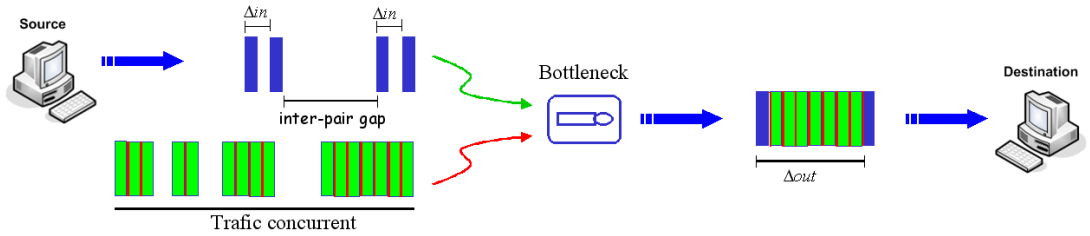


FIG. 1.22 – PGM : la relation entre les dispersions finales et le trafic concurrent.

Dans [Hu03], Hu et al ont montré qu'il existe une relation linéaire entre les dispersions de deux paquets consécutifs à la sortie du goulet d'étranglement (Δ_{out}) et le débit du trafic concurrent. En effet, la dispersion de sortie Δ_{out} est égale à la somme du temps de traitement du premier paquet, c'est-à-dire la taille du paquet divisée par la capacité du goulet L/C, et du temps de traitement du trafic concurrent qui s'introduit entre les deux paquets de la paire, c'est-à-dire la quantité de trafic concurrent ($C_T\Delta_{in}$) arrivé au goulet entre les deux paquets divisé par la capacité du goulet C. Cette relation est valable si et seulement si le premier paquet de la paire ne quitte pas la file du routeur bottleneck avant l'arrivée du deuxième paquet. Cette condition est appelée "*condition JQR*".

Spruce[Din03] et IGI [Hu03, Hu06] implémentent cette technique. *Spruce* est constitué d'un émetteur et d'un récepteur. L'émetteur envoie au récepteur 100 paires de paquets de 1500 octets, de dispersion initiale Δ_{in}. Pour s'assurer de respecter la *condition JQR*, *Spruce* ajuste Δ_{in} au temps de transmission d'un paquet de 1500 octets au goulet d'étranglement. Le récepteur mesure la dispersion finale entre les paquets de chaque paire, puis calcule la bande passante disponible pour chaque paire. *IGI* est aussi constitué d'un émetteur et d'un récepteur. L'émetteur envoie au récepteur une séquence périodique de 60 paquets de 500 octets, de dispersion initiale Δ_{in}. *IGI* débute son exécution par la mesure de la capacité C du chemin (cette mesure est effectuée en utilisant une méthode similaire à celle introduite dans bprobe [CC96]). Les expérimentations menées dans [Hu03] ont montré que, pour respecter la *condition JQR* et ne pas saturer le goulet d'étranglement, la valeur optimale de Δ_{in} est obtenue quand la moyenne des Δ_{out} est égale à Δ_{in}. Ainsi *IGI* comprend une phase de recherche du meilleur écart initial : l'émetteur commence par envoyer des paquets avec un faible Δ_{in} et l'augmente jusqu'à ce que la moyenne des Δ_{out} soit égale à Δ_{in}. *IGI* utilise une séquence périodique de paquets pour réaliser des mesures de dispersion. Dans ce cas, les paires de paquets qui forment la séquence ne sont pas indépendantes et les mesures de Δ_{out} sont corrélées. Pour limiter cet effet, *IGI* sélectionne uniquement les échantillons de $\Delta_{out} > \Delta_{in}$ pour effectuer ses calculs [Hu06].

Une variante de la technique à dispersions de paquets (PGM) est utilisée dans l'outil *Cprobe* [CC96]. En effet, ce dernier envoie un train de N paquets sondes (de taille L) vers la destination. Cette dernière mesure le temps $\Delta(N)$ écoulé entre la réception du premier et du dernier paquet.

La bande passante disponible est calculée comme étant le quotient de la quantité binaire q de données envoyée par la dispersion mesurée $\Delta(N)$:

$$A = \frac{(N-1)L}{\Delta(N)}$$

Cprobe utilise un mode de mesure aller-retour basé sur le Protocole ICMP. l'émetteur envoie des paquets ICMP *echo request* vers la destination qui lui renvoie des paquets ICMP *echo reply*. *Cprobe* analyse les délais aller-retour des trains de paquets et mesure ainsi la bande passante disponible. L'inconvénient de ce genre d'outils est que lors de mauvaises performances rapportées par les mesures, on ne sait pas quel chemin (aller ou retour) pose problème. De plus, de nos jours, les paquets ICMP sont de plus en plus filtrés rendant ainsi ce genre d'outils peu efficaces. Dans [Dov01], Dovrolis et Al ont montré que *Cprobe* et les outils qui implémentent une méthode similaire (*Pipechar* par exemple), ne mesurent pas vraiment la bande passante disponible mais ils mesurent plutôt une métrique appelée ADR (*Asymptotic Dispersion Range*) liée au taux d'utilisation du lien bottleneck et qui correspond à la capacité de bout en bout en l'absence de trafic concurrent.
Bon nombre d'outils fondés sur le *Probe Gap Model* ont été développés, parmi eux on peut citer par exemple : Pipechar [JYCA01], Delphi [Rib00],. . ..

La mesure de la bande passante est actuellement en cours de discussion au sein de l'IETF, de nombreux drafts visant la standardisation de quelques techniques de mesures de la capacité et de la bande passante disponible ont été proposés. Cependant, aucune RFC n'a encore été validée dans ce domaine. L'un de ces drafts [ASU] propose une technique de mesure appelée ABEst [ASC02] qui est fondée sur la mesure de la fenêtre de congestion de TCP.

Malgré le développement de plusieurs outils de mesure de la bande passante ces quelques dernières années, l'outil de mesure le plus largement utilisé reste encore aujourd'hui l'outil MRTG (Multi Router Traffic Grapher) [OR]. En effet, MRTG repose sur une méthode de mesure passive, il permet d'interroger les routeurs et de connaître précisément leur charge en s'appuyant sur SNMP. Connaissant la capacité de chaque lien, il est alors facile d'en déduire la bande passante disponible. MRTG génère des pages HTML qui affichent les résultats sous forme de graphiques (figure 1.23). La granularité de mesure est limitée à celle qu'offre SNMP, habituellement 5 minutes. Pour connaître la bande passante disponible entre deux systèmes, il est donc indispensable de disposer de la liste des routeurs traversés et de bénéficier de droits d'accès en lecture aux MIB de ces derniers.

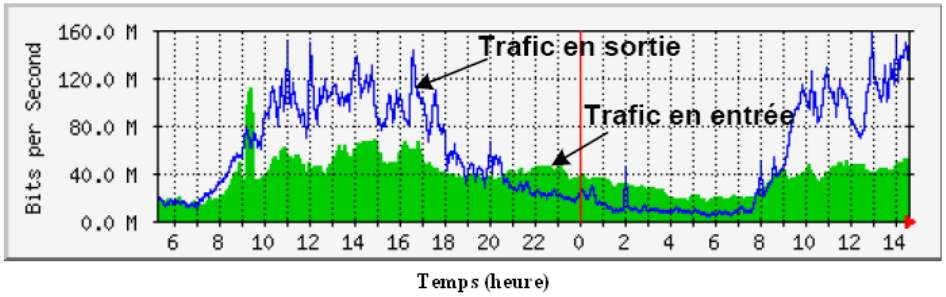


FIG. 1.23 – Exemples de graphiques obtenus avec MRTG.

1.10 Conclusion

Ce chapitre est loin d'être exhaustif, il a pour but de présenter les paramètres jugés essentiels de la qualité de service dans les réseaux. Pour chaque paramètre, nous avons dressé l'état de l'art des techniques et des outils de mesure qui lui sont associés. Le tableau A.2 de *l'annexe A* présente quelques outils de mesure et les métriques de l'IETF pour chaque paramètre de QoS. De plus, il indique des références bibliographiques dans lesquels le lecteur pourra trouver de précieux compléments d'information. Le tableau A.3 présente les caractéristiques de chaque outil de mesure : il indique les paramètres mesurés, la classe de mesure, et précise si l'outil est coopératif ou non (c'est-à-dire s'il est nécessaire de le déployer sur les deux systèmes concernés par la mesure), le protocole sur lequel les mesures sont faites et le système d'exploitation supporté.

Certains paramètres étudiés dans ce chapitre (tels que le délai unidirectionnel, le délai aller-retour, la gigue et les pertes de paquets) semblent être relativement plus faciles à mesurer que d'autres. En effet, les paramètres comme la capacité ou la bande passante disponible nécessitent des algorithmes beaucoup plus compliqués et utilisent des mécanismes plus difficiles à comprendre. De plus, ces derniers se basent sur des hypothèses qui limitent dans certains cas leur utilisation.

Dans la suite de ce manuscrit, nous avons choisi d'étudier les techniques et les outils de mesure de la bande passante disponible afin de mieux comprendre leurs fonctionnement, évaluer leurs performances et améliorer leur précision. Parmi les outils de mesure de ce paramètre, nous nous sommes intéressé particulièrement à quatre d'entre eux, à savoir : Pathload, pathchirp, Spruce et IGI (deux outils dans chaque catégorie avec des méthodes de convergence différentes). Ces derniers seront étudiés en détail au prochain chapitre et leurs performances seront comparées en termes de précision, de temps de convergence et d'intrusivité.

Deuxième partie

Contributions

Chapitre 2

Analyse et étude comparative des techniques et outils de mesure de la bande passante disponible

Résumé

Ce chapitre est consacré à l'étude des différentes techniques et outils de mesure de la bande passante disponible. Si certaines analyses et mesures de cette métrique ont été réalisées, nous avons constaté l'absence d'études comparatives significatives des résultats expérimentaux et l'absence d'un "outil référence" auquel seraient comparées les performances des outils de mesure de la bande passante disponible lors de leur développement. Nous avons donc effectué une analyse et une étude comparative des performances des techniques et des outils de mesure en se basant essentiellement sur les aspects précision, charge réseau induite et temps de réponse. Les tests sont effectués sur une plateforme isolée afin de pouvoir contrôler les différents paramètres du réseau et afin de tester les outils dans les mêmes conditions. Les résultats obtenus montrent que Spruce est l'outil qui offre les meilleures performances au regard des critères étudiés.

2.1 Introduction

L'étude de la bande passante disponible a connu un engouement considérable ces dernières années. Plusieurs techniques et outils de mesure ont été proposés et certaines études visant à comparer les performances des ces outils on été effectuées. Par exemple, Strauss & al [Din03] ont comparé l'outil Spruce à IGI et Pathload. Cette comparaison s'est portée sur les aspects précision, la charge sur le réseau et les problèmes liés aux modèles utilisés. Ils ont testé ces outils sur différents chemins de l'Internet (plus de 400) en comparant leurs résultats à ceux de l'outil MRTG qu'ils ont pris comme référence. Les résultats ont montré que Spruce est plus précis que Pathload et IGI. Ils ont montré aussi que Pathload tend à surestimer la bande passante disponible tandis qu'IGI est complètement imprécis quand le taux d'utilisation du bottleneck

est élevé.
Shreram & al [Shr05] ont évalué les outils de mesure de la bande passante disponible mis à la disposition de la communauté scientifique. Ils ont comparé les outils Abing [Nav03], Spruce, Pathload et Pathchirp sur une plateforme de mesure isolée développée par les chercheurs de l'association CAIDA (Cooperative Association for Internet Data Analysis) en collaboration avec le laboratoire CalNGI (California Next Generation Internet). Leurs résultats ont montré que les outils Pathload et pathchirp sont les plus précis sous les conditions de leurs expérimentations.
Hu & Steenkist [Hu03] ont développé deux outils de mesure de la bande passante disponible à savoir IGI et PTR qu'ils ont testé sur 13 chemins différents de l'Internet, puis ils ont comparé leurs résultats à ceux de Pathload. Iperf [Ipe] est l'outil référence utilisé dans cette comparaison. Cependant, étant donné que l'outil Iperf n'est pas très précis, les résultats de l'évaluation des performances d'IGI et de PTR sont mitigés.
Dans [Lab05], les auteurs ont comparé plusieurs outils de mesure de la bande passante disponible sur une plateforme conçue dans le cadre du projet METROPOLIS [REs01]. Les mesures de référence sont obtenues en utilisant des cartes DAG réputées pour la précision de leurs mesures. Dans cette comparaison les auteurs ont étudié les outils Abing, Spruce, IGI, Pathchirp et Pipchar en terme de précision, de temps de réponse et d'intrusivité. Les résultats de leurs tests ont montré qu'aucun de ces outils testés n'offre de mesures précises et leurs performances sont globalement médiocres.

Les outils de mesures étudiés dans ces différents projets de comparaison sont testés dans certains cas sur des plateformes déployées directement sur Internet, cependant les conditions de ces tests ne sont pas identiques pour tous les outils, ce qui pourrait fausser les résultats. Par ailleurs, d'un point de vue académique, ces outils de mesures sont des implémentations de modèles théoriques en vue de leur validation. Ces modèles sont fondés sur un certain nombre d'hypothèses qu'il convient de respecter lors de ces tests.
Parmi les différents modèles de mesure de la bande passante disponible étudiés jusqu'à aujourd'hui, ressortent deux techniques principales : la technique de l'auto-congestion et la technique à dispersions de paquets qui sont presque à l'opposé l'une de l'autre. Il convient donc de comparer les outils de mesure implémentant ces techniques et de déterminer laquelle des deux est celle qui offre les meilleures performances. Ceci afin de l'améliorer et de l'utiliser comme base pour un nouveau modèle de mesure de la bande passante disponible de bout en bout. Par ailleurs, mis à part les problèmes liées à la comparaison de ces outils de mesure, nous avons constaté aussi l'absence d'un "outil référence" auquel seront comparés les outils de mesure de la bande passante disponible lors de leur développement.

Le but de ce chapitre est donc de présenter les principaux outils de mesure implémentant les deux techniques précédentes, puis d'effectuer une analyse et une étude comparatives de ces outils en se basant essentiellement sur les aspects précision, temps de convergence et intrusivité. Les

tests sont effectués sur une plateforme isolée afin de pouvoir contrôler les différents paramètres du réseau et afin de tester les outils dans des conditions d'expérimentation identiques. Cette plateforme permettra aussi de reproduire les conditions initiales correspondant au différentes hypothèses sous lesquelles sont développés les différents modèles implémentés dans ces outils de mesure. A la fin de ce chapitre, nous tirerons nos conclusions par rapport aux performances de ces deux techniques et nous choisirons la meilleure d'entre elles pour lui apporter des améliorations en vue de l'utiliser comme base pour le développement d'un nouveau modèle qui sera par la suite implémenté dans un nouvel outil de mesure et qui sera validé au chapitre 3.

2.2 Outils de mesure de la bande passante disponible

Bon nombre d'outils de mesure des caractéristiques réseaux (délai, gigue, taux de pertes, etc) ont été développés. Cependant, ce n'est que ces dernières années que les outils de mesure de la bande passante disponible ont été conçus. Parmi les différents outils de mesure de ce paramètre, nous avons choisi d'étudier et d'évaluer les performances de quatre d'entre eux, à savoir Spruce, Pathload, pathchirp et IGI. Notre choix s'est porté sur ces outils car ces derniers implémentent les deux techniques qui nous intéressent mais chacun d'entre eux utilise un algorithme différent pour effectuer les mesures. Ils ont l'avantage d'être simples, leurs algorithmes assez faciles à comprendre et étaient aussi au moment de notre étude, les seuls outils mis à la disposition de la communauté scientifique. Il sont disponibles sous forme de programmes "Open-source" écrits en langage *C* et fonctionnant sous l'environnement Linux. L'objectif de cette section est donc de présenter ces différents outils, de décrire les algorithmes qu'ils implémentent et d'analyser les différentes hypothèses émises lors de leur conception.

2.2.1 L'outil Spruce

Spruce est un outil de mesure de la bande passante disponible dans un chemin de bout en bout, il est fondé sur le "Probe Gap Model" (PGM) [Din03] et utilise la technique de la paire de paquets définie dans le chapitre 1. Spruce suppose que le chemin étudié ne possède qu'un seul bottleneck, autrement dit, le bottleneck reste au niveau du même nœud et ne change pas de place pendant la mesure. Il suppose aussi que ce bottleneck est le lien qui a la plus petite bande passante disponible, donc le lien le plus serré (Tight link) et aussi le lien qui a la plus petite capacité, donc le lien le plus étroit (Narrow link). Spruce fait aussi l'hypothèse que tous les routeurs du chemins de bout en bout implémentent la politique de service FIFO et suppose que le trafic concurrent est fluide et qu'il change lentement pendant la phase de mesure.

Spruce est constitué d'un émetteur et d'un récepteur (figure 2.1). L'émetteur envoie vers le récepteur une séquence de paires de paquets avec une dispersion initiale Δ_{in}. En mesurant la dispersion finale Δ_{out} à l'arrivée, le récepteur calcule la quantité de trafic concurrent qui s'est infiltré entre les paquets de la paire et calcule ensuite la valeur de la bande passante disponible

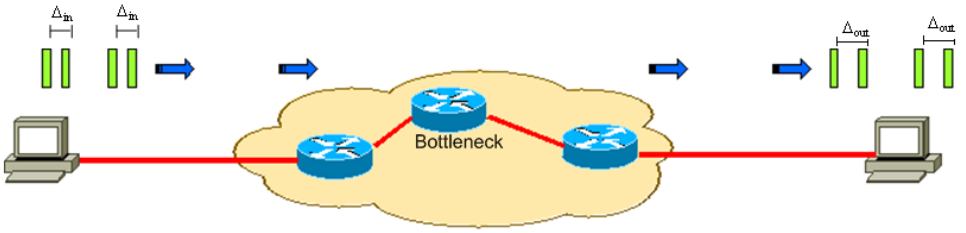


FIG. 2.1 – La technique de la paire de paquets implémentée dans Spruce.

A_i pour chaque paire i comme étant :

$$A_i = C\left(1 - \frac{\Delta_{out}(i) - \Delta_{in}(i)}{\Delta_{in}(i)}\right) \tag{2.1}$$

où C représente la capacité du lien bottleneck (cette dernière est supposée être connue ou peut être mesurée en utilisant l'un des outils spécifique à cette métrique), $\Delta_{in}(i)$ et $\Delta_{out}(i)$ représentent respectivement, la dispersion initiale et la dispersion finale mesurées pour chaque paire i.

La bande passante disponible A du chemin de bout en bout est calculée par Spruce comme étant la moyenne des bandes passantes disponibles A_i de toutes les paires de la séquence envoyée :

$$A = \frac{1}{n}\sum_{i=1}^{n} A_i \tag{2.2}$$

Spruce est configuré par défaut pour envoyer 100 paires de paquets sondes afin de réduire les erreurs épistémiques. Il utilise des paquets UDP de 1500 octets et règle la dispersion initiale Δ_{in} de manière à permettre aux deux paquets de la paire d'être dans la file d'attente du lien bottleneck en même temps (le second paquet arrive dans la file avant que le premier ne la quitte). Dans Spruce, cette dispersion initiale correspond au temps nécessaire au bottleneck de transmettre 1500 octets de données. Il est important aussi que cette valeur soit suffisamment élevée pour permettre aux paquets du trafic concurrent de s'infiltrer entre les paquets de la paire.

Spruce envoie les paires de paquets sondes selon un processus poissonnien afin d'éviter d'être intrusif. Par ailleurs, il implémente un mécanisme qui permet de réduire le trafic de mesure en ajustant la dispersion initiale Δ_{in} de manière à ce que la charge réseau induite soit le minimum entre 240 kb/s et 5% de la capacité du lien bottleneck.

2.2.2 L'outil Pathload

Pathload est un outil de mesure de la bande passante disponible qui implémente le modèle PRM (Probe Rate Model) [Din03] en se basant sur la technique SloPS (Self Loading Periodic Stream)[Dov03a]. Cet outil est composé d'un émetteur et d'un récepteur, il utilise deux types de paquets à savoir les paquets UDP pour les flux sondes et une connexion TCP pour les flux de contrôle.

Pathload envoie vers la destination un ensemble de flux périodiques avec un certain débit R. A l'arrivée, le récepteur mesure les délai unidirectionnels (OWD) de ces flux puis analyse leur variations. Pour mesurer la bande passante disponible, Pathload se base sur une heuristique qui considère que si le délai est jugé constant c'est que forcément le débit des flux sondes est inférieur à la bande passante disponible et que les paquets n'ont donc pas subi d'attente au niveau du lien bottleneck. En revanche, si le délai est croissant alors le débit des flux sondes est considéré comme étant supérieur à la bande passante disponible et que la croissance de ce délai est due à la saturation progressive du lien bottleneck causée par les flux sondes eux-mêmes (figure 2.2).

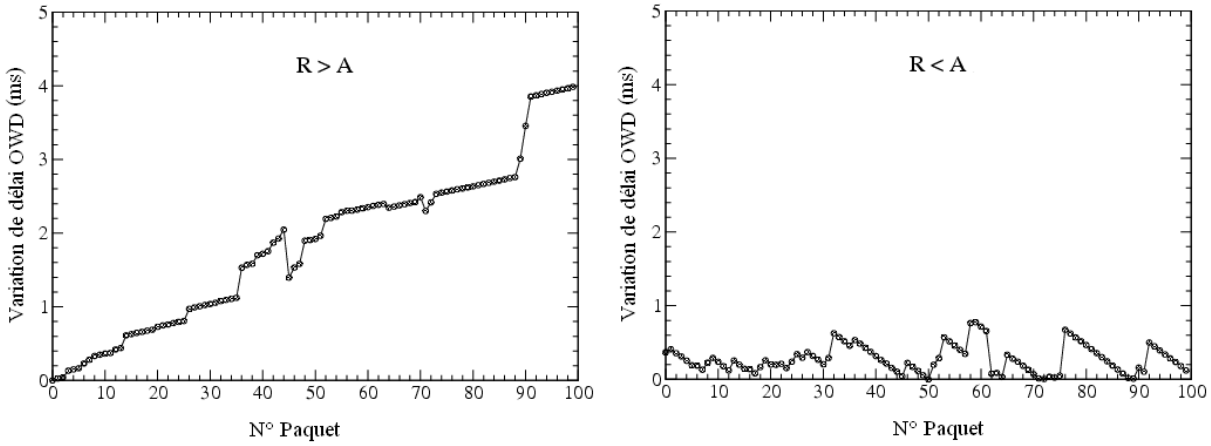


FIG. 2.2 – Tendances des variations de délais unidirectionnels des paquets de Pathload.

Le récepteur demande alors à l'émetteur de lui envoyer un autre flux avec un débit supérieur dans le premier cas et inférieur dans le second. Ce mécanisme est répété et Pathload estime par dichotomie la valeur de la bande passante disponible. Dans le cas où les flux sondes ne présentent aucune tendance, alors Pathload considère que le débit de ces derniers est compris dans un intervalle délimitant la valeur de la bande passante disponible. Cet intervalle est appelé "*Région grise*".

Les flux sondes envoyés par Pathload sont périodiques (de période T), ils sont composés de 100 paquets UDP de même taille L et ont un débit $R = L/T$. Pour varier les débits des flux sondes, Pathload fait varier la taille L des paquets entre 200 et le MTU. L'algorithme de Pathload étant itératif et non déterministe, on ne peut pas prédire son temps d'exécution. En effet, la convergence de l'algorithme peut prendre plus ou moins de temps selon la valeur de la bande passante disponible à estimer. Étant donné que cet algorithme est basé sur les variations des délais et non pas sur les délais eux-mêmes, il n'est pas nécessaire de faire appel à des mécanismes de synchronisation entre l'émetteur et le récepteur. Pathload n'émet aucune hypothèse sur le lien bottleneck et ne nécessite pas la connaissance préalable de la capacité de ce dernier. En revanche, il suppose que le chemin considéré est stable et qu'il ne change pas pendant la durée de la mesure.

2.2.3 L'outil Pathchirp

Pathchirp [Rib03] est un outil de mesure de la bande passante disponible dans un chemin de bout en bout, il utilise une technique d'auto-congestion très similaire à SLoPS (Self Loading Periodic Streams). Il consiste à envoyer des flux de paquets appelés "*Chirps*" d'une source vers une destination. Un "*chirp*" est constitué de N paquets de taille P, espacés d'une manière exponentielle avec un facteur de dispersion γ (figure 2.3).

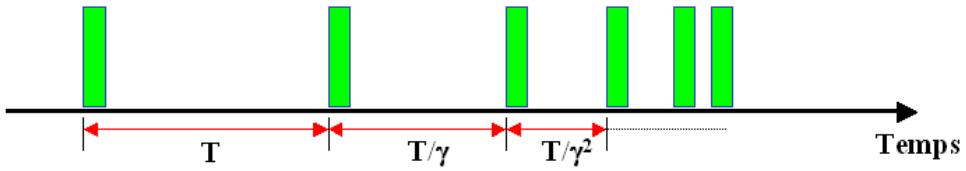


FIG. 2.3 – Chirp : flux de paquets sondes exponentiellement espacés.

L'évaluation de la bande passante disponible se fait par la mesure de l'écart entre les temps d'arrivée des paquets d'un même "*chirp*". L'analyse de l'évolution de l'écart de ces paquets permet de savoir quand le réseau a été saturé. Si la fréquence d'émission des paquets provoque un dépassement de la bande passante disponible, il y a alors mise en file d'attente des paquets dans un ou plusieurs routeurs, ce qui se traduit à la réception par une augmentation de la dispersion entre les paquets. Supposant que le délai d'attente d'un paquet K du "chirp m" dans la file est $q_k^{(m)}$, son temps de transmission par la source est $t_k^{(m)}$, et l'écart séparant deux paquets (k et$k+1$)est $\Delta_k^{(m)}$, le débit instantané du "chirp" au niveau du paquet k est alors :

$R_k^{(m)} = \frac{P}{\Delta_k^{(m)}}$

Si on considère que le trafic concurrent est constant, alors on aura : $q_k^{(m)} = 0$, si $A \geq R_k$ (tel que A est la bande passante disponible du chemin étudié) sinon $q_k^{(m)} > q_{k-1}^{(m)}$, dans le cas contraire.

Sous l'hypothèse que le trafic concurrent est constant, la bande passante disponible est définie simplement comme étant le débit du paquet k pour lequel le délai d'attente à un ou plusieurs nœuds commence à croître. Cette hypothèse n'étant pas réaliste, Pathchirp met en place une technique de détection d'une augmentation significative du délai de mise en file d'attente grâce à l'analyse de la "*signature*" d'un "*chirp*" .

Une *signature* (figure 2.4) représente le délai d'attente de chaque paquet du "*chirp*" en fonction de l'instant de son envoi. Une signature est constituée d'un ensemble "*d'excursions*" à partir de l'axe des temps qui est pris comme référence. Une "*excursion*" est définie comme une succession de paquets ayant subi une mise en file d'attente ($q_k^{(m)} > 0$) (pour un nombre de paquets supérieur à deux). La première "*excursion*" est une "*excursion*" qui se termine (le délai dans la file retourne à zéro après un certain nombre de paquets) car le débit R_k du "*chirp*" est inférieur à la bande passante disponible A. Par contre la dernière "*excursion*" ne se termine pas (les délais continuent d'augmenter pour les derniers paquets du "*chirp*"). Ceci peut être expliqué par le fait que R_k est supérieur à A. A partir de cette signature, on estime pour chaque paquet k la bande

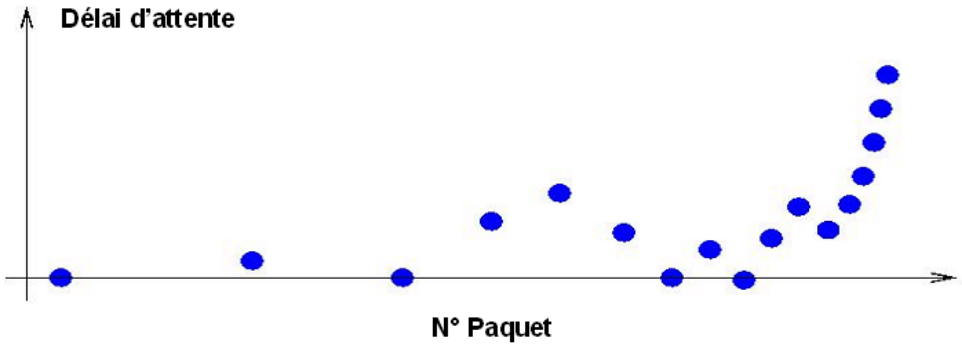


FIG. 2.4 – Signature du délai d'attente des paquets d'un chirp.

passante disponible E_k : Les E_k ont une valeur différente suivant la position du paquet k dans la signature du "*chirp*" :

- les paquets appartenant à une pente ascendante d'une *excursion* qui se termine ont pour valeur $E_k = R_k$ (débit du "*chirp*" au paquet k)
- les paquets appartenant à une pente ascendante d'une *excursion* qui ne se termine pas ont pour valeur $E_k = R_I$ où I représente le point de départ de l'excursion.
- Les paquets appartenant à une pente descendante ou n'appartenant pas à une *excursion* prennent la valeur $E_k = R_I$ où I est le même point défini précédemment.

A partir de l'ensemble de ces estimations par paquets, Pathchirp évalue la valeur de la bande passante disponible pour le "*chirp* m" (de longueur N) par la formule suivante :

$$D_{(m)} = \frac{\sum_{k=1}^{N-1} E_k^{(m)} \Delta_k}{\sum_{k=1}^{N-1} \Delta_k}$$

Enfin, la bande passante disponible du chemin de bout en bout est estimée comme étant la moyenne des $D_{(m)}$ pour M "*chirps*" :

$$A = \frac{\sum_{m=1}^{M} D_{(m)}}{M}$$

Pathchirp utilise des paquets UDP pour construire les "*chirps*", et considère que le chemin est constitué de routeurs équipés de files d'attente adoptant la politique d'ordonnancement FIFO. Pathchirp n'a pas besoin de synchronisation d'horloges entre la source et la destination car les mesures s'effectuent sur les variations des délais.

2.2.4 L'outil IGI

IGI utilise aussi le "Probe Gap Model" pour mesurer la bande passante disponible de bout en bout, mais au lieu d'utiliser d'une manière explicite des trains de paires de paquets, il utilise des trains de paquets périodiques et considère que chaque deux paquets consécutifs constituent une paire, comme le montre la figure 2.5.

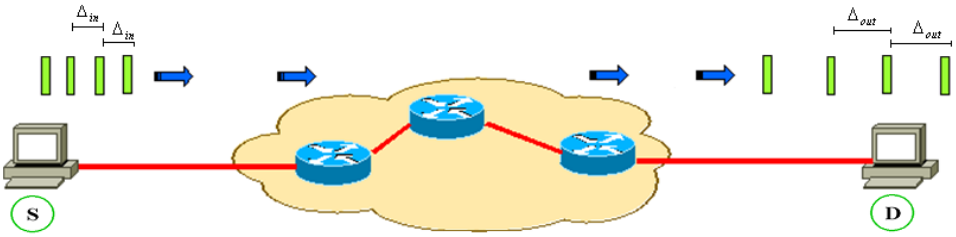


FIG. 2.5 – Train de paquets sondes généré par IGI.

Le modèle implémenté dans cet outil considère deux cas possibles à l'arrivée des paquets sondes au niveau du lien bottleneck :

- Le cas où la file d'attente est vide au départ du premier paquet.
- Le cas où la file d'attente n'est pas vide à l'arrivée du second paquet (des paquets du trafic concurrent s'introduisent entre les deux paquets considérés).

IGI définit une période appelée "*période d'attente*" qui correspond à l'intervalle de temps pendant lequel la file d'attente du bottleneck n'est pas vide. Quand les deux paquets de la paire se trouvent dans la même "*période d'attente*" on parle alors de période JQR (Joint Queuing Region), dans le cas contraire, il s'agit de la période DQR (Disjoint Queuing Region). Dans la région DQR la relation entre la dispersion initiale Δ_{in} et la dispersion finale Δ_{out} est donnée par :

$$\Delta_{out} = \Delta_{in}\frac{q}{C} \tag{2.3}$$

où C est la capacité du lien bottleneck et q la longueur de sa file d'attente à l'arrivée du premier paquet. Dans la région JQR cette relation est donnée par :

$$\Delta_{out} = \Delta_{in}\frac{C_T}{C} + \frac{L}{C} \tag{2.4}$$

où L est la taille du premier paquet et C_T le trafic concurrent qui traverse le lien bottleneck. A partir de ces formules, il est facile de voir que dans le cas de la période DQR il n'y a aucune relation entre la dispersion finale Δ_{out} est le trafic concurrent C_T. En revanche, dans la région JQR, il y a moyen de mesurer le débit du trafic concurrent à partir de la dispersion finale car d'après l'équation 2.4, Δ_{out} augmente linéairement avec l'augmentation du trafic concurrent C_T. En incrémentant la dispersion initiale, la différence entre Δ_{in} et Δ_{out} diminue jusqu'au point où Δ_{out} sera égale à Δ_{in}. Ce point particulier est appelé le "*Turning Point*", c'est à ce niveau que le débit des paires de paquets sondes est le plus proche de la valeur de la bande passante

disponible.

Dans la réalité, le trafic concurrent n'est pas constant et présente des "*bursts*". Une seule paire de paquets sondes ne suffit pas à mesurer correctement la bande passante disponible. Pour résoudre ce problème, IGI envoie vers la destination un train de 60 paquets de 700 octets chacun, tel que :

- M paires de paquets ont une dispersion initiale croissante.
- K paires de paquets ont une dispersion initiale stable.
- N paires de paquets ont une dispersion initiale décroissante.

IGI calcule ensuite la quantité du trafic concurrent en utilisant la formule suivante :

$$C_T = C \frac{\sum_{i=1}^{M} (\Delta_{in}^{+}(i) - t_b)}{\sum_{i=1}^{M+K+N} \Delta_{in}(i)} \tag{2.5}$$

où $\Delta_{in}^{+}(i)$ sont les dispersions initiales croissantes et t_b est le temps nécessaire au lien bottleneck pour transmettre le premier paquet de la paire.

Comme c'est le cas pour Spruce, IGI émet des hypothèses pour valider son modèle. Il suppose par exemple que le trafic concurrent est fluide et qu'il change lentement pendant la mesure et que les routeurs du chemin de bout en bout implémentent la politique de service FIFO. La *condition JQR* définie dans IGI est équivalente à la condition introduite dans Spruce et qui suppose que le deuxième paquet de la paire arrive avant le départ du premier. Avant de mesurer la bande passante disponible IGI mesure d'abord la capacité du lien bottleneck en utilisant un mécanisme identique à bprobe[CC96]. Les performances générale d'IGI dépendent donc aussi des performances de ce mécanisme.

2.3 Évaluation de performances

2.3.1 Méthodologie

Nous présentons dans cette partie les résultats de mesures obtenues avec les outils Spruce, Pathload, Pathchirp et IGI, puis nous analyserons ces derniers en termes de précision, de temps de réponse et de la charge réseau induite par le trafic de mesure puis nous vérifierons leurs répétabilité ainsi que leur intervalles de compatibilité. Ensuite, nous comparerons leurs performances.

Pour ce faire, nous avons mis en œuvre une plateforme d'expérimentation dont la topologie est représentée dans la figure 2.6.

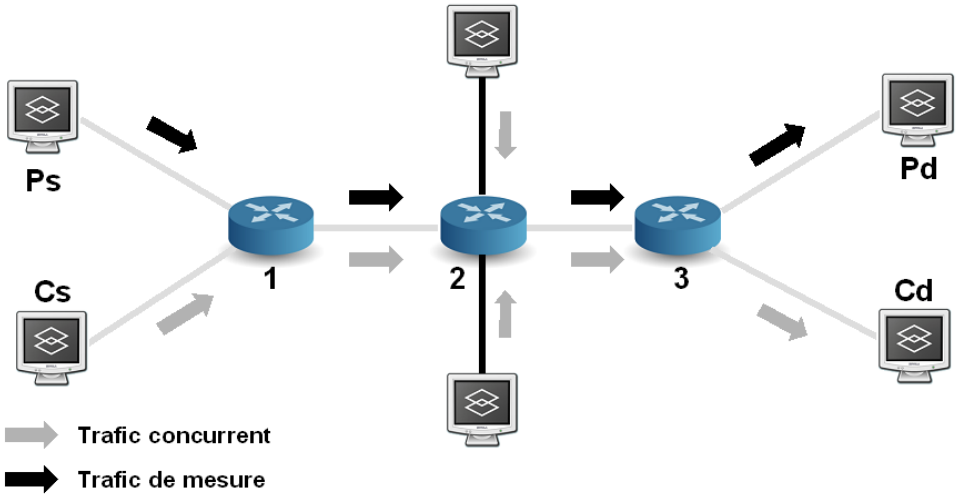


FIG. 2.6 – Plateforme d'expérimentation

La plateforme de tests consiste en un ensemble d'ordinateurs interconnectés via 3 routeurs Cisco . Les différents liens interconnectant l'ensemble des machines sont à 100Mb/s.

L'émetteur et le récepteur de ces différents outils de mesures sont installés respectivement sur P_s et P_d. Les flux de mesures circulent ainsi de P_s à P_d. Installés sur C_s, les générateurs de trafics MGEN [MGEN] et D-ITG [DITG] permettent d'envoyer un trafic concurrent vers C_d. MGEN génère un trafic UDP périodique alors que D-ITG permet de générer un trafic UDP et TCP selon différentes distributions. L'outil de capture Ethereal est installé au niveau des récepteurs. Il permet de vérifier le débit du trafic concurrent produit par les générateurs de trafic et de mesurer le temps de convergence et la charge induite par les outils de mesure. Lors de ces expérimentations, nous réglerons les différents trafics concurrents de façon à ce que le lien (*Routeur2, Routeur3*) soit le goulet d'étranglement (bottleneck).

Pour évaluer les performances de ces outils, nous avons considéré six scénarios, correspondant aux différents types de trafic concurrent utilisés. Dans les premier et deuxième scénarios nous avons considéré des trafics périodiques du type UDP et TCP respectivement. Dans le troisième scénario, nous avons considéré deux sources de trafic TCP. Le trafic concurrent au niveau du bottleneck est donc l'agrégation des flux TCP générés par ces deux sources. Dans le quatrième scénario le trafic est composé d'un flux TCP et d'un flux UDP. En revanche, dans les cinquième et sixième scénarios nous avons utilisé un trafic TCP dont les paquets sont générés avec des instants inter-depart (IDT : Inter Departure Time) exponentiellement distribués, avec une taille de paquets constante pour le cinquième scénario et une taille de paquet uniformément variable pour le sixième.

Dans chaque scénario le débit du trafic concurrent est réglé de manière à faire varier la bande

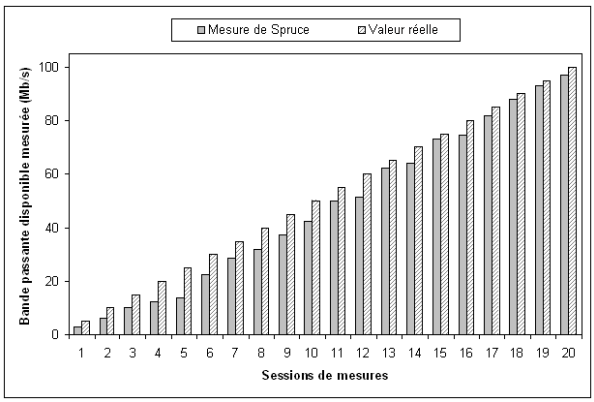
passante disponible du lien bottleneck de 0 et 100 Mb/s (avec un incrément de 5 Mb/s après chaque session de mesure). Pour chaque valeur de cet intervalle, on effectue 30 mesures. La valeur estimée de la bande passante disponible reportée est la moyenne des 30 valeurs obtenues.

2.3.2 Les résultats

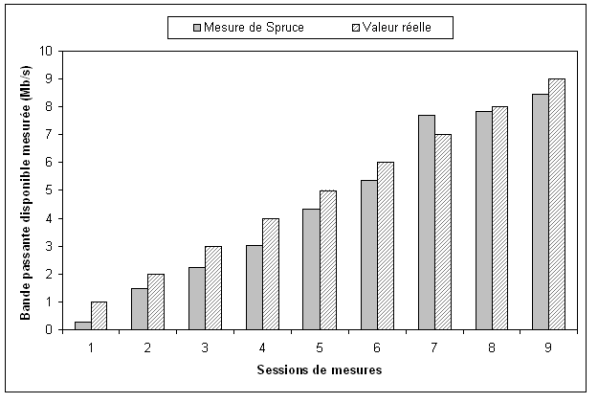
Les résultats obtenus lors des différents tests sont présentés dans les figures 2.7, 2.8 2.9 et 2.10. Ces derniers sont comparés aux valeurs de la bande passante disponible réelle obtenue en tenant compte de la taille des paquets sondes utilisés par chaque outil et prenant aussi en considération l'encapsulation des paquets au niveau des différentes couches de la pile TCP/IP. Ces résultats montrent qu'à part Spruce qui offre des mesures relativement précises, les autres outils offrent des performances médiocres.

En effet, les mesures obtenues en utilisant Pathload ne sont pas stables, ces dernières oscillent, tantôt en surestimant la bande passante disponible et tantôt en la sous estimant, particulièrement pour les valeurs inférieures à 80 Mb/s. Cependant, ces valeurs suivent la tendance générale des variations de la bande passante disponible. Ces remarques sont valables pour les quatre premiers scénarios considérés (trafic concurrent périodique de type UDP, TCP, deux flux TCP et TCP + UDP). En revanche, pour les deux derniers scénarios où le trafic concurrent est un trafic TCP avec des temps inter-départs exponentiellement distribués, Pathload donne des résultats médiocres et il est incapable de suivre les variations de la bande passante disponible. Nous avons constaté lors de ces expérimentations que Pathload arrête prématurément ses mesures lorsque la bande passante disponible est inférieure à 10 Mb/s (soit 10% de la capacité du lien bottleneck). Il n'est donc pas capable d'effectuer des mesures pour les valeurs trop faibles de la bande passante disponible. Ce phénomène peut s'expliquer par le fait que Pathload intègre un mécanisme de détection de pertes de paquets sondes. Si pour un flux donné, plus de 10% des paquets sont perdus, Pathload écarte les mesures effectuées sur ce flux. Si ce phénomène se répète pour plusieurs flux consécutifs, Pathload arrête la mesure. Nous pouvons ainsi supposer que lorsque la bande passante disponible est faible, les flux générés par l'algorithme dichotomique de Pathload ont des débits trop élevés et subissent ainsi des pertes de paquets considérables au niveau du lien bottleneck, ce qui a pour conséquence de rendre les mesures impossibles. Nous avons aussi observé le même phénomène (mesure impossible pour la bande passante disponible inférieure à 10% de la capacité) pour d'autres valeurs de la capacité du lien bottleneck .

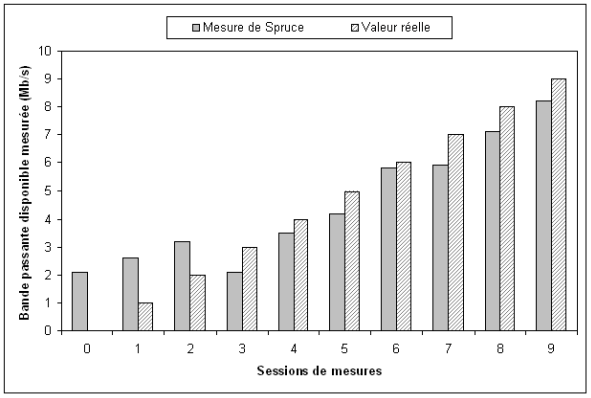
Les oscillations enregistrées par les mesures de Pathload et qui engendrent une surestimation ou une sous estimation de la bande passante disponible sont dues principalement à la nature de l'algorithme implémenté dans Pathload. En effet, ce dernier utilise un algorithme à convergence dichotomique qui fait varier le débit des flux sondes en l'augmentant ou en le diminuant de 50% à chaque fois, ce qui provoque des oscillations qui sont d'autant plus importantes et plus fréquentes lorsqu'il s'agit d'un trafic concurrent non périodique et à débits variables comme le montrent bien les figures 2.8(a) et 2.8(b). Nous supposons aussi que dans ces cas, le temps de convergence de cet outil est beaucoup plus long que pour des trafics concurrents constants.



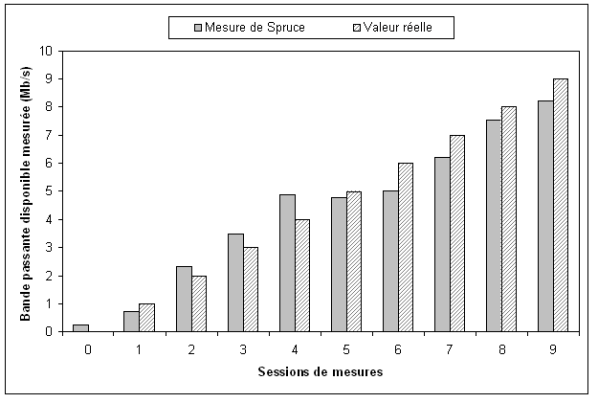
(a) Trafic = UDP



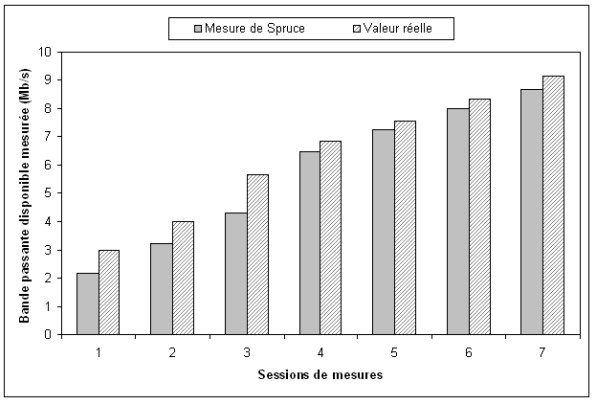
(b) Trafic = TCP



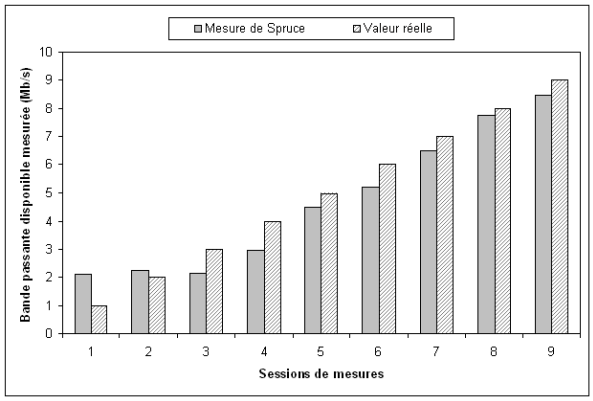
(c) Trafic = 2 flux TCP



(d) Trafic = TCP + UDP

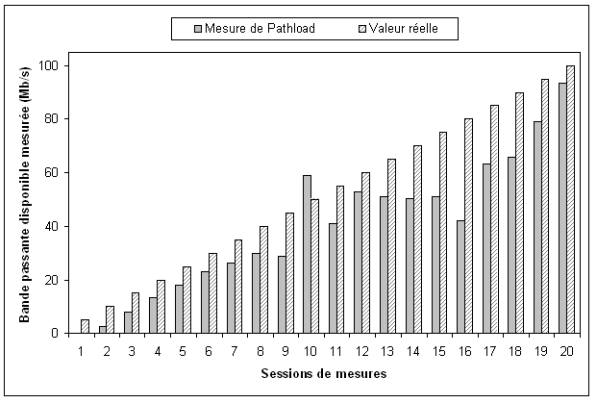


(e) TCP poissonnien, taille paquets variable

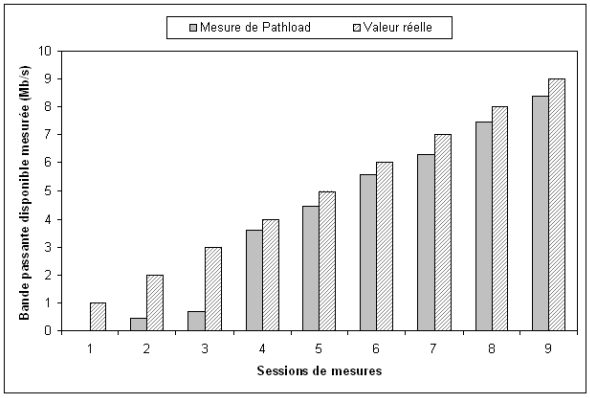


(f) TCP poissonnien, taille paquets constante

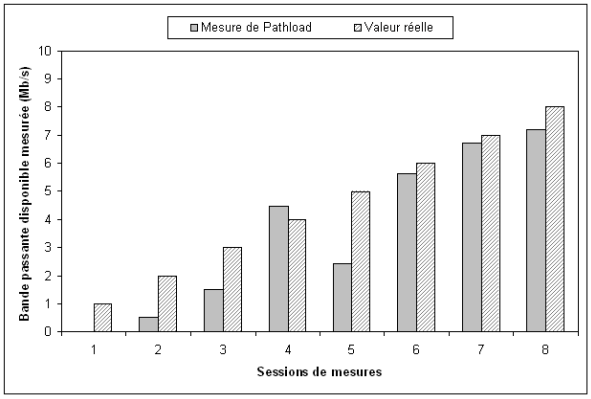
FIG. 2.7 – Résultats de mesures de Spruce



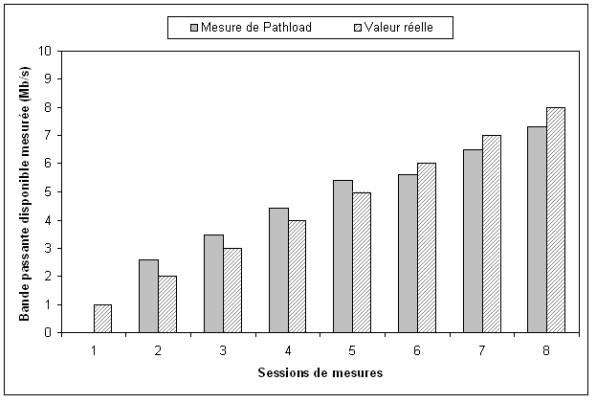
(a) Trafic = UDP



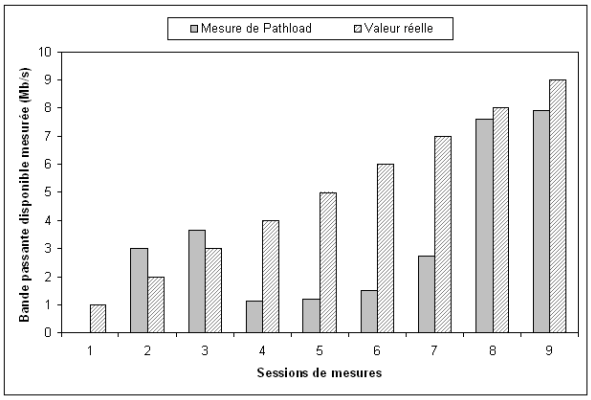
(b) Trafic = TCP



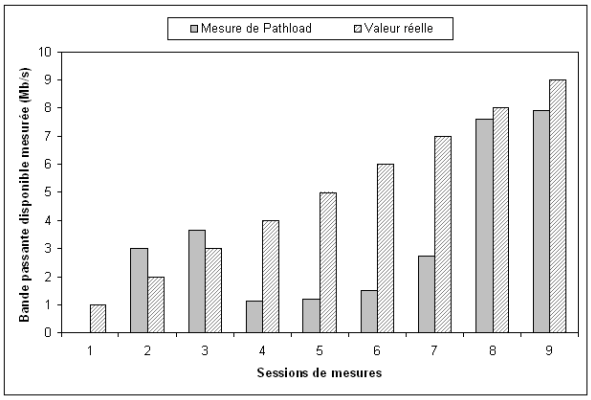
(c) Trafic =2 flux TCP



(d) Trafic = TCP + UDP

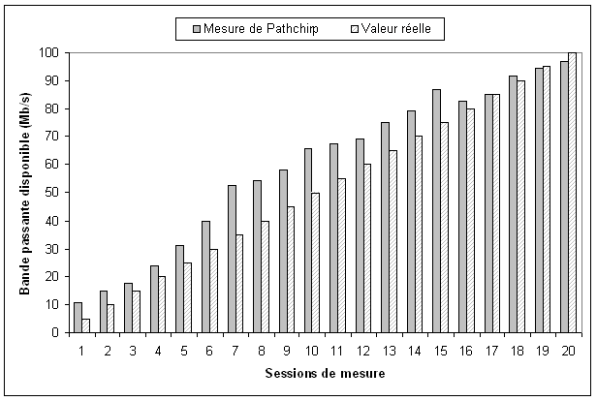


(e) TCP poissonnien, taille paquets variable

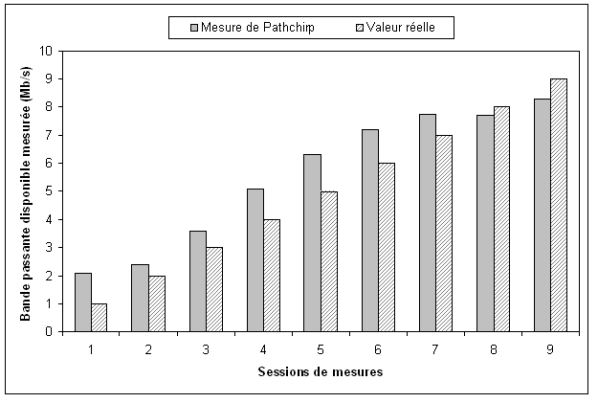


(f) TCP poissonnien, taille paquets constante

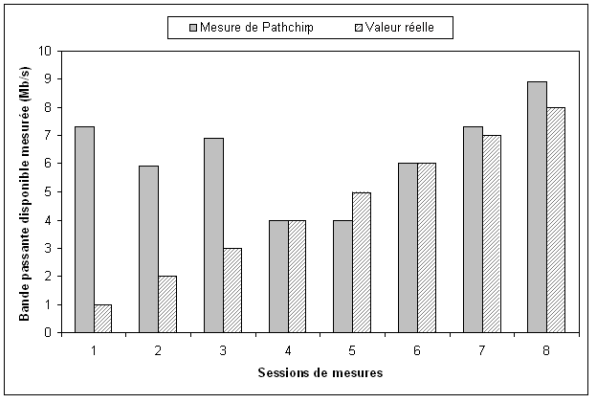
FIG. 2.8 – Résultats de mesures de Pathload



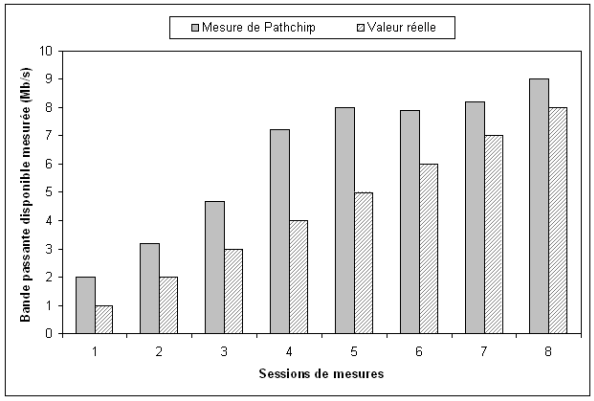
(a) Trafic = UDP



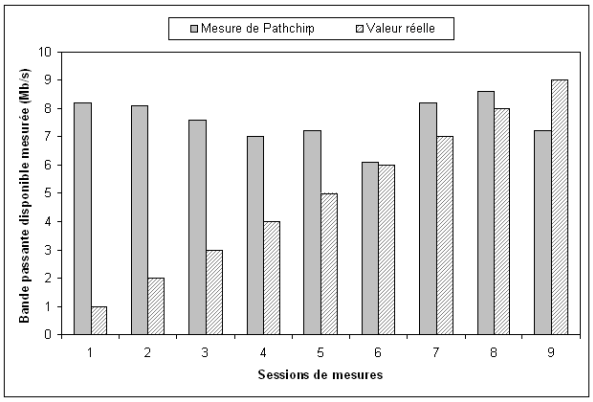
(b) Trafic = TCP



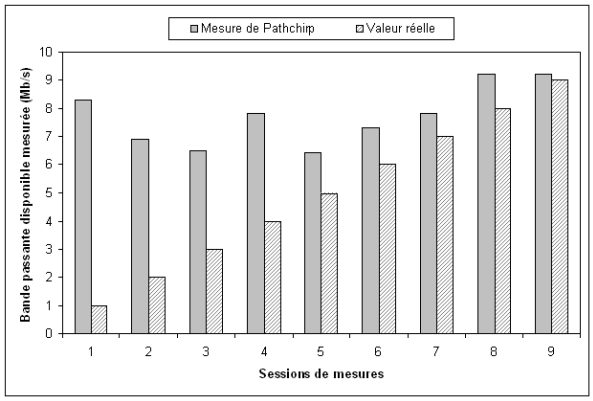
(c) Trafic =2 flux TCP



(d) Trafic = TCP + UDP

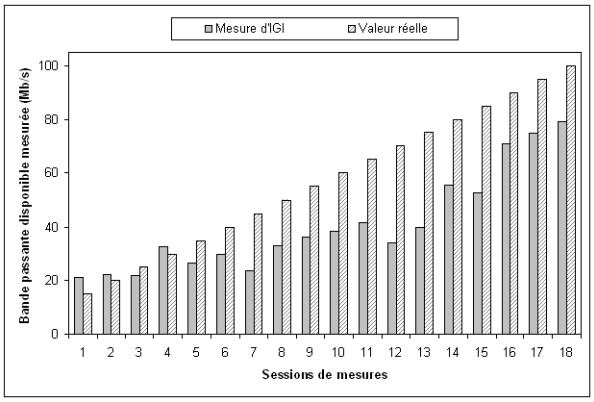


(e) TCP poissonnien, taille paquets variable

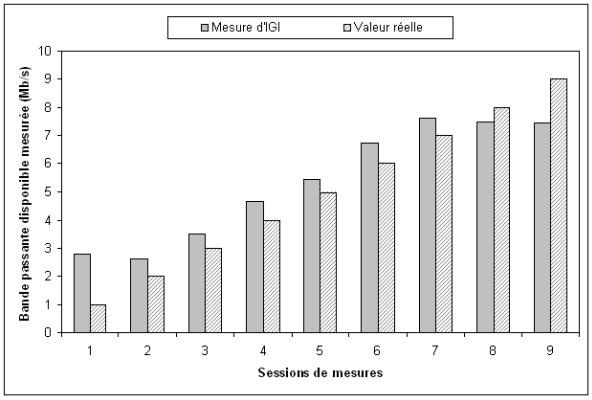


(f) TCP poissonnien, taille paquets constante

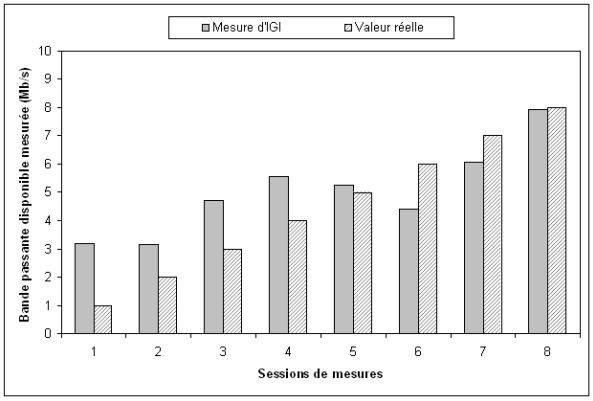
FIG. 2.9 – Résultats de mesures de Pathchirp



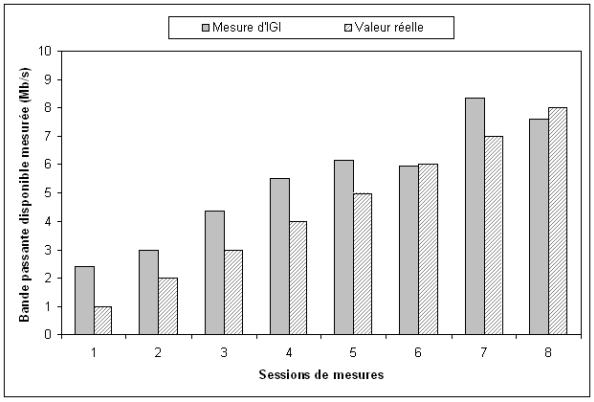
(a) Trafic = UDP



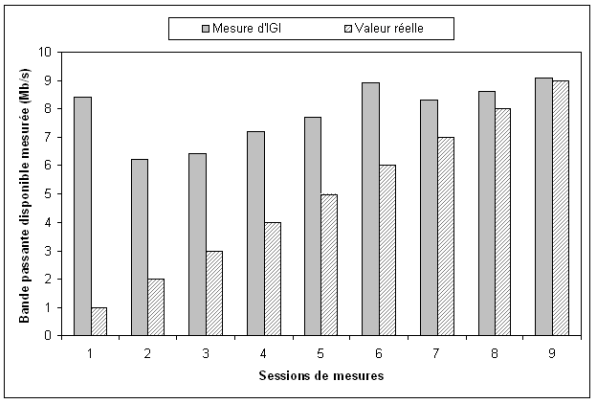
(b) Trafic = TCP



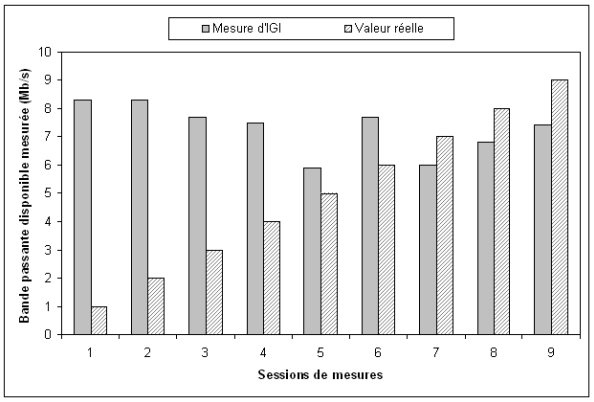
(c) Trafic =2 flux TCP



(d) Trafic = TCP + UDP



(e) TCP poissonnien, taille paquets variable



(f) TCP poissonnien, taille paquets constante

FIG. 2.10 – Résultats de mesures d'IGI

Les résultats de Pathchirp (figure 2.9) montrent que les mesures obtenues surestiment largement la bande passante disponible dans tous les scénarios considérés. Ces résultats montrent que cet outil n'est pas du tout sensible aux variations de la bande passante disponible et qu'il est totalement imprécis pour les valeurs faibles de cette métrique. L'imprécision des mesures s'accentue dans le cas des scénarios 5 et 6 qui utilisent un trafic non périodique à des débits variables.

Pathchirp nécessite de spécifier le temps d'exécution pour chaque session de mesure. Après des tests préliminaires, nous avons constaté que la variation du temps d'exécution a un impact négligeable, du moins non observable, sur les résultats. Les mesures présentées ici ont été obtenues avec une durée de 20 secondes.

Les résultats de l'outil IGI (figure 2.10) sont moins précis que ceux obtenus par Pathchirp. En effet, mis à part le cas où le trafic concurrent est un trafic UDP, les mesures d'IGI surestiment largement la valeur de la bande passante disponible et il est insensible aux variations de cette dernière. Nous pensons que les erreurs de mesures d'IGI sont dues principalement au fait que ce dernier mesure d'abord la capacité du lien bottleneck avant de mesurer la bande passante disponible, des petites erreurs de mesure sur cette capacité se répercuteront forcément sur la mesure de la bande passante disponible (voir chapitre 5).

En revanche, les résultats obtenus par l'outil Spruce montrent que ce dernier mesure la bande passante disponible avec une précision relativement élevée par rapport aux autres outils. En effet, malgré le fait que ses mesures sous-estiment légèrement la bande passante disponible, Spruce est l'outil qui se rapproche le plus des valeurs réelles de cette métrique et c'est l'un des outils qui suit de près les variations de cette dernière. Comme nous pouvons le voir sur la figure 2.7, contrairement aux autres outils qui surestime la bande passante disponible, Spruce la sous-estime. Surestimer cette métrique ferait " croire " à certaines applications qu'elles disposent de beaucoup plus de bande passante que ce qu'il en est en réalité. Ces dernières pourraient donc générer des volumes de trafic trop importants qui perturberaient le bon fonctionnement de ces applications et provoqueraient des congestions sur le réseau. En revanche, sous estimer la bande passante disponible pourrait avoir des conséquences moins graves (pour certaines applications).

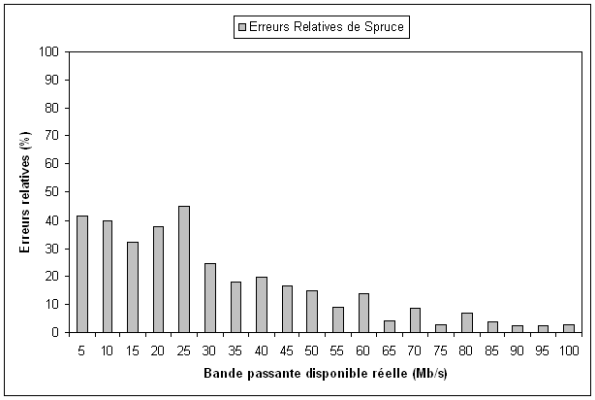
2.3.3 La précision

Pour étudier la précision de ces outils, nous étudions les erreurs de mesure relatives. L'erreur relative est calculée par la formule suivante :

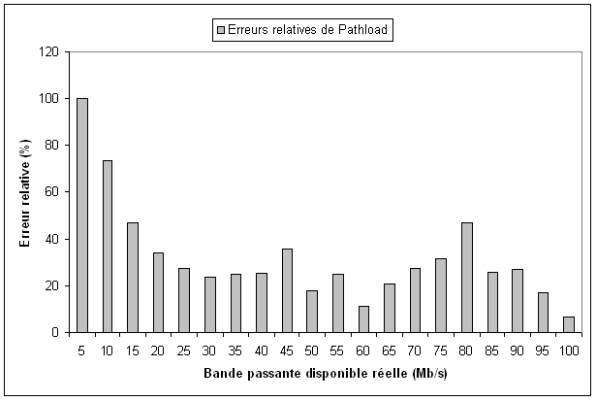
$$Erreur = \frac{|Val_{théorique} - Val_{mesurée}|}{Val_{théorique}}$$

Nous avons remarqué que la plupart des outils étudiés offrent des mesures imprécises dans presque tous les scénarios sauf pour le premier qui utilise un trafic concurrent périodique de type UDP. Nous allons donc effectuer dans ce qui suit une comparaison de ces outils au regard de la

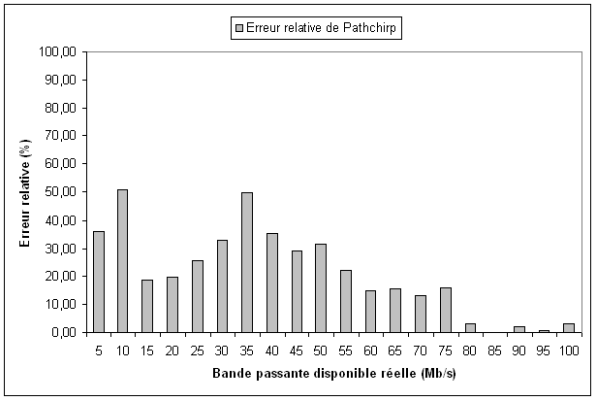
précision en considérant uniquement ce scénario. Les erreurs relatives obtenues sur les mesures de la bande passante disponible sont présentées dans la figure 2.11. Cette figure montre que Pathload offre des mesures avec des erreurs relatives qui varient dans un large intervalle. Ces dernières sont d'autant plus importantes et excèdent les 50% lorsque les taux d'utilisation du lien bottleneck sont élevés (ces erreurs sont beaucoup plus importantes dans les autres scénarios, particulièrement dans le cinquième et le sixième).



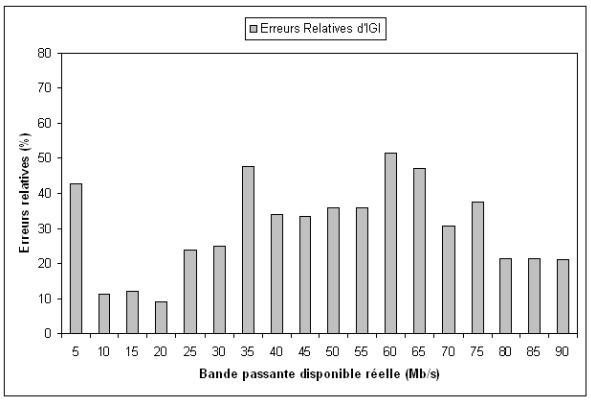
(a) Spruce



(b) Pathload



(c) Pathchirp



(d) IGI

FIG. 2.11 – Erreurs relatives des outils de mesure (UDP)

IGI et Pathchirp présentent des erreurs relatives assez élevées dépassant dans la plupart des cas 40% dans le premier scénario, témoignant ainsi de l'imprécision de ces outils. Dans les autres scénarios ces outils offrent des mesures complètement fausses avec des erreurs relatives exorbitantes lorsque la bande passante disponible est faible. Cependant, comme nous l'avons mentionné précédemment, IGI mesure d'abord la capacité du chemin de bout en bout avant de mesurer la bande passante disponible. Donc la précision de cet outil dépend (entre autres) de

la qualité des mesures de la capacité. Les résultats obtenus par Pathchirp et IGI montrent que dans la majorité des cas ces deux outils surestiment largement la valeur de la bande passante disponible, ce qui rend leur utilisation impossible.

En revanche, Spruce est l'outil qui a globalement l'erreur relative la plus faible et ceci quelque soit le scénario considéré. Les erreurs relatives de Spruce sont inférieures à 20% pour les valeurs de la bande passante disponible supérieures à 30Mb/s. Cependant ces dernières sont beaucoup plus importantes lorsque la bande passante disponible est inférieure à 15 Mb/s.

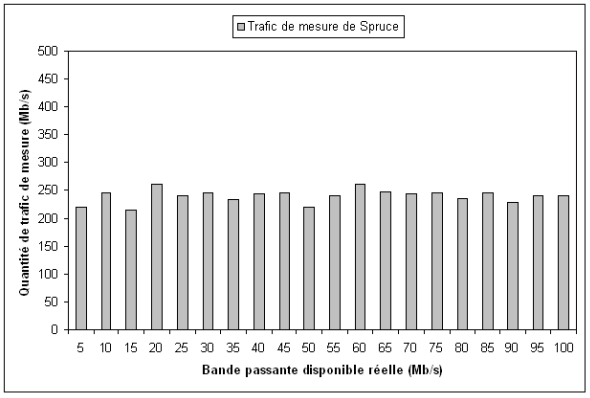
A partir de cette première analyse nous avons constaté que Spruce est relativement précis par rapport aux autres outils et que contrairement à Pathload ce dernier est capable de mesurer les petites valeurs de la bande passante disponible. Cependant, il le fait avec une faible précision. Par contre, Pathlaod, pathchirp et IGI offrent des performances médiocres et montrent des taux d'erreurs trop élevés pour presque tous les scénarios considérés.

Spruce et IGI sont tout les deux basés sur le Probe Gap Model (technique à dispersions de paquets). Cependant, les algorithmes de mesures utilisés sont complètement différents. Spruce utilise un ensemble de paires de paquets envoyées selon un processus poissonnien alors qu'IGI utilise un train périodique de paquets et considère que chaque deux paquets consécutifs constituent une paire.
De cette première analyse nous pouvons déjà déduire que la technique de la paire de paquets implémentée dans Spruce offre de meilleures performances que les autres techniques, du moins en ce qui concerne la précision. Les aspects temps de convergence et intrusivité sont abordés dans la suite de ce chapitre.

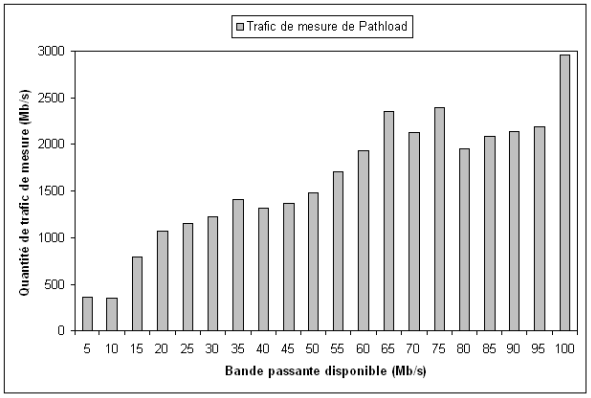
2.3.4 L'intrusivité

L'intrusivité est définie comme étant le rapport de la quantité du trafic généré par l'outil de mesure sur la capacité du chemin. C'est un critère d'évaluation très important car le trafic de mesure est susceptible de perturber le réseau et de biaiser les mesures s'il est trop important. La figure 2.12 présente les débits moyens des flux de mesure générés par les outils étudiés.

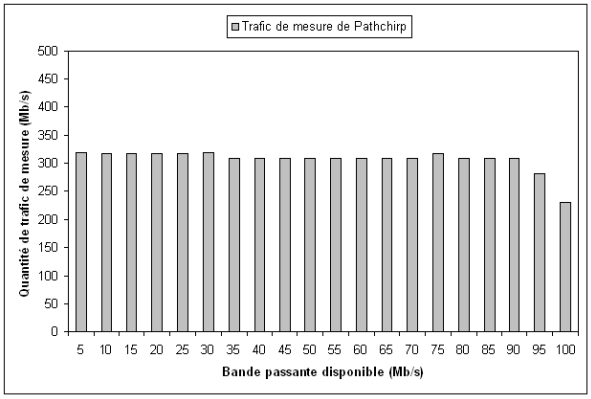
Cette figure montre clairement que Pathload génère beaucoup plus de trafic que les autres outils. Le débit moyen des flux de mesure peut atteindre des valeurs supérieures à 10% de la capacité du chemin. Par exemple, dans le cas d'un chemin de bout en bout de 100 Mb/s ce débit moyen dépasse les 10 Mb/s, ce qui représente une charge réseau considérable. Ces grandes quantités de trafic de mesure s'expliquent par la technique SLoPS utilisée. En effet, cette dernière consiste à saturer le réseau pour en extraire les informations. Si la bande passante disponible est trop large, il est nécessaire de générer une quantité importante de trafic sonde pour remplir le chemin de bout en bout, donc la méthode est très intrusive. Les quantités du trafic sonde présentées dans la courbe de Pathload sont des valeurs moyennes obtenues durant



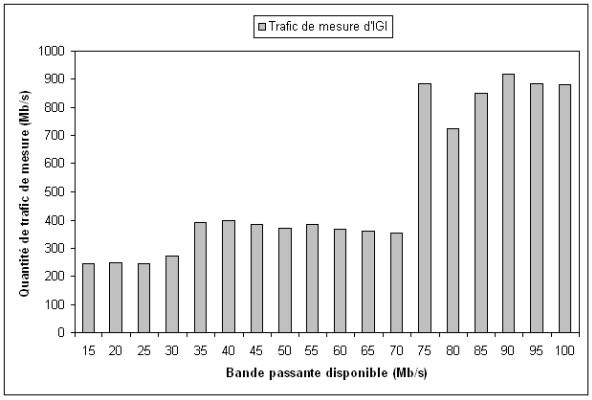
(a) Spruce



(b) Pathload



(c) Pathchirp



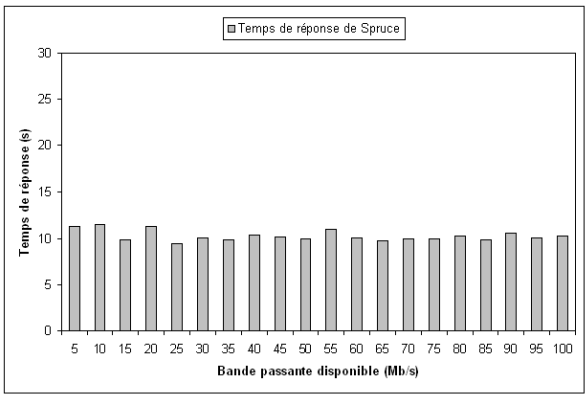
(d) IGI

FIG. 2.12 – Le trafic sonde généré par les outils de mesure (UDP)

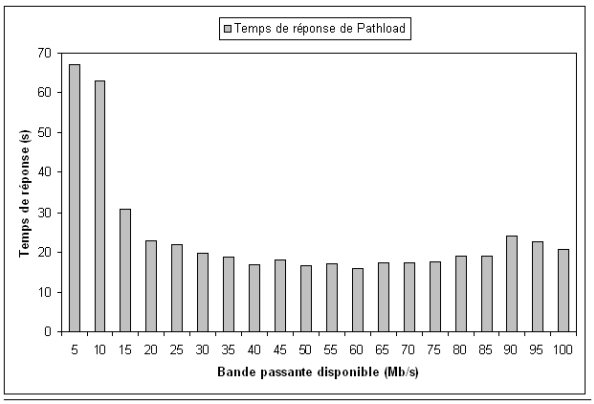
toute la période de mesure, ce qui veut dire donc que les charges réseaux instantanées générées par Pathload peuvent être dans certains cas très élevées dépassant largement les moyennes présentées. Pour les autres outils, les débits sont plus modestes. Les débits de Pathchirp et Spruce sont constants quelle que soit la bande passante disponible, et sont respectivement aux alentours de 300 et 240 kb/s. Le faible débit du trafic de mesure généré par Spruce est dû à un mécanisme qui permet de limiter ce trafic pour qu'il soit inférieur au minimum des deux quantités suivantes : 240Kb/s et 5% de la capacité du chemin de bout-en-bout (ce trafic est donc inférieur à 240 Kb/s dans tous les cas). Le débit d'IGI est de l'ordre de 200 kb/s pour des bandes passantes disponibles inférieures à 80 Mb/s et de 500 kb/s pour les valeurs supérieures. Toutefois, il est important de noter qu'IGI débute son exécution par la mesure de la capacité. Les valeurs présentées ici incluent le trafic relatif à cette mesure. Le volume du trafic utilisé par IGI pour la mesure de la bande passante disponible est ainsi inférieur aux valeurs reportées ici.

2.3.5 Le temps de réponse

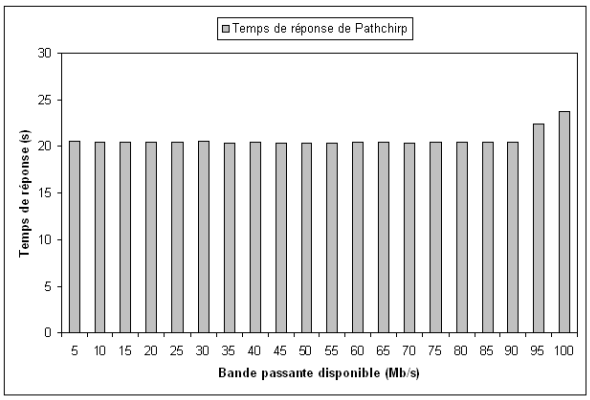
La rapidité avec laquelle un outil est capable de mesurer la bande passante disponible est un critère important. En effet, la bande passante disponible est un paramètre qui varie dans le temps, il est donc essentiel de pouvoir effectuer sa mesure le plus rapidement possible. Les mesures doivent être effectuées d'autant plus rapidement lorsqu'elles sont employées pour configurer dynamiquement des applications. Cette contrainte est moins forte lorsqu'un outil est utilisé dans le cadre de l'ingénierie de trafic (étudier le comportement d'un réseau en vue d'améliorer son architecture, distribuer les charges sur un réseau, etc).



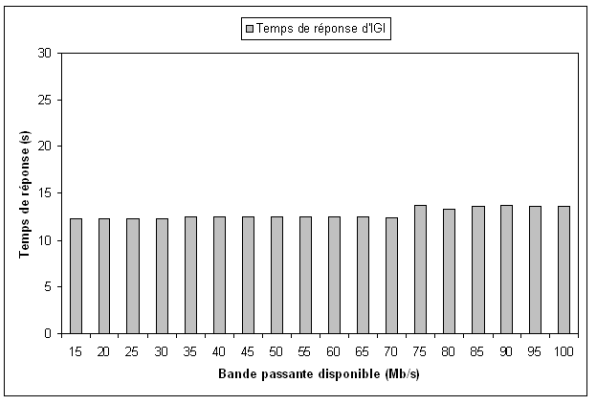
(a) Spruce



(b) Pathload



(c) Pathchirp



(d) IGI

FIG. 2.13 – Le temps de convergence des outils de mesure (UDP)

La figure 2.13 présente les durées moyennes de mesure de chaque outil pour les différentes valeurs de bande passante. Nous constatons que Pathload se distingue par son temps d'exécution variable. De plus, les temps de mesures sont supérieurs aux autres outils lorsque la bande passante disponible est inférieure à 40 Mb/s. Ce comportement peut s'expliquer par l'algorithme

de mesure mis en œuvre : Pathload commence par générer un flux à un débit important puis cherche à faire converger ce débit vers la valeur de la bande passante disponible. Le temps de convergence de l'algorithme est donc plus long lorsque la bande passante disponible est faible. En revanche, Pathload est le plus rapide lorsque la bande passante disponible est supérieure à 60 Mb/s. Pour tous les autres outils, les temps de mesure sont constants quelle que soit la bande passante disponible, respectivement 20s pour Pathchirp, 13s pour IGI et 10s pour Spruce. Notons que le temps de mesure d'IGI est représenté à titre informatif. En effet, il comprend la durée de la mesure préliminaire de la capacité. De plus, il est fixé par l'utilisateur.

2.3.6 Compatibilité des mesures

La compatibilité des outils de mesure permet de savoir dans quel intervalle de valeurs ces derniers sont susceptibles de donner les mêmes résultats. Il s'agit de vérifier la concordance des résultats obtenus avec les différents outils pour chaque valeur théorique donnée. Cette plage de valeurs est définie comme l'intersection des intervalles de confiance calculés pour chaque outil.

Appliqué à nos résultats avec un risque de 5%, le test de normalité de Shapiro & Wilk [SW65] permet de vérifier que les distributions des variables aléatoires X_i représentant les différentes valeurs de la bande passante disponible suivent une loi normale $N(M_i, \sigma_i)$ où M_i est la moyenne des valeurs mesurées et σ_i représente l'écart-type expérimental. Nous avons ensuite calculé les intervalles de confiance à 95% pour chaque outil. Ce qui signifie qu'il existe un risque de 5% pour que la valeur moyenne mesurée de la bande passante disponible ne soit pas dans l'intervalle calculé.

L'intervalle de confiance IC autour de la moyenne d'une distribution gaussienne est calculé comme suit :

$$IC = \left[M_i - \frac{\sigma_i t_{.025}}{\sqrt{n}}, M_i + \frac{\sigma_i t_{.025}}{\sqrt{n}} \right]$$

avec $t_{.}025$ la valeur critique de la distribution du t de Student correspondant au niveau de confiance 95%. Cette formule est appliquée pour calculer les intervalles de confiance de Spruce, IGI et Pathchirp. Pour Pathload nous avons considéré les intervalles de valeurs obtenus lors des mesures car ces derniers sont déjà sous forme d'intervalles de confiance calculés directement par l'outil Pathload et qui correspondent aux "*régions grises*" introduite dans [Dov02a]. Les intervalles de confiance des mesures de chaque outil sont représentés sur la figure 2.14.

Nous avons constaté que globalement, les outils de mesure ne sont pas compatibles. Leurs intervalles de confiance ne se superposent que sur quelques points. Ceci dit, nous avons remarqué une forte compatibilité entre le couple d'outil Pathchirp-IGI alors que le couple Spruce-Pathload ne présente qu'une compatibilité moyenne. Les autres combinaisons d'outils offrent une très faible

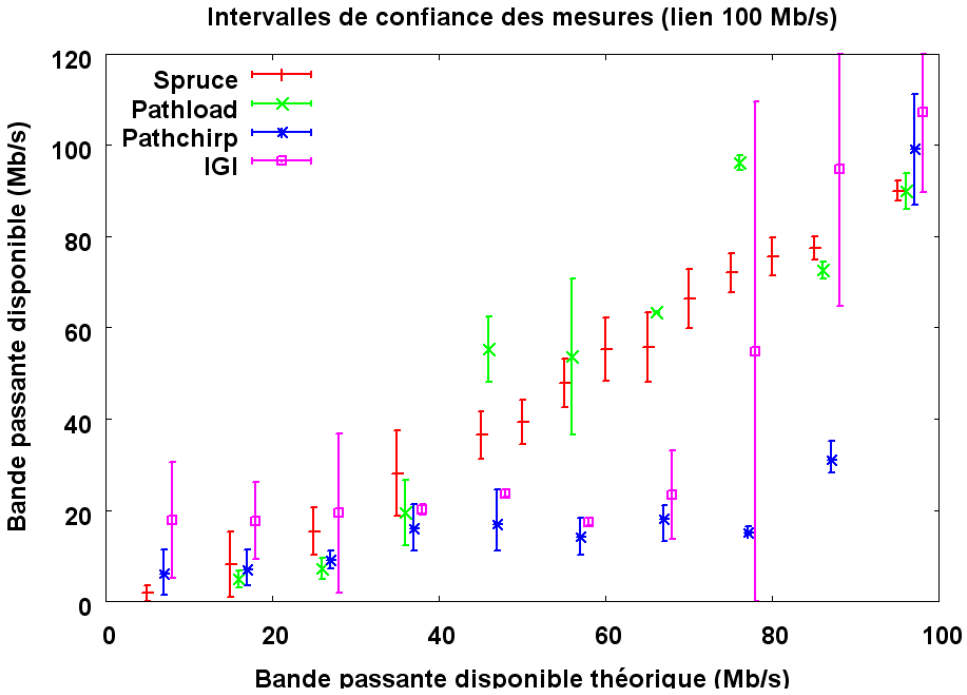


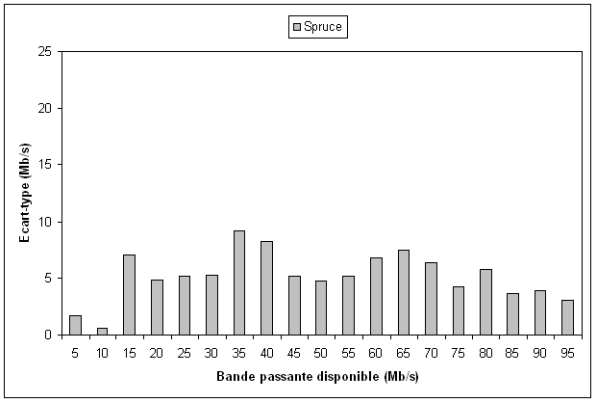
FIG. 2.14 – Intervalles de confiance des mesures (UDP).

compatibilité. Donc IGI et Pathchirp peuvent donner des résultats plus au moins concordants mais qui sont susceptibles d'être tout à fait éloignés de ceux obtenus par Spruce et Pathload. Ceci confirme d'ailleurs les résultats obtenus dans les sections précédentes et qui montraient que IGI et Pathchirp sont les outils qui offraient les plus mauvais résultats de mesures alors que Spruce et Pathload offraient des résultats plus au moins corrects.

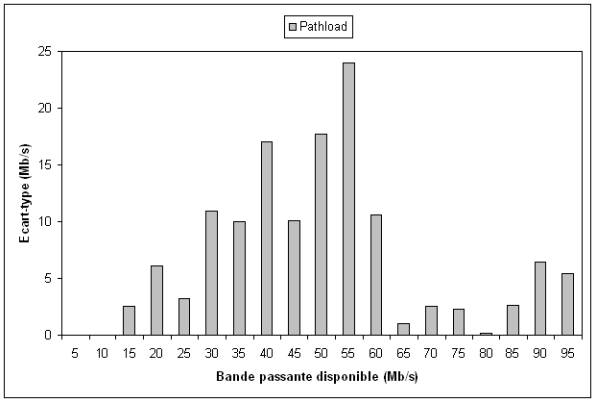
2.3.7 Répétabilité des mesures

La répétabilité (voir Annexe A) mesure l'aptitude d'un outil à reproduire des résultats identiques pour chaque série de mesures effectuées dans des conditions très proches (et donc généralement dans un temps court). L'écart type déterminant la distance moyenne des observations à la moyenne arithmétique, permet de mesurer la répétabilité des observations effectuées. Une valeur élevée de l'écart-type indique une dispersion importante des données et donc une faible répétabilité. La fidélité d'un outil est ainsi inversement proportionnelle à l'écart-type des mesures observées.

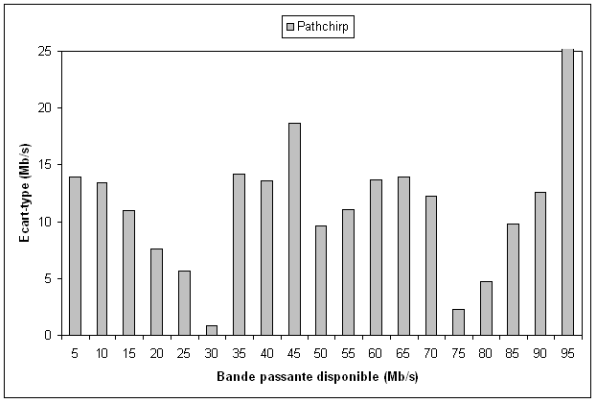
On constate que Pathchirp présente des écarts-type élevés, donc il offre une faible répétabilité des mesures. IGI est reproductible au niveau des valeurs faibles de la bande passante disponible et les mesures de Pathload présentent un niveau de répétabilité élevé pour les valeurs supérieures à 60 Mb/s. En revanche Spruce offre un niveau de répétabilité constant et raisonnable pour toutes les valeurs de la bande passante disponible mesurées. Donc, globalement, Spruce présente une meilleure fidélité par rapport aux autres outils de mesure.



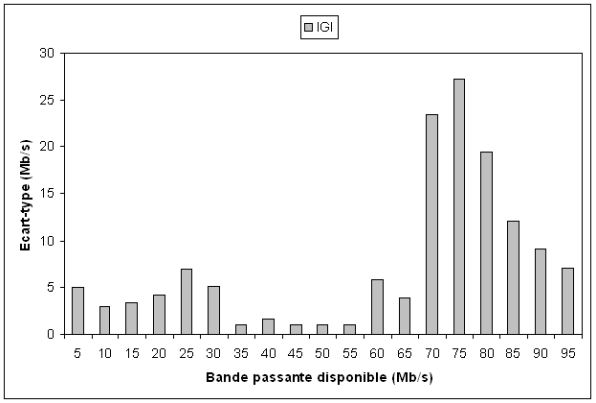
(a) Spruce



(b) Pathload



(c) Pathchirp



(d) IGI

FIG. 2.15 – Les Ecarts-types des résultats de mesure (UDP)

Les résultats présentés dans cette partie montrent que :

1. Pathload est l'outil le plus intrusif et est, dans certains cas, l'outil le moins rapide.
2. IGI et Pathchirp sont imprécis.
3. Spruce offre globalement les meilleures performances en termes de précision, d'intrusivité, de rapidité et de répétabilité. De plus, ses performances sont relativement stables et indépendantes de la valeur de la bande passante disponible.
4. La technique de la paire de paquets implémentée dans Spruce offre les meilleures performances que les autres techniques au regard de la précision, temps de convergence et d'intrusivité.

Cette technique sera améliorée et utilisée au chapitre 3 pour le développement d'un nouvel outil beaucoup plus performant.

2.4 Conclusion

Nous avons présenté dans ce chapitre une analyse et une étude comparative des techniques et des outils de mesure de la bande passante disponible. Nous avons mis en œuvre les différents outils sur une plateforme d'expérimentation dans des conditions identiques. Cette étude nous a permis de comparer les performances de ces outils en termes de précision, d'intrusivité et de rapidité ainsi qu'en terme de compatibilité et répétabilité des mesures.

Les résultats obtenus montrent que Pathload est l'outil le plus intrusif et peut s'avérer très lent. IGI et Pathchirp sont des outils généralement imprécis. Enfin, Spruce est l'outil le plus précis, le plus rapide et fait partie des outils les moins intrusifs. Cet outil offrant les meilleures performances au regard des critères étudiés, il sera considéré, pour l'instant, comme un outil référence auquel sera comparé les performances d'un nouvel outil de mesure de la bande passante disponible dont le modèle et l'implémentation seront présentés dans le prochain chapitre.

L'étude présentée dans ce chapitre s'est concentrée sur les critères qui nous semblaient essentiels. Cependant, la plateforme utilisée n'intégre pas toutes les configurations présentes dans l'Internet (le sans fil par exemple) et cette étude devra être complétée par la prise en compte d'autres paramètres tels que l'effet des équipements Store-and-Forward de couche 2, l'asymétrie et le changement des routes, etc. Par ailleurs, nous pensons indispensable de définir des critères plus pertinents que le débit moyen des flux de mesure pour quantifier l'intrusivité. Enfin, des études complémentaires sur réseaux réels (Internet) devront être menées.

Chapitre 3

Un nouveau modèle déterministe pour la mesure de la bande passante disponible

Résumé

Ce chapitre présente un nouveau modèle déterministe basé sur la technique de la paire de paquets pour la mesure de la bande passante disponible dans un chemin de bout en bout. Le modèle proposé repose sur le "*Probe Gap Model*" qui exploite dynamiquement les informations temporelles obtenues des paquets sondes afin de quantifier les différents paramètres caractérisant un goulet d'étranglement dans un chemin réseau. Les travaux réalisés dans ce domaine [DRM01]ont démontré que la taille des paquets utilisés pour sonder et analyser le chemin étudié est un paramètre très important qui affecte considérablement la mesure de la bande passante disponible. En nous basant sur ces conclusions, nous avons proposé un modèle qui prend en compte ce paramètre et nous l'avons implémenté dans un nouvel outil de mesure appelé IGMPS (Improved Gap Model using Packet Size parameter). Nous avons évalué les performances de cet outil sur une plateforme d'expérimentation selon différents scénarios et nous avons constaté que ce dernier permet de mesurer la bande passante disponible avec une très grande précision qui dépasse largement celle offerte par les autres outils existants. Enfin, nous avons étudié l'effet des différents paramètres du modèle proposé sur la précision de l'outil IGMPS. Nous avons constaté que la taille des paquets sondes est un facteur décisif dans la mesure de la bande passante disponible.

3.1 Introduction

Plusieurs métriques liées au délai, comme par exemple le délai aller-retour, le délai unidirectionnel ou la gigue ont été utilisés ces dernières années pour la reconfiguration dynamique de certaines applications réseaux. Cependant, de nos jours, nous avons noté l'émergence d'autres

métriques caractérisant un goulet d'étranglement dans un chemin de bout en bout, telles que la capacité ou la bande passante disponible. Ces derniers paramètres sont indispensables au bon fonctionnement de plusieurs applications telles que la sélection de routes dans les réseaux overlay, le contrôle de congestion et la spécification des contrats de garantie entre les clients et leurs fournisseurs d'accès (SLA : Service Level Agreement).

Dans ce chapitre, nous présentons un nouveau modèle déterministe basé sur la technique de la paire de paquets pour mesurer la bande passante disponible de bout en bout. Ce modèle établit une relation entre la quantité du trafic concurrent qui s'infiltre entre les paquets de la paire et la dispersion inter-paquets observée tout en tenant compte de la taille des paquets sondes utilisés pour analyser le chemin [Din03, MB00]. Nous avons implémenté le modèle proposé dans un outil de mesure appelé IGMPS (Improved Gap Model using Packet Size parameter) en considérant un cas particulier où les tailles des paquets constituant la paire sont égales. Sur une plateforme d'expérimentation, nous avons effectué des tests de validation selon différents scénarios afin d'étudier le comportement général de l'outil et d'analyser la précision des mesures obtenues. L'atout majeur du modèle proposé est l'introduction du paramètre taille de paquets sondes dans le calcul de la bande passante disponible. Les tests effectués ont montré que l'intégration de ce paramètre dans la formule de calcul d'IGMPS a considérablement amélioré la précision des mesures. Par ailleurs, la comparaison entre l'outil IGMPS et les autres outils de mesure basés sur la technique de la paire de paquets montre que pour des durées de convergence et des charges induites égales, IGMPS offre des mesures beaucoup plus précises.

3.2 Modélisation de la bande passante disponible

Dans cette partie, nous proposons un nouveau modèle basé sur la technique de la paire de paquets pour l'estimation de la bande passante disponible dans un chemin " *multihop* ". Au début, nous commençons par modéliser la capacité du chemin de bout en bout, puis en combinant la formule obtenue avec l'équation du *Probe Gap Model* nous obtenons une formule générique qui modélise la bande passante dans le chemin étudié.

Notre modèle est basé sur la méthode de la paire de paquets, il est donc impératif d'émettre les hypothèses suivantes :

- Le goulet d'étranglement sur le chemin de bout en bout ne change pas d'emplacement pendant la phase de mesure.
- Le trafic concurrent est fluide et varie lentement.
- Les routeurs du chemin de bout en bout implémentent la politique de service FIFO et sont tous " *store-and-forward*".

Les différentes notations et paramètres utilisés dans le reste de ce chapitre sont données dans le tableau 3.1.

En suivant le raisonnement donné dans [LB00, Pax97] et [Pax97a] la capacité du chemin est

Paramètre	Définition
n	Nombre de liens sur le chemin
d_i	Delai de propagation du paquet sur le lien i
C_i	Capacité du lien i
s^k	Taille du paquet k
t_l^k	Instant d'arrivée du paquet k au lien l
q_l^k	Délai d'attente du paquet k dans la file d'attente du lien l
b	Goulet d'étranglement (Bottleneck)

TAB. 3.1 – Définition des différents paramètres

dérivée de deux formules différentes : une formule définissant les instants d'arrivée des paquets à un nœud, une autre formule définissant les délais d'attente des paquet dans les files d'attente.

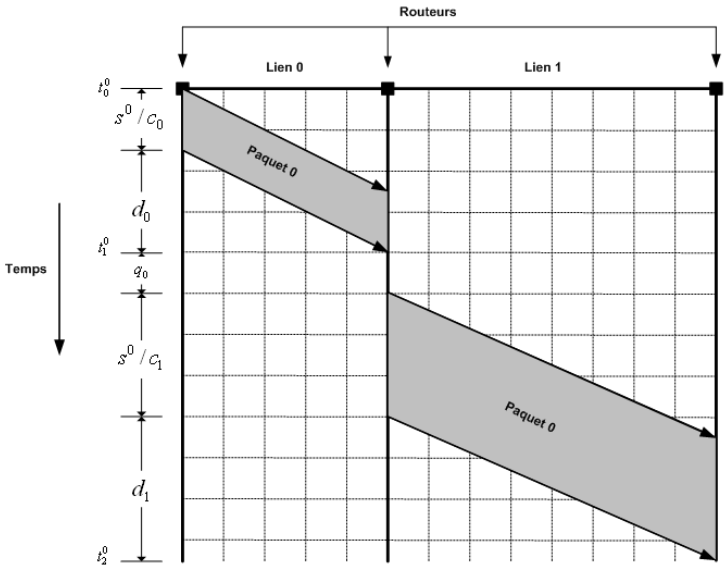


FIG. 3.1 – Modélisation des temps de transfert des paquets sur les liens.

La première formule donne le temps nécessaire à un paquet pour traverser l lien avant d'atteindre sa destination, elle est extraite des figures 3.1 et 3.2. L'instant d'arrivée t_l^k du paquet k au lien l est donné par la formule 3.1 :

$$t_l^k = t_0^k + \sum_{i=0}^{l-1} \left(\frac{s^k}{C_i} + d_i + q_i^k \right) \tag{3.1}$$

où t_0^k est l'instant d'envoi du paquet k et $(0 < i < l)$.

La deuxième formule, définissant les délais d'attente q_l^k dans les différents nœuds du chemin est extraite de la figure 3.2. Cette dernière est donnée par :

$$q_l^k = max \left(0, t_{l+1}^{k-1} - d_l - t_l^k \right) \tag{3.2}$$

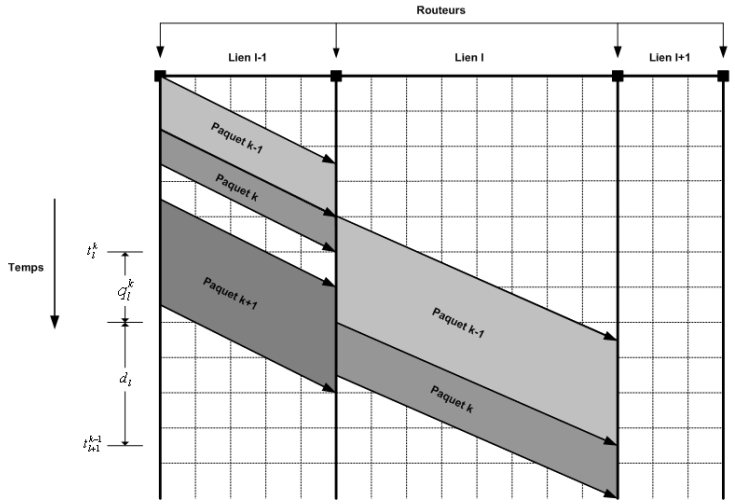


FIG. 3.2 – Modélisation des délais d'attente dans les files d'attente des différents noeuds du chemin.

En remplaçant (3.2) dans (3.1) nous obtenons :

$$t_n^k = t_0^k + \sum_{i=0}^{n-1} \left(\frac{s^k}{C_i} + d_i + max\left(0, t_{i+1}^{k-1} - d_i - t_i^k\right) \right) \tag{3.3}$$

la formule 3.3 représente le délai nécessaire à un paquet k de se déplacer de la source vers le nœud n. Dans la suite de ce chapitre, nous supposons qu'il existe un seul goulet d'étranglement avec une capacité $C_b = C$ et que le délai d'attente q_i n'intervient qu'au niveau du goulet d'étranglement. Dans ce cas, le délai défini dans la deuxième partie de l'équation 3.3 peut être divisé en trois délais différents à savoir (1) le temps que met un paquet pour atteindre le goulet d'étranglement, (2)le temps passé dans le goulet d'étranglement et (3) le temps passé entre le goulet d'étranglement et le nœud de destination n. L'équation 3.3 devient alors :

$$t_n^k = \left[t_0^k + \sum_{i=0}^{b-1} \left(\frac{s^k}{C_i} + d_i \right) \right] + \left[\frac{s^k}{C} + t_{b+1}^{k-1} - t_b^k \right] + \left[\sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} + d_i \right) \right] \tag{3.4}$$

avec $(0 < i < n)$.

Étant donné que la mise en file d'attente ne se produit qu'au niveau du lien bottleneck alors :

$$t_0^k + \sum_{i=0}^{b-1} \left(\frac{s^k}{C_i} + d_i \right) = t_b^k$$

En remplaçant dans 3.4 nous obtenons :

$$t_n^k = t_b^k + \frac{s^k}{C} + t_{b+1}^{k-1} - t_b^k + \sum_{i=b+1}^{n-1} \left(\frac{s^k}{C_i} + d_i \right)$$

$$t_n^k = \left(\frac{s^k}{C}\right) + t_{b+1}^{k-1} + \sum_{i=b+1}^{n-1}\left(\frac{s^k}{C_i} + d_i\right) \tag{3.5}$$

En utilisant l'équation 3.3, t_{b+1}^{k-1} peut être calculé comme étant :

$$t_{b+1}^{k-1} = t_0^{k-1} + \sum_{i=0}^{b}\left(\frac{s^{k-1}}{C_i} + d_i\right)$$

L'équation 3.5 devient alors :

$$t_n^k = \left(\frac{s^k}{C}\right) + \sum_{i=0}^{b}\left(\frac{s^{k-1}}{C_i} + d_i\right) + t_0^{k-1} + \sum_{i=b+1}^{n-1}\left(\frac{s^k}{C_i} + d_i\right)$$

$$t_n^k = \left(\frac{s^{k-1}}{C}\right) + \sum_{i=0}^{b-1}\left(\frac{s^{k-1}}{C_i}\right) + \sum_{i=b}^{n-1}\left(\frac{s^k}{C_i}\right) + t_0^{k-1} + \sum_{i=0}^{n-1}(d_i) \tag{3.6}$$

Étant donné que :

$$\sum_{i=b+1}^{n-1}\left(\frac{s^k}{C_i}\right) = \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \sum_{i=0}^{b}\left(\frac{s^k}{C_i}\right)$$

alors en remplaçant dans l'equation 3.6 nous obtenons :

$$t_n^k = \left(\frac{s^k}{C}\right) + \left(s^{k-1} - s^k\right)\sum_{i=0}^{b}\left(\frac{1}{C_i}\right) + \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) + \sum_{i=0}^{n-1}(d_i) + t_0^{k-1}$$

$$t_n^k = \left(\frac{s^{k-1}}{C}\right) + \left(s^{k-1} - s^k\right)\sum_{i=0}^{b-1}\left(\frac{1}{C_i}\right) + \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) + \sum_{i=0}^{n-1}(d_i) + t_0^{k-1} \tag{3.7}$$

De l'équation 3.7 la capacité du chemin de bout en bout C est dérivée comme suit :

$$C = \frac{s^{k-1}}{t_n^k + (s^k - s^{k-1})/\sum_{i=0}^{b-1}\left(\frac{1}{C_i}\right) - \left(s^k\right)/\sum_{i=0}^{n-1}\left(\frac{1}{C_i}\right) - \sum_{i=0}^{n-1}(d_i) - t_0^{k-1}} \tag{3.8}$$

L'équation 3.8 définit la capacité du goulet d'étranglement. Elle est exprimée en fonction de plusieurs paramètres qui influencent directement cette métrique. Dans ce qui suit, nous combinons cette équation avec celle du "*Probe Gap Model*" afin d'obtenir la formule générique permettant l'estimation de la bande passante disponible dans un chemin de bout en bout. Le *Probe Gap Model* établit la relation entre la dispersion inter-paquets dans la paire et le trafic concurrent qui s'y infiltre. En effet, une paire de paquets sondes est envoyée avec une dispersion initiale $\Delta_{in} = t_0^k - t_0^{k-1}$ pour atteindre la destination avec une dispersion finale $\Delta_{out} = t_n^k - t_n^{k-1}$.
Δ_{out} est aussi le temps nécessaire au goulet d'étranglement pour transmettre le premier paquet

ainsi que les paquets du trafic concurrent qui se sont infiltrés entre les paquets de la paire.

En utilisant le "*Probe Gap Model*", la bande passante disponible est calculée comme étant :

$$A = C\left(1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}}\right) \tag{3.9}$$

En remplaçant la capacité donnée par la formule 3.8 dans l'équation 3.9 nous obtenons :

$$A = \frac{s^{k-1}\left(2t_0^k + t_n^{k-1} - 2t_0^{k-1} - t_n^k\right)}{\left(t_0^k - t_0^{k-1}\right)\left[t_n^k + \frac{\left(s^k - s^{k-1}\right)}{\sum_{i=0}^{b-1}\left(\frac{1}{C_i}\right)} - \frac{\left(s^k\right)}{\sum_{i=0}^{n-1}\left(\frac{1}{C_i}\right)} - \sum_{i=0}^{n-1}(d_i) - t_0^{k-1}\right]} \tag{3.10}$$

Le délai de bout en bout OWD est l'accumulation des délais de transmission $\sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right)$, des délais d'attente dans les files q_i et des délais de propagation d_i :

$$OWD = \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) + \sum_{i=0}^{n-1} q_i + \sum_{i=0}^{n-1} d_i$$

$$OWD = \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) + \sum_{i=0}^{n-1} q_i + d^l$$

ce qui fait que :

$$d^l = \sum_{i=0}^{n-1} d_i = OWD - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \sum_{i=0}^{n-1} q_i \tag{3.11}$$

En supposant que les paquets ne subissent d'attente qu'au niveau du goulet d'étranglement, alors :

$$\sum_{i=0}^{n-1} q_i = q_b$$

où q_b est le délai d'attente au goulet d'étranglement.
L'équation 3.11 s'écrit donc :

$$\sum_{i=0}^{n-1} d_i = OWD - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - q_b$$

L'une des suppositions du "*Probe Gap Model*" est que le deuxième paquet de la paire arrive à la file d'attente du goulet d'étranglement avant que le premier paquet ne la quitte. A partir de cette hypothèse, nous pouvons représenter les instants d'arrivée et de départ des deux paquets

sondes selon le schéma de la figure 3.3 :

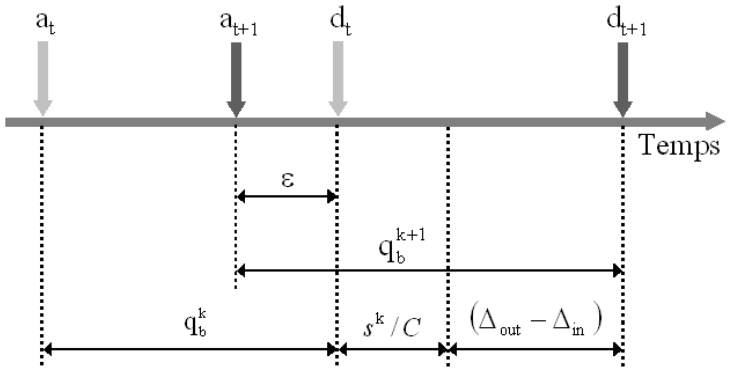


FIG. 3.3 – Modélisation des arrivées et des départs des paquets sondes au niveau du bottleneck.

A partir de cette figure, nous pouvons extraire l'équation définissant le temps d'attente d'un paquet k au niveau du lien bottleneck :

$$q_b^k = \epsilon + \left(\frac{s^{k-1}}{C}\right) + (\Delta_{out} - \Delta_{in}) \tag{3.12}$$

Nous supposons que ϵ est suffisamment petit pour le mettre égal à 0. Ceci nous permettra de faciliter les calculs et de simplifier les formules. Le cas où ϵ est différent de 0 est traité dans le paragraphe 3.6.
L'équation 3.12 s'écrit alors comme suit :

$$q_b^k = \left(\frac{s^{k-1}}{C}\right) + (\Delta_{out} - \Delta_{in}) \tag{3.13}$$

En remplaçant dans l'équation 3.11 nous obtenons :

$$\sum_{i=0}^{n-1} d_i = OWD - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \left(\frac{s^{k-1}}{C}\right) - \left(t_n^k - t_n^{k-1} - t_0^k + t_0^{k-1}\right) \tag{3.14}$$

Le délai unidirectionnel de bout en bout OWD est donnée par :

$$OWD = t_n^k - t_0^k$$

En remplaçant dans l'équation 3.14 nous obtenons :

$$d^l = \sum_{i=0}^{n-1} d_i = t_n^{k-1} - t_0^{k-1} - \sum_{i=0}^{n-1}\left(\frac{s^k}{C_i}\right) - \left(\frac{s^{k-1}}{C}\right) \tag{3.15}$$

En remplaçant 3.15 dans l'équation 3.10 nous obtenons la formule générale exprimant la

bande passante disponible dans un chemin de bout en bout :

$$A = \frac{s^{k-1}\left(2t_0^k + t_n^{k-1} - 2t_0^{k-1} - t_n^k\right)}{\left(t_0^k - t_0^{k-1}\right)\left[\frac{\left(s^k - s^{k-1}\right)}{\sum_{i=0}^{b-1}\left(\frac{1}{C_i}\right)} + \frac{s^{k-1}}{C} + \left(t_n^k - t_n^{k-1}\right)\right]} \tag{3.16}$$

Le modèle présenté dans l'équation 3.16 comporte un paramètre difficile à mesurer. En effet, le paramètre ($\sum_{i=0}^{b-1}\left(\frac{1}{C_i}\right)$) représente la somme des inverses des capacités des liens situés en amont du lien bottleneck. L'estimation de ce paramètre nécessite l'accès aux différents nœuds constituant ces liens et suppose que l'utilisateur possède les droits d'accès nécessaires à cet effet. Or le but des mesures de bout en bout est justement d'éviter l'accès aux nœuds intermédiaires et de mesurer les caractéristiques du chemin sans se soucier de sa topologie interne et encore moins de sa situation géographique. De plus, les différents segments du chemin à mesurer peuvent appartenir à des systèmes autonomes différents, ce qui rendra l'obtention des différentes autorisations requises pour accéder à ces nœuds quasi impossible pour un simple utilisateur. Pour palier ce problème, il suffit de considérer le cas particulier où les paquets sondes constituant la paire sont de taille égale. Ceci permettra d'éliminer le paramètre ($\left(s^k - s^{k-1}\right)/\sum_{i=0}^{b-1}\left(\frac{1}{C_i}\right)$) et il ne restera donc que des paramètres mesurables de bout en bout ainsi que la capacité du lien bottleneck qui, quant à elle, est facile à mesurer en utilisant l'un des outils dédiés à cette métrique tels que Nettimer[LB01] ou Cprobe [CC96] par exemple. Utiliser un outil de mesure pour estimer la valeur de la capacité du lien bottleneck introduira des erreurs et des incertitudes à la sortie de notre modèle. Ce problème d'incertitudes est traité au chapitre 5, des solutions pour atténuer ces incertitudes et pour diminuer les erreurs de mesure y sont proposées aussi.

En considérant le cas particulier où les tailles des paquets sondes sont égales ($s^k = s^{k-1} = S$), alors la bande passante disponible dans un chemin de bout en bout est exprimée par :

$$A = \frac{S\left(2\Delta_{in} - \Delta_{out}\right)}{\left(\Delta_{in}\right)\left[\left(\frac{S}{C}\right) + \Delta_{out}\right]} \tag{3.17}$$

où ($\Delta_{in} = t_0^k - t_0^{k-1}$) est la dispersion inter-paquet initiale, ($\Delta_{out} = t_n^k - t_n^{k-1}$) est la dispersion inter-paquet finale, S est la taille des paquets sondes et C est la capacité du lien bottleneck.
En nous basant sur l'équation 3.17 nous avons développé un outil de mesure appelé IGMPS. L'implémentation et l'évaluation des performances de ce dernier sont abordées dans les sections suivantes.

3.3 IGMPS un nouvel outil de mesure

Avant de présenter l'aspect implémentation de l'outil IGMPS, nous allons d'abord décrire l'architecture générale d'un outil de mesure de la bande passante disponible basé sur l'approche active, ainsi que ces différents modules et les fonctions de chacun d'entre eux.
Nous avons étudié les codes sources des différents outils de mesure présentés dans les premier et deuxième chapitres. Nous avons constaté que l'architecture de ce genre d'outils pouvait être modulaire. Nous avons donc proposé une architecture qui permet aux concepteurs et aux développeurs de ces outils d'apporter plus facilement des modifications et des améliorations. Celle-ci permettra aussi la réutilisation de certains modules lors de développements de nouveaux outils ce qui apportera aussi un gain considérable en temps de développement et rendra le code beaucoup plus lisible.

3.3.1 Architecture d'un outil de mesure de la bande passante disponible

Les outils de mesure de la bande passante étudiés dans ce manuscrit peuvent être classés en deux catégories distinctes : les outils de mesures *unidirectionnelles* et les outils de mesures *aller-retour*.

Outils de mesures unidirectionnelles

En général, les outils et les techniques de mesures unidirectionnelles nécessitent l'utilisation d'une base de temps commune entre les différents sites en synchronisant les horloges des émetteurs et des récepteurs à l'aide de GPS, de modules de réception d'horloge radiofréquence ou encore en synchronisant sur des serveurs NTP. Cependant la plupart des outils de mesure de la bande passante disponible étudié dans les chapitres précédents (c'est valable aussi pour l'outil IGMPS) sont basés sur l'estimation de la variation des délais unidirectionnels. En considérant que les erreurs dues à la dérivé de l'horloge est négligeable (la durée de mesure de ce délai étant faible) la mesure de la variation de délai ne nécessite pas de synchronisation étant donné que les erreurs de synchronisation sont éliminées lorsqu'on calcule les différences de délais.
Donc dans l'architecture générale d'un outil de mesure de la bande passante disponible représentée dans la figure 3.4, l'aspect synchronisation n'est pas pris en considération. La figure 3.4 présente les différentes parties ainsi que les différents composants et relations qui constituent un outil de mesures unidirectionnelles. Cette architecture met en évidence deux parties distinctes, à savoir l'émetteur et le récepteur.

L'émetteur est le programme qui envoie à travers le réseau (le chemin dont les caractéristiques sont à mesurer) les paquets sondes au programme récepteur. De son coté, le récepteur reçoit les paquets envoyés par l'émetteur, analyse ces derniers et calcule par la suite la valeur de la bande passante disponible en appliquant des algorithmes implémentant des modèles qui peuvent être stochastiques ou déterministes en se basant sur une analyse statistique ou sur un ensemble d'heuristiques. Par exemple dans Pathload l'algorithme utilisé est un algorithme déter-

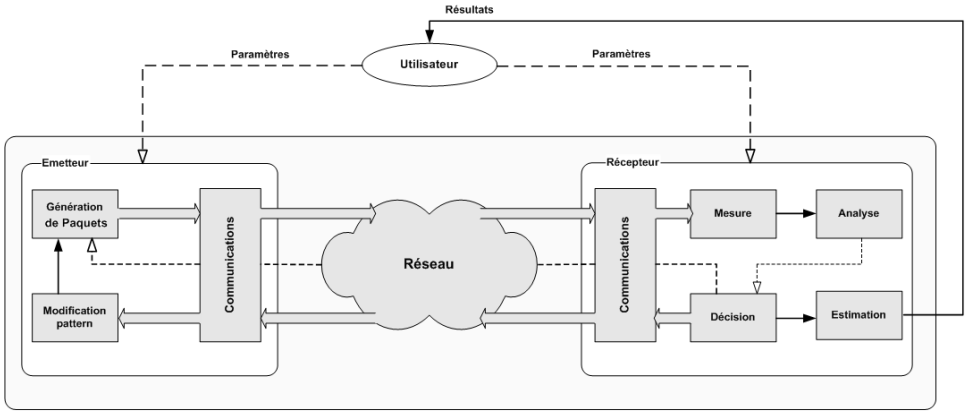


FIG. 3.4 – Architecture d'un outil de mesures unidirectionnelles.

ministe qui utilise une méthode dichotomique pour mesurer la bande passante disponible en se basant sur une heuristique qui considère que la valeur de la bande passante disponible est égale au débit du flux sonde à partir duquel le délai unidirectionnel des paquets sondes commence à augmenter significativement et d'une manière continue.

La partie émettrice comporte trois principaux modules à savoir :

- Le module de *génération de paquets*
- Le module *modification de pattern*
- Le module *unité de communication*

Le module *génération de paquets* est chargé de générer selon *le pattern* les paquets sondes qu'il transmet au récepteur via *l'unité de communication*. Le *pattern* est un modèle caractérisant le processus d'envoi des paquets sondes : processus périodique, poissonien, etc.
Dans l'exemple du *Probe Gap Model* (PGM) et du *Probe Rate Model* (PRM) introduits dans [Din03], les dispersions inter-paquet et les débits des flux sondes générés doivent être d'une grande précision. D'ailleurs, comme nous allons le démontrer dans le chapitre 5, les erreurs de mesures de ces paramètres influencent considérablement la précision des outils de mesure développés. Donc les performances des outils de mesure de la bande passante en général dépendent largement du module de *génération de paquets*. Par conséquent, il est très important de choisir un système d'exploitation capable de fournir une horloge système d'une haute résolution et d'une grande précision. Idéalement, un système d'exploitation temps réel (exp : RTLinux) serait le plus adéquat pour ce genre d'opérations. Une telle plateforme n'offrira pas seulement une grande résolution mais permettra aussi d'affecter au processus de mesure une priorité importante au niveau de l'ordonnanceur des tâches de manière à ce qu'il ne soit pas interrompu par d'autre processus, ce qui permettra par exemple d'éviter les erreurs d'estampillage des paquets. Le choix du langage de programmation est aussi très important. En effet, dans le cas des outils de mesure de la bande passante, le choix du langage se porte généralement sur le langage C

(sous Linux) car ce dernier offre de large possibilités et une multitudes de fonctionnalités dans le domaine de la programmation réseau, en l'occurrence au niveau des sockets.

Quand le module de décision situé au niveau du récepteur demande une modification du modèle de génération des paquets sondes, le module *génération de paquets* utilise de nouveaux paramètres pour générer les paquets sondes selon un nouveau modèle.

Le module *unité de communication* est chargé de transmettre les paquets sondes (émetteur) ainsi que les informations de contrôle (récepteur) à la carte réseau qui les envoie au récepteur/émetteur via le réseau de communication sur le chemin de bout en bout dont les caractéristiques sont à mesurer. La programmation de ce module est basé sur les sockets. En langage C sous l'environnement Linux, le module de communication est généralement développé en utilisant les RAW-sockets qui sont un type de sockets permettant d'extraire les entêtes des paquets sondes permettant ainsi une analyse beaucoup plus fine des paquets.

La partie réceptrice d'un outil de mesure (figure 3.4) est constitué de nombreux modules importants tels que le *module de mesure*, le *module de décision*, le *module d'estimation* et finalement comme pour la partie émettrice, cette partie dispose aussi d'*une unité de communication.* Le *module de mesure* reçoit les paquets sondes arrivant à *l'unité de communication* via le réseau et procède à la mesure des paramètres et des métriques nécessaires à la mesure finale comme par exemple le délai unidirectionnel des paquets sondes, la variation des délais ou bien les délais des flux sondes, etc. Le type de paramètre à mesurer dépend de la technique de mesure implémentée dans l'outil. Les mesures intermédiaires obtenues sont envoyées au *module de décision* qui les analyse en les comparant par exemple à une valeur référence ou en calculant leur variation, etc. Ensuite ce dernier prend une décision concernant la modification du pattern des paquets sondes pour pouvoir extraire les caractéristiques du chemin de bout en bout . Il peut aussi arrêter les mesures si les conditions requises sont satisfaites. La condition d'arrêt peut être par exemple le nombre de paires de paquets sondes reçues dans le cas du *Probe Gap model* (PGM). Une fois les mesures terminées, le *module estimation* procède au calcul de la bande passante en se basant par exemple sur une analyse statistique ou en utilisant un modèle bien défini comme c'est le cas par exemple pour Spruce qui utilise la formule 2.1 pour calculer la bande passante disponible correspondant à chaque paire de paquets, puis il calcule la moyenne de ces valeurs comme étant le résultat de la mesure de la bande passante disponible. Finalement, le résultat final de la mesure est envoyé à l'utilisateur (émetteur).

Outils de mesures aller-retour

Contrairement aux outils de mesures unidirectionnelles, dans les outils de mesures aller-retour l'émetteur et le récepteur sont implémentés sur la même machine.

L'architecture d'un outil de mesures aller-retour est représentée dans la figure 3.5. cette architecture se compose du même ensemble de modules que l'architecture présenté précédemment à la seule différence que ces derniers sont rassemblés sur une même partie qui joue le rôle de l'émetteur et du récepteur en même temps.

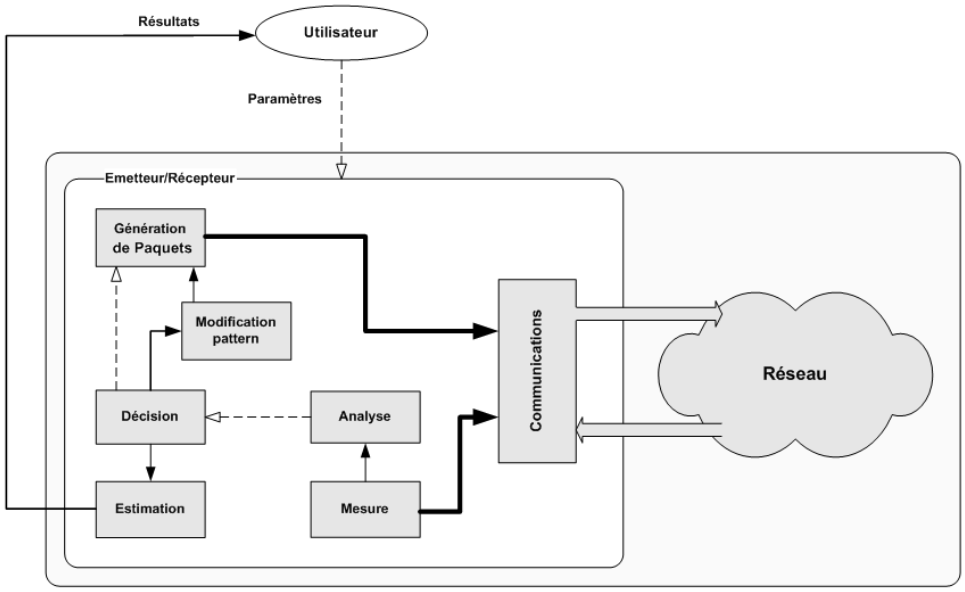


FIG. 3.5 – Architecture d'un outil de mesures aller-retour.

Le mécanisme d'envoi des paquets sondes est basé sur le protocole ICMP. Une utilisation maligne des services ICMP de par le passé a conduit à un filtrage presque systématique de ces paquets de la part de la majorité des serveurs et des utilisateurs, par conséquent ces outils deviennent aujourd'hui de moins en moins efficaces.

Malgré la nécessité de moyens de synchronisation qui peuvent être parfois onéreux ainsi que la nécessité d'avoir accès à la station distante pour installer et lancer la partie réceptrice, les outils de mesures unidirectionnelles reste le meilleur moyen pour mesurer les métriques unidirectionnelles telles que le délai, les pertes de paquets et la bande passante. L'outil de mesure IGMPS qui implémente le modèle proposé dans la section précédente est un outil de mesures unidirectionnelles qui mesure la bande passante disponible dans un seul sens sur le chemin de bout en bout considéré.

3.3.2 Implémentation de l'outil IGMPS

Comparé au Probe Gap Model décrit dans [Din03], notre modèle prend en compte la taille des paquets sondes qui est un paramètre important dans la mesure de la bande passante disponible. L'effet de ce paramètre sur les résultats de mesure en utilisant la technique de paire de

paquets est étudié et analysé en détail dans [DRM01].

La formule (3.17) est un cas particulier de la formule générale, elle est obtenue en mettant la taille du premier paquet de la paire égale à celle du deuxième paquet.
Nous avons implémenté cette formule dans un outil de mesure appelé IGMPS (Improved Gap Model using Packet Size parameter). Ce dernier est composé de deux parties distinctes : l'émetteur et le récepteur. Son fonctionnement général est identique à celui de Spruce, il suppose que la capacité du goulet d'étranglement est connue ou peut être mesurée par l'un des outils spécifiques à cette métrique tels que Nettimer [LB01] ou Cprobe [CC96].

IGMPS choisit la dispersion initiale Δ_{in} de la paire égale au temps que met le goulet d'étranglement à transmettre un paquet de 1500 octets. Cette dispersion est choisie de manière à permettre au trafic concurrent de s'infiltrer entre les paquets de la paire. Par ailleurs, cette dispersion ne doit pas être trop large pour que les paquets de la paire puissent se trouver, en même temps, dans la file d'attente du goulet d'étranglement.

Au niveau du récepteur , IGMPS mesure la dispersion finale Δ_{out} . Cette dernière est égale au temps de transmission d'un paquet de 1500 octets ainsi que les paquets du trafic concurrent arrivés entre les paquets de la paire. L'émetteur d'IGMPS envoie des paires de paquets UDP de 1500 octets en suivant un processus Poissonien. La dispersion entre les paires suit une distribution exponentielle avec une moyenne supérieure à Δ_{in}. L'utilisation d'un processus Poissonien rend IGMPS moins intrusif.
Pour améliorer sa précision, IGMPS mesure la bande passante disponible pour k paires de paquets puis considère leur moyenne comme étant la bande passante disponible de bout en bout. Par défaut, IGMPS configure la valeur de k égale à 100. Cette valeur est un compromis pour réduire les erreurs de mesure et pour obtenir un temps de réponse et une charge réseau raisonnables.

3.4 Validation et évaluation de performances

Nous présentons dans cette partie les résultats de mesures obtenues avec IGMPS, puis nous validerons ce dernier en termes de précision, du temps de réponse et de la charge réseau induite par le trafic de mesure. Ensuite, nous comparons IGMPS à l'outil Spruce qui utilise aussi le *Probe Gap Model*. Pour ce faire, nous avons mis en œuvre une plateforme d'expérimentation dont la topologie est représentée dans la figure 2.6.

3.4.1 Méthodologie

La plateforme de tests utilisée pour la validation d'IGMPS est identique à celle décrite à la section 2.3 du chapitre 2 . Le goulet d'étranglement du chemin de bout en bout considéré (figure

2.6) est situé au niveau du deuxième routeur (*Routeur2, Routeur3*), sa capacité est de 10 Mb/s alors que tous les autres liens interconnectant l'ensemble des machines sont à 100 Mb/s.

Pour évaluer les performances de notre outil, nous avons considéré trois scénarios différents, correspondant aux différents types de trafic concurrent utilisés. Dans le premier et le deuxième scénarios nous avons considéré des trafics périodiques du type UDP et TCP respectivement, cependant, dans le troisième scénario nous avons utilisé un trafic TCP dont les paquets sont générés avec des instants inter-depart (IDT : Inter Departure Time) exponentiellement distribués.

Dans chaque scénario le débit du trafic concurrent est réglé de manière à faire varier la bande passante disponible de 0 à 10 Mb/s (avec un incrément de 1 Mb/s après chaque session de mesure). Pour chaque valeur de cet intervalle, on effectue 30 mesures. La valeur estimée de la bande passante disponible reportée est la moyenne des 30 valeurs obtenues. Les résultats des expérimentations pour chaque scénario sont représentés sur les figures 3.6, 3.7 et 3.8.

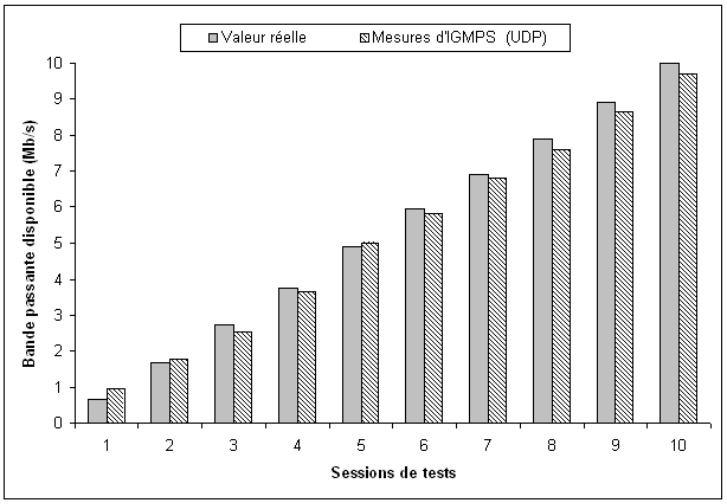


FIG. 3.6 – Résultats de mesure d'IGMPS dans un chemin à 10 Mb/s avec un trafic concurrent UDP.

3.4.2 Les résultats

Les résultats présentés dans ces trois figures montrent clairement que IGMPS mesure la bande passante disponible avec une très grande précision et qu'il répond d'une manière instantanée aux variations du trafic concurrent indépendamment du type de paquets utilisé (UDP ou TCP) et indépendamment de la distribution de ces derniers (périodique, poisson, etc).

L'écart-type détermine la distance moyenne des mesures à la moyenne arithmétique. Plus l'écart-type est grand plus les données sont dispersées autour de la moyenne ce qui indique que l'outil est peu reproductible donc peu fidèle. Pour mieux représenter les résultats obtenus et voir leurs distributions autour de la moyenne nous avons calculé les écarts-types des mesures pour

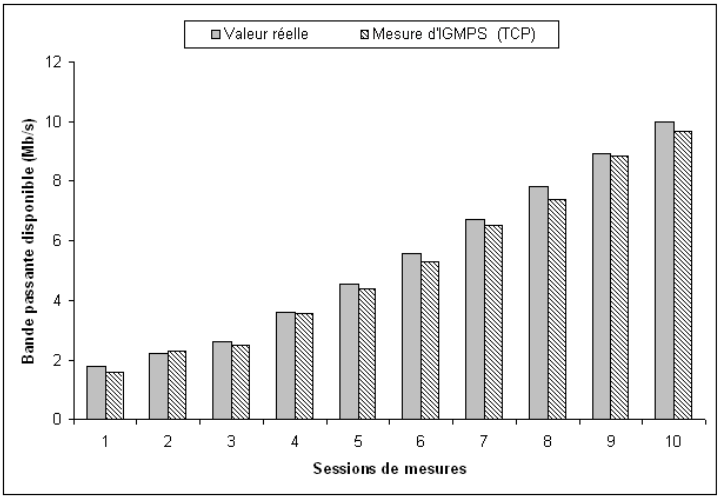


FIG. 3.7 – Résultats de mesure d'IGMPS dans un chemin à 10 Mb/s avec un trafic concurrent TCP.

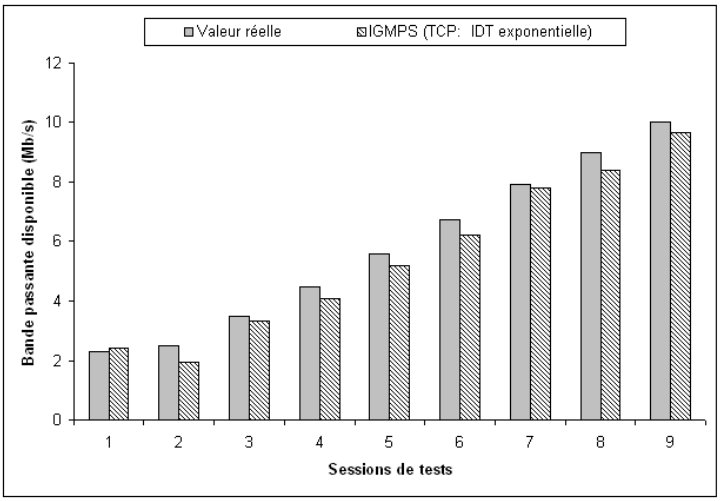


FIG. 3.8 – Résultats de mesure d'IGMPS dans un chemin à 10 Mb/s avec un trafic concurrent TCP dont les IDT sont exponentiellement distribuées.

chaque session et pour chaque scénario. Les résultats sont présentés dans la figure 3.10. Cette figure montre que les moyennes des mesures du deuxième et du troisième scénario sont obtenues avec des écarts-types relativement plus importants que ceux du premier scénario, ce qui signifie que sur les 30 mesures effectuées dans certaines sessions de mesures et dans chacun de ces deux scénarios, IGMPS sous-estime ou surestime considérablement la bande passante disponible.

Étant donné que les mesures sont effectuées dans les mêmes conditions et à des intervalles de mesure assez courts alors les résultats obtenus nous renseignent sur la répétabilité des mesures pour une valeur donnée de la bande passante disponible. En effet, la répétabilté est la variabilité aléatoire des résultats d'une série de déterminations d'un même échantillon effectuées dans les mêmes conditions ou dans des conditions très proches et donc généralement dans un laps de

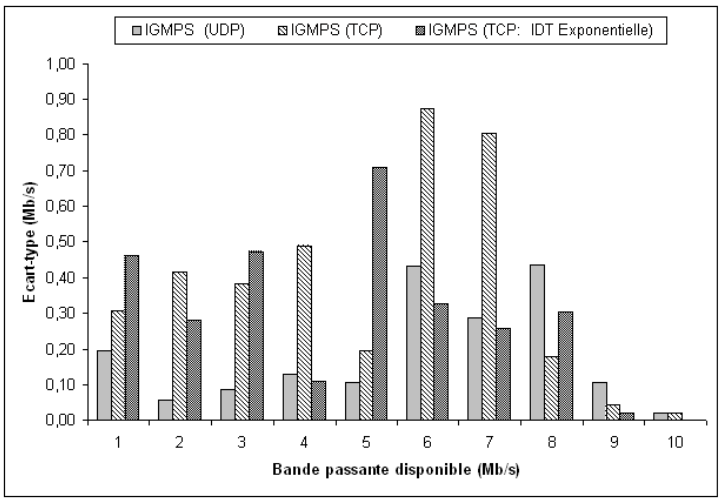


FIG. 3.9 – Ecarts-types des résultats de mesures d'IGMPS dans les différents scénarios.

temps très court, elle est l'une des caractéristiques définissant la fidélité d'un outil (voir Annexe A). Les écarts-types des mesures obtenues nous montrent qu'IGMPS est caractérisé par une faible répétabilité lors de deuxième et troisième scénarios (trafic concurrent du type TCP), alors que dans le premier scénario où le trafic concurrent est du type UDP, l'outil IGMPS montre une très grande répétabilité. Ceci signifie que pour avoir les meilleurs résultats possibles avec un trafic concurrent de type TCP, notre outil a besoin d'un grand nombre de sessions de mesures pour la même valeur de la bande passante disponible afin de réduire les erreurs aléatoires alors que pour un trafic concurrent de type UDP quelques sessions de mesures suffisent pour obtenir des résultats satisfaisants.

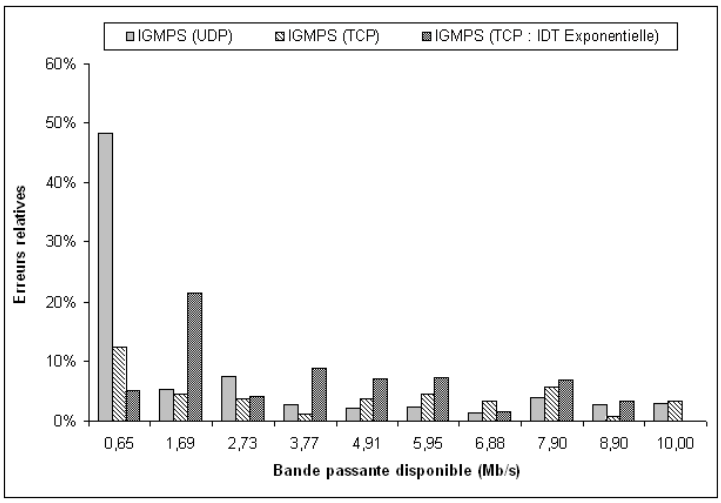


FIG. 3.10 – Erreurs relatives des mesures d'IGMPS dans les différents scénarios.

La figure 3.10 présente les erreurs relatives des résultats obtenus pour chaque scénario. Ces dernières sont définies par :

$$rel\_err = \frac{|abw\_exp - abw\_est|}{abw\_exp}$$

où *abw_exp* est la bande passante disponible réelle et *abw_est* est la bande passante disponible mesurée par IGMPS.

Cette figure (3.10) montre que dans le premier scénario (UDP), à l'exception de la première mesure, IGMPS présente des erreurs relatives faibles qui n'excèdent pas 10% et pour la plupart des autres mesures les résultats montrent moins de 5% d'erreurs relatives. En effet, IGMPS est très précis quand le taux d'utilisation du goulet d'étranglement est inférieur à 70%. En revanche, quand ce taux d'utilisation est supérieur à 70%, IGMPS obtient des mesures légèrement moins précises et qui sont surestimées d'environ 48% quand le taux d'utilisation du lien dépasse les 90%.

Dans le deuxième scénario (Trafic concurrent périodique de type TCP), IGMPS obtient des résultats beaucoup plus précis et ceci quelque soit le taux d'utilisation du lien bottleneck mais comme nous l'avons vu précédemment, ceci est vrai sous la condition que le nombre de session de mesures soit assez élevé pour éliminer les erreurs aléatoires. Dans ce scénario IGMPS obtient des résultats avec des erreurs relatives inférieures à 5% dans la plupart des cas. En ce qui concerne le troisième scénario (trafic concurrent de type TCP avec des IDT exponentiellement distribués), IGMPS est légèrement moins précis, il présente des erreurs relatives proches de 10% dans presque tous les cas. Pour résumer, les mesures d'IGMPS sont plus précises quand le trafic concurrent est périodique et elles sont beaucoup plus répétables quand le trafic concurrent est de type UDP.

En utilisant la même plateforme, nous avons procédé au test d'IGMPS avec un chemin de bout en bout d'une capacité de 1Mb/s puis d'une capacité de 100Mb/s pour compléter l'analyse de performances de notre outil. En augmentant le débit du trafic concurrent de 0 à 1 Mb/s (respectivement de 0 à 100Mb/s) avec un incrément de 100Kb/s (respectivement de 5 Mb/s) à chaque session de mesure, on fait varier la valeur de la bande passante disponible de 1 à 0 Mb/s (respectivement de 100 à 0 Mb/s). Les résultats obtenus montrent qu'IGMPS offre les mêmes performances que les autres scénarios précédents, il présente des résultats très précis et réagit rapidement aux variations du trafic concurrent sauf pour les bandes passantes disponibles étroites où il surestime légèrement les mesures.

Pour résoudre le problème des erreurs de mesures d'IGMPS quand le taux d'utilisation de lien bottleneck est élevé et améliorer ainsi la précision de cet outil, nous avons proposé un algorithme qui calcule le paramètre (mis à 0 dans l'équation 3.12) que nous soupçonnons être à l'origine de cette imprécision. Cet algorithme est présenté dans la section suivante.

3.4.3 Amélioration de la précision d'IGMPS

Dans l'équation 3.12 définie dans la paragraphe 3.2, nous avons mis le paramètre ϵ égal à 0 afin de faciliter les calculs et de simplifier la formule de la bande passante disponible. Cependant, après les résultats préliminaires nous avons constaté (dans le cas d'un trafic concurrent périodique de type UDP) une imprécision d'IGMPS quand le taux d'utilisation du goulet d'étranglement est élevé. Nous soupçonnons le fait d'avoir mis $\epsilon = 0$ être à l'origine de cette imprécision.

En effet quand le taux d'utilisation du goulet d'étranglement est élevé, le premier paquet de la paire risque de trouver du trafic concurrent à son arrivée à la file d'attente ce qui fait que ce paquet doit attendre un temps ϵ non négligeable. En revanche quand le taux d'utilisation du goulet est faible, le premier paquet a plus de chance de ne trouver aucun paquet du trafic concurrent dans la file d'attente, ce qui rend l'hypothèse $\epsilon = 0$ beaucoup plus réaliste.

En considérant le cas ou $\epsilon > 0$, l'équation 3.12 s'écrit comme suit :

$$q_b^k = \epsilon + \left(\frac{s^{k-1}}{C}\right) + (\Delta_{out} - \Delta_{in}) \tag{3.18}$$

Le modèle général définissant la bande passante disponible s'écrira alors :

$$A = \frac{s^{k-1}\left(2t_0^k + t_n^{k-1} - 2t_0^{k-1} - t_n^k\right)}{\left(t_0^k - t_0^{k-1}\right)\left[\frac{\left(s^k - s^{k-1}\right)}{C^{b-1}} - \frac{s^{k-1}}{C} - \left(t_0^k - t_n^{k-1}\right) + \epsilon\right]} \tag{3.19}$$

Pour estimer la bande passante disponible, IGMPS dans sa version améliorée, implémente l'équation 3.19 obtenue en mettant ($S^k = S^{k-1} = S$) dans l'équation 3.18 :

$$A = \frac{S\left(2\Delta_{in} - \Delta_{out}\right)}{(\Delta_{in})\left[\left(\frac{S}{C}\right) + \Delta_{out} + \epsilon\right]} \tag{3.20}$$

Le paramètre ϵ dépend du trafic concurrent et afin de le définir, nous avons proposé une méthode expérimentale basée sur quelques étapes données dans l'algorithme suivant :

1. Calculer la bande passante disponible (abw) en mettant ϵ égal à 0. cette étape permet d'avoir une idée sur la quantité du trafic concurrent.

2. Si $\left(abw > \frac{C}{3}\right)$ alors

   ⊙ mettre ϵ égal à 0.
   ⊙ la bande passante disponible est alors calculée en utilisant l'équation 3.17.

Sinon

⊙ $\epsilon = \alpha\,(\Delta_{out} - \Delta_{in})\left(1 - \frac{abw}{C}\right)$ where $\alpha \in [0, 1]$

⊙ Si $(abw > 0.15C)$ alors

$\alpha = 0.1$

La bande passante est alors donnée par :

$$A = \frac{S\,(2\Delta_{in} - \Delta_{out})}{(\Delta_{in})\left[\left(\frac{S}{C}\right) + \Delta_{out} + (0.1)\,(\Delta_{out} - \Delta_{in})\left(1 - \frac{abw}{C}\right)\right]}$$

⊙ Sinon

$\alpha = 1$

La bande passante disponible est alors calculée par :

$$A = \frac{S\,(2\Delta_{in} - \Delta_{out})}{(\Delta_{in})\left[\left(\frac{S}{C}\right) + \Delta_{out} + (\Delta_{out} - \Delta_{in})\left(1 - \frac{abw}{C}\right)\right]}$$

⊙ Finsi

Finsi

La figure 3.11 présente l'erreur relative obtenue avec IGMPS après l'implémentation de cet algorithme. Cette figure montre que quand le taux d'utilisation du goulet d'étranglement est élevé, IGMPS dans sa deuxième version offre une meilleure précision par rapport à la première. Par exemple, pour la première mesure, la version initiale d'IGMPS présente une erreur relative égale à 49%, en revanche, dans sa version améliorée, IGMPS présente une erreur relative égale à 29% et dans les autres cas, cette version présente une erreur de 3% au lieu de 5%.

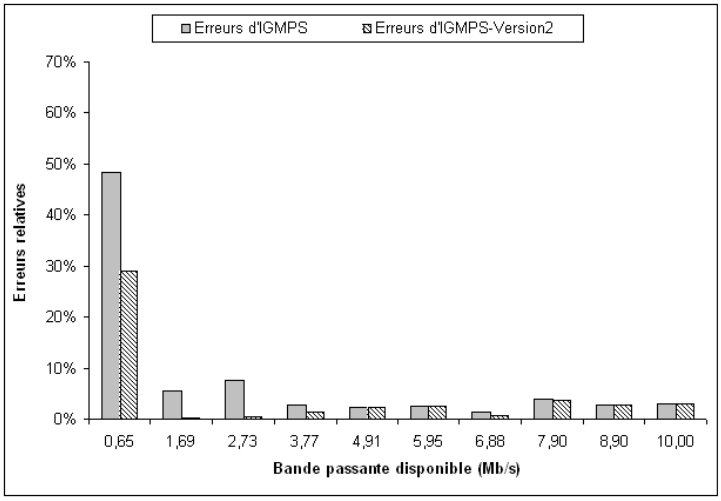


FIG. 3.11 – Comparaison des erreurs relatives des mesures d'IGMPS dans sa première version et sa version améliorée.

3.4.4 Intrusivité et temps de réponse d'IGMPS

Dans l'évaluation de performances d'IGMPS, nous avons considéré deux paramètres importants qui sont l'intrusivité et le temps de convergence d'IGMPS (dans un souci d'éviter la répétition nous utilisons aussi dans ce manuscrit le terme temps de réponse pour désigner ce dernier paramètre).

Afin de mieux montrer les performances d'IGMPS par rapport à ces deux paramètres, nous avons comparé la charge réseau induite par ce dernier et son temps de convergence à ceux de Pathload. Pour rappel, Pathload est un outil qui utilise la technique de l'auto-congestion pour mesurer la bande passante disponible ; son temps de réponse et son trafic de mesure peuvent être dans certains cas excessivement élevés.

La bande passante disponible est un paramètre qui varie rapidement dans le temps, il est alors essentiel de la mesurer aussi rapidement que possible. Pour évaluer les performances d'IGMPS nous avons effectué 30 sessions de mesures pour chaque valeur de la bande passante disponible sur le chemin de bout en bout considéré. Le temps de réponse de notre outil est donc calculé comme étant la moyenne des 30 temps de réponse obtenus pour chaque session de mesure. Ce dernier est d'environ 10 secondes et il est présenté dans la figure 3.12.

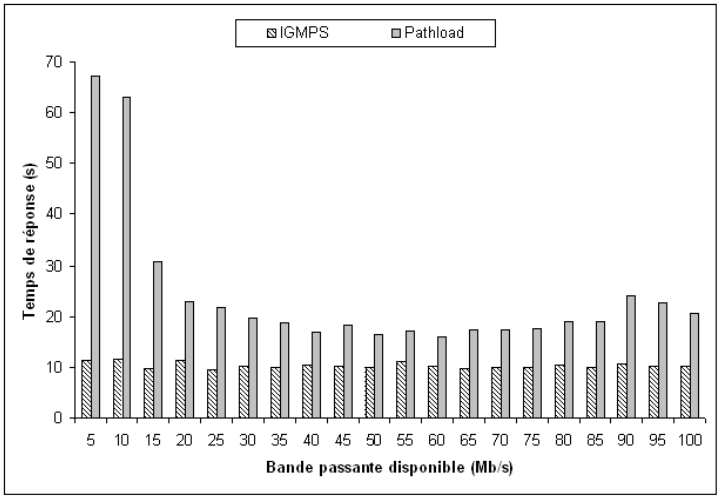


FIG. 3.12 – Temps de réponse d'IGMPS comparé à celui de Pathload.

Par contre, il est à signaler que ce temps de réponse dépend du nombre de paires de paquets sondes utilisées pour sonder le chemin de bout en bout. Par exemple, le temps de réponse obtenu dans nos expérimentations (10 secondes) correspond au temps nécessaire à l'envoi de 100 paires de paquets sondes. L'utilisation de 10 paires de paquets nécessitera moins de temps mais comme la valeur de la bande passante disponible calculée par IGMPS est la moyenne des valeurs instantanées de la bande passante disponible calculée par chaque paire, la diminution du nombre de paires dans une séquence de mesure augmentera les erreurs aléatoires et donnera des mesures

beaucoup moins précises (voir paragraphe 3.7).

Cette figure montre que dans certains cas IGMPS est 7 fois plus rapide que Pathload. Cela confirme la supériorité de la technique de la paire de paquets sur les techniques à auto-congestion.

L'intrusivité quant à elle, est définie comme étant le rapport de la quantité du trafic généré par IGMPS à la capacité du chemin. Le trafic de mesure généré par IGMPS dépend lui aussi de la taille des paquets sondes et du nombre de paires utilisées dans une séquence de mesure. Par défaut IGMPS utilise 100 paires de paquets sondes et la taille de chaque paquet est de 972 octets. La figure 3.13 montre qu'IGMPS génère un trafic de mesure faible et constant dont la moyenne est d'environ 250kb/s. Comparé à IGMPS, l'outil Pathload génère un trafic de mesure beaucoup plus important qui est dans certains cas 12 fois plus élevé que la charge de trafic induite par IGMPS. Cet intrusivité est due à la technique SLoPS [Dov03a] utilisée par pathload et qui consiste à saturer le chemin de bout en bout pour en extraire ses caractéristiques.

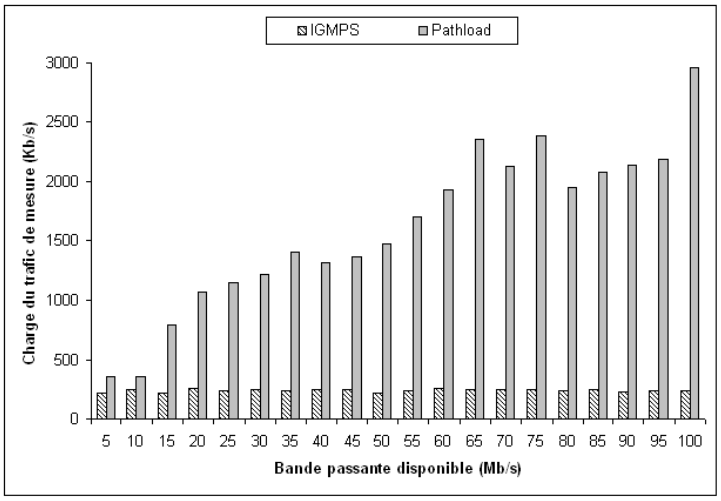


FIG. 3.13 – Charge de trafic de mesure d'IGMPS comparée à celle de Pathload.

La comparaison des performances de ces deux outils de mesure en terme d'intrusivité et de temps de convergence montre qu'IGMPS est plus rapide et moins intrusif que Pathload.

3.5 Les paramètres de performance d'IGMPS

Dans ce paragraphe nous étudierons l'impact de la variation de certains paramètres de configuration d'IGMPS tels que la taille des paquets sondes et le nombre de paires de paquets dans une séquence de mesure sur la précision des résultats de mesures de la bande passante disponible fournis par notre outil.

3.5.1 La taille des paquets sondes

Dans [Dov01], Dovrolis & Al ont signalé l'impact des tailles de paquets sondes sur les performances des outils de mesure. Nous avons étudié cet impact sur la précision de l'outil IGMPS. Pour ce faire, nous avons effectué des mesures de la bande passante disponible sur la même plateforme définie précédemment en faisant varier la taille des paquets sondes d'IGMPS entre 100 et 1500 octets. Pour chaque taille de paquets dans cet intervalle (multiple de 100), nous avons fait varier la valeur de la bande passante disponible entre 0 et 100 Mb/s avec un incrément de 1 Mb/s et nous avons effectué 30 mesures pour chaque valeur de cet intervalle en utilisant un trafic concurrent périodique de type UDP dont la taille des paquets est de 972 octets. Les résultats de mesure de la bande passante disponible sont présentés dans la figure 3.15. Dans cette figure l'axe des abscisses représente la taille des paquets sondes, l'axe des ordonnées représente la valeur réelle de la bande passante disponible et l'axe z représente la valeur mesurée de la bande passante disponible en utilisant IGMPS.

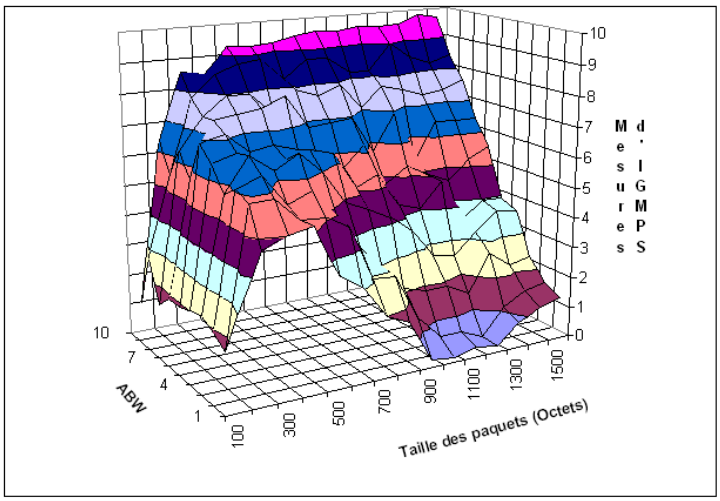


FIG. 3.14 – Effet de la taille des paquets sondes sur les mesures d'IGMPS avec un trafic concurrent dont la taille des paquets est de 972 octets .

La figure 3.14 montre que quand le taux d'utilisation de lien bottleneck est élevé et que la taille des paquet sondes est entre 100 et 700 octets, IGMPS surestime largement la bande passante disponible. En revanche, ce dernier la sous-estime quand la taille des paquets est entre 1000 et 1300 octets. Par contre, quand le taux d'utilisation du lien bottleneck est faible, la précision des mesures augmente d'une manière presque proportionnelle à la taille des paquets sondes. Dans ces deux cas (taux d'utilisation du bottleneck élevé ou faible), nous avons constaté que quand la taille des paquets est entre 800 et 1000 octets ou entre 1400 et 1500 octets, IGMPS fournit des mesures de la bande passante disponible avec une meilleure précision.

En effet, comme nous l'avons expliqué dans le paragraphe 3.3.2, la dispersion inter-paquets est le délai entre l'instant de départ du premier paquet et l'instant de départ du deuxième pa-

quet. Cette dispersion dépend essentiellement de deux paramètres. Le premier paramètre est le délai entre la fin de transmission du dernier bit du premier paquet et le début de transmission du premier bit du deuxième paquet. Le deuxième paramètre est la taille des paquets sondes (dans IGMPS le premier et le deuxième paquets sont de même taille).

Le premier paramètre cité ci-dessus, correspond au temps nécessaire au lien bottleneck pour transmettre 1500 octets de données, il dépend principalement de la capacité de ce dernier. En effet, plus la capacité du lien bottleneck est élevée plus ce temps est faible. Donc avec des paquets sondes de petites tailles ou un lien botteleneck avec une capacité élevée, nous obtiendrons forcément des dispersions inter-paquets trop réduites.

Dans nos expérimentations, nous avons fixé la capacité du bottleneck à 10Mb/s tout au long des tests. Par conséquence , la dispersion inter-paquets ne dépend dans ce cas que de la taille des paquets sondes. Avec des paquets sondes de petites tailles, les dispersions inter-paquets résultantes seront faibles donc plus difficiles à mesurer avec exactitude et plus sensibles aux erreurs. En effet, dans ce cas de figure, IGMPS a tendance à envoyer les paquets de la paire avec une dispersion supérieure à ce qu'elle devrait être ($\Delta_{in} > \Delta_{out}$). Plus cette dispersion est élevée, plus la quantité du trafic concurrent qui s'infiltre entre les paquets de la paire est important. Le récepteur calcule la valeur de la bande passante disponible en utilisant la formule 3.17, cette dernière prend en considération le paramètre Δ_{in} au lieu de Δ'_{in}. La non prise en compte des éventuelles erreurs dans la mesure de la dispersion initiale des paquets dans IGMPS conduit à sous estimer la quantité du trafic concurrent qui s'infiltre entre les paquets de la paire et par conséquent à surestimer la valeur de la bande passante disponible dans le chemin de bout en bout considéré. L'effet de ces erreurs s'accentue quand la capacité du lien bottleneck est élevée, ces dernières affectent considérablement la précision des mesures d'IGMPS.

Les résultats présentés dans la figure 3.15 sont obtenus en utilisant un trafic concurrent de type UDP avec une taille des paquets de 972 octets. Cette figure montre que les résultats de mesures d'IGMPS sont très précis quand la taille des paquets sondes est entre 900 et 1000 octets. En se basant sur ces observation, nous pensons que quand la taille des paquets sondes est suffisamment proche de celle des paquets du trafic concurrent, les paires de paquets interagissent mieux avec ce dernier (trafic concurrent) et par conséquent IGMPS offre des mesures très précises. Afin de confirmer cette hypothèse, d'autres expérimentations ont été réalisées en utilisant un trafic concurrent avec des paquets de plus petite taille. Dans ce scénario le trafic concurrent injecté dans le réseau est un trafic UDP avec des paquets de 460 octets. Les résultats des mesures d'IGMPS pour ce scénario sont illustrés dans la figure 3.16.

Quand le taux d'utilisation du lien bottleneck est élevé, IGMPS surestime la valeur de la bande passante disponible quand il utilise des paquets sondes de petites tailles (entre 100 et 400

octets) et il sous estime cette dernière quand la taille des paquets sondes est entre 700 et 1200 octets. En revanche, quand cette taille est entre 400 et 500 octets (donc très proche de la taille des paquets du trafic concurrent), IGMPS présente des résultats de mesure assez précis. La figure 3.16 montre aussi que quand la taille des paquets sondes est entre 1400 et 1500 octets IGMPS mesure la bande passante disponible avec une bonne précision. Dans ces expérimentations les tests sont réalisés en variant la taille des paquets sondes et en utilisant un trafic concurrent avec une taille de paquets fixe.

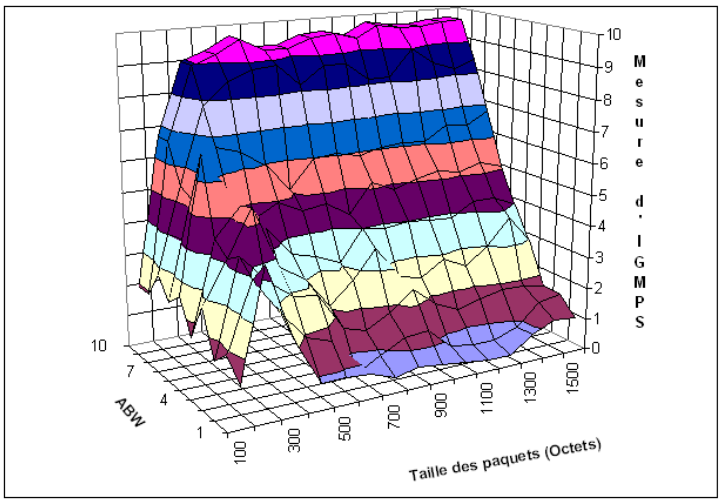


FIG. 3.15 – Effet de la taille des paquets sondes sur les mesures d'IGMPS avec un trafic concurrent dont la taille des paquets est de 460 octets .

Pour mieux mettre en évidence la relation entre la taille des paquets sondes et celle des paquet du trafic concurrent, nous avons procédé à des expérimentations en faisant varier, à chaque session de mesure, la taille des paquets du trafic concurrent et la taille des paquets sondes ainsi que la valeur de la bande passante disponible sur le chemin de bout en bout. En effet, nous avons considéré trois valeurs différentes de la bande passante disponible correspondant à 80%, 50% et 20% de taux d'utilisation du lien bottleneck. Les résultats obtenus pour chaque scénario sont présentés dans les figures 3.17, 3.18 et 3.19 respectivement. Nous avons constaté que dans chacun de ces scénarios la valeur de la bande passante disponible est sous estimée quand les paquets du trafic concurrent sont de petite taille et qu'en revanche cette dernière est surestimée quand les paquets sondes sont assez petits.

Donc IGMPS n'est pas capable de mesurer avec précision la bande passante disponible quand les paquets du trafic concurrent sont petits. Quand la taille des paquets sondes est petite, ces derniers ne parviennent pas vraiment à interagir avec les paquets du trafic concurrent quelque soit leur taille. Les mesures sont imprécises dans ce cas de figure aussi. Cette première constatation confirme déjà nos premiers résultats (figures 3.15 et 3.16). La plus importante observation à partir des figures 3.17, 3.18 et 3.19 est l'apparition dans les courbes représentant les mesures obtenues par IGMPS d'une ligne diagonale presque parallèle au plan (x, y) et dont une projec-

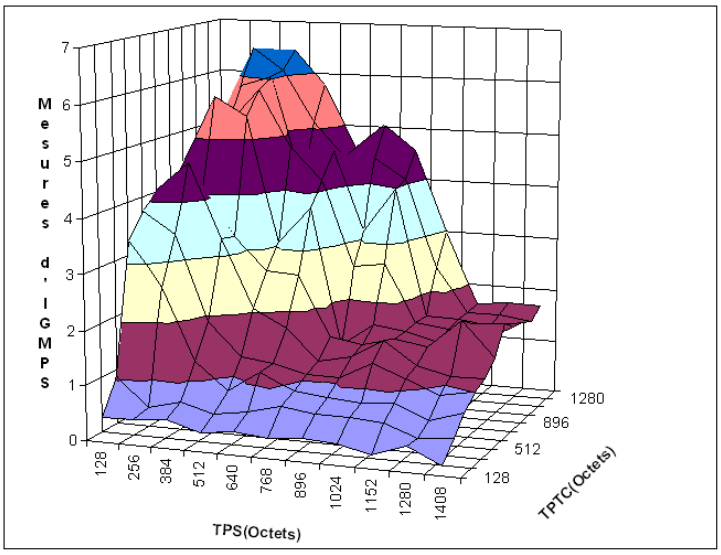


FIG. 3.16 – Effet de la taille des paquets sondes sur IGMPS avec une bande passante disponible de 2 Mb/s.

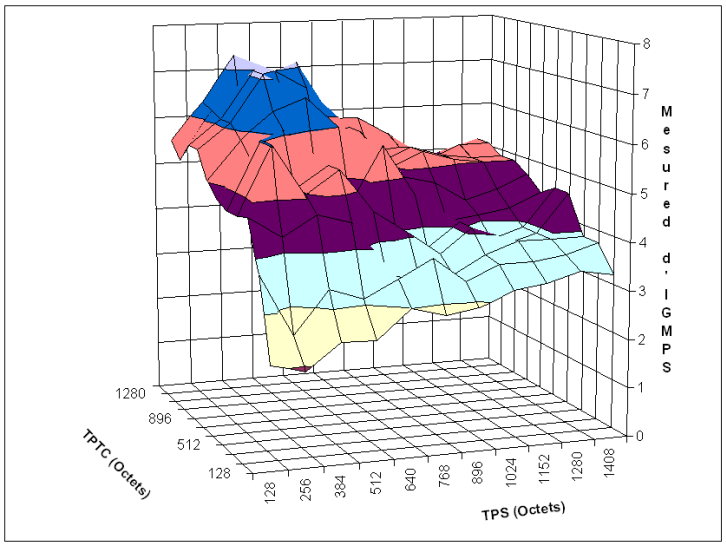


FIG. 3.17 – Effet de la taille des paquets sondes sur IGMPS avec une bande passante disponible de 5 Mb/s.

tion sur ce plan est une droite d'équation $y = x$ (avec x représentant la taille des paquets sondes et y la taille des paquets du trafic concurrent). Nous avons constaté que les mesures représentées par ces lignes horizontales sont beaucoup plus précises que les autres. Ceci signifie que quand les tailles des paquets sondes sont égales ou suffisamment proches de celles du trafic concurrent, IGMPS offre les mesures les plus précises possibles.

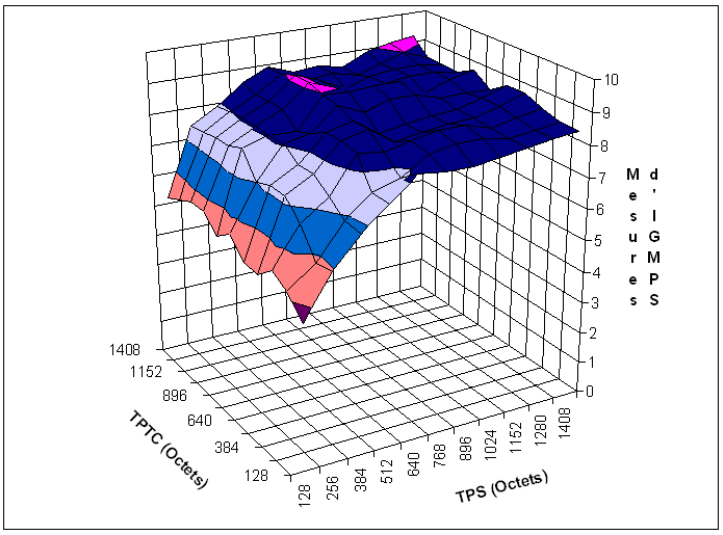


FIG. 3.18 – Effet de la taille des paquets sondes sur IGMPS avec une bande passante disponible de 8 Mb/s.

Proposition1

Pour mesurer la bande passante disponible de bout en bout avec précision en utilisant IGMPS, il faut utiliser des paquets sondes dont la taille est égale ou suffisamment proche de celle des paquets du trafic concurrent.

Afin de vérifier que la *proposition 1* peut se généraliser à d'autres outils de mesure de la bande passante disponible nous avons effectué des expérimentations en utilisant deux autres outils basés sur la technique de paire de paquets à savoir Spruce et IGI.

Comme IGMPS, Spruce est basé sur la technique de la paire de paquets. En revanche, IGI utilise un train de paquets et considère que chaque deux paquets consécutifs constituent une paire. Identiquement aux expérimentations décrites dans la première partie de cette section, nous testons Spruce et IGI en utilisant un trafic concurrent avec une taille de paquets fixe (972 octets) et nous faisons varier la taille des paquets sondes ainsi que la valeur de la bande passante disponible de bout en bout à chaque session de mesure.

Les résultats des mesures sont donnés dans la figure 3.20 pour Spruce et la figure 3.21 pour IGI . La figure 3.20 montre que les résultats de Spruce sont presque identiques à ceux d'IGMPS. En effet, cet outil surestime largement la bande passante disponible quand la taille des paquets sondes est petite et obtient des résultats assez précis quand cette dernière est aux alentours de 1000 octets et de 1500 octets.

En ce qui concerne IGI, la figure 3.21 montre que malgré l'imprécision de ses résultats dans la plus part des cas, ce dernier offre des mesures assez proches de la valeur de la bande passante

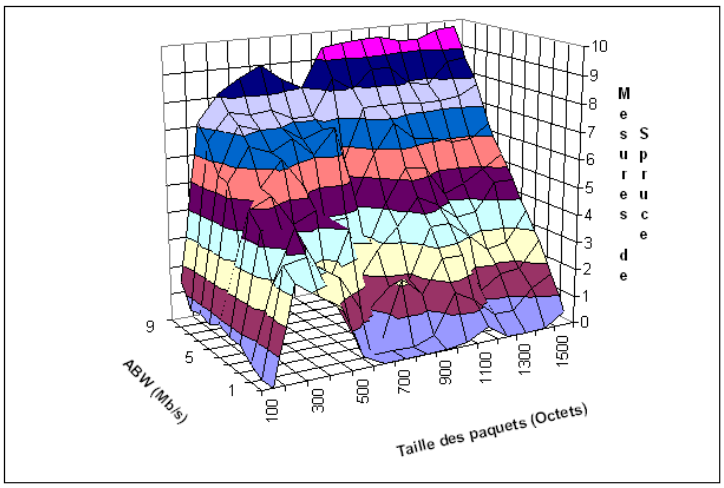


FIG. 3.19 – Effet de la taille des paquets sondes sur les mesures de Spruce.

disponible réelle quand la taille des paquets sondes sont proches de 1000 et 1500 octets. En d'autre mots, les résultats de ces expérimentations montrent que quand la taille des paquets sondes est égale ou proche de la taille des paquets du trafic concurrent les mesures de Spruce et d'IGI sont les plus précises. Ceci confirme donc les résultats obtenus dans nos premières expérimentations et généralise donc la *proposition 1* à d'autres outils utilisant les techniques à dispersion de paquets (paires de paquets et trains de paquets) tels que Spruce et IGI.

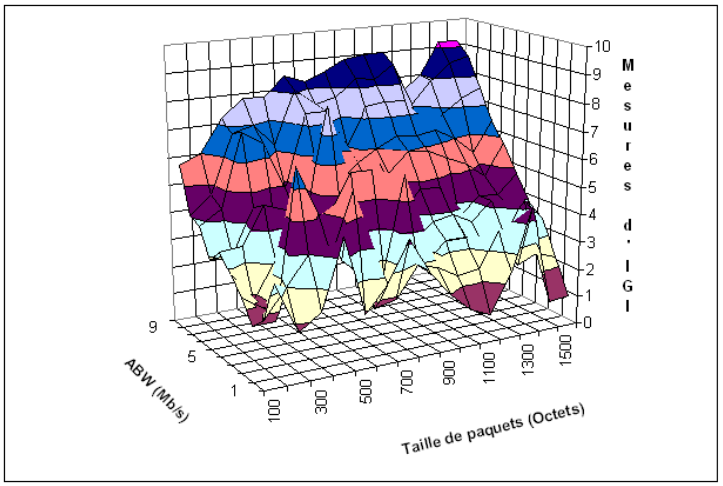


FIG. 3.20 – Effet de la taille des paquets sondes sur les mesures dIGI.

Dans IGMPS, la taille par défaut des paquets sondes est égale au MTU (maximum Transmission Unit) et qui est de 1500 octets dans notre cas (Ethernet). Le choix d'envoyer des paquets sondes de grande taille (tout en évitant la fragmentation) permet d'avoir des dispersions inter-paquet assez larges facilitant ainsi la mesure des délais et réduisant les risques d'erreur d'estampillage des paquets au niveau du récepteur. Cependant, les paquets du trafic Internet

ont des tailles variables. En effet, les auteurs de l'étude menée dans [Cal98] ont montré que les paquets de petites tailles sont prédominants et que les dispersions de leurs tailles présentent des pics au niveau de 44, 552, 576 et 1500 octets. Les paquets de petites tailles (40-44 octets) correspondent généralement aux accusés de réception de TCP (ACK), aux paquets de contrôle de TCP (SYN, FIN et RST) et à certains paquets générés lors de sessions telnet. Par ailleurs, les équipements réseaux qui n'implémentent pas le protocole *Path MTU-Discovery* [RFC1191] ou en cas d'echec de la négociation du MSS (taille maximale de segment TCP) entre deux équipements réseaux distants, la taille par défaut du MSS utilisée est soit 512 ou 536 octets générant ainsi des paquets de taille 552 octets ou 576 octets. Finalement, les tailles de paquets de 1500 octets correspondent au MTU caractérisant les liens Ethernet.

En nous basant sur ces résultats et en prenant en considération notre proposition, il est nécessaire d'améliorer le fonctionnement d'IGMPS pour mesurer avec précision la bande passante disponible dans les conditions réelles de l'Internet. En effet, la taille des paquets du trafic Internet est variable il faut donc faire varier dynamiquement la taille des paquets sondes d'IGMPS autour des valeurs citées précédemment à savoir 40, 560 et 1500 octets.

Etant donnée qu'IGMPS utilise par défaut k=100 paires de paquets sondes, il serait judicieux d'utiliser, par exemple, $k/4$ paires avec une taille de 400 octets, $k/4$ paires de 560 octets et finalement $k/2$ paires avec une taille de 1500 octets afin d'interagir au mieux avec les paquets du trafic concurrent. Comme les mesures d'IGMPS ne sont pas précises quand la taille des paquets sondes est trop petite, il faut éviter d'utiliser des tailles de paquets inférieurs à 400 octets. En effet, ces dernières produisent des dispersions assez faibles rendant les opérations d'estampillage des paquets au niveau du récepteur beaucoup plus difficiles, engendrant ainsi des imprécisions dans la mesure de la bande passante disponible de bout en bout.

En résumé, *l'analyse des résultats des expérimentations effectuées en utilisant les différents outils de mesure ont révélé que pour mesurer la bande passante disponible avec précision en utilisant des techniques à dispersion de paquets, il est nécessaire de sonder le chemin de bout en bout avec un trafic sonde dont la taille des paquets est égale ou suffisamment proche de la taille des paquets du trafic concurrent. Afin d'obtenir de bonnes performances dans les conditions réelles de l'Internet, il est nécessaire de varier dynamiquement les tailles des paquets sondes pour les faire correspondre aux tailles des paquet du trafic Internet.*

Les résultats présentés dans cette étude sont expérimentaux, une étude beaucoup plus formelle est nécessaire afin de montrer l'interaction entre les paquets sondes et le trafic concurrent ainsi que la relation entre leurs tailles respectives. Cette étude formelle est présentée dans le chapitre suivant (chapitre 4), elle définit un modèle stochastique pour la technique de la paire de paquets et établie une relation entre les dispersions initiales des paquets et leurs dispersions finales. En se basant sur ce modèle, nous démontrons en utilisant quelques tests expérimentaux

que la *proposition 1* est valide et pourrait être généralisée à tous les outils de mesure utilisant la technique de la paire de paquets.

3.5.2 Nombre de paires d'une séquence de mesure

Afin d'analyser l'effet du nombre de paires de paquets d'une séquence de mesure sur la précision des estimations de la bande passante disponible en utilisant IGMPS, nous procédons à un ensemble de tests (sur la même plateforme décrite précédemment) en faisant varier de 10 à 100 le nombre de paires utilisées pour sonder le chemin de bout en bout. Les résultats obtenus sont présentés sur la figure 3.22.

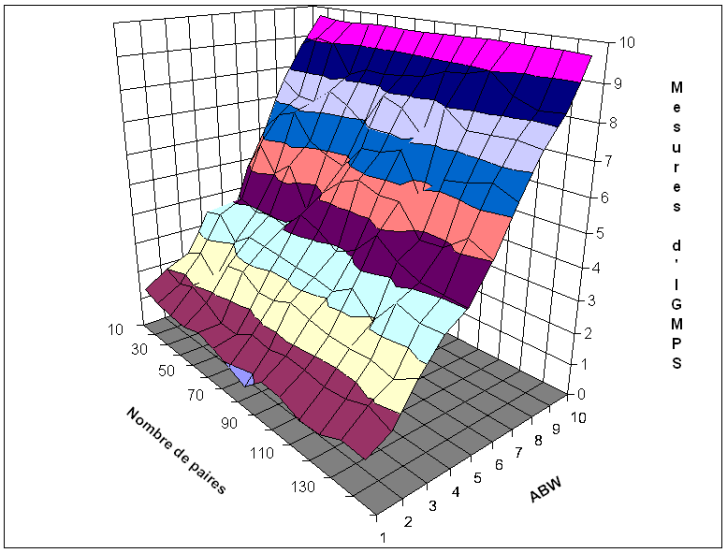


FIG. 3.21 – Effet du nombre de paires sondes sur la précision d'IGMPS

Cette figure montre que dans les conditions de nos expérimentations, quelque soit le taux d'utilisation du lien bottleneck, la variation du nombre de paires dans la séquence de mesure n'affecte pas beaucoup les mesures obtenues. Cependant, il est recommandé d'utiliser un nombre suffisant de paires de paquets sondes lorsqu'il s'agit d'utiliser IGMPS dans les conditions réelles d'Internet, ceci dans le but de réduire d'éventuelles erreurs systématiques. Par ailleurs, les figures 3.23 et 3.24 montrent que la variation du nombre de paires a un effet considérable sur les coûts en charge réseau induite et en temps de réponse d'IGMPS. En effet, l'intrusivité d'IGMPS et son temps de convergence augmentent proportionnellement au nombre de paires de paquets sondes.

3.6 Comparaison entre IGMPS et Spruce

Dans le chapitre 2 nous avons analysé les différents outils de mesure de la bande passante disponible de bout en bout et nous les avons comparés. Dans les conditions de nos expérimentations, nous avons constaté que l'outil Spruce est celui qui offre les meilleures performances au

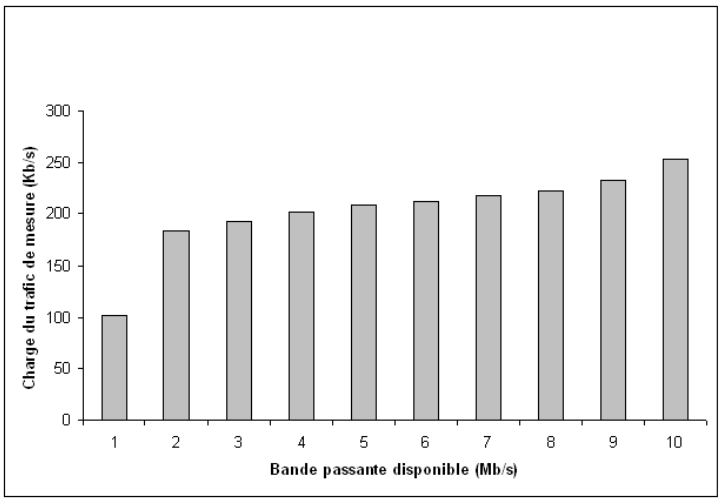


FIG. 3.22 – Effet du nombre de paires sondes sur la charge du trafic de mesure d'IGMPS

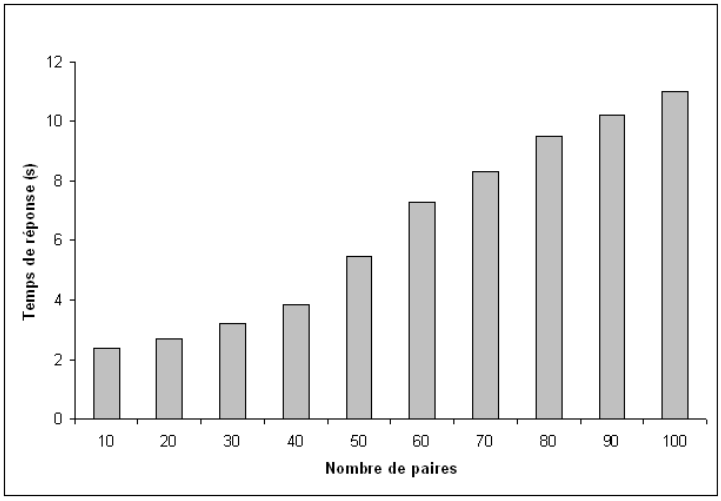


FIG. 3.23 – Effet du nombre de paires sondes sur le temps de réponse d'IGMPS

regard des critères que nous avons considéré (précision, temps de réponse et intrusivité). Les résultats ont montré que ce dernier est l'outil le plus précis, le plus rapide et l'un des outils les moins intrusifs. Afin de valider définitivement les performances de l'outil IGMPS, nous l'avons comparé à l'outil Spruce que nous considérons comme un outil référence dans le domaine des techniques de paires de paquets (voir chapitre 2).

Dans notre comparaison, nous avons utilisé la même plateforme de tests décrite dans la section 3.4.1 et nous avons fait varier la bande passante disponible de bout en bout entre 0 et 10 Mb/s avec un incrément de 1 Mb/s à chaque session de mesure. Les résultats obtenus avec IGMPS et Spruce sont représentés sur la figure 3.25. Les erreurs relatives qui entachent ces résultats sont illustrées dans la figure 3.26.

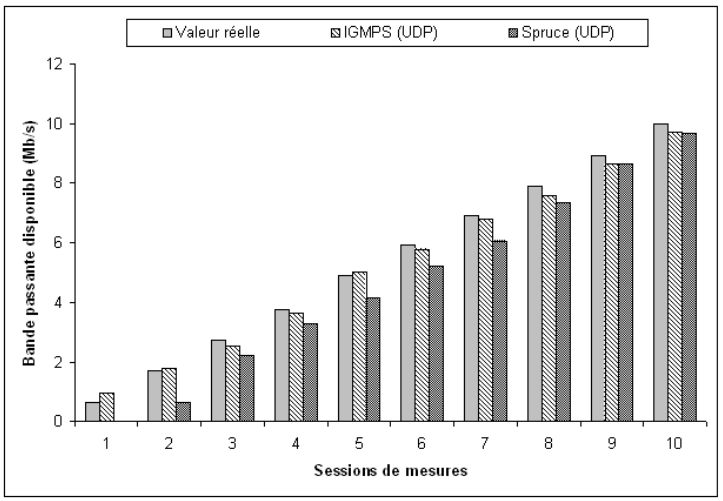


FIG. 3.24 – Comparaison des performances d'IGMPS à celles de Spruce

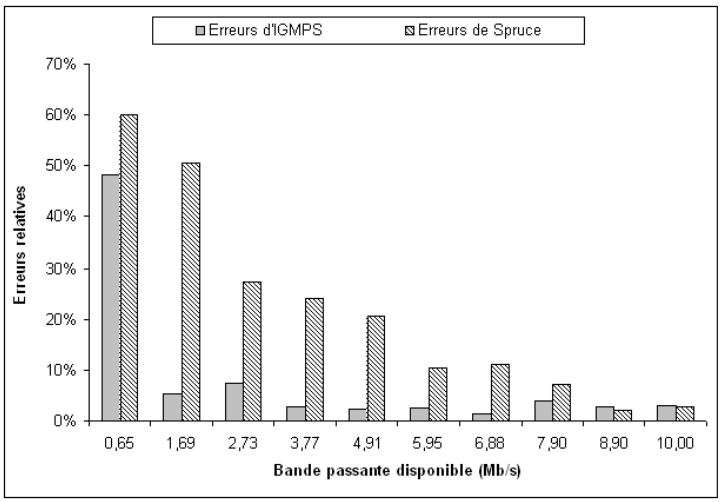


FIG. 3.25 – Comparison des erreurs relatives sur les mesures d'IGMPS et de Spruce

Les résultats montrent qu'IGMPS mesure la bande passante disponible avec une très grande précision et que son erreur relative est inférieure à 3% dans certains cas. Notre outil est plus précis que Spruce. En effet, bien que légèrement imprécis quand le taux d'utilisation du lien bottleneck est élevé, IGMPS offre une meilleure précision que Spruce lors de toutes les sessions de mesures tandis que leurs charges réseaux et leurs temps de convergence respectifs sont presque identiques et sont mesurés aux alentours de 10s et 250Kb/s respectivement.

3.7 Les limites de l'outil IGMPS

Les expérimentations ont montré que l'outil IGMPS offre de bonnes performance au regard des critères étudiés, il présente des résultats assez précis, un temps de réponse relativement court et un trafic de mesure de faible débit. Cependant, le modèle utilisé et la technique implémentée dans IGMPS montrent quelques faiblesse qu'il est honnête de signaler.

En effet IGMPS est un outil de mesure de bout en bout, il nécessite donc d'être installé au niveau de la station émettrice et la station réceptrice. L'accès à cette dernière n'est pas toujours possible pour un simple utilisateur et nécessite souvent d'avoir des privilèges pour pouvoir y accéder, ce qui pourrait donc limiter l'utilisation de notre outil. Cependant, cet inconvénient n'est pas spécifique à IGMPS mais il concerne tous les outils de mesures unidirectionnelles de bout en bout. Toutefois, l'utilisation du protocole OWAMP [Vei06] pourrait bien remédier à ce problème. Le deuxième inconvénient est lié directement au modèle proposé. Ce dernier suppose que tous les routeurs composant le chemin de bout en bout implémentent la politique de service FIFO. L'utilisation d'IGMPS dans un réseau sans fil n'est donc pas possible étant donné que ces derniers n'implémentent pas cette politique de service. Finalement le modèle proposé suppose que le trafic concurrent est fluide et que ce dernier varie lentement. Or cette hypothèse n'est pas vérifiée dans les conditions réelles de l'Internet. Pour remédier à ce problème , nous avons implémenté dans IGMPS un mécanisme de filtrage qui permet d'écarter les paires de paquets dont les dispersion inter-paquets finales sont inférieures ou supérieures à certains seuils (voir paragraphe 3.3.2).

3.8 Conclusion

Dans ce chapitre, nous avons proposé un nouveau modèle déterministe pour l'estimation de la bande passante disponible dans un chemin de bout en bout. Ce modèle est basé sur la technique de la paire de paquets et prend en compte la taille des paquets sondes. Nous avons implémenté ce modèle dans un nouvel outil de mesure appelé IGMPS dont nous avons évalué les performances et comparé à l'outil Spruce. Les résultats ont montré qu'IGMPS offre de meilleures performances pour un temps de réponse et un débit de mesure presque identiques à ceux de Spruce. Les différents tests effectués en utilisant IGMPS ont montré que l'introduction de la taille des paquets dans la formule du *Probe Gap Model* améliore considérablement la précision des mesures obtenues. En effet, nous avons constaté que quand la taille des paquets sondes est égale ou suffisamment proche de la taille des paquets du trafic concurrent, IGMPS offre les résultats les plus précis. Pour utiliser IGMPS dans les conditions réelles de l'Internet et afin d'améliorer sa précision, il est nécessaire de faire varier dynamiquement la taille des paquets sondes pour les faire correspondre aux tailles les plus fréquentes des paquets du trafic Internet.

Les principaux résultats obtenus dans ce chapitre sont expérimentaux et les solutions proposées pour améliorer la précision des mesures sont intuitives. Une étude beaucoup plus formelle

est nécessaire afin d'expliquer l'interaction entre les paquets sondes et le trafic concurrent ainsi que la relation entre leurs tailles respectives. Cette étude formelle est présentée dans le chapitre suivant, elle définit un modèle stochastique pour la technique de la paire de paquets et établit une relation entre les dispersions initiales des paquets et leurs dispersions finales. En se basant sur ce modèle, nous démontrons en utilisant quelques tests expérimentaux que les propositions faites dans ce chapitre sont valides et pourraient être généralisées à tous les outils utilisant la technique de la paire de paquets.

Chapitre 4

Modélisation stochastique de la technique de la paire de paquets

Résumé

Les principaux résultats obtenus dans le troisième chapitre sont expérimentaux. Une étude beaucoup plus formelle est nécessaire afin d'expliquer l'interaction entre les paquets sondes et le trafic concurrent ainsi que pour définir la relation entre leurs tailles respectives. La contribution apportée dans ce chapitre consiste donc en la définition d'un modèle stochastique pour la technique de la paire de paquets qui établit une relation entre les dispersions initiales des paquets et leurs dispersions finales. En se basant sur ce modèle, nous avons démontré en utilisant quelques tests expérimentaux que les propositions faites au chapitre précédent sont valides et pourraient être généralisées à tous les outils de mesure de la bande passante disponible utilisant la technique de la paire de paquets.

4.1 Introduction

Le but de cette section est de définir une relation entre la dispersion initiale et la dispersion finale d'une paire de paquets au niveau du bottleneck en considérant l'hypothèse que le trafic concurrent est de type poissonnien.

Cette hypothèse n'est certes pas exacte dans les conditions réelles de l'Internet. Les études menées sur la caractérisation de ce trafic ont montré que le modèle de Poisson ne peut pas représenter le trafic Internet. Les mesures sur ce trafic ont montré une dépendance longue du trafic. Le modèle Poissonnien classiquement utilisé pour modéliser les processus d'arrivée du trafic est incapable de représenter les rafales et les relations de dépendance qui existent entre les flux, les paquets et les pertes dans l'Internet.

La figure 4.1, introduite dans [Lar05] illustre l'écart à différentes échelles temporelles entre un trafic comportant un caractère auto-similaire (colonne de gauche) et un trafic poissonnien simulé (colonne de droite). Chaque graphique représente l'évolution du débit au cours du temps.

On peut noter que dans les deux cas, les courbes de trafic ne se lissent pas avec la même vitesse lorsque la granularité de l'observation augmente (les graphiques de la première ligne possèdent la granularité la plus fine et la granularité augmente avec les lignes qui suivent). En effet, bien que l'amplitude des oscillations décroît lorsque la granularité d'observation est plus importante, il est clair que pour une granularité d'observation importante (1 seconde par exemple), l'amplitude des oscillations du trafic Internet est plus importante que celle du trafic poissonnien. Cette particularité est le signe de la présence de LRD (Long Range Dependance) dans le trafic Internet. Cela signifie qu'il est indispensable de sur-dimensionner les liens et les tailles des files d'attente dans les routeurs pour prendre en compte cette variance. Cependant, dans la communauté scientifique, il existe des débats qui défendent la théorie du trafic poissonnien. Par exemple, les auteurs dans [Cao01] illustrent que le trafic Internet dans le réseau de cœur devient de plus en plus poissonnien au fur et à mesure de l'augmentation de la capacité des liens observés et de l'agrégation des trafics. Par ailleurs, d'autres observations dans le cadre du projet METROPOLIS [Owe06] rejoignent cette théorie.

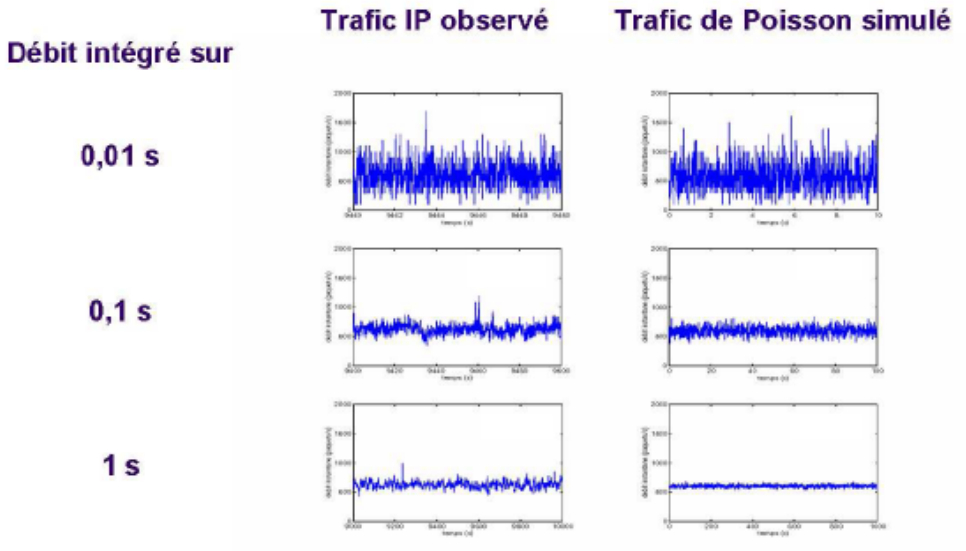


FIG. 4.1 – Comparaison entre les oscillations observables dans un trafic Internet et un trafic poissonnien.

Il est apparu que le trafic dans les réseaux de cœur suit un modèle poissonnien [Owe04b]. En effet, face à la croissance constante du trafic Internet et aux besoins de plus en plus forts des utilisateurs en termes de garantie de service et de performance, les opérateurs ont pour la plupart choisi de sur-dimensionner leur réseau d'un facteur allant généralement de 2,5 à 3 (et parfois plus). Ce choix a la propriété de repousser les phénomènes de congestion en bordure du réseau de l'opérateur. C'est donc dans les réseaux d'accès et de bordure que le trafic a les caractéristiques les plus complexes et les plus néfastes pour les performances globales du réseau [Lar05].

Dans la suite de ce chapitre, nous supposons que les équipements réseaux sont suffisamment

surdimensionnés de manière à pouvoir modéliser le trafic par un modèle poissonnien. En effet, un modèle poissonnien du trafic Internet nous permettra de modéliser la technique de la paire de paquets plus facilement en représentant le goulot d'étranglement par une file d'attente M/D/1 qui considère que le processus des arrivées du trafic est poissonnien et le temps de service d'un client (paquet) est constant. La considération d'un temps de service constant implique que nous avons considéré aussi que les tailles des paquets du trafic concurrent sont égales (or ces dernières suivent une distribution multimodale [CAI]). Pour éviter cette hypothèse, il aurait fallu utiliser la file $M^{[X]}/D/1$ qui est beaucoup plus difficile et complexe à étudier et nous avons donc choisi d'étudier un cas particulier de cette file qui est la file M/D/1 avec un trafic poissonnien à tailles de paquets égales.

Ce chapitre se base sur les travaux de Franx [Fra98] ainsi que ceux de Park & Al [Par05]. Le premier a étudié la file M/D/c en régime transitoire dont nous adaptons les résultats obtenus et nous les appliquons pour un système M/D/1. Les autres ont proposé une modélisation de la relation entre les dispersions initiales et les dispersions finales pour un train de paquets que nous adaptons et nous appliquons pour une seule paire de paquets.

Dans ce chapitre, nous expliquerons d'abord comment décrire le comportement d'une paire de paquets en utilisant le système M/D/1, puis en utilisant les résultats obtenus dans [Fra98] nous définirons la matrice d'état du système au régime transitoire. Le calcul du temps moyen de séjour dans la file d'attente, nous permettra de définir le temps d'attente du deuxième paquet de la paire dans le système à partir duquel sera dérivé le modèle stochastique des délais de la technique de la paire de paquets.

4.2 La file M/D/1

Le Système M/D/1 est formé d'une file FIFO à capacité illimitée et d'un seul serveur. Le processus d'arrivée des clients est supposé poissonnien de taux λ et le temps de service est constant. Comme nous l'avons mentionné précédemment, le trafic concurrent qui entre dans le système est supposé être poissonnien et les tailles de ces paquets sont supposées être égales pour avoir un temps de service constant (figure 4.2).

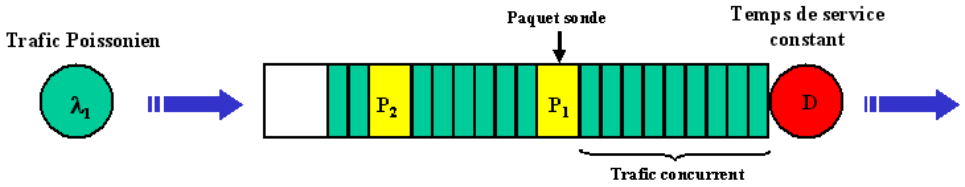


FIG. 4.2 – La file d'attente M/D/1

Nous supposons qu'à l'instant $t = t_0$ le premier paquet de la paire arrive au niveau de la file d'attente et que ce dernier trouve q_1 paquets dans la file et un paquet en service avec un temps de service restant d (contrairement au système M/M/1, il ne suffit pas de savoir qu'un client est

en service pour prédire quand ce service va se terminer. Il faut en plus savoir depuis combien de temps le service a commencé. Si par exemple le temps moyen de service est de 10 secondes et que le service à commencé depuis 8 secondes, ce dernier se terminera au bout de 2 secondes. Cette information est indispensable pour prédire l'évolution du système). S'il n y'a pas de paquet en service à l'instant $t = t_0$ alors $d = 0$. Nous supposons aussi qu'il reste dans la file d'attente q_2 paquets à l'arrivée du second paquets.

Soient W_1 et W_2 les temps d'attente respectifs du premier et du deuxième paquet dans la file (sans le service). On a le temps de service du premier paquet qui est égal à celui du deuxième paquet et est donné par $t_s = S/C$, où S est la taille des paquets de la paire et C la capacité du lien. La figure 4.3 représente l'état du système à l'instant $t = t_0$ correspondant à l' arrivée du premier paquet et à l'instant $t = t_0 + \Delta_{in}$ correspondant à l'arrivée du deuxième paquet de la paire.

A partir de cette figure il est possible de calculer la quantité $\Delta_{out} - \Delta_{in}$ telle que :

$$\Delta_{out} - \Delta_{in} = (W2 + S/c) - (W1 + S/c)$$

$$\Delta_{out} - \Delta_{in} = W2 - W1 \tag{4.1}$$

La taille des paquets du trafic concurrent est supposée être constante, elle est choisie égale à L. On suppose aussi que les paquets sondes ont une taille différente de longueur $S \geq L$ tel que S est multiple de L. (afin que la quantité S/L soit un entier).
L'idée est donc de considérer le système du point de vue du second paquet tel que W_2 est le temps d'attente de ce paquet dans la file M/D/1 en régime transitoire et le nombre de paquets dans le système à l'état initial est $N_0 = q_1 + S/L + 1$ (à l'état initial on a q_1 paquets dans la file plus le premier paquet sonde et un paquet en service).

En mettant $t_0 = 0$, le temps d'attente W_2 du second paquet sonde correspond au temps d'attente d'un client qui arrive dans le système M/D/1 à l'instant $t = \Delta_{in}$. Sous l'hypothèse $t_0 = 0$, il est possible de trouver une relation entre Δ_{in} et Δ_{out} en définissant le temps d'attente dans le système M/D/1 à l'instant $t = \Delta_{in}$ en régime transitoire (d'où l'idée d'utiliser la file M/D/1).

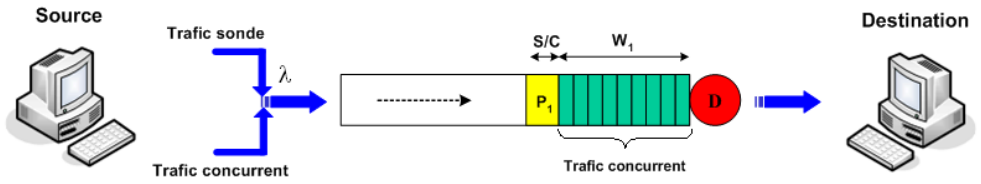
4.3 Concepts théoriques de la file M/D/1

4.3.1 Nombre moyen de paquets dans la file M/D/1

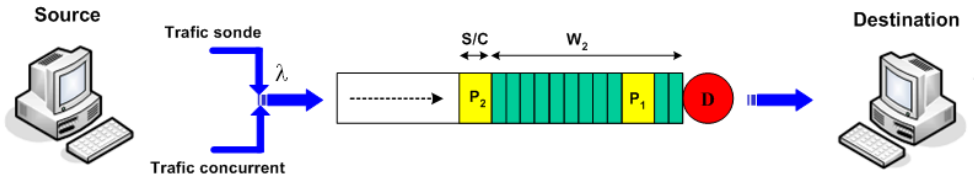
soit $\pi_j(t)$ la probabilité que le nombre de paquets dans le système à l'instant t est égal à j. $\pi_j(t)$ est définie comme suit :

Proposition 1

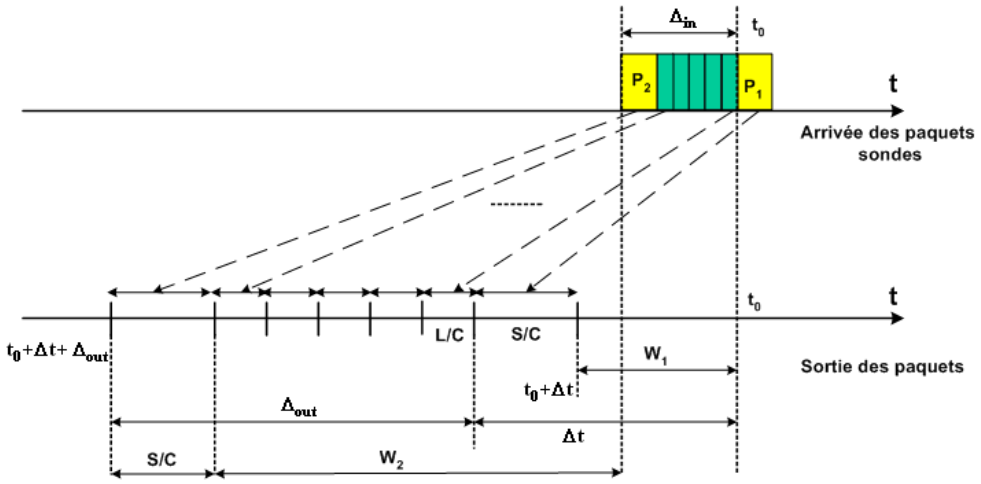
pour tout $t \in (0, D]$ tel que $D = L/C$ avec L la taille des paquets du trafic concurrent,



(a) A l'instant $t = t_0$



(b) A l'instant $t = t_0 + \Delta_{in}$



(c) Les instants de sortie des paquets sondes

FIG. 4.3 – L'état du système M/D/1 à l'arrivée des paquets sondes et leurs instants de sortie

$$\pi_j(t) = \begin{cases} \frac{(\lambda t)^{j-N_0+K(t)}}{(j-N_0+K(t))} e^{-\lambda t}, & j \geq N_0 - K(t) \\ 0, & j < N_0 - K(t) \end{cases}$$

avec $N_0 = q_1 + S/L + 1$, λ est le taux d'arrivée des paquets du trafic concurrent et $K(t)$ est le nombre de paquets qui ont quitté le système avant l'instant t [Fra98].

Démonstration

A tout instant $t \in (0, D]$, avec $D = L/C$, les seuls paquets qui peuvent quitter le système depuis l'instant $t = 0$ sont ceux qui sont en service (dans le serveur) à cet instant là. En effet, initialement à l'instant $t = 0$, nous avons $N_0 = q_1 + S/L + 1$ paquets avec un temps résiduel d, donc :

$$K(t) = \begin{cases} 1 & d \leq t \leq D \\ 0 & 0 \leq t < d \end{cases}$$

Étant donné que le processus des arrivées des paquets du trafic concurrent est poissonien, nous avons éventuellement i nouvelles arrivées dans le système avec une probabilité $\frac{(\lambda t)^i}{i} e^{-\lambda t}$.

Afin d'avoir j paquets dans le système à l'instant t, il nous faut $j - N_0 + K(t)$ nouvelles arrivées durant l'intervalle de temps $(0, t]$. Par conséquent, pour tout $t \in (0, D]$

$$\pi_j(t) = \begin{cases} \frac{(\lambda t)^{j-N_0+K(t)}}{(j-N_0+K(t))} e^{-\lambda t}, & j \geq N_0 - K(t) \\ 0, & j < N_0 - K(t) \end{cases}$$

de la proposition 1, on peut en déduire que le vecteur des probabilités d'état

$$\pi(t) = (\pi_0(t), \pi_1(t), \pi_2(t), \dots)$$

peut être calculé facilement pour tout $t = \Delta_{in} \in (0, D]$. Cependant, quand $\Delta_{in} > D$, nous ne disposons d'aucune information quant au nombre de paquets qui ont quitté le système depuis l'instant $t = 0$ (dans ce cas $K(t)$ ne nous est d'aucune utilité).

pour déterminer le vecteur des probabilités d'état du système pour tout $\Delta_{in} > D$, nous exploitons le fait que chaque paquet en attente de service à l'instant t quittera le système à l'instant $t + D$ et que chaque paquet présent dans le système à l'instant $t + D$ est soit arrivé durant l'intervalle de temps $(t, t+D]$, soit il était déjà en attente de service dans la file à l'instant t, d'où la proposition suivante :

Proposition 2

(Vecteur de probabilités d'état du système M/D/1 à tout instant t [Fra98])
Le vecteur de probabilités d'état du système M/D/1 : $\pi(t) = (\pi_0(t), \pi_1(t), \pi_2(t), \dots)$ est donné par :

$$\pi(t) = \pi(t \bmod D) P^{\lfloor t/d \rfloor}$$

où

$$P = \begin{pmatrix} e^{-\lambda D} & \lambda D e^{-\lambda D} & \dots & \frac{(\lambda D)^{j-1}}{(j-1)!}e^{-\lambda D} & \dots \\ e^{-\lambda D} & \lambda D e^{-\lambda D} & \dots & \frac{(\lambda D)^{j-1}}{(j-1)!}e^{-\lambda D} & \dots \\ 0 & e^{-\lambda D} & \dots & \frac{(\lambda D)^{j-2}}{(j-2)!}e^{-\lambda D} & \dots \\ 0 & 0 & e^{-\lambda D} & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Démonstration

soit $N(t)$ le nombre de paquets présents dans le système à l'instant t, on a :

$$\begin{aligned} \pi_j(t+D) &= P(N(t)=0)P(j \text{ arrivées pendant } (t,t+D] \,|\, N(t)=0) \\ &\quad + \sum_{i=1}^{j+1} p(N(t)=i)P(j+1-i \text{ arrivées pendant } (t,t+D] \,|\, N(t)=i) \end{aligned}$$

$$\pi_j(t+D) = \pi_0(t)\frac{(\lambda D)^j}{j!}e^{-\lambda D} + \sum_{i=1}^{j+1}\pi_i(t)\frac{(\lambda D)^{j+1-i}}{(j+1-i)!}e^{-\lambda D} \tag{4.2}$$

Pour tout i et $j \in \mathbb{N}^*$, nous définissons la matrice $P = [p_{ij}]$ comme suit :

$$p_{ij} = \begin{cases} \frac{(\lambda D)^{j-1}}{(j+1-i)!}e^{-\lambda D}, & \text{pour } i = 1 \\ \frac{(\lambda D)^{j+1-i}}{(j+1-i)!}e^{-\lambda D}, & \text{pour } 2 \le i \le j+1 \\ 0, & \text{sinon} \end{cases} \tag{4.3}$$

A partir de la formule 4.2 et de la matrice définie par l'expression 4.3 nous pouvons facilement voir que :

$$\pi(t+D) = (\pi_0(t+D), \pi_1(t+D), \pi_2(t+D), \dots)$$

$$\pi(t+D) = (\pi_0(t), \pi_1(t), \pi_2(t), \dots) \begin{pmatrix} e^{-\lambda D} & \lambda D e^{-\lambda D} & \dots & \frac{(\lambda D)^{j-1}}{(j-1)!}e^{-\lambda D} & \dots \\ e^{-\lambda D} & \lambda D e^{-\lambda D} & \dots & \frac{(\lambda D)^{j-1}}{(j-1)!}e^{-\lambda D} & \dots \\ 0 & e^{-\lambda D} & \dots & \frac{(\lambda D)^{j-2}}{(j-2)!}e^{-\lambda D} & \dots \\ 0 & 0 & e^{-\lambda D} & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Donc

$$\pi(t+D) = \pi(t)P \tag{4.4}$$

En appliquant la formule 4.4 d'une manière itérative, $\pi(t+mD)$ peut être calculé, pour tout $m \in \mathbb{N}$, comme suit :

$$\pi(t+mD) = \pi(t)P^m \tag{4.5}$$

De la formule 4.5 on obtient :

$$\pi(t) = \pi(t \bmod D)P^{\lfloor t/D \rfloor}$$

A partir des propositions 1 et 2, le vecteur de probabilités d'état du système $(\pi(\Delta_{in}))$ peut donc être calculé quand le deuxième paquet sonde arrive dans la file d'attente quelque soit la valeur de Δ_{in}.

4.3.2 Temps d'attente moyen virtuel dans la file M/D/1

Soit la variable aléatoire $W(T)$ qui représente le temps d'attente d'un paquet arrivé dans la file d'attente du système M/D/1 à l'instant $t = T > 0$. Ce paquet est appelé *paquet virtuel* car ce dernier n'appartient pas nécessairement au trafic poissonnien qui entre dans le système. Il peut être par exemple un paquet du trafic sonde.

Admettons que $W_2 = W(\Delta_{in})$ avec initialement N_0 paquets dans le système. En définissant le temps d'attente $W(T)$ d'un *paquet virtuel* quelque soit son instant d'arrivée T, il serait facile de définir, à partir de la formule 4.1, la relation entre la dispersion initiale Δ_{in} et la dispersion finale Δ_{out} des paquets sondes.

La distribution du temps d'attente moyen virtuel d'un paquet dans la file $P(W(T) \leq kD - v)$ pour tout $k \in \mathbb{N}$ et $v \in (0, D]$ peut être déduite à partir de la position de ce paquet dans la file d'attente à l'instant $t = T + D - v$.

En admettons que $N_{priorT}(t)$ est la variable aléatoire qui définit le nombre de paquets qui arrivent dans la file avant l'instant T et qui y sont encore à l'instant t, alors la proposition suivante est vraie :

Proposition 3 (relation entre $W(T)$ et $N_{priorT}(t)$)

Pour tout $k \in \mathbb{N}$ et $v \in (0, D]$,

$$P[W(T) \leq kD - v] = P[N_{priorT}(T + D - v) < k].$$

Démonstration

La démonstration de cette proposition est un cas particulier des résultats obtenus dans [Fra98]. Pour démontrer cette proposition nous allons considérer deux cas différent :

- ⊙ le cas où $N_{priorT}(T + D - v) < k$
- ⊙ et le cas où $N_{priorT}(T + D - v) \geq k$

Quand $N_{priorT}(T + D - v) < k$, c'est-à-dire que quand le nombre de paquets qui arrivent dans la file avant l'instant $t = T$ et qui y sont toujours à l'instant $t = T + D - v$ est inférieur à k, alors $(k-1)D$ unités de temps après on aura $N_{priorT}(T + kD - v) < k - (k-1)$ c'est-à-dire $N_{priorT}(T + kD - v) < 1$ car pendant cette période on aura $(k-1)$ paquets qui quitteront le système. L'expression $N_{priorT}(T + kD - v) < 1$ veut tout simplement dire que tous les paquets qui sont arrivés dans le système avant l'instant $t = T$ l'ont tous quitté à l'instant $t = T + kD - v$ y compris le paquet virtuel (le paquet sonde). Ce qui veut dire que le temps d'attente de ce paquet virtuel (qui est arrivé dans le système à l'instant $t = T$) est inférieur à $kD - v$. Donc,

$$N_{priorT}(T + D - v) < k \Longrightarrow W(T) \leq kD - v. \tag{4.6}$$

Par ailleurs, si $N_{priorT}(T + D - v) \geq k$ c'est-à-dire que si le nombre de paquets qui sont dans le système à l'instant $t = T + D - v$ est supérieur à k alors le temps d'attente du paquet virtuel est supérieur à $kD - v$ unités (il sera toujours dans la file à l'instant $t = T + D - v$. D'où :

$$N_{priorT}(T + D - v) \geq k \Longrightarrow W(T) > kD - v. \tag{4.7}$$

En combinant 4.6 et 4.7 on obtient l'équivalence suivante :

$$N_{priorT}(T + D - v) < k \Longleftrightarrow W(T) \leq kD - v. \tag{4.8}$$

En terme de probabilités, l'expression 4.8 s'écrit comme suit :

$$P[W(T) \leq kD - v] = P[N_{priorT}(T + D - v) < k].$$

La fonction cumulative (CDF) de la variable aléatoire $W(T)$ (temps d'attente virtuel) est dérivée de la façon suivante :

Proposition 4

La fonction cumulative $F_T(x) = P\{W(T) \leq x\}$ de la variable aléatoire $W(T)$ est donnée par :

$$F_T(x) = \begin{cases} \sum_{j=0}^{\lfloor \frac{x}{D} \rfloor - N_0 + K(T + (x \bmod D))} \frac{(\lambda T)^j}{j!} e^{-\lambda T}, & \text{si } (x \bmod D) \leq D - T \\ \\ \sum_{j=0}^{\lfloor \frac{x}{D} \rfloor} Q_{\lfloor \frac{x}{D} \rfloor - j}(T - D + (x \bmod D)) \frac{[\lambda(D - (x \bmod D))]^j}{j!} e^{-\lambda(D - (x \bmod D))}, & \text{sinon} \end{cases}$$

avec

$$Q_m(T) = \sum_{i=0}^{m+1} \pi_i(t)$$

Démonstration

Soit $N(t)$ le nombre de paquets dans le système à l'instant t. Étant donné que l'arrivée du second paquet sonde est complètement indépendante de l'arrivée des paquets du trafic concurrent, alors :

$$P(N(t) = i \,|\, \text{le deuxième paquet sonde arrive à l'instant T}) = \pi_i(t)$$

,

pour $0 \leq t < T$.

De la proposition 3 nous avons : Pour tout $k \in \mathbb{N}$ et $v \in (0, D]$,

$$P[W(T) \leq kD - v] = P[N_{priorT}(T + D - v) < k]. \tag{4.9}$$

Puisque nous avons choisi $T > 0$ et $0 < v \leq D$ alors on a $T + D - v > 0$. Cependant, nous pouvons distinguer deux cas possibles : le cas où $T - v \leq 0$ et le cas où $T - v > 0$.

Dans le premier cas, puisqu'on a $T - v \leq 0$ alors $T + D - v \in (0, D]$. Donc à l'instant $t = T + D - v$, le système contient $N_0 - K(T + D - v)$ paquets qui restent dans la file, plus les paquets qui arrivent dans la file avant l'arrivée du second paquet sonde (les paquets qui arrivent durant l'intervalle $(0, T)$). Soit $A(t_1, t_2)$ le nombre de paquets qui arrivent dans le système dans l'intervalle (t_1, t_2), alors :

$$N_{priorT}(T + D - v) = N_0 - K(T + D - v) + A(0, T), \text{avec } T - v \leq 0.$$

En remplaçant $N_{priorT}(T + D - v)$ dans l'equation 4.9 (proposition 3), on aura :

$$P[W(T) \leq kD - v] = P[N_0 - K(T + D - v) + A(0, T) < k]$$

. donc

$$P[W(T) \leq kD - v] = P[A(0, T) < k - N_0 + K(T + D - v)], \text{avec } T - v \leq 0.$$

Étant donné que $A(0, T)$ est une variable aléatoire qui suit une loi de Poisson (arrivées des paquets d'un trafic concurrent poissonien) et qui est indépendante de N_0 et de $K(T + D - v)$, alors :

$$P[W(T) \leq kD - v] = \sum_{k - N_0 + K(T+D-v) - 1}^{j=0} \frac{(\lambda T)^j}{j!} e^{-\lambda T}, \text{pour } T - v \leq 0.$$

Dans le deuxième cas, on a $T - v > 0$ ce qui fait que $T + D - v > D$. On a tous les paquets qui sont en service (dans le serveur) à l'instant $t = T - v$ qui quitteront forcément le système à l'instant $t = T + D - v$. En revanche, tous les paquets qui ont été dans la file d'attente et qui attendaient d'entrer dans le serveur à l'instant $t = T - v$ seront toujours dans le système à

l'instant $t = T + D - v$. Soit $L_q(t)$ la longueur de la file d'attente (nombre de paquets dans la file) à l'instant t. A l'instant $t = T + D - v$ il y aura exactement $L_q(T-v) + A(T-v, T)$ paquets dans le système (les paquets de la file d'attente plus ceux du serveur) qui sont arrivés avant le deuxième paquet sonde. Ce qui fait que pour $T - v > 0$ on a :

$$N_{priorT}(T + D - v) = L_q(T - v) + A(T - v, T).$$

En posant des conditions sur le nombre de paquets qui arrivent dans l'intervalle de temps $[T - v, T]$ on a :

$$P[W(T) \leq kD - v \,|\, A(T - v, T) = j] = P[L_q(T - v) < k - j], \text{pour } T - v > 0 \tag{4.10}$$

Étant donné que $A(T - v, T)$ et $L_q(T - v)$ sont des variables aléatoire disjointes, l'application du *théorème des probabilités totales* ($P(A) = \sum_{k=0}^{n-1} P(A \,|\, B_k)P(B_k)$) sur la formule 4.10 donnera :

$$P[W(T) \leq kD - v] = \sum_{j=0}^{k-1} P[L_q(T - v) < k - j] \frac{(\lambda v)^j}{j!} e^{-\lambda v}, \text{pour } T - v > 0.$$

On a

$$\begin{aligned} P[L_q(T - v) < k - j] &= \sum_{i=0}^{k-j-1} \pi_i(T - v) \\ &= Q_{k-j-1}(T - v) \end{aligned}$$

avec $Q_m(t) = \sum_{i=0}^{m+1} \pi_i(t)$ qui peut être calculé en utilisant la formule 4.5.

Ceci nous donne :

$$P[W(T) \leq kD - v] = \sum_{j=0}^{k-1} Q_{k-j-1}(T - v) \frac{(\lambda v)^j}{j!} e^{-\lambda v} \tag{4.11}$$

En remplaçant $x = kD - v$ et $k = \lfloor \frac{x}{D} + 1 \rfloor$ dans 4.11, la fonction cumulative $F_T(x) = P\{W(T) \leq x\}$ de la variable aléatoire $W(T)$ représentant le temps d'attente d'un paquet qui arrive dans la file à l'instant $t = T$ est donnée par :

$$F_T(x) = \begin{cases} \sum_{j=0}^{\lfloor \frac{x}{D} \rfloor - N_0 + K(T + (x \bmod D))} \frac{(\lambda T)^j}{j!} e^{-\lambda T}, & \text{si } (x \bmod D) \leq D - T \\ \\ \sum_{j=0}^{\lfloor \frac{x}{D} \rfloor} Q_{\lfloor \frac{x}{D} \rfloor - j}(T - D + (x \bmod D)) \frac{[\lambda(D - (x \bmod D))]^j}{j!} e^{-\lambda(D - (x \bmod D))}, & \text{sinon} \end{cases}$$

4.4 Le modèle M/D/1 des dispersions finales

Dans cette section nous présentons un modèle stochastique des délais de la technique de la paire de paquets en nous basant sur les différentes propositions énoncées. Ce modèle définit la relation entre la dispersion initiale et la dispersion finale de la paire de paquets.

Pour une valeur donnée de la dispersion initiale Δ_{in} et un nombre initial q_1 de paquets dans la file, selon la formule 4.1 nous avons :

$$\Delta_{out} - \Delta_{in} = W2 - W1$$

tels que W_2 est le temps d'attente du second paquet dans la file M/D/1 au régime transitoire et $N_0 = q_1 + S/L + 1$ est le nombre de paquets dans le système à l'état initial avec L la taille des paquets du trafic concurrent et S la taille des paquets sondes. Étant donné que le second paquet arrive à l'instant $t = T = \Delta_{in}$ et sachant qu'à l'instant $t = 0$ il y a N_0 paquets dans la file nous pouvons écrire $W_2 = W(T = \Delta_{in}) = W(\Delta_{in})$.
W_1 est le temps d'attente du premier paquet dans la file, il correspond au temps T_{q1} nécessaire au traitement des q_1 paquets qui se trouvent déjà dans la file, donc $W_1 = W(T_{q_1})$.

D'après la définition de l'état initial du système, le variable aléatoire W_2 dépend de q_1, ce qui fait que si nous voulons exprimer l'expression 4.1 en terme d'espérance mathématique nous obtiendrons :

$$\mathbb{E}[\Delta_{out}] = \Delta_{in} + \mathbb{E}[\mathbb{E}[W_2|q_1]] - \mathbb{E}[W_1] \tag{4.12}$$

avec $\mathbb{E}[W_2|q_1]$ l'espérance conditionnelle de W_2 sachant q_1 qui est elle même une variable aléatoire (ce qui explique le terme $\mathbb{E}[\mathbb{E}[W_2|q_1]]$).

D'après les définitions précédentes nous avons :

$$\mathbb{E}[W_2|q_1] = \mathbb{E}[W(\Delta_{in})|q_1]$$

et d'après la définition de l'espérance mathématique nous avons :

$$\mathbb{E}[Y] = \int_0^\infty xF(x)\,dx = \int_0^\infty (1 - F(x))\,dx$$

avec $F(x)$ la fonction de densité de probabilité de Y.

donc :

$$\mathbb{E}[W_2|q_1] = \mathbb{E}[W(\Delta_{in})|q_1] = \int_0^\infty (1 - F_{\Delta_{in}}(x))\,dx$$

avec $F(x)$ la fonction de densité de probabilité de $W(\Delta_{in})$ qui peut être calculée en utilisant la proposition 4 en remplaçant T par Δ_{in}.

La formule 4.12 définie une relation entre la dispersion initiale Δ_{in} et la dispersion finale

Δ_{out}. Cependant, le calcul de $\mathbb{E}[\mathbb{E}[W(\Delta_{in})|q_1]]$ et de $\mathbb{E}[W(T_{q1})]$ n'est pas du tout pratique. Pour contourner cette difficulté, nous considérons deux cas extrêmes concernant Δ_{in}. Nous allons considérer le cas où $\Delta_{in} \longrightarrow 0$ et le cas où $\Delta_{in} \longrightarrow \infty$.

Cas 1 : $\Delta_{in} \longrightarrow 0$

Selon la loi de Poisson, quand Δ_{in} est très petit nous avons le nombre de paquets qui arrivent dans le système pendant l'intervalle $(0, \Delta_{in}]$ donné par :

$$A(0,\Delta_{in}) = \begin{cases} 1, & \text{avec un probabilité } \lambda\Delta_{in}e^{-\lambda\Delta_{in}} \\ 0, & \text{avec un probabilité } e^{-\lambda\Delta_{in}} \end{cases}$$

En utilisant le développement limite de $e^{\lambda\Delta_{in}}$ quand $\Delta_{in} \longrightarrow 0$ et pour tout $\Delta_{in} \ll \frac{1}{\lambda}$, nous obtenons :

$$A(0,\Delta_{in}) = \begin{cases} 1, & \text{avec un probabilité } \lambda\Delta_{in} + o(\Delta_{in}) \\ 0, & \text{avec un probabilité } 1 - \lambda\Delta_{in} + o(\Delta_{in}) \end{cases}$$

Pour une valeur donnée de q_1, la fonction de densité de probabilité de la variable aléatoire $W(\Delta_{in})$ est calculée comme suit :

$$F_{\Delta_{in}}(x) = (1-\lambda\Delta_{in})I(x - N_0D) + \lambda\Delta_{in}I(x-(N_0+1)D) + o(\Delta_{in})(1 - I(x - MD))$$

avec $I()$ la fonction indicatrice et M un nombre suffisamment grand et indépendant de Δ_{in}.

Étant donné que :

$$\begin{aligned} \mathbb{E}[W_2|q_1] &= \int_0^{\infty} (1 - F_{\Delta_{in}}(x))\, dx, \\ &= \lambda D\Delta_{in} + N_0D + o(\Delta_{in}). \end{aligned}$$

En considérant que le temps d'attente W_1 du premier paquet est le temps nécessaire au traitement des q_1 paquets arrivés avant lui, alors $W_1 = (N_0 - S/L)D$. Donc on aura :

$$\begin{aligned} \mathbb{E}[\Delta_{out}] &= \Delta_{in} + \mathbb{E}[W_2|q_1] - W_1 && (4.13) \\ &= \lambda D\Delta_{in} + SD/L + o(\Delta_{in}) && (4.14) \end{aligned}$$

Si r est le débit du trafic concurrent qui arrive dans le système, alors $r = \lambda L$. Étant donné

que $D = L/C$ alors l'equation 4.13 s'écrit come suit :

$$\mathbb{E}[\Delta_{out}] = \frac{r}{C}\Delta_{in} + \frac{S}{C} + o(\Delta_{in}) \tag{4.15}$$

Cas 2 : $\Delta_{in} \longrightarrow \infty$

Quand $\Delta_{in} \longrightarrow \infty$, on a l'equation 4.12 qui s'écrit comme suit :

$$\mathbb{E}[\Delta_{out}] - \Delta_{in} = \mathbb{E}[W_2] - \mathbb{E}[W_1]$$

Étant donné que pour Δ_{in} suffisamment grand, on a W_2 qui est indépendante de N_0, alors $\mathbb{E}[W_2]$ converge vers le temps d'attente moyen des paquets dans le système M/D/1 calculé en régime stationnaire. Donc : $\mathbb{E}[W_2] \sim \mathbb{E}[W_1]$.

De la formule précédente on en déduit que :

$$\mathbb{E}[\Delta_{out}] \sim \Delta_{in} \tag{4.16}$$

Ce qui exprime la stabilité de la file.

Récapitulatif

A partir des equations 4.14 et 4.15 nous déduisons que :

⊙ Pour une dispersion initiale Δ_{in} suffisamment petite ($\Delta_{in} < \frac{1}{\lambda}$) on a :

$$\mathbb{E}[\Delta_{out}] = \frac{r}{C}\Delta_{in} + \frac{S}{C} + o(\Delta_{in})$$

⊙ Pour une dispersion initiale Δ_{in} suffisamment large ($\Delta_{in} \geq \frac{1}{\lambda}$) on a :

$$\mathbb{E}[\Delta_{out}] \sim \Delta_{in}$$

Remarque

Les propositions et les démonstrations introduites dans les sections précédentes de ce chapitre sont dérivées principalement des travaux de Franx [Fra98] et de Park [Par05].

4.5 Évaluation du modèle

4.5.1 Méthodologie

Dans cette section nous évaluons le modèle défini dans la section précédente et nous le comparons aux résultats expérimentaux obtenus sur une plateforme de mesure. Le modèle stochastique introduit dans ce chapitre définit la relation entre la dispersion initiale Δ_{in} et la dispersion finale Δ_{out} d'une paire de paquets. Pour vérifier la concordance des résultats de ce modèle par rapport

à la réalité, nous avons développé un outil appelé *Gapper* constitué d'un émetteur et d'un récepteur. Ce dernier permet d'envoyer des paires de paquets avec une certaine dispersion initiale et de mesurer leurs dispersions finales au niveau du récepteur. Les résultats du modèle et ceux obtenus expérimentalement sont comparés selon différents scénarios. En effet, les expérimentations effectuées tiennent compte de la capacité du lien bottleneck, de la bande passante disponible ainsi que de la taille des paquets du trafic concurrent et les scénarios considérés correspondent aux différentes valeurs de ces deux derniers paramètres.

Le modèle étudié est valide uniquement dans le cas d'un seul lien (le lien bottleneck en l'occurrence). Nous avons donc choisi d'effectuer les expérimentations sur une plateforme constituée seulement de deux routeurs et dont la topologie est représentée dans la figure 4.4.

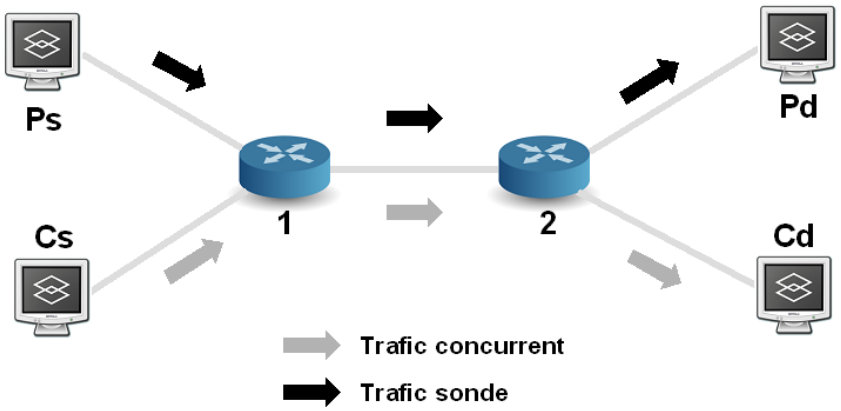


FIG. 4.4 – Plateforme d'expérimentation

La topologie considérée consiste en un ensemble de stations de travail interconnectées via deux routeurs constituant un seul lien (Routeur1, Routeur2). L'ensemble des liens reliant les stations de travail aux différents routeurs sont à 100 Mb/s, seul le lien (Routeur1, Routeur2) est à 10 Mb/s constituant ainsi le lien bottleneck. L'émetteur et le récepteur de l'outil *Gapper* sont installés respectivement sur P_s et P_d. Les paires de paquets circulent donc de P_s à P_d. Le générateur de trafic MGEN installé sur C_s permet d'envoyer un trafic concurrent de type UDP vers C_d. L'outil de capture Ethereal (renommé Wireshark dans sa dernière version) est installé au niveau du récepteur, il permet de vérifier le débit du trafic concurrent généré par MGEN.

Nous avons considéré trois différents scénarios correspondant à trois valeurs de la bande passante disponible à savoir 1 Mb/s, 5Mb/s et 9 Mb/s. Dans chaque scénario nous avons pris en compte 3 différents cas pour lesquels nous avons utilisé des paires de paquets de 1000 octets et un trafic concurrent dont la taille des paquets est de 500 octets, 1000 octets et 1500 octets respectivement. Pour chaque cas, nous faisons varier la valeur de la dispersion initiale Δ_{in} entre 0 et 5 millisecondes et nous récupérons au niveau du récepteur les résultats des mesures de la dispersion finale Δ_{out} qui correspondent à chaque valeur de Δ_{in}. Nous avons ensuite comparé

ces derniers aux résultats théoriques obtenus par le modèle stochastique. Les résultats de la comparaison pour chaque scénario sont présentés dans les figures 4.5, 4.6 et 4.7.

4.5.2 Résultats

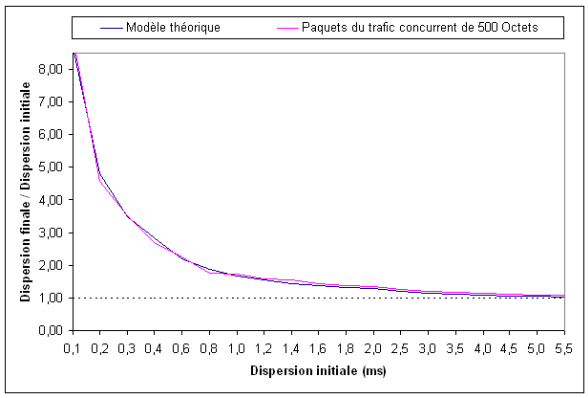
Pour comparer les résultats du modèle théorique aux résultats des expérimentations obtenus à l'aide de l'outil Gapper, nous traçons les courbes de $\Delta_{out}/\Delta in$ par rapport à Δ_{in} pour les différents scénarios considérés. Les figures 4.5, 4.6 et 4.7 montrent que le rapport $\Delta_{out}/\Delta in$ est élevé quand Δ_{in} est faible ($\Delta_{in} \longrightarrow 0$) et que ce rapport tend rapidement vers 1 quand Δ_{in} augmente. Pour tous les scénarios considérés et ceci quelque soit la taille des paquets du trafic concurrent, nous remarquons que les courbes du modèle théorique correspondent bien à celles obtenues expérimentalement, ce qui montre que le modèle présenté correspond bien à la réalité et ceci quelque soit la taille des paquets du trafic concurrent .

Les tests réalisés dans le cadre de ces expérimentations, utilisent un trafic concurrent Poissonien. Nous n'avons pas effectué de test avec d'autres types de trafic, toutefois, nous pensons que les résultats seraient légèrement différents et que ces derniers montreraient quelques erreurs par rapport aux résultats du modèle étant donné que ce dernier est basé sur le système M/D/1 qui utilise un trafic Poissonien. Nous constatons aussi à partir de ces figures, que le rapport $\Delta_{out}/\Delta in$ dépend de la bande passante disponible. Ce rapport converge plus rapidement vers 1 quand la bande passante disponible est grande, ceci est du au fait que la quantité du trafic concurrent dans ce cas est faible et donc les paquets de ce dernier sont transmis au niveau du bottleneck en un temps inférieur ou égal à la dispersion initiale Δ_{in}. Ce qui a pour conséquence d'obtenir le rapport $\Delta_{out}/\Delta in$ qui tend vers 1 beaucoup plus rapidement.

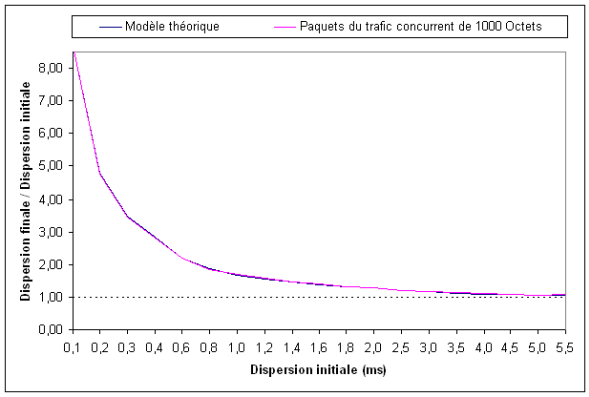
4.6 Effet de la taille des paquets du trafic concurrent

Les courbes présentées dans la section précédente ne permettent pas d'étudier l'effet de la taille des paquets du trafic concurrent sur la mesure de la dispersion finale Δ_{out}. Pour étudier cet effet, nous analysons dans cette section la différence entre les résultats des dispersions finales obtenus expérimentalement et ceux du modèle théorique. Les résultats de cette comparaison sont représentés dans la figure 4.8.

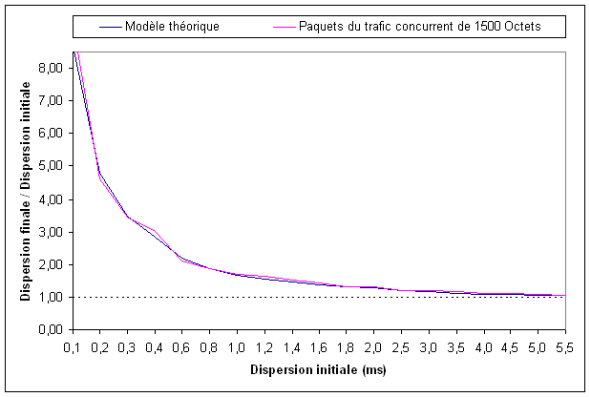
Cette figure présente les résultats de comparaison des dispersions finales Δ_{out} en fonction des dispersions initiales Δ_{in} variant entre 0 et 5 millisecondes. Or dans la réalité, que ce soit au niveau de l'outil Gapper ou de l'outil IGMPS, cette dispersion initiale est généralement comprise entre $0,8\,ms$ et $1,6\,ms$ (cet intervalle est valable uniquement dans le cas où la capacité $C = 10$ Mb/s, pour les autres cas, les valeurs de Δ_{in} sont complètement différentes). En effet, ce choix de valeur de Δ_{in} est imposé par la définition même de la technique de la paire de paquets qui exige que la dispersion initiale entre les paquets de la paire soit suffisamment petite pour permettre au deuxième paquet d'arriver dans la file d'attente du bottleneck avant le départ du premier



(a) Paquets de trafic concurrent de 500 octets

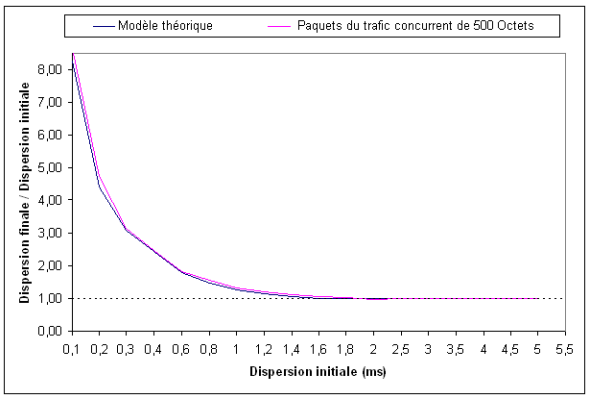


(b) Paquets de trafic concurrent de 1000 octets

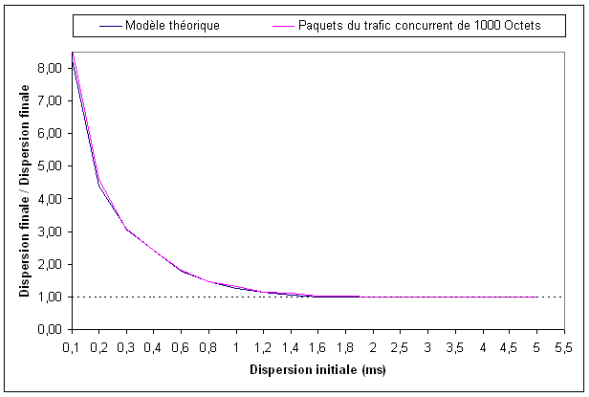


(c) Paquets de trafic concurrent de 1500 octets

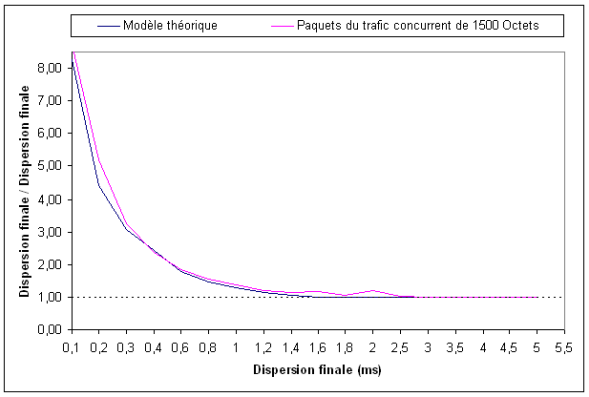
FIG. 4.5 – Scénario 1 (1 Mb/s) : Le rapport de la dispersion finale sur la dispersion initiale



(a) Paquets de trafic concurrent de 500 octets

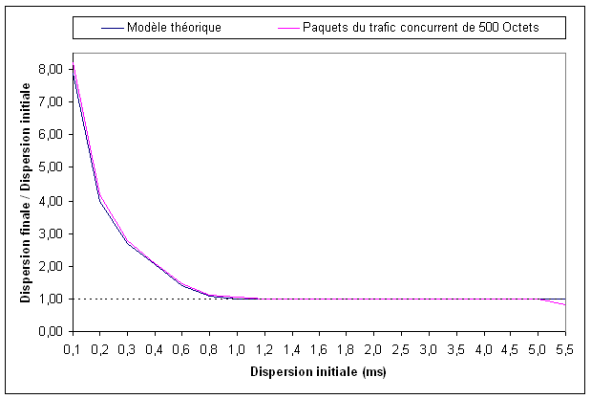


(b) Paquets de trafic concurrent de 1000 octets

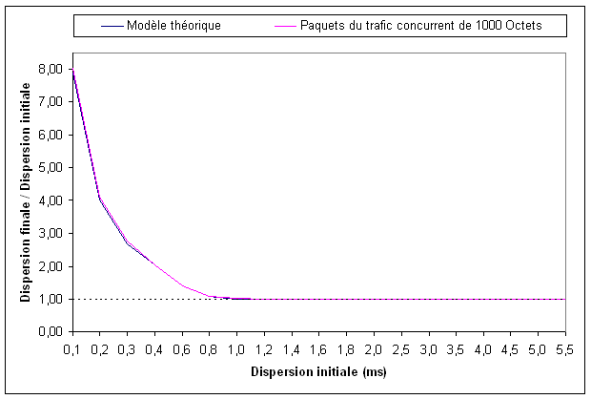


(c) Paquets de trafic concurrent de 1500 octets

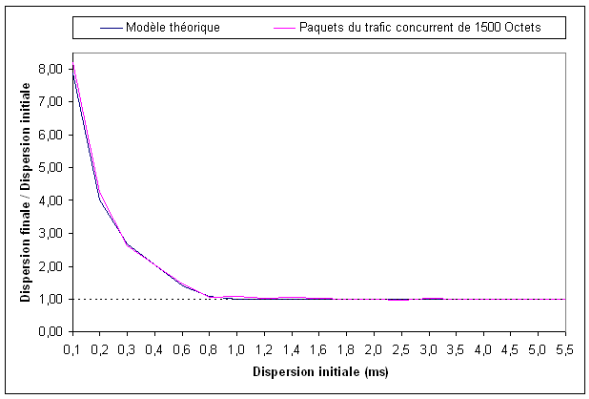
FIG. 4.6 – Scénario 2 (5 Mb/s) : Le rapport de la dispersion finale sur la dispersion initiale



(a) Paquets de trafic concurrent de 500 octets

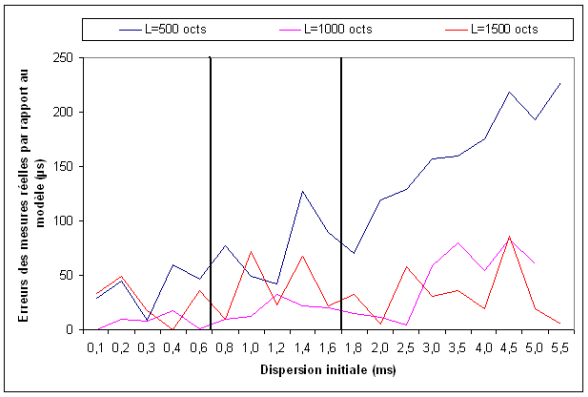


(b) Paquets de trafic concurrent de 1000 octets

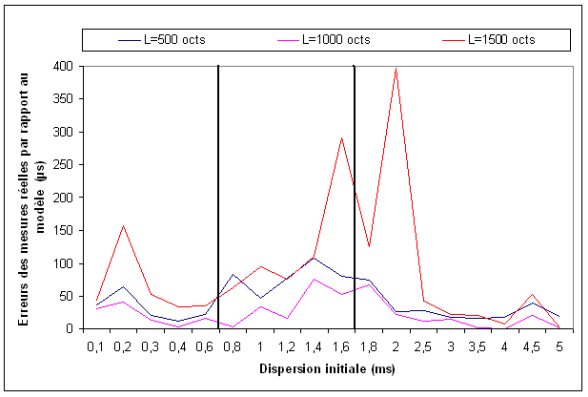


(c) Paquets de trafic concurrent de 1500 octets

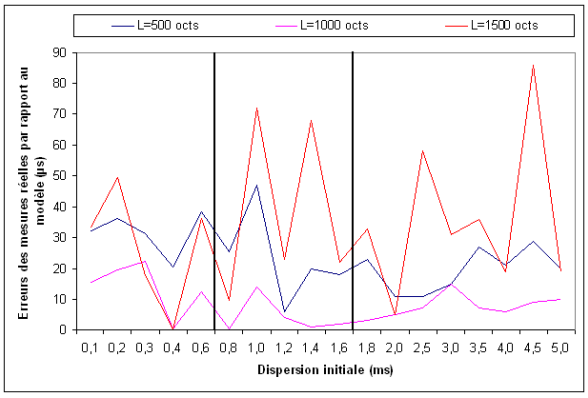
FIG. 4.7 – Scénario 3 (9 Mb/s) : Le rapport de la dispersion finale sur la dispersion initiale



(a) Scénario 1 (1 Mb/s)



(b) Scénario 2 (5 Mb/s)



(c) Scénario 3 (9 Mb/s)

FIG. 4.8 – Erreurs des mesures réelles par rapport au modèle théorique

paquet. Cependant, cette dispersion initiale doit être suffisamment importante pour permettre aux paquets du trafic concurrent de s'infiltrer entre les paquets de la paire. Donc, choisir des valeurs de Δ_{in} en dehors de cette intervalle risque de ne pas satisfaire les hypothèses considérées dans la technique de la paire de paquets.

La figure 4.8 montre que pour les trois scénarios considérés, quand la dispersion initiale est dans l'intervalle de temps défini précédemment, la courbe des erreurs des dispersions finales correspondant au trafic concurrent dont la taille des paquets est $L = 1000$ octets est celle qui montre les plus faibles valeurs. Donc les dispersion finales mesurées pour $L = 1000$ octets sont celles qui se rapprochent le plus du modèle.

Étant donné que les paquets sondes utilisés dans nos expérimentations ont une taille $S = 1000$ octets, nous déduisons que pour avoir les résultats les plus proches des résultats du modèle (les résultats les plus précis) il est nécessaire d'avoir la taille des paquets sondes qui est égale à la taille des paquets du trafic concurrent (ou suffisamment proche d'elle).

En connaissant la dispersion initiale Δ_{in}, la taille des paquets sondes S et la capacité du lien bottleneck C, la seule information supplémentaire nécessaire pour le calcul de la bande passante disponible est bien la dispersion finale Δ_{out}. Comme la précision de mesure de la bande passante disponible dépend largement de la mesure de Δ_{out} (voir chapitre 5), nous déduisons de la constatation précédente que pour mesurer la bande passante disponible avec précision, il est nécessaire que la taille des paquets sondes soit égale ou suffisamment proche de la taille des paquets du trafic concurrent, ce qui démontre la proposition 1 introduite au chapitre précédent (chapitre 3).

Étant donné que les résultats obtenus dans nos expérimentations ne dépendent pas du processus d'envoi des paires de paquets sondes (Poissonien, Périodique, ...), nous pouvons donc généraliser la proposition 3.1 à tous les outils de mesure de la bande passante disponible qui utilisent la technique de la paire de paquets. Les résultats des chapitres 3 et 4 ont donc permis d'arriver à la proposition suivante :

proposition

Pour mesurer la bande passante disponible dans un chemin de bout en bout avec précision en utilisant la technique de la paire de paquets, il est nécessaire de régler la taille des paquets sondes pour qu'elle soit égale ou suffisamment proche de la taille des paquets du trafic concurrent.

Pour que cette proposition soit applicable dans les conditions réelles de l'Internet, il est nécessaire d'utiliser des paquets sondes avec des tailles qui sont dynamiquement variables afin de les faire correspondre aux tailles les plus fréquentes des paquets du trafic Internet.

4.7 Conclusion

Dans ce chapitre, nous avons présenté un modèle stochastique qui se base sur l'étude du système de file d'attente M/D/1 pour établir une relation entre les dispersions initiales des paquets sondes et leurs dispersions finales pour la technique de la paire de paquets. Pour valider ce modèle nous avons développé un outil appelé *Gapper* qui permet d'envoyer un ensemble de paires de paquets sondes d'une source vers une destination et de mesurer les dispersions finales de ces dernières en fonctions de leurs dispersions initiales. Les résultats obtenus ont montré la validité du modèle pour tous les scénarios. Ils ont aussi montré que lorsque la taille des paquets sondes est égale à la taille des paquets du trafic concurrent ou est suffisamment proche, les dispersions finales correspondent le mieux aux résultats du modèle et sont donc les plus précises, confirmant ainsi les propositions et les résultats obtenus expérimentalement au chapitre précédent. Les chapitres 3 et 4 nous ont montré que la taille des paquets sondes est un paramètre très important pour la mesure de la bande passante disponible. Cependant, y'a-t-il d'autres paramètres importants qui pourraient influencer les résultats de mesure de cette métrique ? C'est à cette question que nous tenterons de répondre au chapitre suivant.

Chapitre 5

Analyse de sensibilité appliquée à la mesure de la bande passante disponible

Résumé

Le modèle proposé dans le chapitre 3 utilise plusieurs paramètres qui peuvent parfois être mesurés avec une forte incertitude. L'importance de ces paramètres et leurs effets sur la sortie du modèle peuvent être déterminés par une analyse de sensibilité globale ou par une étude de propagation d'incertitudes. L'objectif de cette étude est de déterminer quels sont les paramètres susceptibles de perturber la sortie du modèle afin d'apporter par la suite des solution pour réduire les erreurs et améliorer ainsi la précision des mesures. Les résultats obtenus montrent que les dispersions inter-paquets initiales et finales sont les paramètres les plus importants lors de la mesure de la bande passante disponible en utilisant les techniques à dispersion de paquets.

5.1 Introduction

La simulation de phénomènes physiques (lorsque les expérimentations sont impossibles ou trop onéreuses) et leur interprétation reposent généralement sur des modèles mathématiques destinés à les mimer ou à prédire leurs comportements. Ces modèles font correspondre d'une manière déterministe à un ensemble de paramètres d'entrée une ou plusieurs variables de sortie. La connaissance du phénomène physique étudié est souvent imparfaite et les moyens utilisés pour la numérisation du modèle correspondant sont limités (puissance de calcul, techniques de programmation, etc). De plus les paramètres d'entrée du modèle ne sont pas connus d'une façon exacte et sont souvent entachés d'erreur, ce qui fait qu'à la plupart de ces modèles sont associées différentes sources d'incertitudes [Jac05]. Les modèles mathématiques décrivant les phénomènes réels contiennent deux sortes d'incertitude : l'une qui est liée au modèle lui-même due à la

description physico-mathématique inexacte du système et désignée par le terme *"incertitudes de modèle"* et l'autre liée aux paramètres du modèle et due aux erreurs d'estimation et de procédure adoptée dans la collecte des données expérimentales utilisées.

Les incertitudes de modèle sont dues au passage du phénomène physique au modèle mathématique (incertitudes sur la structure même du modèle mathématique). En effet, la définition du modèle théorique génère une incertitude due à la nature même du phénomène étudié, qui peut ne jamais être connue ni comprise parfaitement par l'homme d'autant plus que ce phénomène peut ne pas être stable et peut évoluer dans le temps. L'outil théorique utilisé pour définir le phénomène (théorie des graphes, réaction chimique, calculs réseaux, etc) n'est pas forcément en adéquation avec ce dernier, de même que les hypothèses et les conditions initiales utilisées ne sont pas nécessairement exactes. L'impossibilité de prendre en compte la globalité du phénomène physique fait que le passage du modèle théorique au modèle mathématique entraîne des approximations qui sont sources d'incertitudes. L'utilisation de ce modèle mathématique nécessite souvent que ce dernier soit discrétisé afin de pouvoir en tirer une solution numérique. Cette discrétisation engendre aussi des approximations qui sont sources d'incertitudes supplémentaires. Enfin, le calcul informatique de la valeur numérique de la sortie du modèle est sujet au incertitudes dues à la précision finie du calculateur utilisé. Ces différentes incertitudes (épistémiques) ne peuvent être réduite que par l'approfondissement et l'amélioration de la connaissance du phénomène physique étudié et par le choix d'outil mathématiques (ou physique) beaucoup plus adéquats [Abr93].

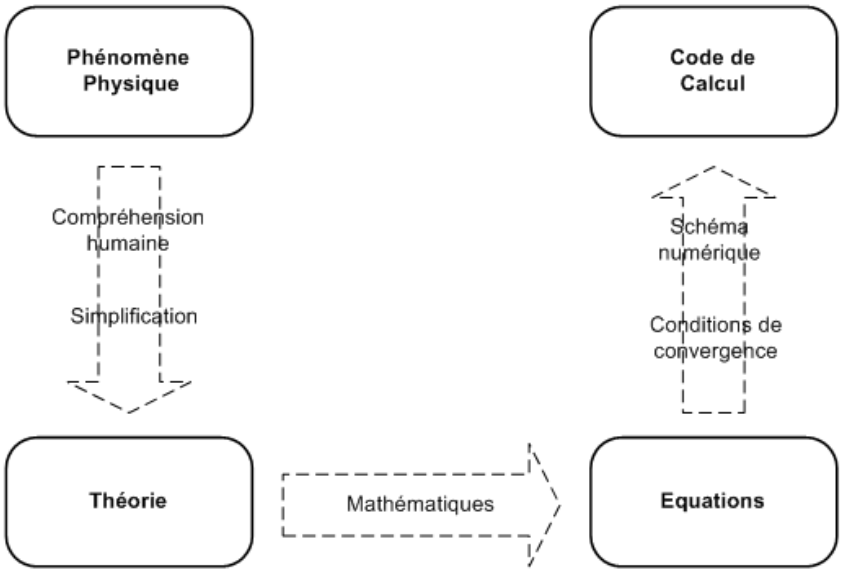


FIG. 5.1 – Passages du phénomène physique au modèle mathématique numérique.

La littérature correspondant à l'étude des incertitudes de modèle traite de trois axes de

recherche différents :

- ⊙ la définition des différentes sources d'incertitudes lors de l'élaboration d'un modèle,
- ⊙ la sélection de modèle lorsqu'on est en présence de plusieurs modèles concurrents
- ⊙ et finalement l'étude d'incertitudes issues de l'utilisation d'un modèle simplifié.

Dans notre travail, nous nous intéressons plutôt au premier axe de recherche qui s'intéresse à la définition des différentes sources d'incertitudes sur un modèle. Ces incertitudes sont dues soit à l'incertitude des paramètres d'entrée du modèle soit à sa structure qui ne reproduit pas vraiment le phénomène physique ou qui le fait mais seulement d'une manière partielle. Malheureusement, dans la littérature, aucune méthode quantitative n'est donnée pour déterminer les incertitudes liées à la structure même du modèle. Un moyen d'étudier ce genre d'incertitudes est de déterminer la distribution des incertitudes sur la sortie du modèle en connaissant les distributions des incertitudes sur les différents paramètres d'entrée de ce dernier (propagation d'incertitudes sur la sortie du modèle). En effet, les valeurs des paramètres d'entrée ne sont jamais connues avec exactitude, et très souvent ils sont définis avec un important degré d'incertitude pouvant influencer les prédictions, même si le modèle est correct d'un point de vue structurel. Donc il est primordial pour l'utilisateur d'un modèle de savoir comment réagit la sortie de ce dernier aux facteurs d'entrée. Cette recherche de la compréhension de l'effet des perturbations des paramètres d'entrée sur le modèle peut se faire soit par l'étude de propagation d'incertitudes sur la sortie ou bien par une analyse de sensibilité (SA : Sensitivity Analysis) [Sal00, Rod05].

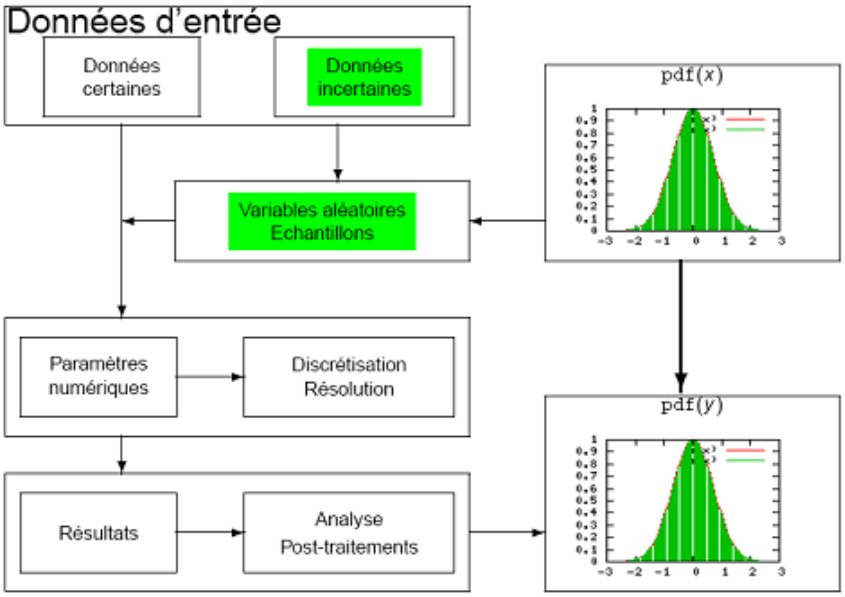


Fig. 5.2 – Propagation d'incertitudes.

Le modèle de mesure de la bande passante disponible proposé dans le chapitre 3 et implémenté dans IGMPS prend plusieurs paramètres en entrée, tels que la taille des paquets sondes,

la capacité du lien bottleneck et les instants d'arrivée et de départ des paquets sondes, etc. Pour effectuer les mesures, IGMPS envoie vers la destination des paires de paquets estampillés. A la réception, les paquets sont encore estampillés et la dispersion finale entre les paquets est calculée. IGMPS analyse cette dispersion et calcule la bande passante disponible à partir de la capacité du lien bottleneck. Cette capacité est sensée être connue ou peut être mesurée en utilisant l'un des outils spécifiques à cette métrique. Les outils de mesure de capacité dont on dispose actuellement sont généralement entachés d'incertitudes et l'estampillage des paquets à l'envoi et à la réception est fait de façon imprécise. Ces différentes erreurs qui caractérisent les variables d'entrée de notre modèle induisent des incertitudes sur la réponse en sortie de ce dernier ce qui génère des mesures imprécises.

Le but de ce chapitre est d'étudier les différentes source d'incertitudes du modèle déterministe proposé puis, en utilisant les différentes méthodes d'analyse de sensibilité, nous tenterons de déterminer et d'analyser comment réagit la sortie de ce modèle aux perturbations et aux incertitudes de mesure des paramètres d'entrée. Les informations obtenues de cette analyse de sensibilité nous permettrons de savoir quels sont les paramètres qui ont le plus d'influence sur les résultats de mesure de la bande passante disponible. Ces résultats seront par la suite utilisés pour tenter d'améliorer les processus de développement des techniques et des outils de mesure de cette métrique en se basant sur des méthodes qui permettrons de diminuer les incertitudes de mesure sur les paramètres désignés par l'analyse de sensibilité comme étant des paramètres clés dans la mesure de la bande passante disponible .

5.2 Analyse de sensibilité

Soit un modèle mathématique constitué d'un ensemble de n variables d'entrée $X = (X_1, X_2, \ldots, X_n)$ et d'une variable de sortie Y telles que :

$$\begin{aligned} f : \Re^n &\longrightarrow \Re \\ Y &\longrightarrow Y = f(X) \end{aligned}$$

L'analyse de sensibilité permet d'étudier comment les perturbations sur les variables d'entrée du modèle engendrent des perturbations sur la réponse de ce dernier. Les méthodes de l'analyse de sensibilité peuvent être classées en trois catégories :

- ⊙ les méthodes de Screening,
- ⊙ l'analyse de sensibilité locale,

⊙ et l'analyse de sensibilité globale.

Les méthodes de screening permettent une analyse qualitative des variables d'entrée, elles permettent d'analyser l'importance de ces variables et d'établir une hiérarchie au sein de ces dernières en fonction de leur influence sur la variabilité de la réponse du modèle [Nea90]. Ces méthodes sont très efficaces quand le modèle à analyser a un nombre considérable de paramètres d'entrée [Jol02].

L'analyse de sensibilité locale s'attache à déterminer l'impact local des facteurs d'entrée sur le modèle. Elle repose sur le principe consistant à calculer les dérivées partielles des fonctions de sorties par rapport aux variables d'entrée. Ces calculs se font numériquement en faisant varier les entrées du modèle dans un intervalle très restreint autour d'une valeur nominale. Cette méthode peut être perçue comme un cas particulier de l'approche "One-Factor-At-A-Time" (OAT), car quand un facteur varie, tous les autres sont maintenus constants. Généralement, quand on applique cette méthode, les relations entre les entrées et les sorties sont supposées être linéaires et par conséquent, l'effet de la relative variation de la sortie du modèle en faisant varier le facteur d'entrée autour de sa valeur de base peut être estimée par la relation :

$$S_i = \frac{\ln Y - \ln Y_b}{\ln X_i - \ln X_{ib}}$$

où S_i est la mesure de sensibilité pour un facteur i donné, Y_b et Y sont respectivement les sorties correspondant au facteur de base X_{ib} et au facteur X_i (facteur de base augmenté ou diminué d'un certain pourcentage de variation).

L'analyse de sensibilité globale est une méthode quantitative basée sur l'estimation de la contribution de chaque paramètre d'entrée d'un modèle à la variance observée dans les sorties ; elle étudie aussi l'interaction entre ces différentes variables d'entrée [Rat01, Sou99]. Les techniques de l'analyse de sensibilité globale s'affranchissent des hypothèses classiques de linéarité que supposent les principes de régression et de corrélation.

5.3 Objectifs de l'analyse de sensibilité

Les enjeux de l'analyse de sensibilité (AS) diffèrent d'un domaine à l'autre et ses objectifs peuvent être :

⊙ l'identification d'un jeu de paramètres importants pouvant servir au calibrage d'un modèle,
⊙ juger de la qualité d'un modèle en essayant de repérer les fausses hypothèses posées lors de l'interprétation des phénomènes,
⊙ identifier parmi plusieurs modèles celui ou ceux qui décrivent mieux un système réel donné.

Cependant, selon Kleijnen [Kle95], l'AS peut être considérée comme une investigation systématique de la réaction d'un modèle vis-à-vis des valeurs extrêmes de ses paramètres ou des changements drastiques dans sa structure. Si dans certains cas la structure du modèle peut ne pas être remise en question, sa sensibilité ou le degré de variation des sorties par rapport aux incertitudes de ses paramètres d'entrée est d'importance et doit être connue. Selon [Jol02], les paramètres dans un modèle mathématique complexe sont parfois très nombreux, et tous n'ont pas le même degré d'influence sur les sorties du modèle. Il y en a qui sont beaucoup plus importants que d'autres. Ainsi, une analyse de sensibilité peut aider à prédire l'effet de chaque paramètre sur les résultats du modèle et à les classer suivant leur degré de sensibilité [Sal00]. La connaissance de cette information est très importante pour l'utilisateur d'un modèle, elle renseigne ce dernier sur le niveau de précision que doit avoir chaque paramètre et l'attention qu'on doit lui accorder lors de sa détermination sur le terrain ou au laboratoire [Jol02].

Dans la suite de ce chapitre, nous nous intéresseront uniquement aux méthodes de l'analyse de sensibilité globale vue que c'est cette technique qui nous permettra de mesurer la part d'influence de chaque paramètre d'entrée dans la variation de la sortie du modèle proposé en s'affranchissant de toute hypothèse sur ce dernier.

5.4 Les méthodes de l'analyse de sensibilité globale

Dans [Sal00], les différentes technique d'analyse de sensibilité globale on été introduites, elles sont classées en :

- ⊙ méthodes fiabilistes de type FORM et SORM traitant de l'analyse de sensibilité basée sur l'étude de risque,
- ⊙ les méthodes bayesiennes,
- ⊙ les méthodes graphiques
- ⊙ et enfin les méthodes basées sur l'étude de la variance.

Dans notre cas, on ne s'intéressera qu'aux méthodes basées sur la variance qui consistent à déterminer quelle part de variance de la sortie du modèle est due à la variance de chaque paramètre d'entrée.

De nos jours différentes méthodes basées sur l'étude de variance, les unes plus complexes que d'autres, existent pour effectuer l'analyse de sensibilité globale d'un modèle mathématique. Parmi elles deux techniques principales sont utilisées :

- ⊙ La méthode de Monte Carlo appelée aussi Méthode de Sobol,

⊙ La méthode FAST (Fourier Amplitude Sensitivity Test)

Des variantes de ces méthodes ont été développées afin d'accélérer la convergence lors des simulations. Parmi ces dernière on y trouve la méthode par Hypercube Latin (LHS) appelée aussi méthode de McKay[McK79], les méthodes de Quasi-Monte carlo [Nie92] et les méthode de Quasi-Monte Carlo randomisées [Owen98]. L'ouvrage de référence de Saltelli [Sal00] décrit en détails ces différentes techniques d'analyse de sensibilité et propose des exemples d'applications dans de nombreux domaines tels que le nucléaire, la chimie et l'économie par exemple.

5.4.1 La méthode de Sobol

Rappel sur la méthode de Monte Carlo [Sal00]

Dans beaucoup de problèmes scientifiques, on est amené à calculer une intégrale du type

$$I = \int_D f(x)\, dx \tag{5.1}$$

où D est un espace fermé de plus au moins grande dimension, et f une fonction intégrable. Soit $x_1, \ldots, x_n$ un N-échantillon d'une variable aléatoire uniforme sur D. Nous supposons cet échantillon pris de manière totalement aléatoire. Grace à la loi forte des grands nombres qui assure que la moyenne d'une suite de variables aléatoires indépendantes de même espérance et de variances finies converge presque sûrement vers l'espérance alors une approximation de I par la méthode de Monte Carlo est faite par :

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} f(X_i)$$

Étant données les hypothèses de la loi forte des grands nombres (même espérance et variances finies), toute espérance mathématique d'une variable aléatoire $f(X)$ de densité de probabilité μ :

$$E\big[f(X)\big] = \int Df(x) d\mu(x) = \int Df(x)\mu(x)dx,$$

peut être estimée par :

$$\hat{E}\big[f(X)\big] = \frac{1}{N} \sum_{i=1}^{N} f(X_i)$$

où $(x_i)_{i=1\ldots N}$ est un N-échantillon de réalisations de la variable aléatoire X.

La méthode de Monte Carlo avec échantillonnage aléatoire est la méthode de base pour calculer l'intégrale (5.1). Comme indiqué précédemment, des méthodes alternatives améliorent toutefois la convergence.

Estimation des indices de sensibilité par Monte Carlo

Considérons un N-échantillon de réalisations des variables d'entrée $(X_1, \ldots, X_p)$ de notre modèle et Y la variable de sortie de ce dernier :

$$\hat{X}_N = (X_{k1}, \ldots, X_{kp})_{k=1\ldots N}$$

L'espérance de Y, $E[Y] = f_0$, et sa variance, $V(Y) = V$, sont estimées par :

$$\hat{f}_0 = \frac{1}{N} \sum_{i=1}^{N} f(X_{k1}, \ldots, X_{kp}), \tag{5.2}$$

$$\hat{V}_0 = \frac{1}{N} \sum_{i=1}^{N} f^2(X_{k1}, \ldots, X_{kp}) - \hat{f}_0^2. \tag{5.3}$$

L'estimation des indices de sensibilité nécessite l'estimation d'espérance de variance conditionnelle. Nous présentons une technique d'estimation due à Sobol [Sal00].
L'estimation des indices de sensibilité de premier ordre consiste à estimer la quantité :

$$V_i = V(E[Y|X_i]) = \underbrace{E[E[Y|X_i]^2]}_{U_i} - E[E[Y|X_i]]^2 = U_i - E[Y]^2,$$

la variance de Y étant estimée classiquement par (5.3).
Sobol propose d'estimer la quantité U_i, c'est-à-dire l'espérance du carré de l'espérance de Y conditionnellement à X_i, comme une espérance classique, mais en tenant compte du conditionnement à X_i en faisant varier dans les deux appels à la fonction f toutes les variables sauf la variable X_i. Ceci nécessite deux échantillons de réalisations des variables d'entrée, que nous notons $\tilde{X}^{(1)}_{(N)}$ et $\tilde{X}^{(2)}_{(N)}$:

$$\begin{aligned}\hat{U}_i = \frac{1}{N} sum_{k=1}^{N} f\left(x_{k1}^{(1)}, \ldots, x_{k(i-1)}^{(1)}, x_{ki}^{(1)}, x_{k(i+1)}^{(1)}, \ldots, x_{kp}^{(1)}\right) \\ \times f\left(x_{k1}^{(2)}, \ldots, x_{k(i-1)}^{(2)}, x_{ki}^{(1)}, x_{k(i+1)}^{(2)}, \ldots, x_{kp}^{(2)}\right)\end{aligned}$$

Les indices de sensibilité de premier ordre sont alors estimés par :

$$\hat{S}_i = \frac{\hat{V}_i}{\hat{V}} = \frac{\hat{U}_{ij} - \hat{f}_0^2}{\hat{V}}$$

Pour les indices de sensibilité de second ordre $S_{ij} = \frac{V_{ij}}{V}$, où :

$$V_{ij} = V(E[Y|X_i, X_j]) - V_i - V_j = U_{ij} - E[Y]^2 - V_i - V_j,$$

les quantités $U_{ij} = E[E[Y|X_i, X_j]^2]$ de la même manière, en faisant varier toutes les variables

sauf X_i et X_j :

$$\hat{U}_{ij} = \frac{1}{N} sum_{k=1}^{N} f\left(x_{k1}^{(1)}, \ldots, x_{k(i-1)}^{(1)}, x_{ki}^{(1)}, x_{k(i+1)}^{(1)}, \ldots, x_{k(j-1)}^{(1)}, x_{kj}^{(1)}, x_{k(j+1)}^{(1)}, \ldots, x_{kp}^{(1)}\right) \times f\left(x_{k1}^{(2)}, \ldots, x_{k(i-1)}^{(2)}, x_{ki}^{(1)}, x_{k(i+1)}^{(2)}, \ldots, x_{k(j-1)}^{(2)}, x_{kj}^{(1)}, x_{k(j+1)}^{(2)}, \ldots, x_{kp}^{(2)}\right)$$

L'indice S_{ij} est alors estimé par :

$$\hat{S}_{ij} = \frac{\hat{U}_{ij} - \hat{f}_0^2 - \hat{V}_i - \hat{V}_j}{\hat{V}}$$

Et ainsi de suite pour les indices de sensibilité d'ordre supérieur.

5.4.2 La méthode FAST (Fourier Amplitude Sensitivity Test)

La méthode FAST (Fourier Amplitude Sensitivity Test) a été développée par Cukier et al. [Cuk73, Cuk75] et [Cuk78], ainsi que Schaibly et Shuler [Sch73]. Considérons une fonction

$$f(x) = f(x_1, \ldots, x_p)$$

où $x \in [0,1]^p$, et le modèle à variables aléatoires $Y = f(X_1, \ldots, X_p)$ associé. Cukier et al. montrent qu'il est possible d'obtenir une décomposition de la variance de Y, semblable à la décomposition de Sobol [Jac05], en utilisant la transformée de Fourier multi-dimensionnelle de f. Le calcul d'une telle décomposition multi-dimensionnelle étant trop complexe pour être réalisé en pratique, l'idée de la méthode FAST est de remplacer les décompositions multi-dimensionnelles par des décompositions unidimensionnelles le long d'une courbe parcourant l'espace $[0,1]^p$. Cette courbe est définie par un ensemble d'équations paramétriques :

$$x_i(s) = g_i(\sin(\omega_i s)) \text{ pour } i = 1, \ldots, p,$$

où g_i sont des fonctions à déterminer, permettant un recouvrement uniforme de $[0,1]^p$, et où $(\omega_1, \ldots, \omega_p) \in \mathbb{N}^{*p}$ est un ensemble de fréquences entières linéairement indépendantes (aucune n'est une combinaison linéaire des autres). Ainsi, lorsque s varie dans $\mathbb{R}$, le vecteur $(x_1(s), \ldots, x_p(s))$ décrit une courbe qui parcourt $[0,1]^p$. Cukier et al. montrent que l'on a alors :

$$f_0 = \int_{[0,1]^p} f(X)\, dx = \lim_{T \to \infty} \frac{1}{2\pi} \int_{-T}^{T} f(X(s))\, ds$$

Les fréquences $(\omega_1, \ldots, \omega_p)$ étant entières, la courbe ne remplit pas l'espace $[0,1]^p$ mais est périodique de période 2π, d'où :

$$f_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(X(s))\, ds$$

Si on applique ces idées au calcul de la variance V d'un modèle :

$$Y = f(X_1, \ldots, X_p),$$

en notant $f_0 = E[Y]$, on obtient

$$\begin{aligned} V &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f^2(X(s))\,ds - f_0^2 \\ &\simeq \sum_{j=-\infty}^{\infty}(A_j^2 + B_j^2) - (A_0^2 + B_0^2) \\ &\simeq 2\sum_{j=1}^{\infty}(A_j^2 + B_j^2) \end{aligned}$$

où A_j et B_j sont des coefficients de Fourier définis par :

$$A_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(X(s))\cos(js)\,ds,$$

$$B_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(X(s))\sin(js)\,ds.$$

Cukier et al. [Cuk78] expliquent en utilisant une analyse Hilbertienne compliquée, que la part de la variance due à une variable X_i est la somme des carrés des coefficients de Fourier A_j et B_j attribués à la fréquence ω_i relative à X_i et à ses harmoniques :

$$V_i = 2\sum_{p=1}^{\infty}(A_{p\omega_i} + B_{p\omega_i}).$$

L'indice de sensibilité S_i est alors défini par :

$$S_i = \frac{\sum_{p=1}^{\infty}(A_{p\omega_i} + B_{p\omega_i})}{\sum_{j=1}^{\infty}(A_j^2 + B_j^2)}$$

où les intégrales A_j et B_j sont estimées par une méthode de Monte-Carlo classique, et où M, harmonique maximum considérée, est évaluée en fonction des propriétés suivantes :

⊙ plus M est grand, mieux les indices reflètent l'effet des variables,

⊙ mais aussi plus M est grand, plus le nombre de simulations sera élevé.

Le choix de M revient à faire un compromis entre la qualité des indices et le coût de leur estimation. Cukier et al.[Cuk78] ont déterminé de façon empirique que le meilleur compromis pour M était 4 ou 6, et ce quelque soit la dimension du modèle.

La définition des fonctions g_i et les fréquences ω_i utilisés dans l'estimation des indices de sensibilité ainsi que le choix du paramètre M, sont détaillés dans [Sal00] et [Jac05].
Les méthodes de Sobol et FAST sont les deux méthodes les plus couramment utilisées en analyse de sensibilité pour estimer les indices de sensibilité. Toutefois, une troisième méthode, antérieure à celles-ci, a été proposée par McKay[McK79].

5.5 L'analyse de sensibilité appliquée à la mesure de la bande passante disponible

5.5.1 Le modèle de mesure de la bande passante disponible

Dans le chapitre 3, nous avons présenté un nouveau modèle déterministe basé sur les délais des paquets sondes pour mesurer la bande passante disponible dans un chemin de bout en bout. Ce modèle établit une relation entre la quantité du trafic concurrent qui s'infiltre entre les paquets de la paire et la dispersion inter-paquets observée tout en tenant compte de la taille des paquets sondes utilisés pour analyser le chemin.

A partir de deux équations différentes, l'une définissant les instants d'arrivée des paquets à un nœud

$$t_l^k = t_0^k + \sum_{i=0}^{l-1} \left(\frac{s^k}{C_i} + d_i + q_i^k \right) \tag{5.4}$$

et l' autre définissant les délais d'attente des paquet dans les files d'attente

$$q_l^k = max \left(0, t_{l+1}^{k-1} - d_l - t_l^k \right), \tag{5.5}$$

nous avons défini la formule générique pour l'estimation de la bande passante disponible dans un chemin réseau de bout en bout :

$$A = \frac{s^{k-1} \left(2t_0^k + t_n^{k-1} - 2t_0^{k-1} - t_n^k \right)}{\left(t_0^k - t_0^{k-1} \right) \left[\frac{\left(s^k - s^{k-1} \right)}{\sum_{i=0}^{b-1} \left(\frac{1}{C_i} \right)} + \frac{s^{k-1}}{C} + \left(t_n^k - t_n^{k-1} \right) \right]} \tag{5.6}$$

Cette dernière formule est exprimée en fonction de plusieurs paramètres d'entrée les uns plus importants que d'autres. Pour des raisons pratiques ce modèle à été simplifié en considérant un cas particulier où les deux paquets de la paire ont la même taille. On a obtenu alors la formule

suivante :

$$A = \frac{S\,(2\Delta_{in} - \Delta_{out})}{(\Delta_{in})\left[\left(\frac{S}{C}\right) + \Delta_{out}\right]} \tag{5.7}$$

Cette dernière formule à été implémentée dans un outil de mesure de la bande passante disponible appelé IGMPS. L'outil IGMPS à été testé sur une plateforme de mesure selon plusieurs scénarios et les résultats obtenus (que nous rappelons ici dans la figure 5.3) ont démontré que ce dernier permet de mesurer la bande passante disponible avec une grande précision. Toutefois, lorsque le taux d'utilisation du lien bottleneck est trop élevé, IGMPS montre des résultats un peu moins précis en surestimant la valeur de la bande passante disponible de 29 %. (dans les autres cas l'erreur de mesure est aux alentours de 5%).

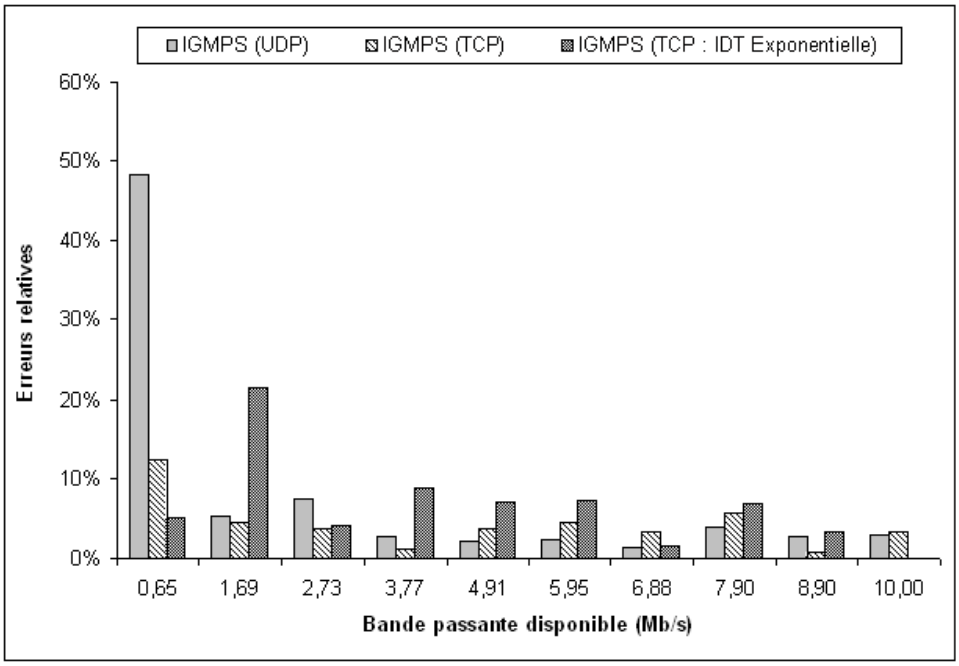


FIG. 5.3 – Erreurs relatives des mesures d'IGMPS dans les différents scénarios.

Dans ce qui suit, nous tenterons de déterminer l'origine de ces erreurs et de voir quels sont les paramètre d'entrée qui participent à la perturbation de la sortie de notre modèle causant ainsi une imprécision dans les mesures d'IGMPS. Pour ce faire, nous appliquerons l'une des méthodes de l'analyse de sensibilité sur notre modèle (équation 5.6) puis nous appliquerons la même méthode sur le modèle simplifié (équation 5.7) pour voir si les paramètres incriminés dans le modèle générique sont les mêmes qui influencent la sortie du modèle simplifié. Cette étude nous permettra de désigner les paramètres les plus importants dans notre modèle, ce qui nous permettra d'améliorer la précision de l'outil IGMPS en portant une attention particulière aux paramètres incriminés par l'analyse de sensibilité et en mesurant ces derniers avec beaucoup plus de précision.

Pour effectuer l'analyse de sensibilité sur notre modèle nous n'avons pas besoin de développer

notre propre outil étant donné que plusieurs outils dédiés ont été déjà développés et qui sont couramment utilisés par la communauté scientifique dans les domaines de l'évaluation des risques, la prise de décision et la validation de modèles, etc. Parmi ces outils on peut citer par exemple GEM-SA [Hag06], SimEnv [Fle05] et Simlab [Sal97]. Pour définir les indices de sensibilité de notre modèle nous avons choisi d'utiliser l'outil Simlab de Saltelli car par rapport aux autres outils ce dernier offre une interface graphique simple et une documentation disponible qui facilitent son utilisation. L'outil Simlab et ses différents modules sont décrits dans la section suivante.

5.5.2 L'outil Simlab pour l'analyse de sensibilité

Simlab est un outil de simulation utilisé pour calculer les indices de sensibilité et étudier les incertitudes de la sortie d'un modèle par rapport à ses paramètres d'entrée en se basant essentiellement sur les méthodes à décomposition de variance (Monte Carlo, FAST, etc).
Les méthodes d'analyse de sensibilité basées sur Monte Carlo consistent à simuler l'exécution du modèle un certain nombre (plus au moins élevé) de fois en utilisant des paramètres d'entrée dont la distribution de probabilité a été bien définie. Les résultats de cette évaluation sont ensuite utilisés pour :

1. Calculer les incertitudes à la sortie du modèle (prédire le comportement du modèle) en connaissant seulement les distributions de probabilité des paramètres d'entrée.
2. Désigner les variables d'entrée responsables de ces incertitudes et calculer les indices de sensibilité correspondant.

En général, l'analyse à base de méthodes à décomposition de variance est effectuée en 4 étapes sur Simlab :

1. Définition des paramètres d'entrée ainsi que leur distribution de probabilité.
2. A partir des distributions choisies pour les paramètres d'entrée dans la première étape, un ensemble de points est sélectionné en utilisant une méthode d'échantillonnage prédéfinie.
3. Dans cette étape, les simulations du modèle sont lancées en utilisant l'ensemble des points échantillonnés comme entrée. Un ensemble de points de sortie correspondant aux différents points d'entrée est produit.
4. A partir de ces points de sortie du modèle le calcul d'incertitudes et des indices de sensibilité est effectué.

L'outil Simlab est composé de trois modules :

1. Le module du pré-processeur statistique qui couvre la première et la deuxième étape.
2. Le module de spécification du modèle qui effectue la troisième étape.
3. Le module post processeur statistique qui couvre la quatrième étape.

Simlab est disponible sur l'environnement Windows, à son démarrage il offre une interface composée de trois panels correspondant aux différents modules définis précédemment :

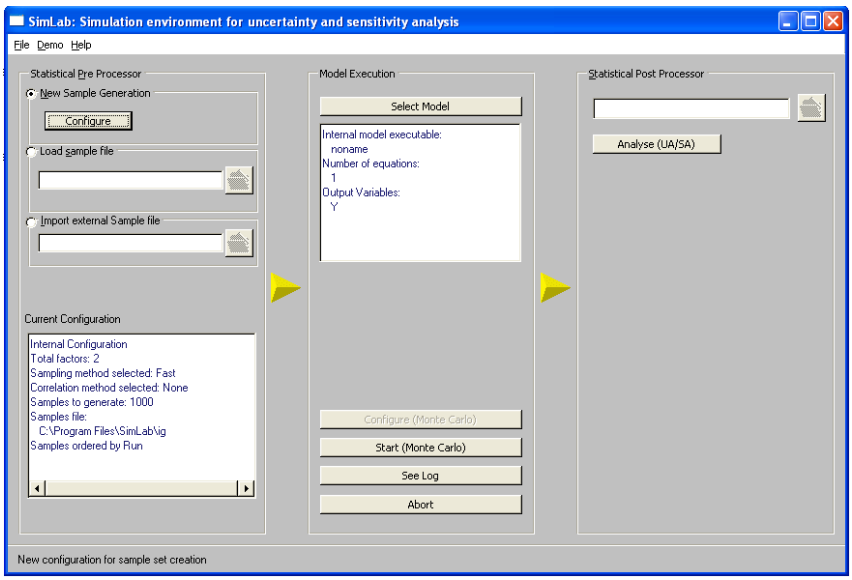


FIG. 5.4 – L'interface de Simlab et les différents modules la composant.

Le pré-processeur statistique

Le pré *processeur* permet à l'utilisateur de définir la liste des paramètres d'entrée du modèle ainsi que la fonction de distribution de probabilité de chacun d'entre eux. Il permet aussi d'établir une éventuelle relation entre les paramètres corrélés. Le pré-processeur dispose d'un grand nombre de fonctions de distribution pour les paramètres (normale, uniforme, etc).
Une fois que le paramètre est défini et que sa distribution est choisie, l'utilisateur sélectionne la méthode d'échantillonnage (méthode d'analyse de sensibilité) à utiliser ainsi que le nombre d'exécution du modèle pour générer l'ensemble des points de sortie de ce dernier après les simulations. De nombreux choix de méthodes d'analyse de sensibilité sont disponibles dans Simlab et parmi elles on trouve la méthode de Monte Carlo avec échantillonnage aléatoire. Bon nombre d'autres méthodes alternatives ont été proposées pour améliorer la convergence, nous pouvons citer par exemple, les méthodes de simulation pseudo probabilistes qui consistent en un échantillonnage non totalement aléatoire comme la méthode par Hypercube Latin (LHS) [McK79] ou la méthode de Quasi Monte Carlo [Owen98] basée sur les séquences LP_τ de Sobol [Sal00]. Simlab permet aussi d'utiliser les méthodes FAST [Sal00] et FAST étendue [SB98], la méthode de Morris [Jac05] et finalement il permet l'utilisation d'un ensemble de valeurs réelles connues ou mesurées pour les paramètres d'entrée.

Le module de spécification du modèle

Ce module permet à l'utilisateur de définir le modèle sur lequel seront effectués les calculs des indices de sensibilité ainsi que les incertitudes. Simlab dispose d'un éditeur d'équations qui permet de définir des modèles assez simples. En revanche, pour des modèles beaucoup plus compliqués, ce dernier permet d'importer le modèle à partir d'un fichier externe dont le format est spécifique à Simlab ou bien en utilisant des packages mathématiques ou statistiques prédéfinis comme ceux qui sont utilisés dans Matlab par exemple.
Une fois que le modèle est défini, l'utilisateur peut lancer les simulations pour obtenir l'ensemble des points de sortie du modèle sur lequel seront effectués l'analyse de sensibilité et le calcul d'incertitudes.

Le post processeur statistique

Ce module permet d'effectuer la dernière étape de l'analyse en utilisant Simlab. Ce module utilise l'ensemble des points de sortie résultant des simulations sur le modèle pour effectuer l'analyse de sensibilité et le calcul d'incertitudes sur la sortie du modèle. L'analyse de sensibilité est effectuée dans ce module par la méthode choisie dans le pré processeur statistique. Les résultats sont fournis à l'utilisateur sous forme de tableaux et de courbes.

5.5.3 Choix de la méthode de l'analyse de sensibilité

L'analyse comparative des différentes méthodes d'analyse de sensibilité a été effectuée dans [Jac05]. Elle est fondée sur un ensemble d'applications numériques ainsi que sur une étude bibliographique approfondie. Elle montre que la méthode de McKay ne permet d'estimer que les indices de premier ordre et est relativement lourde en temps de calcul, même pour un échantillon de petite taille. La méthode FAST permet d'estimer les indices de sensibilité de premier ordre pour tous types de modèles. La comparaison a montré que l'un des avantages de cette méthode est que les indices de sensibilité peuvent être calculés indépendamment les uns des autres, à partir d'un même échantillon de simulations, ce que ne permet pas la méthode de Sobol qui nécessite deux échantillons. Par contre, la méthode de Sobol étant stochastique, elle permet d'obtenir un intervalle de confiance sur les estimations d'indices, ce que ne permet pas FAST, puisque pour une série de fréquences donnée, les estimations d'indices sont déterministes. Saltelli et Bolado [SB98] ont comparé FAST et Sobol sur un certain nombre de modèles. Ils ont conclu que FAST était, d'un point de vue complexité de calcul, plus avantageuse que Sobol. Néanmoins, FAST est parfois sujette à un biais (que Saltelli et Bolado assigne au choix des fréquences), tandis que Sobol converge toujours vers la vraie valeur des indices de sensibilité.

En s'appuyant sur les résultats de cette comparaison, nous avons choisi d'utiliser la méthode Sobol pour estimer les indices de sensibilité de notre modèle aux paramètres d'entrée car c'est cette dernière qui offre les résultats les plus précis. La méthode de Sobol est un peu gourmande en temps de calcul mais ceci ne devrait pas poser de problème étant donné que notre modèle

n'est pas trop complexe et que le nombre de paramètres étudiés n'est pas trop élevé.

5.5.4 Méthodologie

Afin d'effectuer l'analyse de sensibilité sur le modèle proposé, nous avons choisi d'utiliser l'outil *Simlab* avec la méthode de *Sobol*. Nous allons commencer notre analyse sur le modèle générique de mesure de la bande passante disponible donné par la formule 5.6. Comme nous l'avons vu précédemment, cette formule fait intervenir plusieurs paramètres importants tels que la taille des paquets sondes, leurs instants de départ et d'arrivée ainsi que la capacité du lien bottleneck. Cette formule fait aussi intervenir le paramètre

$$(S^k - S^{k-1})/(\sum_{i=0}^{b-1} 1/C_i)$$

qui nécessitent la connaissance des capacités des liens se trouvant en amont du lien bottleneck ($C_i, i = 0, \ldots, b-1$ avec b le lien bottleneck).

Dans cette analyse il serait plus intéressant d'étudier l'effet des dispersions initiales et finales représentées par ($\Delta_{in} = t_0^k - t_0^{k-1}$) et ($\Delta_{out} = t_n^k - t_n^{k-1}$) que d'étudier l'effet des instants de départ et d'arrivée eux-mêmes. Ceci est dû au fait que la technique utilisée dans IGMPS est une technique à dispersions de paquets. Il serait donc beaucoup plus judicieux de faire intervenir ces paramètres dans l'étude du modèle générique proposé. En ce qui concerne les instants de départ et d'arrivée des paquets sondes, leurs éventuelles erreurs et leurs effets sur la mesure de la bande passante disponible seront étudiés dans la section 5.6.

Au lieu donc d'utiliser la formule 5.6 et pour prendre en compte les différentes dispersions des paquets sondes, nous utilisons pour notre analyse la formule suivante :

$$A = \frac{s^{k-1}\,(2\Delta_{in} - \Delta_{out})}{\Delta_{in}\left[\dfrac{\left(s^k - s^{k-1}\right)}{\sum_{i=0}^{b-1}\left(\dfrac{1}{C_i}\right)} + \dfrac{s^{k-1}}{C} + \Delta_{out}\right]} \tag{5.8}$$

avec ($\Delta_{in} = t_0^k - t_0^{k-1}$) et ($\Delta_{out} = t_n^k - t_n^{k-1}$).

Comme mentionné précédemment, le modèle générique proposé fait intervenir le paramètre $(S^k - S^{k-1})/(\sum_{i=0}^{i=b-1} 1/C_i)$ où l'indice i compte le nombre de liens en amont du lien bottleneck. Étant donné que les capacités des liens considérés dans le modèle sont supposées être mesurées en utilisant l'un des outils spécifiques à cette métrique comme par exemple *Nettimer* et *Cprobe* (dont les mesures sont entachées d'erreurs) alors, à priori, plus le nombre de liens en amont du lien

bottleneck est élevé plus l'erreur sur la mesure de la bande passante disponible est importante. Afin de prendre en compte l'effet du nombre i de liens en amont du lien bottleneck, nous considérons dans notre étude plusieurs scénarios : un scénario avec $i = 2$, un autre scénario avec $i = 5$ et finalement un scénario avec $i = 10$. Les chemins réseaux de bout en bout correspondant à chacun de ces scénarios sont représentés dans la figure 5.5.

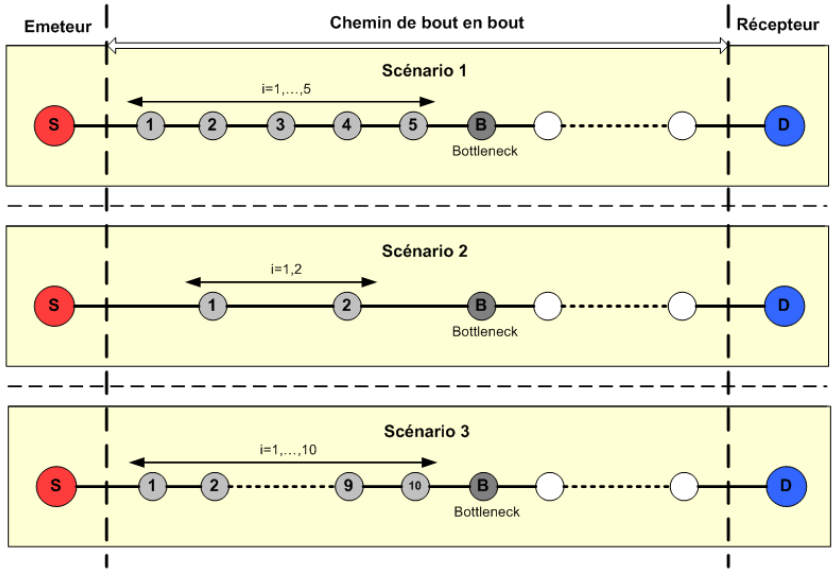


FIG. 5.5 – Les chemins réseaux considérés dans les différents scénarios.

Dans l'étude du modèle générique, le paramètre $(S^k - S^{k-1})/(\sum_{i=0}^{i=b-1} 1/C_i)$ ne doit pas être nul. Donc, dans ce cas la taille S^k du paquet k doit être différent de la taille S^{k-1} du paquet $k-1$, $(S^k \neq S^{k-1})$. Nous avons donc choisi d'utiliser dans le modèle générique des paires de paquets avec la taille du premier paquet $S^{k-1} = 1000$ octets et la taille du deuxième paquet $S^k = 1500$ octets.

Notre analyse de sensibilité se portera ensuite sur le modèle simplifié en considérant le cas particulier du modèle générique où les tailles des paquets sondes sont égales $(S^k = S^{k-1} = S)$. Le modèle simplifié est donné par la formule 5.7. les résultats de l'analyse du modèle générique et ceux du modèle simplifié seront ensuite comparés. Nous terminons ensuite notre étude par une analyse de sensibilité appliquée au *Probe Gap Model* implémenté dans l'outil *Spruce*. Une comparaison des résultats de ce dernier à ceux obtenus pour IGMPS nous permettra de vérifier si ces deux modèles sont sensibles aux mêmes paramètres d'entrée.

5.5.5 Les distributions des paramètres d'entrée du modèle

Conformément aux étapes définies dans la section 5.5.2 pour faire une analyse de sensibilité en utilisant Simlab, il est nécessaire de définir les différents paramètres d'entrée du modèle au niveau du module pré-processeur statistique et de choisir la distribution de probabilité de chacun

d'entre eux. Le tableau 5.1 récapitule l'ensemble des paramètres d'entrée utilisés par IGMPS ainsi que leurs définitions et leurs distributions de probabilités. Ces paramètres et leurs distributions sont appliqués aussi pour l'analyse de sensibilité du *Probe Gap Model* présentée dans la section 5.5.6.

Paramètre	Définition	Distribution
C	Capacité du lien bottleneck	Normale(10 e+006, 1 e+006)
C_i	Capacité du lien i en amont du bottleneck	Normale(100 e+006, 5 e+006)
s^k	Taille du paquet k	1500 octets
s^{k-1}	Taille du paquet $k-1$	1000 octets
Δ_{in}	Dispersion initiale	Normale(1,2 e-003, 6 e-005)
Δ_{out}	Dispersion finale	Normale(1,2 e-003, 6 e-005)

TAB. 5.1 – Définition des différents paramètres et leurs distributions de probabilités

Le choix des paramètres des distributions de probabilité pour chaque entrée du modèle est fait de sorte à reproduire dans les simulations lors de l'analyse de sensibilité les mêmes conditions que celles offertes par la plateforme d'expérimentation lors de la validation d'IGMPS.
Les distributions de probabilités des paramètres d'entrée sont représentées par les figures 5.6, 5.7 et 5.8 :

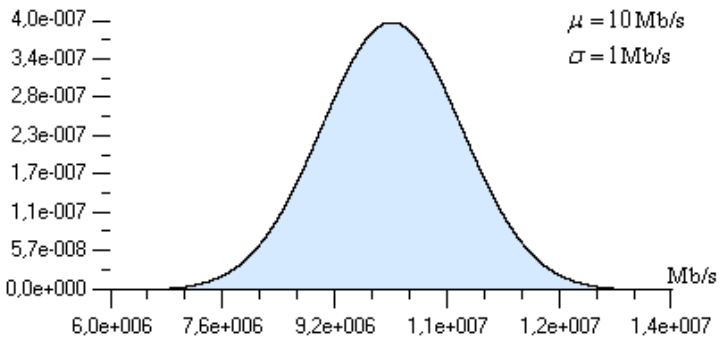


FIG. 5.6 – Distribution de probabilité de la capacité du bottleneck.

Étant donné que les tailles des paquets sondes S^k et S^{k-1} sont connues avec exactitude ($S^k = 1500$ octets) et ($S^{k-1} = 1000$ octets), ces deux paramètres ne devraient pas intervenir dans la perturbation de la sortie du modèle générique. Donc, à priori, aucun indice de sensibilité ne sera calculé pour ces deux paramètres (cette remarque est valable pour le paramètre S représentant la taille des paquets sondes dans le cas du modèle simplifié avec $S = S^k = S^{k-1} = 1500$ octets).

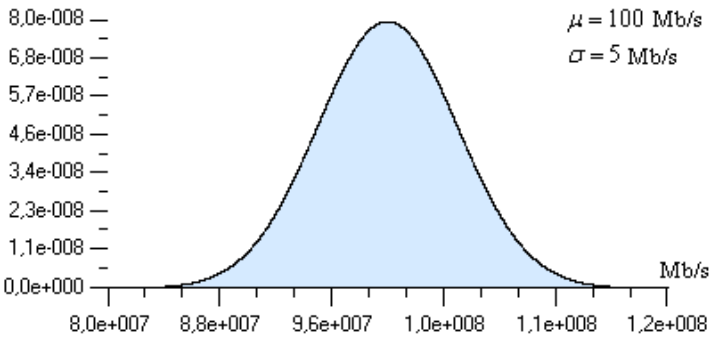


FIG. 5.7 – Distribution de probabilité des capacité des liens en amont du bottleneck.

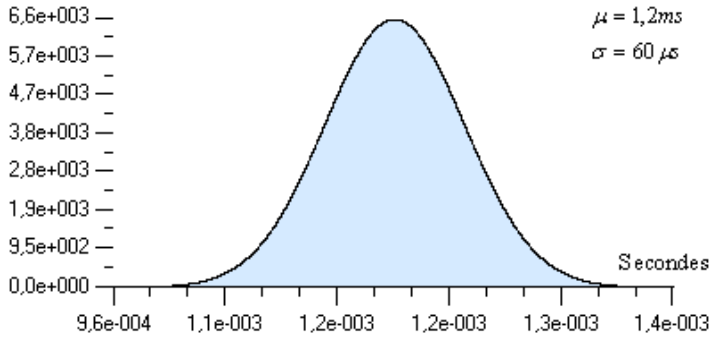


FIG. 5.8 – Distribution de probabilité des dispersions initiale et finale.

5.5.6 Les résultats

Le modèle générique

Les résultats de l'analyse de sensibilité sur le modèle générique sont obtenus en utilisant Simlab et en appliquant la méthode de Sobol selon les trois scénarios expliqués précédemment. Dans le premier scénario nous avons considéré un chemin de bout en bout avec 5 liens en amont du lien bottleneck. Les résultats de l'analyse de sensibilité pour ce scénario sont présentés dans la figure 5.9.

La figure 5.9 montre que les dispersions inter-paquets initiale et finale sont principalement à l'origine des perturbations et des incertitudes sur la sortie du modèle proposé pour l'estimation de la bande passante disponible avec respectivement, 41% et 56% de part de responsabilité. De faibles erreurs de mesure sur ces paramètres engendreront des perturbations considérables sur l'estimation de la bande passante disponible vue le degré de sensibilité élevé de la sortie aux perturbations de ces deux paramètres. En revanche, nous avons constaté que la capacité du lien bottleneck est un paramètre qui n'influence pas beaucoup la mesure de la bande passante disponible en utilisant IGMPS. Aussi, les capacités des liens en amont du lien bottleneck participent à la perturbation de la sortie de notre modèle mais pas avec un grand niveau d'influence (de l'ordre de 1% de la perturbation globale).

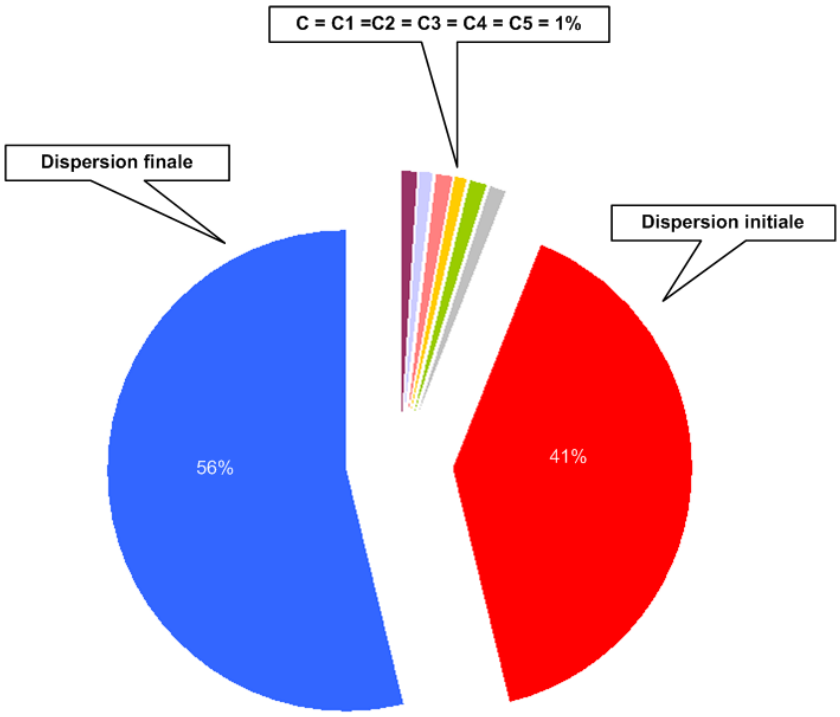


FIG. 5.9 – Scénario 3 : dix liens en amont du lien bottleneck.

Dans les deux autres scénarios nous avons considéré un chemin de bout en bout avec 2 et 10 liens en amont du lien bottleneck. Les résultats correspondant sont représentés respectivement sur la figure 5.10 et la figure 5.11.

Ces deux figures montrent les mêmes tendances que pour le premier scénario ; c'est-à-dire les paramètres dispersions initiale Δ_{in} et dispersion finale Δ_{out} sont à l'origine des perturbations enregistrées à la sortie du modèle générique. Ces résultats montrent aussi que l'influence de la perturbation de la capacité du bottleneck et des capacités des liens en amont de ce dernier sont minimes. Toutefois, si le nombre de ces liens est trop élevé (troisième scénario) l'influence de leur perturbation commence à se ressentir au niveau de la sortie du modèle. En effet, ce n'est pas la perturbation individuelle de chacune de ces capacités qui pose problème (étant donné que chacune d'elles ne participe à la perturbation de la sortie qu'avec 1% d'erreurs) mais c'est plutôt l'accumulation de ces dernières qui risque de perturber la sortie du modèle. Cependant, cette perturbation reste faible par rapport à celle de Δ_{in} et de Δ_{out}.

Comme mentionné précédemment dans nos hypothèses, les tailles des paquets sondes S^k et S^{k-1} n'apparaissent pas dans le calcul des indices de sensibilité. En effet, ces dernières, étant connues avec exactitude, elles n'ont aucune influence sur la perturbation de la sortie du modèle.

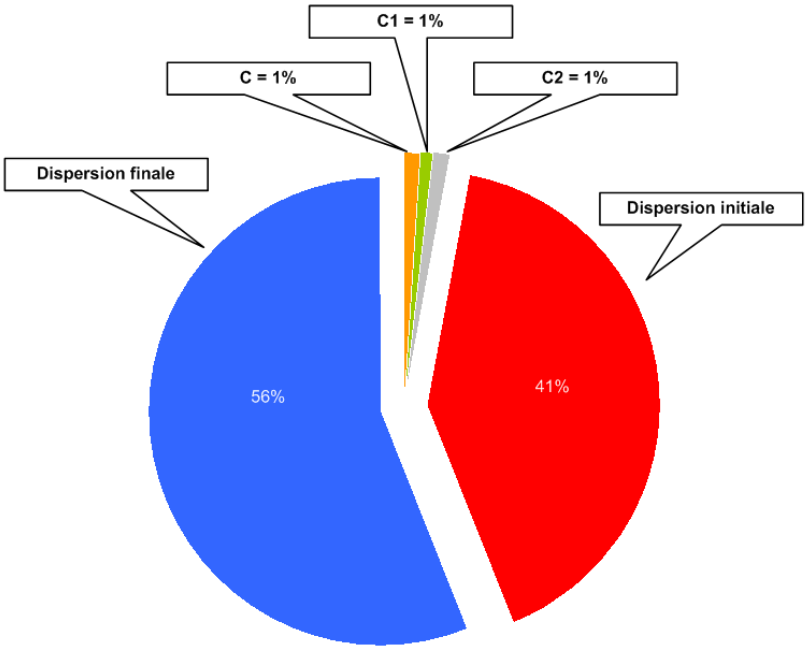


FIG. 5.10 – Scénario 2 : deux liens en amont du lien bottleneck.

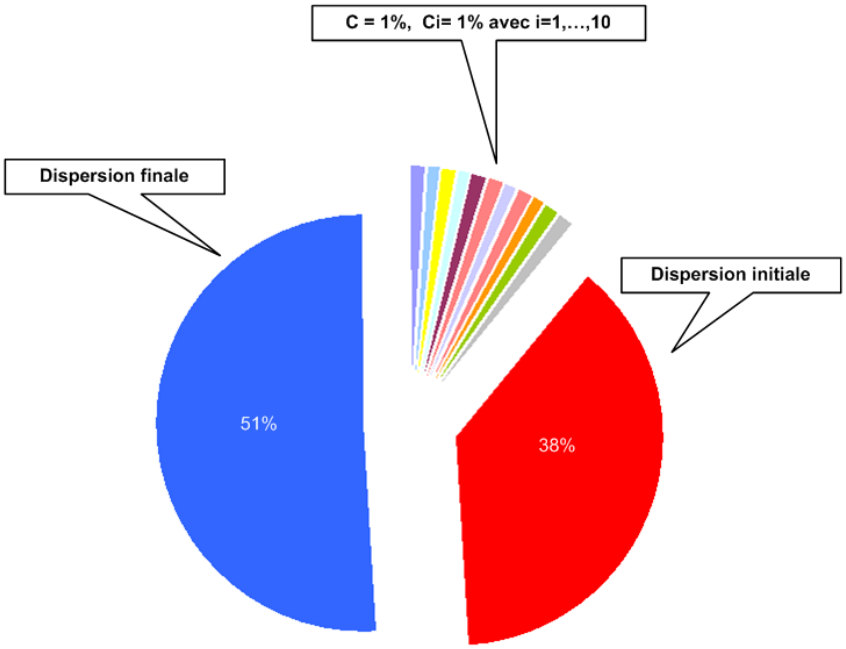


FIG. 5.11 – Scénario 3 : dix liens en amont du lien bottleneck.

Remarque

Dans les conditions réelles de l'Internet, un chemin réseau de bout en bout est généralement composé de moins de 16 liens. Donc, nous supposons que le nombre de liens en amont du lien bottleneck est souvent inférieur à 15. Donc, l'influence de la perturbation de l'ensemble des

capacités considérées dans le modèle générique est inférieure à 16% de la perturbation globale enregistrée sur la sortie de ce dernier ($\sum_{i=0}^{b-1} \Delta C_i \leq 16\%$ de ΔA). Ceci reste à démontrer avec des expérimentations dans les conditions réelles de l'Internet.

Le modèle simplifié

Le modèle simplifié implémenté dans IGMPS est donné par la formule 5.6. Ce dernier est un cas particulier du modèle générique où les tailles des deux paquets de la paire sont égales ($S^k = S^{k-1}$). Ce modèle ne prend pas en compte les capacités des liens en amont du lien bottleneck. Les résultats de l'analyse de sensibilité sur le modèle simplifié sont présentés dans la figure 5.12.

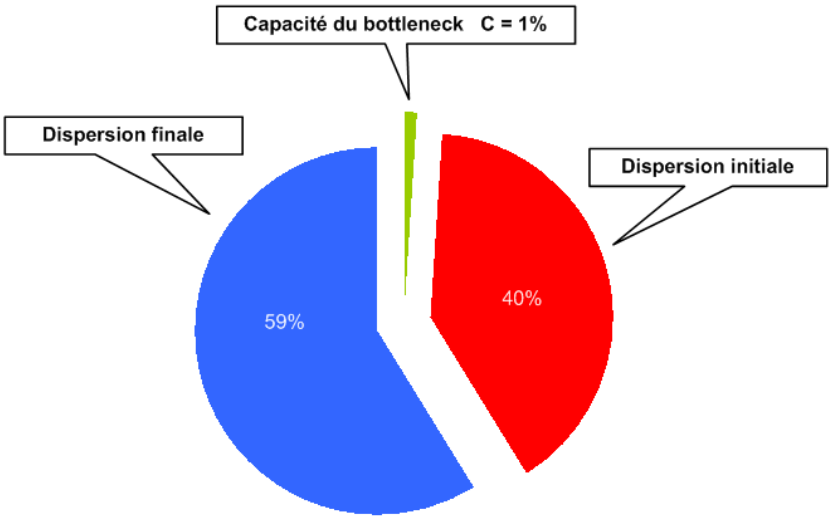


FIG. 5.12 – Analyse de sensibilité du modèle simplifié.

Ces résultats ressemblent beaucoup à ceux obtenus pour le modèle générique. En effet, ces résultats montrent que les erreurs sur la sortie de ce modèle sont dues principalement aux erreurs sur les dispersions initiale et finale (avec respectivement 40% et 59% de l'erreur globale) et que la contribution de la capacité du lien bottleneck est négligeable par rapport à ces derniers.

Le Probe Gap Model (Spruce)

Le *Probe Gap Model* implémenté dans l'outil Spruce utilise presque les mêmes paramètres d'entrée définis dans la section 5.5.5. Dans l'analyse de sensibilité de ce modèle, les paramètres d'entrée sont définis dans *Simlab* avec les mêmes distributions de probabilités que celles choisies pour le modèle générique. Les résultats de cette analyse sont représentés dans la figure 5.13.

Les résultats obtenus lors de l'analyse de sensibilité du Probe Gap Model sont complètement différents de ceux obtenus pour les modèles génériques et simplifié. Alors que la capacité du lien bottleneck a un effet négligeable sur les incertitudes des sorties des modèles précédents, les erreurs de mesures de ce paramètre affectent considérablement la sortie du *Probe Gap Model*. En

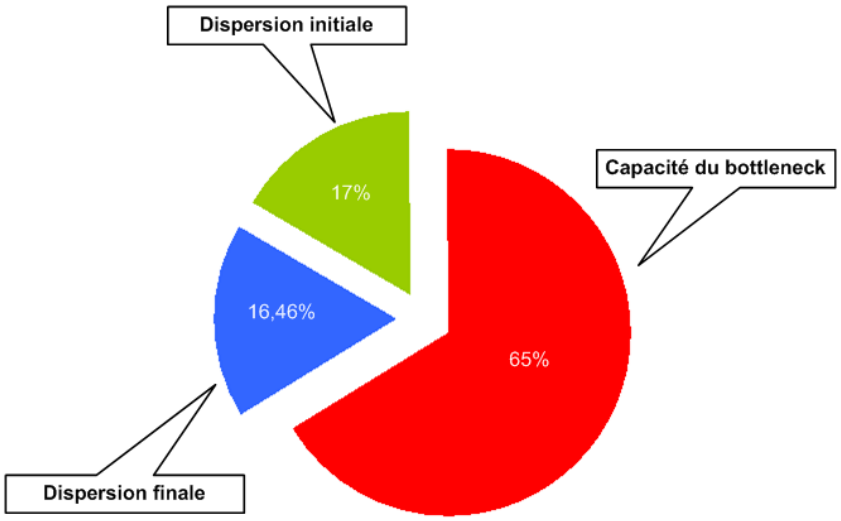


FIG. 5.13 – Analyse de sensibilité du Probe Gap Model (Spruce).

effet, dans ce modèle, la capacité du lien bottleneck est supposée être mesurée en utilisant des outils spécifiques (*Nettimer* ou *bprobe* par exemple). Ces outils offrent des estimations souvent imprécises de la capacité du chemin de bout en bout. Ces erreurs de mesures sont responsables de 65% de l'erreur globale obtenues sur la sortie du *Probe Gap Model*.

En revanche, les indice de sensibilité calculés pour les dispersions initiale et finale sont moins importantes que ceux calculés pour les modèles précédents. Toutefois, ces derniers sont suffisamment élevés pour affecter considérablement la sortie du *Probe Gap Model* (respectivement de l'ordre de 17% et 16,46% de l'erreur globale).

Donc contrairement à IGMPS qui n'est sensible qu'aux dispersions inter-paquets, Spruce est sensible à tous ses paramètres d'entrée et plus particulièrement à la capacité du lien bottleneck.

Analyse

L'utilisation de l'analyse de sensibilité pour étudier les différents modèles considérés (modèle générique, modèle simplifié et le Probe Gap Model) a permis de désigner les paramètres d'entrée les plus importants pour chacun d'entre eux. Les indices de sensibilité calculés nous ont renseigné sur le taux de contribution de chacun de ces paramètres sur l'erreur globale à la sortie de chacun de ces modèles. Pour les modèles que nous avons proposés (générique et simplifié), ce sont les erreurs de mesure des dispersions inter-paquets au niveau de l'émetteur et du récepteur qui posent problème. En effet, une petite perturbation de ces paramètres engendrera des perturbations considérables sur la sortie de ces modèles. En revanche, les erreurs de mesure de la capacité du lien bottleneck et des capacités des liens en amont de ce dernier ont un effet négligeable par rapport aux dispersions initiales et finales.

Le passage du modèle générique au modèle simplifié (qui est un cas particulier de ce dernier) comporte en pratique deux avantages. En effet, étant donné que les mesures sont réalisées de bout en bout nous sommes donc sensés ne pas avoir à mesurer les paramètres des liens intermédiaires constituant le chemin entre la source et la destination. Pour respecter cette condition, nous avons implémenté le modèle simplifié dans IGMPS (qui nécessite seulement la connaissance de la capacité de bout en bout) au lieu du modèle générique. Le deuxième avantage est souligné par l'analyse de sensibilité. En effet comme mentionné précédemment, lorsque le nombre de liens en amont du bottleneck est élevé, l'accumulation des erreurs de mesure de leur capacités affecte la sortie du modèle générique. Donc l'utilisation d'un modèle qui ne nécessite pas la mesure de ces capacités réduira considérablement les erreurs de mesures de la bande passante disponible dans ce genre de cas.

Cette analyse de sensibilité montre aussi qu'en plus d'être le plus précis, IGMPS a un avantage supplémentaire sur Spruce. En effet la précision de ce dernier dépend largement de la capacité du lien bottleneck mesurée par des outils externes tel que Nettimer par exemple. L'amélioration de la précision de cet outil passe donc nécessairement par l'amélioration de la précision des outils de mesure de la capacité des chemins de bout en bout (les performances ne dépendent pas de l'outil lui-même).

5.6 Analyse des erreurs et des incertitudes de mesures

5.6.1 Incertitudes à l'envoi des paquets

Les outils de mesure étudiés dans cette section nécessitent d'envoyer des paquets en respectant un temps inter-paquets précis ou à un débit donné. Le non-respect de ces contraintes temporelles est susceptible de générer une incertitude dans les mesures. Cette incertitude dépend de la latence pour estampiller un paquet, le déplacer de l'espace utilisateur vers l'espace noyau, c'est-à-dire de l'application au pilote de l'interface réseau, et le transmettre sur l'interface réseau. De plus, l'ordonnanceur du système d'exploitation peut assigner les ressources à d'autres processus entre les opérations d'estampillage et d'envoi. Ce phénomène peut être atténué de façon significative s'il est pris en compte lors de la conception du logiciel (affectation d'une priorité élevée à la tâche d'envoi, programmation temps-réel, etc.) .

Nous avons estimé la latence correspondant au passage d'un paquet de l'espace utilisateur à l'espace noyau. Cette latence est représentée sur la figure 5.14. La latence à mesurer est $\Delta_t = t_f - t_d$.

Les différentes opérations d'estampillage sont effectuées par l'appel système *gettimeofday()*. Cet appel ajoute une latence supplémentaire dans la mesure de Δ_t. En effet, la date t_d sera estimée par t_1, et t_f par t_2. En notant $t_{gettimeofday()}$ le temps d'exécution de l'appel *gettimeofday()*,

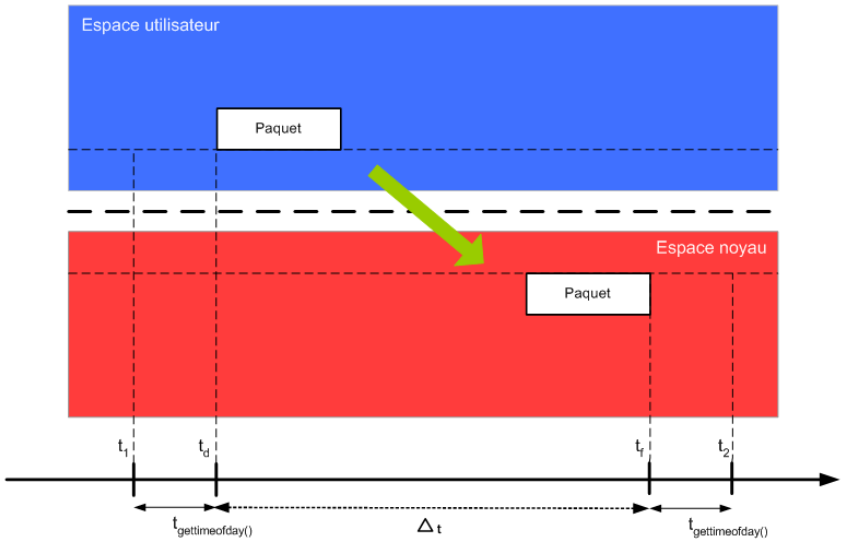


FIG. 5.14 – Passage du paquet de l'application au pilote de l'interface réseau.

Taille des paquets IP (octets)	Nombre d'expériences	$t_2 - t_1(\mu s)$	$\Delta t(\mu s)$
500	1000	0 - 41	0 - 39
1500	1000	0 - 50	0 - 48

TAB. 5.2 – Mesure de la latence des paquets entre l'espace utilisateur et les buffers de l'interface réseau

on a alors

$$\Delta_t = t_2 - t_1 - 2t_{gettimeofday()} \tag{5.9}$$

Estimation du temps d'exécution de l'appel système gettimeofday()

Le temps d'execution de *gettimeofday()* a été estimé en réalisant un programme qui exécute successivement deux appels à la fonction et qui calcule la différence des valeurs renvoyées. Cette expérience a été répétée 100000 fois. Les temps d'exécution obtenus varient de 1 à $6\mu s$. Il faut noter que ces valeurs dépendent de la puissance de la machine sur laquelle sont effectuées les mesures, dans notre cas nous avons utilisé des Pentium IV, 2, 5 GHz.

Estimation du temps de passage de l'application au pilote de l'interface réseau

Les mesures ont été effectuées pour des paquets de tailles différentes. Les résultats sont présentés dans le tableau 5.2. Les valeurs de Δ_t sont calculées en utilisant l'equation 5.9 de la façon suivante :

- ⊙ La valeur maximale de Δ_t est obtenue pour la valeur minimale de $t_{gettimeofday()}$, soit 1ms.
- ⊙ La valeur minimale de Δ_t est obtenue pour la valeur maximale de $t_{gettimeofday()}$, soit 6ms.

L'obtention de valeurs minimales nulles s'expliquent par la durée relativement longue (par rapport à la valeur du temps $t_2 - t_1$ mesurée) de l'exécution de l'appel $t_{gettimeofday()}$. Dans ce cas la valeur minimale de Δ_t n'est pas mesurable. Par ailleurs, les valeurs maximales obtenues pour des paquets de 1500 octets sont supérieures à celles qui sont mesurées pour des paquets de 500 octets, en raison d'un volume de données plus important à copier.

Il est important de noter que pour 87 % et 91 % des mesures effectuées respectivement avec des paquets de 500 et 1500 octets, Δ_t est inférieur ou égal à 20 μs. Les résultats obtenus montrent une concentration des valeurs autour de 10 μs.

Impact sur les outils de mesure étudiés

IGMPS, Spruce et IGI sont les outils les plus susceptibles d'être affectés par l'incertitude liée à la date d'envoi des paquets. En effet, pour ceux-ci, il est impératif d'envoyer des paquets avec un temps inter-paquets Δ_{in} donné, de l'ordre d'une centaine à plusieurs centaines de μs. Une variation de Δ_t de l'ordre de 10 μs entraîne ainsi une erreur relative importante sur Δ_{in}. La valeur de Δ_{in} étant utilisée pour calculer la valeur de la bande passante disponible A, toute erreur sur Δ_{in} entraîne des erreurs sur l'estimation de A. Cette erreur peut être quantifiée en appliquant la méthode différentielle logarithmique. Nous détaillons ici le calcul de cette erreur pour les outils IGMPS et Spruce :

IGMPS estime la bande passante disponible en utilisant la formule 5.6. Nous définissons $\Delta(\Delta_{in})$ l'erreur sur Δ_{in}. En supposant qu'il n'y a pas d'erreur sur la mesure de C et Δ_{out}, on a :

$$\Delta A = \left| \frac{S^2 \Delta_{out}/C + S\Delta_{out}^2}{\Delta_{in}^2 \left[S/C + \Delta_{out}\right]^2} \right| \Delta(\Delta_{in})$$

Ainsi, pour une capacité C de 97,5 Mb/s (niveau IP), une erreur $\Delta(\Delta_{in})$=10 μs (erreur la plus fréquente), une bande passante disponible A de 50 Mb/s, nous obtenons une incertitude $\Delta A = 4,16$ Mb/s, soit 8,32% de A.

pour estimer la bande passante disponible Spruce utilise la formule suivante :

$$A = C\big(1 - \frac{\Delta_{in} - \Delta_{out}}{\Delta_{in}}\big) \tag{5.10}$$

Ainsi l'erreur sur la bande passante disponible (mesurée par Spruce) en connaissant l'erreur sur la dispersion initiale est donnée par :

$$\Delta A = \left| - C \frac{\Delta_{out}}{\Delta_{in}^2} \right| \Delta(\Delta_{in})$$

Donc, pour une capacité C de 97,5 Mb/s, une erreur $\Delta(\Delta_{in})$=10 μs , une bande passante

disponible A de 50 Mb/s, nous obtenons une incertitude $\Delta A = 11,78$ Mb/s, soit 23% de A.

Les contraintes sur les temps inter-paquets à respecter par Pathchirp sont déterminées par la plage des débits instantanés balayée par les *chirps* et la taille des sondes. Par défaut, Pathchirp envoie des paquets de 1000 octets à des débits instantanés variant de 10 à 200 Mb/s, ce qui correspond à des temps inter-paquets variant de 40 à 830 μs environ. Une erreur sur les temps inter-paquets à l'envoi peut conduire Pathchirp à générer les paquets à des débits inférieurs aux débits désirés et ainsi à associer les *excursions* des délais mesurés à l'arrivée [Rib00] à des débits théoriques supérieurs aux valeurs réelles. Cette erreur sera plus importante pour les grandes valeurs de bande passante disponible (c'est-à-dire les temps inter-paquets les plus petits) et causera, pour ces valeurs, une surestimation de la bande passante disponible.

Pour effectuer ses mesures, Pathload envoie des flux de 100 paquets. La contrainte essentielle étant de générer des flux ayant chacun un débit moyen donné et le résultat est donné sous forme d'intervalle avec une valeur A_{min} et A_{max} , l'incertitude étudiée ici est donc déjà prise en compte dans le résultat.

5.6.2 Incertitudes à la réception des paquets

De la même façon, tous les outils estampillent les paquets à leur arrivée au destinataire. L'incertitude de l'opération d'estampillage dépend du temps pour recevoir un paquet sur l'interface réseau, puis dans le système d'exploitation, déplacer le paquet de l'espace noyau à l'espace utilisateur, estampiller le paquet, effectuer éventuellement d'autres tâches avant l'arrivée du second paquet, et la présence éventuelle d'interruptions causées par l'ordonnanceur du système qui assigne les ressources à d'autres processus concurrents. En adoptant une démarche similaire à celle du paragraphe précédent, nous avons estimé Δ_t, le temps de passage d'un paquet depuis son arrivée dans le système d'exploitation jusqu'à l'espace utilisateur. Les valeurs obtenues varient entre 5 et 65 μs. Δ_t est inférieur ou égal à 25 μs pour, respectivement, 80 % et 91 % des mesures effectuées avec des paquets de 500 et 1500 octets.

Impact sur les outils de mesure étudiés

IGMPS, Spruce et Pathchirp s'affranchissent de cette incertitude en estampillant les paquets au niveau du pilote de l'interface réseau dans le noyau du système d'exploitation. Pour cela, IGMPS, Spruce et pathchirp utilisent l'option *SO_TIMESTAMP* dans la socket de réception. Cette option permet de réduire considérablement les erreurs sur la mesure des dispersions finales. Dans le cas d'IGMPS, en supposant qu'il n'y a pas d'erreurs sur la mesure de la capacité C du lien bottleneck ni sur la dispersion initiale Δ_{in}, l'erreur sur la mesure de la bande passante disponible en connaissant l'erreur sur la dispersion finale Δ_{out} est donnée par :

$$\Delta A = \left| \frac{S^2\Delta_{in}/C + 2S\Delta_{in}\Delta_{out} - 2S\Delta_{in}^2}{\Delta_{in}^2\left[S/C + \Delta_{out}\right]^2} \right| \Delta(\Delta_{out})$$

Donc, pour une capacité C de 97,5 Mb/s , une erreur $\Delta(\Delta_{out})$=10 μs et une bande passante disponible A de 50 Mb/s, nous obtenons une incertitude $\Delta A = 2,08$ Mb/s, soit 4,16% de A. Cette erreur est deux fois moins importante que celle provoquée par l'erreur sur la dispersion initiale, ce qui confirme les résultats de l'analyse de sensibilité qui ont montré qu'IGMPS est plus sensible aux perturbations engendrées par la dispersion initiale que par celles engendrées par la dispersion finale.

En revanche, l'erreur de mesure de la bande passante disponible pour Spruce est donnée par :

$$\Delta A = \left| - C\frac{\Delta_{out}}{\Delta_{in}} \right| \Delta(\Delta_{out})$$

Pour une capacité C de 97,5 Mb/s , une erreur $\Delta(\Delta_{out})$=10 μs et une bande passante disponible A de 50 Mb/s, nous obtenons une incertitude $\Delta A = 1$ Mb/s, soit 2% de A. Spruce paraît donc être moins sensible aux erreurs dues aux dispersions finales des paquets que ne l'est IGMPS.

Pour effectuer ses mesures, Pathload mesure les délais des paquets à leur arrivée et en détecte la variation. Les délais et leurs variations observés sur notre plateforme et sous différents scénarios sont de l'ordre de plusieurs dizaines voire centaines de millisecondes. Aussi, nous pouvons supposer que l'incertitude étudiée ici est négligeable. En revanche, IGI s'affranchi de ces erreurs en réalisant les opérations d'estampillage avec la librairie *libpcap* [Lib]. En effet, cette dernière permet de récupérer les paquets directement au niveau du pilote de la carte réseau, ce qui réduit considérablement les erreurs à la réception des paquets.

5.6.3 Incertitude résultant de l'estimation de la capacité du lien bottleneck

Les outils basés sur le Probe Gap model (IGMPS, Spruce et IGI) font l'hypothèse que la capacité du chemin est connue (c'est-à-dire mesurée préalablement), et utilisent cette valeur pour estimer la bande passante disponible. L'incertitude sur la mesure de la capacité s'ajoutera ainsi à l'estimation de la bande passante disponible. Nous quantifions ici cette erreur pour IGMPS et Spruce en appliquant la méthode différentielle logarithmique et en supposant qu'il n'y a pas d'erreurs sur les dispersions inter-paquets.

IGMPS estime la valeur de la bande passante disponible en utilisant la formule 5.6. Ainsi l'erreur sur la bande passante disponible est donnée par :

$$\Delta A = \left| \frac{S^2(2\Delta_{in} - \Delta_{out})/C^2}{\Delta_{in}\left[\frac{S}{C} + \Delta_{out}\right]^2} \right| \Delta C$$

Ainsi, pour une capacité C de 97,5 Mb/s (niveau IP), une erreur $\Delta C = 10$ Mb/s (environ 10% de C), une bande passante disponible A de 30 Mb/s, et en fixant $\Delta_{in} = 120\mu s$, nous obtenons une incertitude $\Delta A = 1$ Mb/s, soit 3,33% de A pour une taille de paquets sondes $S = 1500 octets$.

En revanche, l'erreur sur la bande passante disponible en utilisant Spruce est donnée par :

$$\Delta A = \left| 2 - \frac{\Delta_{out}}{\Delta_{in}} \right| \Delta C$$

Ainsi, pour une capacité C au niveau IP de 97,5 Mb/s, une erreur $\Delta C = 10$ Mb/s (environ 10% de C), une bande passante disponible A de 30 Mb/s, et en fixant $\Delta_{in} = 120\mu s$, nous obtenons une incertitude $\Delta A = 3$ Mb/s, soit 10% de A pour une taille de paquets sondes $S = 1500 octets$.

Nous remarquons donc que les erreurs de mesure sur la capacité du lien bottleneck affectent beaucoup plus les mesures de Spruce que celles d'IGMPS, les incertitudes de ce dernier sont trois fois moins importantes. Par ailleurs l'étude des erreurs d'IGI montre que pour les mêmes conditions citées précédemment, ce dernier mesure la bande passante disponible avec une incertitude $\Delta A = 7$ Mb/s, soit 23.33 % de A. Cette valeur est donc très importante et explique une bonne part de l'erreur relative d'IGI présentée dans le chapitre 2. Ce point constitue un avantage pour IGMPS qui est l'un des outils les moins sensibles aux erreurs de mesure de la capacité du lien bottleneck et ces résultats confirment bien les résultats obtenus par l'analyse de sensibilité qui incriminent beaucoup plus les paramètres liés aux délai et dont les erreurs sont provoquées par les opérations de datation que les paramètres liés à la capacité (capacité du lien bottleneck et les capacités des liens en amont de ce dernier). L'utilisation des outils Nettimer et bprobe par exemple pour la mesure de la capacité du chemin de bout en bout considéré n'affectera pas considérablement les résultats d'IGMPS, malgré l'imprécision de ces derniers.

5.6.4 Incertitudes dues à la corrélation des paquets

Ce phénomène est spécifique à IGI. En effet, ce dernier utilise des train (flux) périodiques de paquets sondes. Les paires de paquets qui forment ces flux ne sont pas indépendantes : si une paire est constituée des paquets k et $k+1$ alors la paire suivante est formée des paquets $k+1$ et $k+2$. Les dispersions Δ_{out} mesurées par le récepteur sont ainsi corrélées c-à-d que s'il y'a un problème de précision lors de la mesure de la dispersion obtenue pour la paire $(k, k+1)$, il y'a de forte chance que la dispersion obtenue pour la paire $(k+1, k+2)$ soit mesurée avec imprécision aussi. Pour atténuer cet effet, IGI ne prend pas en compte les Δ_{out} inférieurs ou égales à Δ_{in}, il considère les paquets de cette paire comme étant des paquets du trafic concurrent. Ceci a pour effet de surestimer le débit du trafic concurrent et donc de sous-estimer la bande passante disponible.

5.7 Analyse

L'analyse de sensibilité et le calcul d'incertitudes appliqués aux outils de mesure de la bande passante disponible de bout en bout basés sur les techniques à dispersions de paquets montrent que les erreurs et les perturbations enregistrées à la sortie des modèles étudiés sont dues principalement aux paramètres liés à la mesure des délais. L'analyse de sensibilité a incriminé les dispersions inter-paquets initiales et finales et le calcul d'incertitudes nous a fourni un peu plus de détails en désignant les erreurs dues à l'estampillage des paquets sondes au niveau de l'émetteur et du récepteur comme étant la vraie source d'erreurs et d'incertitudes des mesures fournies par ces différents outils.

L'étude détaillée des codes sources de ces outils de mesure a montré qu'une grande part de ces erreurs est due à des problèmes de programmation. En effet, pour l'estampillage des paquets sondes au niveau de l'émetteur et du récepteur l'appel système *gettimeofday()* est utilisé. Au niveau de l'émetteur, l'exécution de cette fonction rajoute une latence supplémentaire en plus des latences dues au déplacement du paquet de l'espace utilisateur vers l'espace noyau (de l'application au pilote de la carte réseau en traversant toutes les couches de la pile réseau), la latence due au transfert du paquet du noyau vers la carte réseau et l'éventuelle latence qui serait provoquée par l'ordonnanceur du système d'exploitation en affectant les ressources à d'autres processus. Au niveau du récepteur, c'est l'acheminement des paquets dans le sens inverse c'est-à-dire de la carte réseau vers l'application qui rajoute des latences supplémentaires.

En d'autres mots, lorsque le programme envoie un paquet, entre le moment où il donne l'ordre d'envoi de ce dernier (le paquet est estampillé à ce moment là) et le moment du départ effectif de ce dernier de la carte réseau il y'a un laps de temps. La même chose se produit lors de la capture d'un paquet ; entre le moment de son arrivée à la carte réseau et le moment de son estampillage au niveau de l'application il y a aussi un laps de temps. Ces délais sont variables et dépendent du matériel utilisé pour la mesure (système d'exploitation, vitesse du processeur, etc) ce qui rend leur prédiction impossible. C'est donc ces délais qui rendent les outils de mesure peu précis.

Pour améliorer les performances de ces outils, il est donc nécessaire de réduire au maximum ces laps de temps en procédant de la manière suivante :

- ⊙ Il est possible de réduire les erreurs de mesure en utilisant un système d'exploitation temps réel qui permettrait de traiter le processus de mesure avec une priorité élevée tout en évitant les éventuelles interruptions qui retarderaient la réception / l'envoi du paquet sonde de / vers l'interface réseau.

- ⊙ Le choix du langage de programmation est aussi très important. Il est possible de réduire

les différents délais en choisissant un langage de programmation qui permettrait l'utilisation de routines de bas niveau permettant une interaction directe avec le matériel. Très souvent, le langage C est le langage de prédilection pour développer ces librairies de fonctions permettant l'accès direct au matériel (c'est pourquoi nous avons choisi ce langage pour développer notre outil de mesure IGMPS).

⊙ Une des solutions pour réduire ces délais (que se soit au niveau de l'émetteur ou du récepteur) consiste à utiliser les *RAW-Sockets* au niveau du langage de programmation. Le *RAW-Socket* est un type de *sockets* de communications permettant de faire passer un paquet de la couche applications directement à l'interface réseau sans le faire passer à travers les autres couches de la pile TCP/IP. La définition des différents entêtes du paquet reste dans ce cas à la charge du programmeur. A la réception du paquet c'est l'application qui se charge aussi d'extraire les différents entêtes. En langage C par exemple, on trouve une implémentation des $RAW-Sockets$ sous forme de librairies de fonctions appelée *Libpcap* sous $Linux$ et $Wincap$ sous $Windows$. Ces dernières offrent une multitude de fonctions permettant la capture des paquets et leur estampillage au niveau du pilote de l'interface réseau.

⊙ En langage C, il est possible aussi d'utiliser au niveau du récepteur des *sockets* avec l'option $SO-TIMSTAMP$ à l'aide de la fonction *Setsocketopt()*. Cette option permet au récepteur d'estampiller les paquets au niveau du pilote de l'interface réseau avant que ce dernier ne soit acheminé vers l'application en traversant les couches de la pile réseau.

La combinaison d'un système d'exploitation temps réel et ces différentes techniques de programmation permettrait de réduire considérablement les erreurs de mesures dues aux problèmes d'estampillage des paquets sondes.

La métrologie logicielle mise en place en utilisant des programmes spécifiques offre aujourd'hui des résultats insatisfaisants malgré les nombreux efforts de la communauté scientifique à la rendre beaucoup plus performante et précise. Une alternative à cette catégorie de métrologie consiste en l'utilisation de nouvelles techniques fondées sur des outils matériels. Ces techniques commencent déjà à être utilisées. Cependant, pour l'instant ces dernières sont déployées uniquement dans le cadre des mesures passives en utilisant les cartes DAG par exemple.

5.8 Conclusion

Dans ce chapitre, nous avons effectué une analyse de sensibilité sur les différents modèles de mesure de la bande passante disponible basés sur la technique de la paire de paquets. Parmi les différentes méthodes existantes, nous avons choisi la méthode de Sobol car comparée au autres

techniques cette dernière offre les résultats les plus précis. Les résultats obtenus ont montré que les dispersions inter-paquets initiales et finales sont les paramètres les plus importants lors de la mesure de la bande passante disponible en utilisant IGMPS. Ces derniers sont principalement à l'origine des perturbations et des incertitudes sur les sorties des modèles proposés (générique et simplifié). De faibles erreurs de mesure sur ces paramètres engendreront des perturbations considérables sur l'estimation de la bande passante disponible. Pour améliorer la précision des outils de mesure fondés sur la technique de la paire de paquets, nous avons proposé quelques solutions qui permettrons de réduire les incertitudes sur la mesure des différentes dispersions inter-paquets qui sont dues principalement à l'estampillage des paquets sondes au niveau de l'émetteur et du récepteur.

Conclusion et perspectives

Conclusion

Dans ce travail de thèse, nous nous sommes intéressés à la mesure de la bande passante disponible dans un chemin de bout en bout qui est un paramètre de QoS qui a pris de l'importance au cours de ces dernières années. Nous avons, dans un premier temps, étudié les différentes techniques mises en œuvre pour mesurer ce paramètre. Puis, nous avons décrit quelques outils, que nous avons jugés intéressants, qui implémentent ces techniques. Une analyse comparative de ces outils à été effectuée sur une plateforme d'expérimentation isolée et sous des conditions identiques. Cette étude nous a permis de comparer leurs performances en termes de précision, de temps de réponse et d'intrusivité. Les résultats obtenus ont montré que Spruce est l'outil le plus précis, le plus rapide et l'un des outils les moins intrusifs. Ils ont montré aussi que dans les conditions de nos expérimentations la technique de la paire de paquets implémentée dans Spruce offre des avantages incontestables sur les techniques à auto-congestion au regard de la précision, du temps de convergence et de la charge de trafic de mesure injectée dans le réseau.

En nous basant sur le principe de la technique de la paire de paquets, nous avons développé un nouveau modèle déterministe qui exploite dynamiquement les informations temporelles obtenues des paquets sondes pour mesurer la bande passante disponible. Les travaux réalisés dans ce domaine ont démontré que la taille des paquets sondes utilisés pour analyser le chemin de bout en bout est un paramètre important susceptible d'affecter la mesure de la bande passante disponible. En nous basant sur ces conclusions, nous avons proposé un modèle qui prend en compte ce paramètre et nous l'avons implémenté dans un nouvel outil de mesure appelé IGMPS. Nous avons évalué les performances de cet outil sur la plateforme d'expérimentation selon différents scénarios et nous avons constaté que ce dernier permet de mesurer la bande passante disponible avec une précision dépassant largement celle offerte par les autres outils existants. Une étude plus détaillée de cet outil nous a permis de raffiner la constatation de l'influence de la taille des paquets sondes : elle est en relation avec la taille des paquets du trafic concurrent et cette relation a un impact direct sur la précision des mesures offertes.
Les expérimentations menées en vue d'explorer cette relation ont montré que pour obtenir les résultats les plus précis, il est nécessaire que la taille des paquets sondes soit égale ou suffisamment proche de la taille des paquets du trafic concurrent. Afin d'expliquer ce phénomène, nous avons étudié d'une manière beaucoup plus formelle l'interaction entre les paquets sondes et les paquets du trafic concurrent et la relation entre leurs tailles respectives. En nous basant sur l'étude du

système de file d'attente M/D/1, nous avons défini un modèle stochastique pour la technique de la paire de paquets qui établit une relation entre les dispersions initiales des paquets et leurs dispersions finales. Cette étude nous a permis de constater que le même phénomène précédent est présent aussi dans le modèle théorique proposé. Ce qui a permis de généraliser les propositions faites pour l'outil IGMPS à l'ensemble des outils implémentant la technique de la paire de paquets. Le modèle stochastique présenté dans ce travail nous a confirmé qu'il existe une relation entre le trafic de mesure et le trafic concurrent qui se manifeste d'une manière évidente lorsque les tailles des paquets de ces deux trafics sont égales ou suffisamment proches. Cependant, il ne nous a pas permis d'expliquer ce phénomène ni d'interpréter ce qui se passe réellement dans les files d'attentes des routeurs du chemin de bout en bout.

Dans ce travail de thèse, nous avons apporté des améliorations aux performances de la technique de la paire de paquets en proposant un modèle qui intègre le paramètre taille des paquets sondes et qui prend en compte certaines caractéristiques du trafic concurrent. Toutefois, les mesures obtenues sont imprécises dans certains cas, particulièrement quand le taux d'utilisation du goulet d'étranglement est élevé. Nous nous sommes donc intéressées à la définition des différentes sources d'incertitudes ainsi qu'aux erreurs qui sont à l'origine de ces imprécisions. Nous avons donc effectué, dans un premier temps, une analyse de sensibilité sur les différents modèles de mesure de la bande passante disponible basés sur la technique de la paire de paquets. Les résultats obtenus ont montré que les dispersions inter-paquets initiales et finales sont les paramètres les plus importants lors de la mesure de cette métrique. Ces derniers sont principalement à l'origine des perturbations et des incertitudes sur les sorties des modèles proposés, ils sont dus essentiellement aux erreurs d'estampillage des paquets sondes au niveau de l'émetteur et du récepteur. Finalement, une étude de la propagation d'incertitudes sur les sorties des modèles considérés a montré que de faibles erreurs de mesure sur ces deux paramètres engendreront des perturbations considérables sur l'estimation de la bande passante disponible. Une analyse détaillée des codes sources des différents outils de mesure a montré qu'une grande part des erreurs d'estampillage des paquets sondes est due généralement au système d'exploitation utilisé ainsi qu'à certains problèmes de programmation. L'intégration de quelques librairies spéciales lors de la programmation de l'outil et l'utilisation d'un système d'exploitation temps réel lors des mesures permettraient de réduire considérablement les erreurs et les incertitudes de mesures dues à l'estampillage des paquets sondes.

Tous les résultats de mesure présentés dans ce mémoire sont obtenus sur une plateforme d'expérimentations isolée. Le choix d'une telle plateforme peut être contestable, cependant, ce choix peut être expliqué par le souci de contrôler parfaitement le réseau ce qui permet d'évaluer facilement et rapidement nos propositions dans différentes configurations et dans toutes sortes de scénarios. De plus, une telle plateforme permet de tester les outils de mesure dans des conditions identiques, de sorte à ce que les tests soient reproductibles donc simples à soumettre à

la comparaison. La plateforme de mesure utilisée nous a permis de prendre en considération certaines hypothèses inhérentes à la technique de la paire de paquets qui ne sont pas toujours réalisables dans les conditions réelles de l'Internet. Par exemple, cette technique suppose que le trafic concurrent est fluide et que ce dernier change lentement. Cette hypothèse n'est pas vérifiée dans un réseau comme Internet ce qui a pour conséquence un manque d'interaction entre le trafic sonde et le trafic concurrent provoquant ainsi une sous-estimation ou une surestimation de la métrique mesurée. Cette technique suppose aussi que tous les routeurs constituant le chemin de bout en bout adoptent la politique de service FIFO. L'utilisation d'outils de mesure basés sur cette technique dans les réseaux sans fil ad hoc est donc impossible, étant donné que ces derniers n'implémentent pas cette politique.

Nous n'avons pas eu la possibilité d'accéder à des infrastructures ou des plateforme de mesure sur Internet (telles que PlanetLab, Metropolis, etc) ce qui a fait que nous n'avons pas pu valider certaines de nos propositions. Par exemple, pour avoir les mesures les plus précises possibles, nous avons recommandé d'utiliser des paquets sondes avec une taille égale ou proche de la taille des paquets du trafic concurrent. Pour que cette proposition soit valide dans l'Internet, il faudrait que la taille des paquets sondes soit variable étant donné que la taille des paquets du trafic Internet n'est pas fixe. Cependant, nous n'avons pas pu vérifier cette proposition dans les conditions réelles de l'Internet et nous avons considéré que cette dernière s'impose comme une évidence. Pour compléter l'étude menée dans ce mémoire et pour valider d'autres propositions moins évidentes que la précédente, il est donc nécessaire de recourir à des tests sur une infrastructure réelle supportant Internet. Il est nécessaire aussi de compléter notre analyse par la considération d'autres scénarios et d'autres paramètres tels que l'effet des nœuds en aval du goulet d'étranglement sur la technique de la paire de paquets, l'effet des équipements Store-and-forward de niveau 2, l'asymétrie et le changement des routes, etc.

Perspectives

Les travaux menés tout au long de cette thèse ainsi que les résultats obtenus permettent de dégager plusieurs perspectives scientifiques directement liées à l'utilisation et à l'extension de la technique de la paire de paquets à d'autres domaines de recherche dans le cadre des réseaux.

En premier lieu et en restant toujours dans le domaine de la mesure de la bande passante disponible, il serait utile de tester l'outil de mesure IGMPS dans les conditions réelles de l'Internet. En effet, comme mentionné dans le chapitre 3, la taille des paquets du trafic Internet est variable. Des études sérieuses visant à définir la nature de ce trafic ont démontré que les distributions des tailles des paquets de ce dernier sont multimodales. Les résultats de ces études ont démontré aussi la prédominance des paquets de 40, 552, 576 et 1500 octets. Selon nos propositions, il est donc nécessaire de faire varier les tailles des paquets d' IGMPS de façon à ce qu'elles soient les plus proches de ces valeurs pour obtenir une meilleure qualité de mesure. Des expérimentations sur Internet sont nécessaires afin de configurer les différents paramètres inhérents au modèle et afin de valider définitivement cette proposition. Il en résulterait probablement des propositions d'amélioration de l'outil.

A l'issue de ces travaux de thèse, nous avons constaté que malgré l'amélioration apportée à la précision des outils de mesure de la bande passante disponible fondés sur la technique de la paire de paquets, les mesures de ces derniers sont toujours sujettes à des erreurs qui sont dues soit aux limites des techniques de programmation utilisées, soit aux difficultés de modélisation du comportement du trafic concurrent qui traverse le réseau. Nous pensons donc que la métrologie réseaux telle qu'elle est aujourd'hui c'est-à-dire une métrologie mise en place à l'aide d'outils logiciels peut s'avérer inefficace dans certains cas. Une alternative à cette catégorie de métrologie consiste en l'utilisation de nouvelles techniques fondées sur des outils matériels. Ces techniques commencent déjà à être utilisées mais pour l'instant elles sont déployées uniquement dans le cadre des mesures passives en utilisant les cartes DAG par exemple. Ces dernières capturent le trafic sur un lien, le filtrent et le classifient (protocoles, applications, etc) et mesurent certaines caractéristiques de ce dernier (débit, délai, etc). Cependant, une amélioration considérable pourrait être apportée à ces outils matériels afin de les utiliser dans le cadre de la métrologie active de bout en bout. En effet, il est possible d'envisager par exemple de concevoir une carte matérielle de métrologie qui permettrait de générer des paquets sondes et de les envoyer à l'autre bout du

chemin étudié, d'en recevoir des paquets et de les traiter d'une manière autonome en appliquant des algorithmes de mesure embarqués. L'estampillage des paquets et les différents traitements sur ces derniers seront effectués indépendamment de la station de travail accueillant la carte permettant ainsi une datation très précise des paquets sondes et par conséquent l'obtention de mesures très précises des métriques considérées.

Dans le cadre de l'extension de la technique de la paire de paquets à d'autres domaines, nous envisageons deux autres utilisations possibles de cette technique.

Dans un chemin de bout en bout il est parfois très intéressant de connaître et de localiser avec précision le goulet d'étranglement. Il serait alors plus facile pour un administrateur réseaux d'intervenir et de régler le problème ou du moins de connaître le routeur qui est à l'origine de ce dernier. Aussi, d'un point de vue métrologie, il est plus facile d'effectuer les mesures des caractéristiques du chemin de bout en bout en ciblant et en attaquant directement le routeur goulet d'étranglement. La conception et la réalisation d'un outil permettant cette localisation pourraient être basées sur la technique de la paire de paquets. En effet, la technique de la paire de paquets récursive qui pourrait être mise en place à l'aide de mesures aller-retour basées sur ICMP (comme c'est le cas pour la technique de mesure de la capacité de bout en bout introduite dans la section 1.9.1), permettrait d'envoyer un ensemble de paires de paquets avec une certaine dispersion initiale en ciblant chaque routeur du chemin de bout ne bout. En recevant les paquets ICMPS time-exceeded de chaque paire, l'émetteur mesurerait les dispersions finales afin de les comparer. Le goulet d'étranglement correspondrait au routeur qui aurait provoqué la dispersion finale la plus élevée (en moyenne).

Le dernier point de ces perspectives pourrait être une contribution dans le domaine des réseaux sans fil. En effet, dans les réseaux sans fil en mode infrastructure, nous avons constaté que lors du Handover, le point d'accès est sélectionné en se basant uniquement sur le rapport signal sur bruit. Dans certains cas, cette méthode peut s'avérer inefficace. Dans une BSS (Base Station Subsystem) les mobiles ont tendance à se connecter au point d'accès le plus proche étant donné que la puissance du signal dépend fortement de la distance. Ce qui provoque des congestions sur certains points d'accès alors que d'autres restent presque vides. Pour résoudre ce problème, il est possible de développer un nouvel algorithme basé sur la mesure qui permettrait de rééquilibrer les charges (Load Balancing) sur les différents points d'accès en soulageant ainsi les points d'accès saturés. Avant une association ou une réassociation (lors du Handover), un mobile pourrait par exemple envoyer une paire de paquets avec une dispersion initiale à chacun des points d'accès se trouvant dans son entourage. A la réception des acquittements, le mobile mesurerait les dispersion finale de chaque paire et les comparerait, il choisirait alors de se connecter au point d'accès présentant des acquittements avec une dispersion finale la moins élevée (une dispersion finale importante révèle un état de saturation du point d'accès). L'avantage d'utiliser la technique

de la paire de paquets réside dans le fait que cette dernière exploite les variations du délai au lieu du délai lui même, ce qui la rend moins sensible aux perturbations de l'environnement du mobile. Dans cette étude, il serait essentiel de prendre en compte d'autres paramètres liés à la technologie sans fil et de prendre en considération les différentes contraintes inhérentes à la mobilité.

Annexe A

La métrologie et notions associées

A.1 Définition de la métrologie

La métrologie est l'ensemble des techniques et des savoir-faire qui permettent d'effectuer des mesures et d'avoir une confiance suffisante dans leurs résultats. La mesure est nécessaire à toute connaissance, à toute prise de décision et à toute action. La logique de toute activité est "observer/mesurer, comprendre, prévoir/agir, mesurer/vérifier".
Mesurer est indispensable pour la recherche : toute recherche vise à modéliser les phénomènes, et doit quantifier des grandeurs dans des unités connues et définies, afin de quantifier leurs relations et leurs interactions, et reproduire des phénomènes. La mesure du temps, de l'espace est née de l'astronomie dans l'antiquité (mesure de la circonférence terrestre par *Eratosthène*). Aujourd'hui on mesure la courbure de l'univers et les propriétés des particules élémentaires.

> *"Ce qui est admirable, ce n'est pas que le champ des étoiles soit si vaste, c'est que l'homme l'ait mesuré."* *(Anatole France)*

La métrologie est utilisée depuis de nombreuses années dans plusieurs domaines (physiques, chimie, électronique, etc). Cependant, son utilisation dans le domaine des réseaux est très récente. En effet, dans un premier temps, les opérateurs ont voulu mesurer l'utilisation effective de leurs équipements afin de connaître leurs taux de charge pour pouvoir les ajuster aux besoins de leurs clients et minimiser ainsi leurs coûts. Ils cherchaient à avoir plus de contrôle sur leurs réseaux, afin de détecter rapidement les problèmes de congestion ou éventuellement les pannes d'équipements. Ils voulaient, de plus, avoir une notion de la qualité qu'ils fournissent à leurs clients. Par ailleurs, les utilisateurs ont voulu avoir de meilleures performances, ces dernières se traduisent par les courts délais induits par le réseau de communications lors de l'exécution de leurs applications, une faible gigue et une large bande passante. Au fur et à mesure de l'évolution de l'Internet, le client est devenu de plus en plus exigeant et a commencé à demander à son fournisseur d'accès des garanties de services (SLA = Service Level Agreement). Ainsi, il est

devenu nécessaire de pouvoir contrôler ces SLA, aussi bien du côté opérateur que du côté client, ceci afin d'assurer la bonne supervision du contrat liant les deux parties. Il fallait donc avoir des outils pour pouvoir vérifier les différentes contraintes sur le trafic réseau. Ainsi est née la métrologie réseau.

Actuellement, cette discipline émergente trouve des applications dans de nombreux domaines du monde des réseaux comme par exemple dans :

- ⊙ Optimisation de la QoS et des performances :
  - ◇ La classification des flux et du trafic, soit pour pouvoir trier les flux en fonction de la qualité de service qu'ils requièrent, soit, par rapport à des problèmes de routage, pour pouvoir les encapsuler dans des trafics qui empruntent tous le même chemin.
- ⊙ Ingénierie du trafic :
  - ◇ Le dimensionnement des réseaux qui permet de mettre en place des capacités suffisantes pour assurer en permanence un service adéquat à tous les utilisateurs.
  - ◇ Analyse du comportement des équipement vis à vis du trafic pour pouvoir comprendre les mécanismes de routage et de transport assurant le contrôle des flux, d'erreurs et de congestion, ceci dans le but d'améliorer les mécanismes déjà existants ou de concevoir nouveaux protocoles.
  - ◇ Modélisation et analyse du comportement du trafic pour pouvoir adapter le réseau aux caractéristiques du trafic observé (caractérisation et modélisation du trafic).
- ⊙ Tarification :
  - ◇ La tarification et les SLA qui permettent de définir des coûts de service en relation avec les performances du réseau et les ressources consommées, etc.

A.2 Quelques définitions associées à la métrologie

A.2.1 La justesse

C'est la partie de l'écart entre la valeur mesurée expérimentalement et la valeur vraie qui dépend uniquement des erreurs systématiques (erreurs agissant toujours dans le même sens, elle contribue à toujours surévaluer ou toujours sous-évaluer la valeur mesurée) : c'est le défaut d'étalonnage, de calibrage, de zéro d'un appareil, etc.

Le biais est un terme qui est souvent utilisé (comme traduction du terme anglais Bias) et qui est relié à la justesse (plus le biais est faible, plus la méthode est juste). Si x est le résultat analytique et que l'on peut disposer de la valeur x_c, valeur certifiée de l'échantillon de référence, le biais Δ est donnée par $\Delta = x - x_c$.
x doit représenter la moyenne d'un grand nombre de mesures de façon à minimiser l'influence des erreurs aléatoires. Une erreur est aléatoire lorsque, d'une mesure à l'autre, la valeur obtenue peut être surévaluée ou sous-évaluée par rapport à la valeur réelle.

A.2.2 La fidélité

La fidélité est l'aptitude d'une méthode ou d'un outil à donner des résultats les plus proches possibles lors d'analyses ou de mesures répétées d'un même échantillon, deux notions distinctes lui sont associées :

- ⊙ *Répétabilité :* Variabilité aléatoire des résultats d'une série de détermination d'un même échantillon effectuée dans des conditions très proches (et donc généralement dans un temps court).
- ⊙ *Reproductibilité :* Variabilité aléatoire des résultats de plusieurs déterminations d'un même échantillon, effectuées de manière espacée dans le temps, donc dans des conditions qui peuvent être expérimentalement légèrement différentes.

En métrologie réseaux on parlera beaucoup plus de reproductibilité des outils de mesure que de répétabilité car l'état du réseau change d'une manière continue (même dans un petit laps de temps, les caractéristiques du réseau ne sont pas constantes).

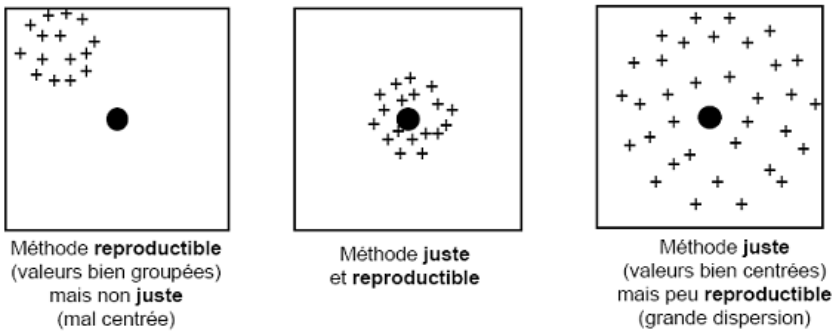


FIG. A.1 – Justesse et reproductibilité.

Sur la figure A.1, la valeur réelle de la variable à mesurer est représentée par le point central et les valeurs estimées de cette même variable sont représentées par des croix. Quand la méthode ou l'outil de mesure utilisé est juste et reproductible les croix ont tendance à se regrouper autour du point central, au contraire quand l'outil de mesure est peu reproductible la dispersion des croix autour du point central est plus importante. Cette dispersion de valeurs autour d'un point peut être estimée par une caractéristique statistique, appelée Écart-type, qui détermine la distance moyenne des observations à la moyenne arithmétique représentée par le point central. Plus l'Écart-type est grand plus les données sont dispersées autour de la moyenne, ce qui indique que l'outil est peu reproductible donc peu fidèle. Au contraire, plus l'écart-type est faible plus les données se rapprochent de la valeur réelle, ce qui veut dire que l'outil est plus reproductible, donc plus fidèle. L'écart-type est donnée par la formule suivante :

$$S = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n}}$$

avec X_i la i^{me} valeur obtenue sur une série de n mesures d'un échantillon, $\overline{X}$ est la valeur moyenne des n mesures.

A.2.3 La précision

La définition du terme *précision* varie d'un auteur à l'autre et ce terme inclut la reproductibilité ou fidélité et également la notion de biais. Cependant, il faut noter que la norme NF X 07-001 "*vocabulaire international des termes fondamentaux et généraux de métrologie*" ne donne pas de définition du terme *précision* mais considère qu'une méthode ou un outil est précis quand ce dernier est juste et reproductible. Le terme *précision* devrait être utilisé avec précaution et uniquement dans un sens subjectif.

A.3 Quelques outils de mesure et leurs caractéristiques

Le tableau de la figure A.2 présente quelques outils de mesure et les métriques de l'IETF pour chaque paramètre de QoS. De plus, il indique des références bibliographiques dans lesquels le lecteur pourra trouver de précieux compléments d'information. Le tableau de la figure A.3 présente les caractéristiques de chaque outil de mesure : il indique les paramètres mesurés, la classe de mesure, et précise si l'outil est coopératif ou non (c'est-à-dire s'il est nécessaire de le déployer sur les deux systèmes concernés par la mesure), le protocole sur lequel les mesures sont faites et le système d'exploitation supporté.

Légende de la figure A.3 :
Colonne de paramètres mesurés : "u" = unidirectionnel, "b" = bidirectionnel, "ar" =aller-retour.
Colonne classe de mesure : "a" = active, "p" = passive.
Colonne type d'environnement : "c" = coopératif, "nc" = non-coopératif.
Colonne système d'exploitation : "*" = la plupart, "U" = la plupart des Unix.

	Outils de mesure	Métriques de l'IETF	Compléments d'information
Bande passante totale	Pathchar [Jac97], Bing [Bey95], Clink [Dow99], Pchar [Mah99], Nettimer [LB00]		[Dow99] [Jia99]
Bande passante disponible	Cprobe [CC96], Iperf [Ipe], Pathload [JD02a] [JD02b], Pipechar [JYCA01]	draft [ASU]	
Bande passante minimale	Bprobe [CC96], Pathrate [Dov] [DRM01], Nettimer [LB01], Sprobe [Sar01]		[Pax97]
Délai unidirectionnel		RFC 2679	[Jia99] [Pax97] [Pax98] [JS99]
Délai aller-retour	ping	RFC 2681	[JS99]
Variation de délai	Iperf [Ipe]	RFC 3393	[HvB98]
Pertes de paquets	Ping, Sting [Sav99] Iperf [Ipe]	RFC 2680 RFC 3357	[JS99]
Route	Traceroute [Jac]		

FIG. A.2 – Récapitulatif des paramètres de QoS, des techniques/outils de mesure, et des métriques définies par l'IETF.

Outil	Délai aller-retour	Délai unidirectionnel	Variation de délai	Pertes de paquets	Bande passante disponible	Bande passante totale	Bande passante minimale	Classe de mesure	Type d'environnement	Protocole	Système d'exploitation
Ping	u			ar				a	nc	ICMP	*
Sting				b				a	nc	TCP	FreeBSD Linux
Bprobe							u	a	c	ICMP	SGI Irix
Cprobe					u			a	c	ICMP	SGI Irix
Sprobe							b	a	nc	TCP	FreeBSD Linux
Pathchar						u		a	nc	UDP	U
Bing						u		a	nc	ICMP	U
Clink						u		a	nc	UDP/ICMP	Linux
Pchar						u		a	nc	UDP/ICMP	U
Pathload					u			a	c	UDP	U
Pathrate							u	a	c	UDP	U
Pipechar NCS					u	u		a	c/nc	UDP/ICMP	U
Nettimer						u	u/b	a/p	c/nc	TCP	Linux
Iperf			b (UDP)	b (UDP)	b (TCP)			a	c	TCP/UDP	*

FIG. A.3 – Caractéristiques des outils de mesure

Bibliographie

[Abr93] L.R Abramson. *Model uncertainty from a regulatory point of view.* In Workshop Model Uncertainty : its Characterization and Quantification, Maryland, USA, 1993.

[Ada00] A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S. B. Moon, V. Paxson, D. Towsley. *The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior.* IEEE Communications, Vol. 38, n° 5, pp. 152-159, May 2000.

[ASC02] T. Anjali, C. Scoglio, L. C. Chen, I. F. Akyildiz, G. Uhl. *Abest : An available bandwidth estimator within an autonomous system.* In Proc. IEEE Globecom, 2002.

[ASU] T. Anjali, C. Scoglio, G. Uhl, A. Sciuto, J. A. Smith. Available Bandwidth Measurement in IP Networks. Internet Draft, Expiration date April 2003.

[Att02] A. P'asztor, D.Veitch. On the Scope of End-to-End Probing Methods. IEEE Communications Letters, vol. 6(11), 509-511, 2002.

[Att03] A. P'asztor. Accurate Active Measurement in the Internet and its Applications. Thèse de doctorat de l'université de Melbourne, Australie, 2003. http ://www.cubinlab.ee.mu.oz.au/ attila/pub/PhDfinal.pdf.

[Bao98] Y. Bao. Quality of Service control for real-time multimedia applications in packet-switched networks. Thèse de doctorat de l'université de Delaware, May 1998.

[BDS95] I.Busse, B. Deffner, H. Schulzrinne. *Dynamic QoS Control of Multimedia Applications based on RTP.* In Proceedings of First International Workshop on High Speed Networks and Open Distributed Platforms, St. Petersburg, Russia, June 1995.

[Bey95] P. Beyssac. BING, a Bandwidth measurement tool based on PING.

http ://spengler.econ.duke.edu, 1995.

[BS98] M. S. Borella, D. Swider. *Internet Packet Loss : Measurement and Implications for End to End QoS.* In Proceedings of ICPP Workshops on Architectural and OS Support for Multimedia applications/Flexible Communication Systems/Wireless Networks and Mobile Computing, 1998.

[Cac99] R. Caceres, N. Duffield, D. Towsley, J. Horowitz. *Multicast-based Inference of Network-internal loss characteristics.* IEEE Transactions on Information Theory, vol. 45, n° 7, pp. 2462-2480, November 1999.

[CAI] CAIDA. Cooperative Association for Internet Data Analysis. http ://www.caida.org.

[Cal98] K. Claffy, G. Miller, K. Thompson. *The Nature of the Beast : Recent Traffic Measurements from an Internet Backbone.* In Proceedings of INET, Genève, Juillet 1998.

[Cam96] A.T. Campbell. A Quality of Service Architecture. Thèse de doctorat du Département d'Informatique de l'Université de Lancaster, UK, 1996.

[Cao01] J. Cao, W. S. Cleveland, D. Lina, D. X. Sun Internet Traffic Tends Toward Poisson and Independent as the Load Increases. Bell-labs, Technical report, http ://cm.belllabs.com/cm/ms/departments/sia/doc/lrd2poisson.pdf, USA, 2001

[CC96] R. Carter and M. Crovella. Measuring bottleneck link speed in packet switched networks. Technical Report 1996-006, Boston University, March 1996.

[CL99] F. Cheong, R. Lai. *QoS Specification and mapping for distributed multimedia systems : a survey of issues.* The Journal of Systems and Software, 45 :127-139, 1999.

[Cor04] J.E.Eduardo Gonzalez. Mesure de paramètres de qualité de service dans les réseaux IP. Thèse de doctorat de l'université Rennes 1, Novembre 2004.

[Cuk73] R.I. Cukier, C.M. Fortuin, K.E. Shuler, A.G. Petschek, J.H. Schaibly. *Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients theory.* Journal Chemical Physics, 59 :3873-3878, 1973.

[Cuk75] R.I. Cukier, K.E.Shuler, J.H. Schaibly. *Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients - analysis of the approximations.* Journal

Chemical Physics, 63 :1140- 1149, 1975.

[Cuk78] R.I. Cukier, R.I. Levine, K.E. Shuler. *Nonlinear sensitivity analysis of multiparameter model systems.* Journal Computational Physics, 26 :1-42, 1978.

[DC02] C. Demichelis, P. Chimento. IP Packet Delay Variation Metric for IPPM. RFC 3393, November 2002.

[Dem02] I. Demeure. Une contribution à la conception et à la mise en ouvre d'applications sous contraintes de QoS temporelles, réparties, adaptables. Thèse de doctorat de l'Université des Sciences et Technologies de Lille, December 2002.

[Dif] DiffServ.Groupe de travail Differentiated Services de l'IETF. http ://www.ietf.org/html.charters/diffserv-charter.html.

[Din03] J.Strauss, D.Katabi, F.Kaashoek. *A Measurement Study of Available Bandwidth Estimation Tools.* In The proceedings of Internet Measurements Conference (IMC'03), Floride, 2003.

[DITG] Distributed Internet Traffic Generator. http ://www.grid.unina.it/software/ITG.

[Dov] C. Dovrolis. Pathrate. http ://www.pathrate.org.

[Dov01] C. Dovrolis, P. Ramanathan, D. Moore. *What Do Packet Dispersion Techniques Measure ?.* In Proceedings of the IEEE Infocom, Anchorage, USA, April 2001.

[Dov02a] M. Jain, C. Dovrolis. *Pathload : A Measurement Tool for End-to-end Available Bandwidth.* In Proceedings of the 3rd Passive and Active Measurements Workshop, Fort Collins, USA, March 2002.

[Dov02b] R. S. Prasad, C. Dovrolis, B. A. Mah. *The Effect of Layer-2 Switches on Pathchar-like Tools.* In Proceedings of 2nd Internet Measurement Workshop (IMW'02), Marseille, France, Novembre 2002.

[Dov03a] R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy. *Bandwidth Estimation : Metrics, Measurement Techniques, and Tools.* IEEE Network, November 2003.

[Dov03b] M. Jain and C. Dovrolis. *End-to-End Available Bandwidth : Measurement methodology, Dynamics, and Relation with TCP Throughput.* IEEE/ACM Transactions on Networking, 11(4) :537-549, August 2003.

[Dov03c] R. Prasad, C. Dovrolis, B. Mah. *The Effect of Layer-2 store-and-forward devices on per-hop capacity estimation.* In Proceedings of IEEE Infocom, San Francisco, USA, April 2003.

[Dov06] M. Jain, C. Dovrolis. *Available bandwidth measurement as simple as running wget.* In Passive and active measurement (PAM) workshop, Adelaide, Australia, Mars 2006.

[Dow99] A.B. Downey. *Using Pathchar to estimate Internet link characteristics.* In Proceedings of ACM SIGCOMM'99, pages 222-223, 1999.

[DRM01] C. Dovrolis, P. Ramanathan, D. Moore. *What do packet dispersion techniques measure ?.* In Proceedings of INFOCOM, pages 905-914, April 2001.

[DS97] C. Diot, A. Seneviratne. *Quality of service in heterogeneous distributed systems.* In Proceedings of the 30th Hawaï International Conference on System Sciences, Hawaï, January 1997.

[E800] ITU-T, Recommendation E.800. *Terms and Definitions Related to the Quality of Telecommunications Services.* International Telecommunication Union.

[Fle05] M.Flechsig, U.Böhm, C.Rachimow, T.Nocke. *Techniques for quality assurance of models in a multi-run simulation environment.* In Proceedings of the 4th International Conference on Sensitivity Analysis of Model Output, Los Alamos, NM, USA, 2005.

[Fra98] G.J. Franx. The transient M/D/c queuing system. http ://www.cs.vu.nl/ franx.

[Hag06] O.Hagan. *Bayesian Analysis of Computer Code Outputs : A Tutorial.* Reliability Engineering and System Safety, Volume 91, Elsevier Science, 2006.

[Hu03] N. Hu, P. Steenkiste. Evaluation and Characterization of Available Bandwidth Probing Techniques. In *the IEEE JSAC Special Issue in Internet Measurement, Mapping, and Modeling*, 21(6), August 2003.

[Hu06] N.Hu. NetworkMonitoring and Diagnosis Based on Available Bandwidth Measurement. Thèse de doctorat de Carnegie Mellon University, Pittsburgh, USA, 2006.

[HvB98] A. Hafid, G. von Bochmann. *Quality-of-Service adaptation in distributed multimedia applications.* Multimedia Systems, 6(5) :299-315, 1998.

[I380] ITU-T, Recommendation I.380. Internet Protocol Data Communication Service - IP Packet Transfer and availability Performance Parameters. International Telecommunication Union.

[Int] IntServ.Groupe de travail Integrated Services de l'IETF.
http ://www.ietf.org/html.charters/intserv-charter.html.

[Ipe] Iperf. http ://dast.nlanr.net/projects/iperf.

[Jac] V. Jacobson. Traceroute. ftp ://ftp.ee.lbl.gov/traceroute.tar.gz.

[Jac05] J.Jacques. Contributions à l'analyse de sensibilité et à l'analyse discriminante généralisée. Thèse de doctorat de l'université Joseph Fourier, Grenoble1, décembre 2005.

[Jac88] V. Jacobson. *Congestion avoidance and control.* In Proceedings of ACM SIGCOMM'88, pages 314-329, august 1988.

[Jac97] V. Jacobson. Pathchar, a tool to infer characteristics of Internet paths. April 1997.

[Jia99] W. Jiang. Detecting and measuring asymmetric links in an IP network. Technical Report CUCS009-99, January 1999.

[Jol02] M.Jolicoeur. Screening designs sensitivity of a nitrate leaching model (ANIMO) using a one-at-a-time method. USA : State University of New York at Binghampton, 2002

[JS99] W. Jiang, H. Schulzrinne. QoS Measurement of Internet Real-Time Multimedia Services. Technical Report CUCS015-99, Columbia University, New York, December 1999.

[JYCA01] G. Jin, G. Yang, B. Crowley, D. Agarwal. *Network Characterisation Service (NCS).* In proceedings of the 10th IEEE Symposium on High Performance Distributed

Computing, August 2001.

[Kiw04] D.Kiwior, J.Kingston, A.Spratt. *PATHMON, A Methodology for Determining Available Bandwidth over an Unknown Network.* IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication, Princeton, New Jersey, USA, Avril 2004.

[Kle95] J.P.C.Kleijnen. Sensitivity analysis and related analyses : a survey of statistical techniques. Netherlands : School of Management and Economics, Tilburg University, 1995.

[KR02] R. Koodli, R. Ravikanth. One-way Loss Pattern Sample Metrics. RFC 3357, August 2002.

[Lab05] Y. Labit, P. Owezarski, N. Larrieu *Evaluation of active measurement tools for bandwidth estimation in real environment.* In Proceedings of the 3rd IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON'05), Nice, France, Mai 2005

[Lar04b] N. Larrieu, P. Owezarski. *De la métrologie pour l'ingénierie des réseaux de l'Internet.* Revue Technique et Sciences Informatiques, numéro thématique Réseaux et Protocoles , vol. 23, n. 5-6/2004, septembre 2004.

[Lar04c] N. Larrieu. *A measurement based networking approach for improving Internet congestion control.* IFIP World Computer Congress (WCC'04), Student Forum, Toulouse (France), Août 2004.

[Lar05] N. Larrieu. Contrôle de congestion et gestion du trafic à partir de mesures pour l'optimisation de la QoS dans l'Internet. Thèse de doctorat de l'INSA de Toulouse, Juillet 2005.

[Lar05a] N. Larrieu, P. Owezarski. *Measurement based networking approach applied to congestion control in the multi-domain Internet.* In proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM'2005), Nice, France, Mai 2005.

[Lar05b] N. Larrieu, P. Owezarski. *Contrôle de congestion et gestion du trafic à partir de mesures.* In proceedings of Colloque Francophone sur l'Ingénierie de Protocoles

CFIP'05, Bordeaux, France, 2005.

[Lar05c] N. Larrieu. Monitoring based approach for congestion control aiming at improving Internet QoS. Rapport LAAS, 3 p., Mars 2005, IEEE INFOCOM 2005, Student Workshop.

[Lar05d] N.Larrieu, P.Owezarski. Towards a measurement based networking approach for Internet QoS improvement. Computer Communications, Vol.28, Issue 3, February 2005.

[LB00] K. Lai, M. Baker. *Measuring link bandwidths using a deterministic model of packet delay.* In Proceedings of ACM SIGCOMM, Stockholm, Sweden, 2000.

[LB01] K. Lai, M. Baker. *Nettimer : A tool for measuring bottleneck link bandwidth.* In Proceedings of USENIX Symposium on Internet Technologies and Systems, San Francisco, USA, March 2001.

[Lel00] A. Lelevé. Contribution à la téléopération de robots en présence de délais de transmission variables. Thèse de doctorat de l'Université de Montpellier II, December 2000.

[Lib] Tcpdump and Libpcap. http ://www.tcpdump.org.

[Mah05] A.Mahul. Apprentissage de la qualité de service dans les réseaux multiservices : Application au routage optimal sous contraintes. Thèse de doctorat de l'université de Blaise pascal, Clermont-Ferrand, 2005.

[MB00] B.Melander, M.Björkman. *A new end-to-end Probing and analysis method for Estimating Bandwidth Bottlenecks.* In Proceedings of IEEE Global Internet Symposium (GLOBCOM'00), San Francisco, USA, 2000.

[McK79] M.D. McKay, R. Beckman, W. Conover. *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code.* Technometrics, 21(2) :239-245, 1979.

[MGEN] B.Adamson. MGEN : Multi-Generator. http ://cs.itd.nrl.navy.mil/work/mgen/index.php.

[Mic03] F.Michaut. Adaptation des applications distribuées à la Qualité de Service fournie par le réseau de communication. Thèse de doctorat de l'université Henri Poincaré Nancy

1, Novembre 2003.

[Mic05] F.Michaut, F.Lepage. *Application-oriented Network Metrology : Metrics and Active Measurement Tools.* IEEE Communications Surveys and Tutorials Volume 7, Number 2, pages 2-24, Second Quarter 2005.

[Mil91] D.L. Mills. *Internet Time Synchronisation : the Network Time Protocol.* IEEE Transactions on Communications, 39(10) :1482-1493, October 1991.

[MIR] MIREHD. Multimédia interactif et réseaux haut débit. http ://www.enst-bretagne.fr/espacedoc/service/fiches/projets/MIREHD.htm.

[Nav03] J.Navratil. *ABwE : A Practical Approach to Available Bandwidth.* In Proceedings of Passive and Active Measurement network (PAM'03), La Jolla, USA, Avril 2003.

[Nea90] A.Nearing, L.A.Deer-Ascough, J.M.Laflen. *Sensitivity analysis of the WEPP hillslope profile erosion model.* Transactions ASAE 33 (3), p. 839-849, 1990.

[Net01] Netsizer. http ://www.netsizer.com.

[NGN] TF-NGN. Task Force Next generation Networks. http ://www.terena.nl/tech/task-forces/tf-ngn.

[Nie92] H. Niederreiter. Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphie, USA, 1992.

[NIM] NIMI.National Internet Measurement Infrastructure.
http ://www.ncne.nlanr.net/nimi.

[OF97] R. Oboe, P. Fiorini. *Issues on Internet-based teleoperation.* In Proceedings of SYROCCO'97, 1997.

[OR] T. Oetiker, D. Rand. MRTG : Multi Router Traffic Grapher.
http ://www.people.ee.ethz.ch/ oetiker/webtools/mrtg.

[Owe01] P. Owezarski. *Que nous dit la métrologie sur le futur d'Internet ?* In Proceedings of JRES 2001, Lyon, December 2001.

[Owe03b] P. Owezarski, D. Andreu, C. Fricker, K. Salamatian, C. Chekroun, N. Benameur, P. Olivier, J. Roberts, F. Guillemin. Projet METROPOLIS. Sousprojet 1 : Rapport d'état de l'art. Rapport du projet RNRT METROPOLIS, janvier 2003.

[Owe03c] P. Owezarski, P. Abry, K. Salamatian, D. Kofman, A. Aussem, F. Guillemin, P. Robert. Métrologie des réseaux de l'Internet. Rapport final de l'Action Spécifique du département STIC du CNRS Num 88, décembre 2003.

[Owe04a] P. Owezarski, N. Larrieu. *A trace based method for realistic simulations.* IEEE International Conference on Communications (ICC'2004), Paris, France, Juin 2004.

[Owe04b] P. Owezarski, N. Larrieu. *Internet traffic characterization - An analysis of traffic oscillations.* In proceedings of the 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'04), Toulouse (France), Juin 2004.

[Owe06] P.Owezarski. Contribution de la métrologie Internet à l'ingénierie des réseaux. Rapport d'habilitation à diriger les recherches, LAAS-CNRS, Toulouse, France, Septembre 2006.

[Owe06a] P. Owezarski, N. Larrieu, L. Bernaille, W. Saddi, F. Guillemin, A. Soule, K. Salamatian. *Distribution of traffic among applications as measured in the French Metropolis project.* Annals of telecommunication, 2006.

[Owe06b] P.Owezarski, N.Larrieu. Techniques et outils de métrologie pour l'Internet et son trafic. Revue Techniques pour l'ingénieur, 2006.

[Owen98] A.Owen. *Monte Carlo extension of quasi-monte carlo.* In 1998 Winter Simulation Conference, Washington DC, USA, 1998.

[PAM00] V. Paxson, A. Adams, M. Mathis. *Experiences with NIMI.* In Proceedings of Passive and Active Measurements (PAM), 2000.

[Par05] K.J.Park, H.Lim, C.H.Choi. Stochastic analysis of packet-pair probing for network bandwidth estimation. Thèse de doctorat de Seoul National University, Corée du sud, 2005.

[Pat03] S.Patarin. Pandora : support pour des services de métrologie à l'échelle d'Internet.

Thèse de doctorat de l'université Pierre et Marie Curie - Paris 6, Juin 2003.

[Pax97] V. Paxson. Measurements and Analysis of End-to-End Internet Dynamics. Thèse de doctorat du Computer Science Division, University of California, Berkeley, and Information and Computing Sciences Division Lawrence Berkeley National Laboratory, University of California, April 1997.

[Pax97a] V. Paxon. *End-to-end internet packets dynamics.* In Proceedings of ACM SIGCOMM, Cannes, France. Septembre 1997

[Pax98] V. Paxson. *On calibrating measurements of packet transit times.* In Proceedings of SIGMETRIC'98, 1998.

[Pch] Pchar. http ://employees.org/ bmah/software/pchar.

[PV02] A. Pasztor, D. Veitch. *Active probing using packet quartets.* In Proceedings of Internet Measurement Workshop, Marseille, France, November 2002.

[Qia06] W.Qiang, C.Liang. *FEAT : Improving Accuracy in End-to-end Available Bandwidth Measurement.* in the proceddings of the IEEE Global Telecommunications Conference (GLOBECOM'06), San Francisco, USA, Novembre 2006.

[Rad04] A.Radu. Evaluation de la qualité de service par l'utilisateur final dans les systèmes mobiles. Thèse de doctorat de l'université de Marne-La-Vallée, 2004.

[Rat01] M.Ratto, S.Tarantola, A. Saltelli. *Sensitivity analysis in model calibration : GSA-GLUE approach.* Computer, Physics and Communications 136 (3), p. 212-224. 2001.

[RC00] L.M. Rojas-Cardenas. Architecture de transport à ordre et fiabilité partiels pour les applications multimédias adaptatives à temps contraint. Thèse de doctorat du LAAS CNRS, 2000.

[REs01] REseau National de Recherche en TElEcommunications. METROPOLIS, METROlogie Pour l'Internet et ses Services, http ://www.telecom.gouv.fr/rnrt/, http ://www-rp.lip6.fr/metrologie, 2001.

[RFC1191] J.Mogul, S.Deering. Path MTU-Discovery. RFC1191, November 1990. http ://www.ietf.org/rfc/rfc1191.txt.

[RFC1574] S.Hares, C.Wittbrodt. Essential Tools for the OSI Internet. RFC1574, Février 1994. http ://www.ietf.org/rfc/rfc1574.txt.

[RFC2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis. Framework for IP Performance Metrics. RFC2330, Mai 1998, http ://www.ietf.org/rfc/rfc2330.txt

[RFC2678] J.Mahdavi, V.Paxson. IPPM Metrics for Measuring Connectivity. RFC2678, September 1999. http ://www.ietf.org/rfc/rfc2678.txt.

[RFC2679] G.Almes, S.Kalidindi, M.Zekauskas. A one-way Delay Metric for IPPM. RFC2679September 1999. http ://www.ietf.org/rfc/rfc2679.txt.

[RFC2680] G.Almes, S.Kalidindi, M.Zekauskas. A one-way Packet loss Metric for IPPM. RFC2680.September 1999. http ://www.ietf.org/rfc/rfc2680.txt.

[RFC2681] G.Almes, S.Kalidindi, M.Zekauskas. Round-Trip Delay Metric for IPPM Metric. RFC2681. September 1999.http ://www.ietf.org/rfc/rfc2681.txt.

[RFC3155] S.Dawkins, G.Montenegro, M. Kojo, V. Magret. End-to-end Performance Implications of Links with Errors. RFC3155, August 2001. http ://www.ietf.org/rfc/rfc3155.txt.

[RFC3357] R.Kooldli, Ravikanth. One-way Loss Pattern Sample Metrics for IP Performance Metrics. RFC3357, August 2002.http ://www.ietf.org/rfc/rfc3357.txt.

[RFC3432] V.Raisanen, G.Grotefeld, A.Morton. Network performance measurement for periodic streams. RFC3432, November 2002. http ://www.ietf.org/rfc/rfc3432.txt.

[Rib00] V.Ribeiro, M.Coates, R.Riedi, S.Sarvotham, B.Hendricks, R.Baraniuk. *Multifractal crosstraffic estimation.* in Proceedings of ITC Specialist Seminar, Monterey, USA, 2000.

[Rib03] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L.Cottrell *PathChirp : Efficient Available Bandwidth Estimation for Network Paths.* In Proceedings of Passive and Active Measurement network (PAM'03), La Jolla, USA, Avril 2003

[RIP] RIPE. Réseaux IP Européens. http ://www.ripe.net.

[RNR] RNRT VTHD. Vraiment Très Haut Débit. http ://www.vthd.org.

[Rod05] f.Rody, D.Xanthoulis. *Analyse de sensibilité du modèle mathématique Erosion productivity Impact calculator (EPIC) par l'approche One-Factor-At-A-Time (OAT).* Biotechnologie, agronomie, société et environnement (Biotechnol. agron. soc. Environ) , 9(3), 179-190, 2005.

[Sal00] A.Saltelli, K.Chan, E.M.Scott. Sensitivity Analysis. Wiley Series in Probability and Statistics, John Wiley and Sons, LTD, New York, 2000.

[Sal97] N.Gigolioli, A.Saltelli. A Simlab1.1, Software for sensitivity and uncertainty analysis, tool for sound modelling. http ://simlab.jrc.cec.eu.int/docs/html/index.html, Italy, 1997.

[Sar01] S. Saroiu. Sprobe : A fast tool for measuring bottleneck bandwidth in uncooperative environments. http ://sprobe.cs.washington.edu, 2001.

[Sav99] S. Savage. *Sting : A TCP-based Network Measurement Tool.* In Proceedings of USENIX Symposium on Internet Technologies and Systems, pages 71-79, Boulder, USA, October 1999.

[SB98] A.Saltelli R. Bolado. *An alternative way to compute fourrier amplitude sensitivity test (fast).* Computational Statistics Data Analysis, 26 :445-460, 1998.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RTP : A Transport Protocol for Real-Time Applications, RFC 1889, January 1996.

[Sch73] J.H. Schaibly, K.E. Shuler. *Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients : II Applications.* Journal Chemical Physics, 59 :3879-3888, 1973.

[Shr05] A.Shriram, M.Murray, Y.Hyun, N.Brownlee, A.Broido, M.Fomenkov, k.claffy. *Comparison of Public End-to-End Bandwidth Estimation Tools on High-Speed Links.* In Proceedings of Passive and Active Measurement PAM'2005, Boston, Mars 2005.

[Sou03] A. Soudani. Integration des fonctionnalités multimédias dans les systèmes répartis temps réels : Application au système de communication industrielle. Thèse de doctorat

de l'Université du Centre, Monastir et Université Henri Poincaré, Nancy, Février 2003.

[Sou99] M.Soutter, A.Musy. *Global sensitivity analysis of three pesticide leaching models using a Monte Carlo Approach.* Transactions ASAE 43 (4), p. 883-895, 1999.

[Sur] Surveyor. http ://www.advanced.org/surveyor.

[SW65] S.Shapiro, M.Wilk Test de Shapiro et Wilk. http ://en.wikipedia.org/wiki/Shapiro-Wilk-test.

[Tou02] L.Toumi. Algorithmes et mécanismes pour la qualité de service dans des réseaux hétérogènes. Thèse de doctorat de l'Institut National Polytechniques de Grenoble, 2002.

[Vei06] H.Veiga, R.Valadas, P.Salvador, A.Nogueira, T.Pfeiffenberger, F.Strohmeier. *OWAMP Performance and Interoperability Tests.* 4th International Workshop on Internet Performance, Simulation, Monitoring and Measurement (IPS-MoMe 2006), Salzburg, Février 2006.

[VG96] A. Vega-Garcia. Mécanisme de contrôle pour la transmission de l'audio sur l'Internet. Thèse de doctorat de l'Université de Nice-Sophia Antipolis, INRIA, October 1996.

[Vic04] P.Vicat-Blanc Primet, B.Gaidioz, M.Goutelle. *Approches alternatives pour la différenciation de service IP.* Revue Technique et Science Informatiques TSI, Réseaux et protocoles, 651- 674, 2004.

[Wu97] Q. Wu. Contribution à l'intégration de communications multimédias dans un environnement MMS. Thèse de doctorat de l'Université Henri Poincaré, Nancy, June 1997.

Résumé

Les travaux menés dans cette thèse s'intéressent particulièrement à la mesure de la bande passante disponible qui est un paramètre très important pour le bon fonctionnement de plusieurs applications réseaux et dont la détermination avec précision reste jusqu'à aujourd'hui un défi à relever. Ces travaux visent donc à améliorer les techniques de mesure de ce paramètre en proposant un nouveau modèle déterministe basé sur la technique de la paire de paquets. Ce dernier est implémenté dans un nouvel outil de mesure appelé IGMPS. L'évaluation de performances de cet outil ont montré que ce dernier permet de mesurer la bande passante disponible avec une très grande précision. Par ailleurs, une analyse de sensibilité et un calcul des incertitudes sur les modèles étudiés ont montré que les erreurs dues à l'estampillage des paquets sondes au niveau de l'émetteur et du récepteur sont principalement à l'origine de l'imprécision des mesures fournies par les différents outils de mesure de la bande passante disponible.

Mots-clés: Réseaux de communication, Qualité de service, Métrologie réseaux, Bande passante disponible, Analyse de sensibilité.

Abstract

This thesis work is focused on the end-to-end available bandwidth measurement that has attracted extensive attentions this last decade. This parameter is useful for several network applications, however, its measurement with good accuracy still stays a challenge to take up. To improve the performance of the available bandwidth measurement techniques, we developed a new deterministic model of packet pair delays that takes into account the probing packet size parameter ; and implemented it in a new measurement tool called IGMPS. Through measurements on several network testbed configurations, we evaluated IGMPS and found that it provides available bandwidth measurements with high accuracy. Using sensitivity and uncertainty analysis to study the proposed model, we investigated the sources of observed errors on the measurement tools. We found that these errors are likely to be inherent in delay measurement. Indeed, the timestamping operations at the sender end the receiver are mainly at the origin of the inaccuracy of the estimates provided by the available bandwidth measurement tools.

Keywords: Communication networks, Quality of service, Network measurement, Available bandwidth, Sensitivity analysis.