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

Ahmed Ait Ali — Fabien Michaut — Francis Lepage

CRAN (Centre de recherche en Automatique de Nancy) UMR-CNRS 7039
Université Henri Poincaré, Nancy I
Faculté des sciences et techniques - BP 239
54506 Vandoeuvre Cedex
{ahmed.aitali, fabien.michaut, francis.lepage}@cran.uhp-nancy.fr

RÉSUMÉ: La bande passante disponible est un critère primordial pour le bon fonctionnement de nombreuses applications et technologies réseau. Il existe plusieurs outils qui permettent la mesure de cette métrique cependant on déplore l'absence d'études comparatives de ceux-ci. Cet article propose une étude expérimentale comparative des outils Pathload, Spruce, Pathchirp et IGI. Il se propose aussi d'expliquer et d'analyser les erreurs et les incertitudes dont ces outils font l'objet. Cet article met en évidence les qualités de l'outil Spruce. En effet, celui-ci semble offrir les meilleures performances en terme de précision, de rapidité de mesure et d'intrusivité.

ABSTRACT: Available bandwidth is a key metric for various network applications and technologies. Several available bandwidth measurement tools exist, however there is a lack of comparative studies. The purpose of this paper is to perform a comparative study of Pathload, Pathchirp, Spruce and IGI tools. This paper aims also to explain measurement errors and uncertainties of these tools. It highlights performances of Spruce, which seems to be the best tool in terms of measurement accuracy, measurement time and intrusiveness.

MOTS-CLÉS: Bande Passante Disponible, Métrologie réseau, Qualité de Service

KEYWORDS: Available Bandwidth, Network Metrology, Quality of Service

1. Introduction

L'Internet, initialement conçu pour véhiculer de simples données (textes, emails etc.), est aujourd'hui un réseau universel transportant des informations de natures variées (voix, vidéo etc.) et fortement contraintes en terme de Qualité de Service (QoS). En effet, pour fonctionner dans de bonnes conditions, de nombreuses applications (téléphonie, vidéoconférence, télé-robotique, etc.) doivent bénéficier d'une QoS minimale qui s'exprime habituellement par des métriques telles que la bande passante, le délai, les pertes de paquets, etc.

Parmi ces critères, la bande passante disponible revêt une importance primordiale pour le fonctionnement de nombreuses technologies et applications réseau: par exemple, l'adhésion dynamique des membres dans certaines applications *Peer-to-Peer* est basée sur la bande passante disponible entre les membres. La bande passante disponible est déterminante pour le choix des taux de compression dans les applications audio et vidéo. Enfin, ce paramètre joue un rôle prépondérant dans l'élaboration des contrats de service entre les opérateurs et leurs clients (*SLA: Service Level Agreement*).

Certaines analyses et mesures de cette métrique ont été réalisées. Cependant, on déplore l'absence d'études comparatives significatives des résultats expérimentaux. On constate aussi l'absence d'un "outil référence" auquel seraient comparées les performances des outils de mesure de bande passante disponible lors de leur développement. Le but de cet article est ainsi de présenter les différentes techniques de mesure de bande passante disponible et d'étudier les outils de mesure qui les implantent. Cette analyse expliquera l'effet de l'encapsulation au niveau de la couche liaison sur la grandeur réelle à mesurer et mettra en lumière les incertitudes de mesure dont ces outils font l'objet. Il présentera aussi une étude comparative des performances de ces outils en se basant essentiellement sur les aspects précision et charge induite.

La deuxième partie de cet article introduira les définitions des métriques relatives à la bande passante et présentera l'effet de l'encapsulation sur la mesure de celles-ci. La troisième partie présentera les méthodes et outils de mesure de la bande passante disponible et proposera une classification de ceux-ci. Une étude comparative de ces outils sera présentée dans la quatrième partie. La cinquième partie mettra en évidence la nature et la cause des incertitudes de mesure. Enfin, nos conclusions et les perspectives à nos travaux seront énoncées dans la sixième partie.

2. Bande passante et bande passante disponible

2.1. Définitions

Le terme "bande passante" regroupe quatre paramètres différents, à savoir la bande passante totale d'un lien ou capacité d'un lien, la bande passante minimale d'un chemin ou 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, \dots, 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 $C = \min C_i$

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 :

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

2.2. Effet de l'encapsulation au niveau de la couche 2

La valeur de la capacité d'un lien est différente selon le 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 [2]. Le temps de transmission T_3 d'un datagramme IP de taille L_3 octets est :

$$T_3 = \frac{(L_3 + H_2) \times 8}{C_i^{N2}} \quad \text{avec } H_2 \text{ la taille totale des entêtes (en octets).}$$

La valeur de la bande passante au niveau 3 est ainsi:

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

Cette équation met en évidence que la bande passante au niveau 3 dépend de la taille des paquets IP et de la taille des entêtes ajoutées par la couche 2. 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).

3. Techniques et outils de mesure de bande passante disponible

Cette section présente les techniques et outils de mesure étudiés et propose de les classer en deux catégories: la première comprend les techniques "basées sur l'auto-congestion" qui permettent une estimation directe de la bande passante disponible. La seconde regroupe les techniques basées sur "l'évaluation du débit du trafic concurrent", qui estiment de manière indirecte la bande passante disponible.

3.1. Techniques et outils basés sur l'auto-congestion (ou modèle PRM¹ [3])

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 . (cf. figure 1).

¹ Probe Rate Model

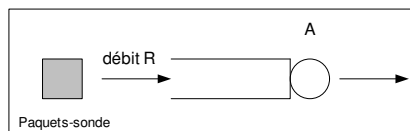


Figure 1. Illustration du modèle PRM

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.

Pathload [1] [4] et Pathchirp [5] implantent ce modèle. Pathload utilise la technique SLoPS (Self-Loading Periodic Stream) qui consiste à envoyer une rafale de paquets sondes d'une source vers une destination. 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 inférieur dans le second cas. Ce mécanisme est répété et on approche par dichotomie la valeur de la bande passante disponible. Pathchirp propose d'envoyer des sondes selon un processus exponentiel (cf. figure 2). L'avantage principal de cette approche est de minimiser le volume du trafic de mesure. En effet, un seul train de paquets permet de sonder le réseau à des débits différents.

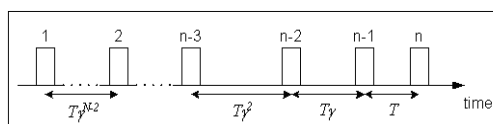


Figure 2. Train de paquets sondes utilisé par Pathchirp

3.2. Techniques et outils basés sur l'évaluation du débit du trafic concurrent

Les techniques et outils de cette catégorie supposent la capacité du chemin C connue, et font l'hypothèse forte que le lien serré et le lien étroit² sont confondus en un lien appelé goulet d'étranglement. 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: $A=C-C_T$.

² Le lien "étroit" est le lien du chemin dont la capacité est la plus faible. Le lien "serré" est le lien du chemin dont la bande passante disponible est la plus faible.

Ces techniques et outils se fondent sur le *Probe Gap Model (PGM)* [3]. Ce modèle 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.

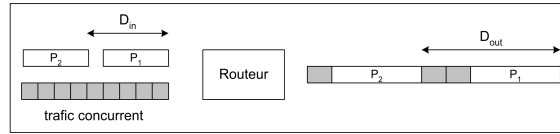


Figure 3. *Probe Gap Model (PGM)*

Dans [8], 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 D_{out} et le débit du trafic concurrent (cf. figure 3). En effet, la dispersion de sortie D_{out} est égale à la somme du temps de traitement du premier paquet, c'est-à-dire la taille du paquet divisé 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 $C_T D_{in}$ arrivé au goulet entre les deux paquets divisé par la capacité du goulet C . Cette relation est valable si et seulement si la file du routeur "goulet" ne se vide pas entre l'instant auquel le premier paquet quitte le routeur et l'instant auquel arrive le deuxième paquet. Cette condition est appelée condition *Joint Queuing Region (JQR)* [7]. La bande passante disponible est alors exprimée par :

$$A = C - C_T = \frac{C(D_{in} - D_{out}) + L}{D_{in}} \quad (1)$$

Le *Probe Gap Model (PGM)* est implanté dans les outils Spruce [3] et IGI [7].

a. *Spruce*

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 D_{in} . Pour s'assurer de respecter la condition *JQR*, Spruce ajuste D_{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 en utilisant la formule :

$$A = C - C_T = \left(2 - \frac{D_{out}}{D_{in}}\right) \cdot C \quad (2)$$

La valeur finale renvoyée à l'utilisateur est la moyenne des 100 estimations.

b. *IGI (Initial Gap Increasing)*

IGI est 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 D_{in} . IGI débute son exécution par la mesure de la capacité du chemin C^3 .

³ Cette mesure est effectuée en utilisant la méthode introduite dans Nettimer [8].

Les expérimentations menées dans [8] ont montré que, pour respecter la condition JQR et ne pas saturer le goulet d'étranglement⁴, la valeur optimale de D_{in} est obtenue quand la moyenne des D_{out} est égale à D_{in} . Ainsi IGI comprend une phase de recherche du meilleur écart initial: l'émetteur commence par envoyer des paquets avec un faible D_{in} et l'augmente jusqu'à ce que la moyenne des D_{out} soit égale à D_{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 D_{out} sont corrélées. Pour limiter cet effet, IGI sélectionne uniquement les échantillons de $D_{out} > D_{in}$ pour effectuer ses calculs.

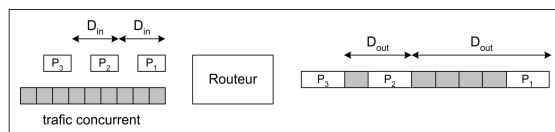


Figure 4. Initial Gap Increasing (IGI)

4. Etude comparative des performances

L'objectif de cette étude est de comparer les performances des outils de mesure décrits précédemment en termes de précision, de rapidité et "d'intrusivité". Pour cela, nous avons mis en place une plateforme d'expérimentation qui permet de tester les outils dans des conditions identiques. Elle consiste en un ensemble d'ordinateurs interconnectés via deux commutateurs (*switchs*) :

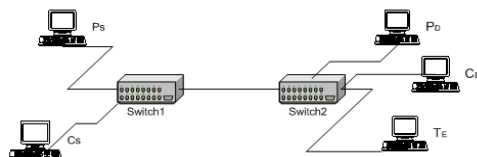


Figure 5. Plateforme d'expérimentation

Les émetteurs et récepteurs des outils de mesure sont installés respectivement sur P_S et P_D . Les flux de mesures circulent ainsi de P_S à P_D . Installé sur C_S , le générateur de trafic MGEN [9] permet d'envoyer un trafic concurrent (flux périodique de paquets) vers C_D . L'outil de capture Ethereal est installé sur T_E . Il permet de vérifier le débit du trafic généré par MGEN et de mesurer le temps d'exécution et la charge induite par chaque outil.

4.1. Expérimentations

Le débit du trafic concurrent est réglé de manière à faire varier la bande passante disponible de 0 et 100 Mb/s. Pour chaque valeur de cet intervalle, on effectue 30

⁴ Une augmentation de D_{in} mène à une diminution du domaine de validité du modèle PGM. D_{in} doit être inférieure ou égale au temps de transmission d'un paquet-sonde au goulet d'étranglement. Un écart initial trop petit conduit à saturer le goulet d'étranglement. En effet, lorsque la valeur de D_{in} est trop faible, peu de paquets du trafic concurrent s'intercalent entre les paquets sonde et il y a sous-estimation de C_T .

mesures avec chaque outil. La valeur estimée de la bande passante disponible reportée est la moyenne des 30 valeurs obtenues. Les résultats des expérimentations sont représentés sur la figure 6.

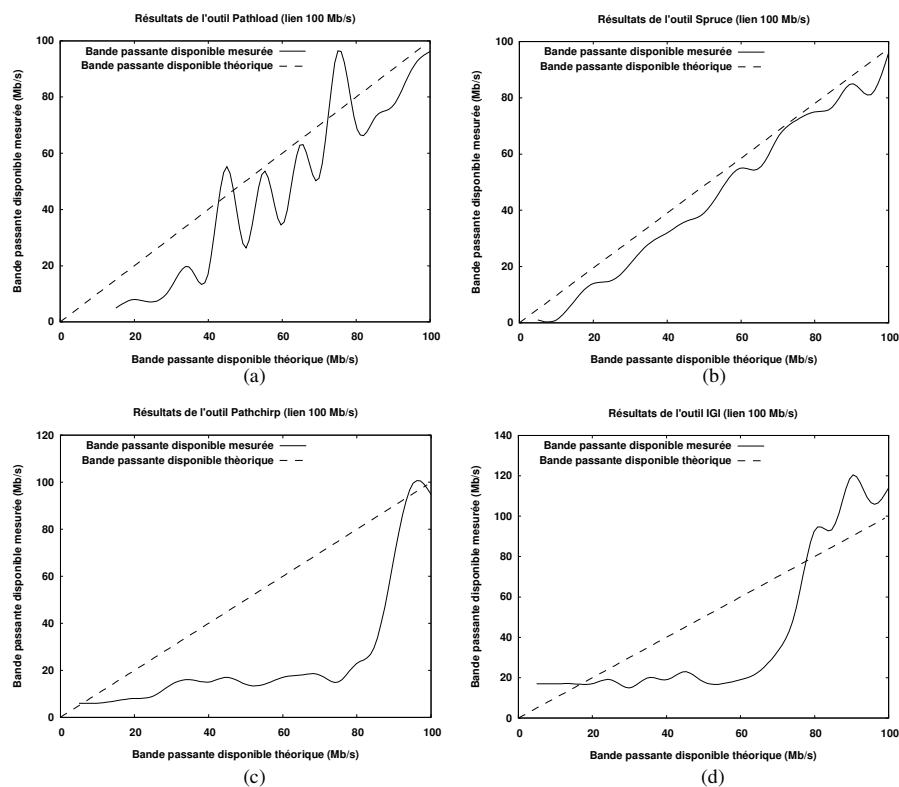


Figure 6. Bande passante disponible estimée par les différents outils

Les courbes de la bande passante disponible théorique sont calculées en fonction de la taille des paquets sondes utilisés par chaque outil afin de tenir compte de l'encapsulation. La courbe obtenue avec Pathload oscille pour les valeurs de bande passante inférieures à 80 Mb/s. Globalement, Pathload sous-estime la bande passante disponible théorique à l'exception de quelques points. Cependant, l'allure de la courbe suit les variations des valeurs théoriques.

Remarque: Nous avons constaté lors de ces expérimentations que Pathload stoppe prématurément son exécution lorsque la bande passante disponible est inférieure à 15 Mb/s (soit 15% de la capacité du chemin), et n'est donc pas capable d'effectuer des mesures pour ces valeurs. Ceci peut s'expliquer par le fait qu'il

intègre un mécanisme de détection des pertes de paquets-sonde: Si pour un SLoPS, 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. On peut ainsi supposer que lorsque la bande passante disponible est faible, les flux générés ont un débit "trop important" et subissent des pertes de paquets, ce qui a pour conséquence de rendre la mesure impossible. Nous avons observé un phénomène identique (mesure impossible pour des bandes passantes inférieures à 15% de la capacité) pour d'autres valeurs de capacité.

L'allure de la courbe obtenue avec Pathchirp ne suit pas les variations des valeurs théoriques. Pathchirp sous-estime largement la bande passante disponible pour les valeurs inférieures à 95 Mb/s et réagit très peu (voir pas du tout) pour les valeurs inférieures à 75 Mb/s.

Remarque: 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.

La courbe obtenue avec Spruce suit les variations des valeurs théoriques. L'écart entre les courbes est d'ailleurs assez faible. Cependant, Spruce sous-estime légèrement les valeurs théoriques dans tous les cas. IGI ne réagit aux variations de la bande passante disponible que pour les valeurs théoriques supérieures à 60 Mb/s. La bande passante disponible est sous-estimée pour les valeurs comprises entre 20 et 80 Mb/s, surestimée pour les autres.

De cette première observation des mesures, nous pouvons constater que Spruce semble être l'outil qui réagit le mieux aux variations de la bande passante disponible.

4.2. Justesse (précision)

Pour étudier la justesse des outils, nous étudions à présent les erreurs de mesure relatives. L'erreur relative est calculée par la formule suivante:

$$\varepsilon = \frac{|valeur_théorique - valeur_mesurée|}{valeur_théorique}$$

Les erreurs relatives des outils sont représentées sur la figure suivante.

L'erreur relative de Pathload est très importante (supérieure à 50%) lorsque la bande passante disponible est inférieure à 40 Mb/s. Elle ne dépasse plus 20% pour les valeurs de bande passante disponible supérieures à 75 Mb/s.

Les résultats obtenus pour Pathchirp montrent que cet outil est très imprécis. Dans la majorité des cas, l'erreur relative est supérieure à 50%. Elle n'est inférieure à 20% que lorsque la bande passante disponible est supérieure à 90Mb/s.

Spruce est l'outil qui a globalement l'erreur relative la plus faible. Celle-ci est inférieure à 20% pour les valeurs de bande passante disponible supérieures à 35

Mb/s. L'erreur relative est cependant très importante lorsque la bande passante disponible est inférieure à 15 Mb/s.

L'erreur relative de IGI est particulièrement importante lorsque la bande passante est inférieure à 15 Mb/s, certaines valeurs dépassant les 200%! L'erreur relative est inférieure à 20% lorsque la bande passante disponible est comprise entre 15 et 25 Mb/s et supérieure à 75 Mb/s. Pour cet outil, il est difficile de tirer des conclusions étant donné qu'il base ses calculs sur une mesure préliminaire de la capacité qui est elle-même sujette à erreur. Ce point est détaillé dans la partie suivante (cf. § 5).

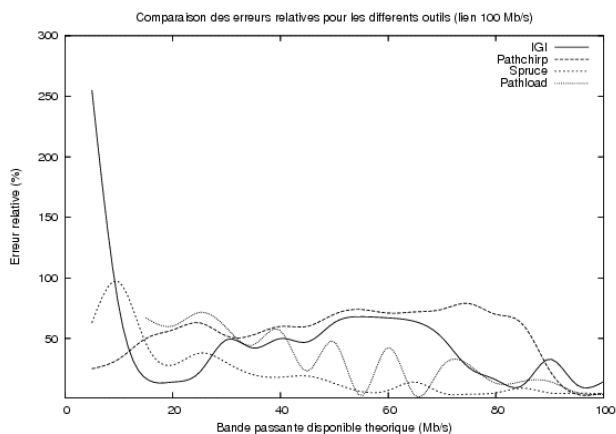


Figure 7. Comparaison des erreurs relatives

On constate ainsi que contrairement à Pathload, Spruce est capable de mesurer des petites valeurs de bande passante disponible. Cependant, il le fait avec une faible précision. Pour les valeurs de la bande passante disponible dépassant 30Mb/s, les erreurs relatives sur les mesures effectuées par Spruce sont inférieures à 20%. En revanche, Pathchirp et IGI montrent des taux d'erreurs excessivement élevés. En offrant la meilleure justesse⁵, Spruce prend l'avantage une fois de plus sur les autres outils.

4.3. Intrusivité

L'intrusivité représente la quantité de trafic généré par un outil de mesure. 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 ci-dessous représente les débits moyens des flux de mesure générés par les différents outils.

⁵ Pathload n'offre pas une excellente précision, mais cette dernière reste bien supérieure à celles offertes par Pathchirp et IGI.

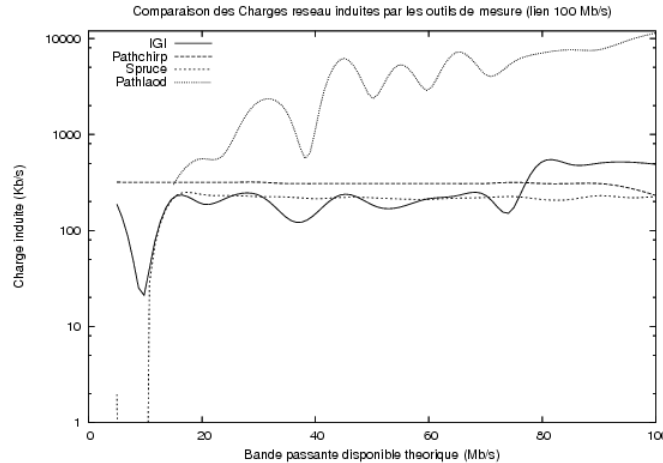


Figure 8. La charge réseau induite par chaque outil (échelle logarithmique)

La figure 8 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. Ceci s'explique par la technique des SLoPS (cf. § 3.1).

Pour les autres outils, les débits sont plus modestes. Les débits de Pathchirp et Spruce sont constants quelle que soit la bande passante disponible, respectivement de 300 et 220 kb/s. Le débit de 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 que 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 pour la mesure de la bande passante disponible est ainsi inférieur aux valeurs reportées ici.

4.4. Rapidité

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 rapidement. Les mesures doivent être effectuées d'autant plus rapidement lorsqu'elles sont employées pour configurer dynamiquement des applications⁶.

⁶ 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, distribution des charges sur un réseau, vérifier des SLA, etc.).

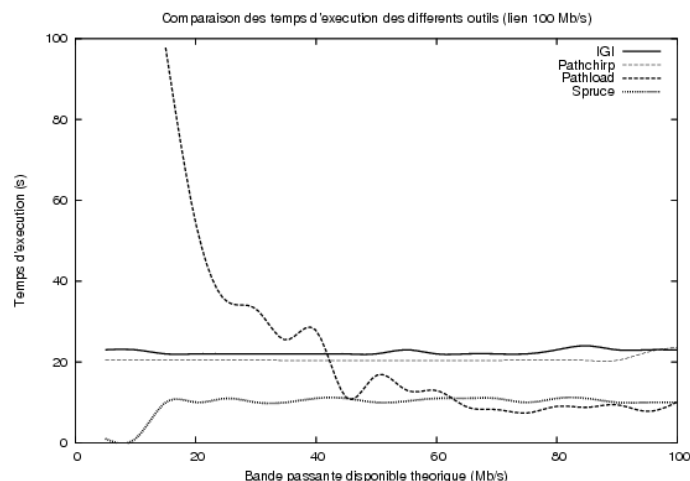


Figure 9. Temps d'exécution des outils de mesure

La figure 9 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 et IGI et 10s pour Spruce. Notons que le temps de mesure de 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.

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) Pathchirp est imprécis. (3) Spruce offre globalement les meilleures performances en termes de justesse, d'intrusivité et de rapidité. De plus, ses performances sont relativement stables et indépendantes de la valeur de la bande passante disponible. (4) A priori, IGI ne semble pas être un outil précis. Toutefois, la qualité des résultats obtenus avec cet outil dépend de la mesure préliminaire de la capacité. Il est ainsi difficile de conclure quant à la précision de cet outil.

Dans le paragraphe suivant, nous analysons les erreurs et incertitudes dont ces outils font l'objet.

5. Analyse des incertitudes de mesure

5.1. Incertitude à l'envoi des paquets

Les outils de mesure étudiés dans cet article 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'envoi⁷.

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 10. La latence à mesurer est $\Delta t = t_f - t_d$.

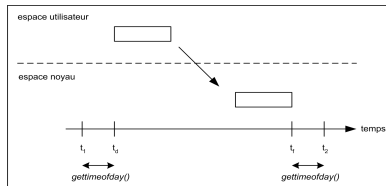


Figure 10: Passage du paquet de l'application au pilote de l'interface

Les différentes opérations d'estampillage sont effectuées par l'appel système `gettimeofday()`. Cet appel ajoute une latence supplémentaire dans la mesure de Δt . En effet, la date t_d sera estimée par t_1 , t_f par t_2 . En notant $t_{\text{gettimeofday}()}$ le temps d'exécution de l'appel `gettimeofday()`, on a alors $\Delta t = t_2 - t_1 - 2 \times t_{\text{gettimeofday}()}$ (3)

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

Le temps d'exécution 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 100 000 fois. Les temps d'exécution obtenus varient de 1 à 6 μs . Il faut noter que ces valeurs dépendent de la puissance de la machine sur laquelle sont effectuées les mesures, ici des Pentium IV, 2,5 GHz⁸.

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 1. Les valeurs de Δt sont calculées de la façon suivante (cf. formule 3) :

⁷ 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.) [11].

⁸ Ces valeurs sont cohérentes avec la valeur maximale obtenue dans [11] de 10 μs sur des machines moins puissantes (Pentium III, 500 MHz).

- Valeur maximale obtenue pour la valeur minimale de $t_{\text{gettimeofday()}}$, soit $1\mu\text{s}$.
- Valeur minimale obtenue pour la valeur maximale de $t_{\text{gettimeofday()}}$, soit $6\mu\text{s}$.

Taille des paquets IP (en octets)	Nombre d'expériences	t_2-t_1 (en μs)	Δt (en μs)
500	1000	0 – 41	0 – 39
1500	1000	0 – 50	0 – 48

Tableau 1. Mesure de la latence espace système - buffers interface réseau

Remarques : (1) L'obtention de valeurs minimales nulles s'expliquent par la durée relativement longue (par rapport à la valeur de temps mesurée) de l'exécution de l'appel *gettimeofday()* (la valeur minimale n'est pas mesurable). (2) 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, Δt est inférieur ou égal à $20\mu\text{s}$. L'histogramme (non présenté ici), montre une concentration des valeurs autour de $10\mu\text{s}$.

Impacts sur les outils étudiés

Spruce et IGI sont a priori les outils les plus susceptibles d'être affectés par l'incertitude liée à l'envoi des paquets. En effet, pour ceux-ci, il est impératif d'envoyer des paquets avec un temps inter-paquets D_{in} donné de l'ordre d'une centaine à plusieurs centaines de μs . Une variation de Δt de l'ordre de $10\mu\text{s}$ entraîne ainsi une erreur relative importante sur D_{in} . La valeur de D_{in} étant utilisée pour calculer l'estimation de la bande passante disponible A , toute erreur sur D_{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 l'outil Spruce:

Spruce estime la bande passante disponible en utilisant la formule (5). Nous définissons ΔD_{in} l'erreur sur D_{in} . En supposant qu'il n'y ait pas d'erreur sur la mesure de C et D_{out} , on a :

$$\Delta A = \left| -\frac{C \cdot D_{out}}{D_{in}^2} \right| \Delta D_{in}$$

Ainsi, pour une capacité C de $97,5 \text{ Mb/s}$ (niveau IP), une erreur $\Delta D_{in}=10\mu\text{s}$ (erreur la plus fréquente), une bande passante disponible A de 50 Mb/s , nous obtenons une incertitude $\Delta A = 11,78 \text{ Mb/s}$, soit 23%. Cette valeur correspond aux valeurs du § 4.2. 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 $830\mu\text{s}$ 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 [5] 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, l'incertitude étudiée ici est donc négligeable.

5.2. Incertitude 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 Δt , le temps de passage d'un paquet depuis son arrivée dans le système d'exploitation jusqu'à l'espace utilisateur⁹. Les valeurs obtenues varient entre 5 et 65 μ s. Δt est inférieur ou égal à 25 μ s pour, respectivement, 80 % et 91 % des mesures effectuées avec des paquets de 500 et 1500 octets.

Impacts sur les outils étudiés

Spruce, IGI 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, Spruce et Pathchirp utilisent l'option *SO_TIMESTAMP* dans la socket de réception et IGI réalise les opérations d'estampillage avec la librairie *libpcap*. Pour effectuer ses mesures, Pathload mesure les délais des paquets à leur arrivée et en détecte la variation. Les délais et leurs variations observés de nos jours sur les réseaux WAN sont de l'ordre de plusieurs dizaines voire centaines de ms. Aussi, nous pouvons supposer que l'incertitude étudiée ici est négligeable.

5.3. Incertitude résultant de l'estimation de la bande passante minimale

Les outils basés sur le modèle PGM (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 IGI en appliquant la méthode différentielle logarithmique.

⁹ L'estimation de Δt n'est pas détaillée ici pour des raisons de place.

IGI estime la bande passante disponible en utilisant la formule (3). Ainsi, l'erreur sur A est¹⁰:

$$\Delta A = \left| 1 - \frac{D_{out}}{D_{in}} \right| \Delta C$$

Ainsi, pour une capacité C de 97,5 Mb/s (niveau IP), une erreur $\Delta C = 20$ Mb/s (environ 20% de C), une bande passante disponible A de 40 Mb/s et en fixant D_{in} tel que:

$$D_{in} = \frac{L}{C}, \text{ cette valeur correspond à la valeur maximum de } D_{in} \text{ pour IGI.}$$

Nous obtenons une incertitude $\Delta A = 11,60$ Mb/s, soit 29%. Cette valeur est donc très importante et explique une bonne part de l'erreur relative de IGI présentée dans le § 4.2.

5.4. Incertitudes dues à la corrélation des paquets

Ce phénomène est spécifique à IGI. En effet, ce dernier utilise des 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 D_{out} mesurées par le récepteur sont ainsi corrélées. Pour atténuer cet effet, IGI ne prend pas en compte les D_{out} inférieures ou égales à D_{in} . Ceci a pour effet de surestimer le débit du trafic concurrent et donc de sous-estimer la bande passante disponible.

6. Conclusion

Nous avons présenté dans cet article une analyse et une étude comparative des techniques et des outils de mesure de la bande passante disponible. Nous avons introduit les différentes méthodes utilisées dans le domaine et nous avons mentionné l'effet de l'encapsulation au niveau de la couche 2. Les différents outils ont été mis en oeuvre sur une plateforme d'expérimentation dans des conditions similaires. Cette étude a permis de comparer les performances de ces outils en termes de précision, d'intrusivité et de rapidité. Nous avons, de plus, tenté d'analyser et d'expliquer les sources d'erreurs et d'incertitudes dont ces outils font l'objet.

Les résultats obtenus montrent que Pathload est l'outil le plus intrusif et peut s'avérer très lent. Pathchirp est un outil généralement imprécis. A priori, IGI n'offre pas une bonne précision. Toutefois, comme nous l'avons expliqué, les résultats obtenus pour ce dernier doivent être nuancés car celui-ci intègre un mécanisme de mesure de capacité qui est susceptible de dégrader sa précision et diminuer sa rapidité.

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 fera l'objet d'études complémentaires notamment sur des réseaux "réels".

¹⁰ En supposant, qu'il n'y a pas d'erreur sur D_{in} et D_{out} .

L'étude présentée dans cet article s'est concentrée sur les critères qui nous semblaient essentiels. Cependant, 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. Ces différents points feront l'objet de nos futurs travaux.

7. Bibliographie

- [1] M. Jain, C. Dovrolis, "Pathload: a measurement tool for available bandwidth estimation", *Proc. PAM'02*, 2002.
- [2] R.S.Prasad, M.Murray, C.Dovrolis, K.C.Claffy "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools", *IEEE Network*, November-December 2003 issue.
- [3] Jacob Strauss, Dina Katabi, and Frans Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," *The Internet Measurements Conference*, Floride, 2003.
- [4] M.Jain, C, Dovrolis, "End-to-End Available Bandwidth: Measurement methodology, Dynamics, and Relation with TCP Throughput". *IEEE/ACM Transactions in Networking*, Août 2003.
- [5] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", *Passive and Active Measurement Workshop 2003*.
- [6] B. Melander, M.Björkman, "A new end-to-end Probing and analysis method for estimating bandwidth bottlenecks" *IEEE GLOBECOM'00*.
- [7] Ningning Hu, Peter Steenkiste. "Evaluation and Characterization of Available Bandwidth Probing Techniques". *In the IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, Vol. 21(6), Août 2003.
- [8] Kevin Lai and Mary Baker, "Nettimer: A Tool for Measuring Bottleneck Link Bandwidth", *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Mars 2001.
- [9] Mgen "Multigenerator tool set", <http://mgen.pf.itd.nrl.navy.mil>
- [10] S. McCanne and V. Jacobson. "The BSD Packet Filter: A New Architecture for User-Level Packet Capture". In *Proc. Winter '93 USENIX Conference*, San Diego, CA, January 1993.
- [11] D. Veitch, A. Pasztor, "A Precision Infrastructure for Active Probing," in *Proc. Passive and Active Measurements (PAM) Workshop*, 2001.
- [12] R. carter M. Corvella, "Measuring bottlenck link speed in packet-switched networks" *technical report 1996-006*, Boston University, Mars 1996.