Première partie

Sous Projet 2

Classification et Dimensionnement

Chapitre 1

Usage des clients ADSL

1.1 Introduction

On s'intéresse dans ce chapitre à l'usage des clients ADSL et plus particulièrement à leur comportement en termes de débit et d'utilisation des applications les plus populaires de l'Internet, incluant les applications pair à pair (*peer-to-peer*). Dans le livrable précédent du sous-projet 2, un certain nombre d'éléments sur le trafic ADSL tel qu'il est observé dans le réseau de France Télécom ont été fournis. Les points suivants ressortent clairement de cette précédente étude :

- Prépondérance du trafic pair à pair (p2p), qui représente près de 80% du trafic observé sur un lien descendant du cœur de réseau vers plusieurs plaques ADSL.
- Activité très intense en termes de signalisation pour le trafic p2p. Les messages de signalisation sont en général de petite taille et arrivent en avalanche ; en regroupant ces messages de manière astucieuse, il est possible d'introduire la notion d'appel dans un réseau p2p, que soit à l'initiative d'un client recherchant un contenu dans un réseau p2p ou que ce soit un appel de "maintenance" initialisé par un nouveau membre se connectant à un réseau (échange de tables de connectivité et de contenus).
- Les transferts de fichiers sur une seule connexion TCP sont en général de taille limitée (de l'ordre de quelques dizaines de Megaoctets), ce qui a tendance à éliminer tout dépendance à long terme dans le trafic. Ce phénomène est intimement lié à la manière dont fonctionne les protocoles p2p, en particulier eDonkey qui est le protocole p2p dominant dans les observations effectuées dans le cadre du projet Metropolis. Ces protocoles mettent en œuvre des télé-chargements en parallèle et asynchrones de portions ("chunks") de fichiers.
- Une part importante (30%) des éléphants (flots correspondant aux transferts de données) est composée de flots longs ne comportant que des segments d'acquittement de petite taille. Comme on observe le trafic descendant, ceci indique que des masses importantes de données sont télé-chargées à partir de terminaux connectés aux plaques ADSL observées par des clients extérieurs à ces plaques. Cette observation révèle que des terminaux de clients ADSL jouent le rôle de serveurs et donc que l'on assiste à une symétrisation des usages avec la dissémination des protocoles p2p sur tout l'Internet.

Dans ce chapitre, on examine de manière plus détaillée l'usage des clients ADSL et plus particulièrement les applications qu'ils utilisent. On adopte une approche par agrégation des usages pour

regrouper les clients par classes. Lees clients dans une même classe ont des comportements homogènes vis à vis des usages des applications et des débits, aussi bien journalier que sur des périodes plus longues (la semaine ou le mois). Ceci permet de faire ressortir l'existence de clients qui ont des usages intensifs de leur connexion ADSL aussi bien dans le sens remontant que descendant. Ces clients engendrent une part importante du trafic, mais sont pour l'instant peu nombreux. Avec l'accoutmance à l'ADSL, la proportion de ces clients hyper-actifs a tendance à croître. Les modifications de la legislation sur le p2p et l'évolution des abonnements (facturation au volume) risquent cependant de modifier la typologie des clients dans l'avenir.

1.2 Cadre expérimental

Des sondes sont installées en sortie de BAS (*broadband access server*) et observent le trafic descendant et remontant vers plusieurs plaques ADSL (cf. la figure 1.1). Une telle sonde permet d'observer le trafic d'environ 2000 clients et d'analyser le niveau applicatif, c'est à dire d'avoir une connaissance de l'application utilisée par tel ou tel client. Au total, 8 sondes sont installées sur des sites différents du réseau de France Telecom (géographiquement éloignés et avec des densités de populations différentes). Le trafic analysé dans ce rapport est observé depuis janvier 2003.

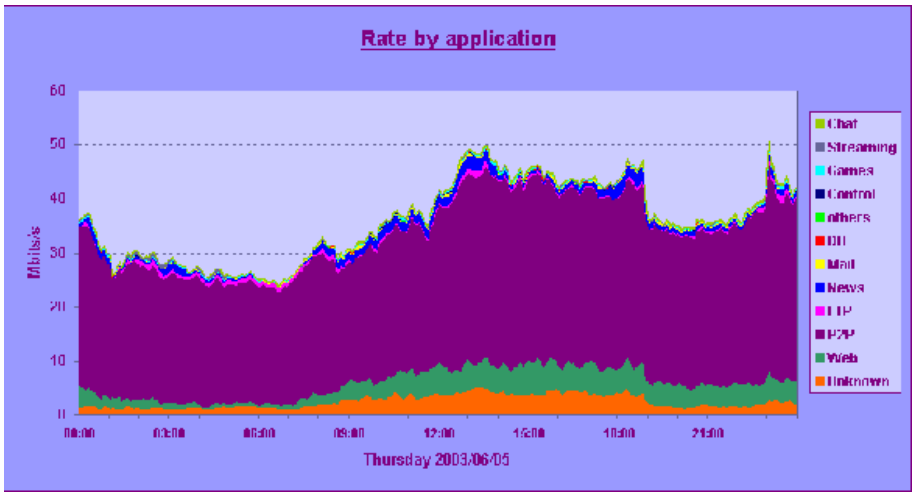


FIG. 1.1 – Cadre expérimental.

Les données collectées fournissent les volumes de données échangées dans les deux directions et pour 12 types d'applications (Web, p2p, ftp, News, Mail, DB, Control, Games, Streaming, Chats, Autres et Inconnues). Les données sont regroupées par périodes de 6 minutes pour chaque client. La plupart des applications correspondent à des ports TCP bien connus, saut évidemment pour les deux dernières classes. Comme la plupart des protocoles p2p utilisent des ports dynamiques, les application p2p sont reconnues à partir d'une analyse du contenu des paquets IP (niveau applicatif).

Le but de cette étude est de mieux comprendre l'usage des clients en termes d'applications et plus précisément de créer des profils d'usage des clients ADSL. Par exemple, une classe de clients peut

être caractérisée par un usage intensif du courrier électronique et par un faible trafic p2p. Ainsi, on recherche dans les données collectées une combinaison des applications qui permet de caractériser une classe de clients. On commence par l'observation de l'activité d'un client par son trafic sur les douze applications introduites ci-dessus.

1.3 Segmentation des clients selon leur usage du réseau

1.3.1 Description du trafic

La figure 1.2 représente la distribution du trafic total mensuel par application (tous les jours et tous les clients) pour un site en Février 2003 ; les volumes sont donnés en octets. Environ 90% du trafic sont dus aux applications p2p et Inconnues. Tous les sites observés présentent des distributions similaires.

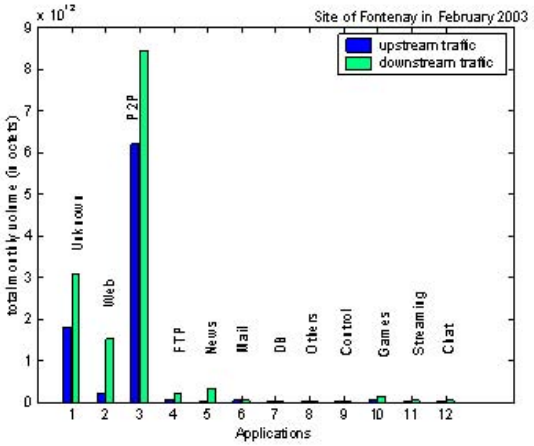


FIG. 1.2 – Volume de trafic par application.

La figure 1.2 montre clairement que les volumes échangés varient de plusieurs ordres de grandeur suivant l'application. Comme notre objectif est de comparer le niveau d'utilisation de chaque application, nous avons normalisé nos données par les statistiques de rang : pour une application donnée, son volume est caractérisé suivant son rang par rapport aux autres volumes (tous les utilisateurs inclus) sur une même application. Le rang est ensuite normalisé entre 0 et 1. Cette caractérisation est simple et robuste, et fournit l'information souhaitée : un client avec une forte (resp. faible) utilisation d'une application sera caractérisé par un rang proche de 1 (resp. 0) pour cette application.

1.3.2 Segmentation par des cartes auto-organisées

Les données collectées sont segmentées en utilisant des cartes auto-organisées (*self-organizing maps*, SOM). Ce type d'outil est utile pour l'analyse exploratoire de données par l'intermédiaire de nœuds placés sur une gille bidimensionnelle, celle-ci formant la carte proprement dite. Ce type de

représentation conserve les proximités : des observations proches dans l'espace multidimensionnel d'entrée sont associées à des nœuds proches sur la grille. Après l'apprentissage, la grille est segmentée selon plusieurs amas, chacun étant composé de nœuds avec des comportement similaires. La segmentation est effectuée à l'aide d'un algorithme d'amas agglomératif et simplifie grandement l'interprétation de la carte.

Les cinq sites ont été analysés séparément dans un premier temps. Nous présentons les résultats pour le site de Fontenay en Février 2003 (2292 clients) avec une carte carrée de 7x7 nœuds avec des voisinages hexagonaux. La figure 1.3 représentent les 7 amas de clients avec des comportements similaires révélés après analyse. Chaque amas est identifié par un numéro.

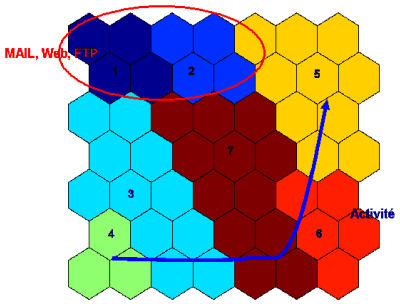


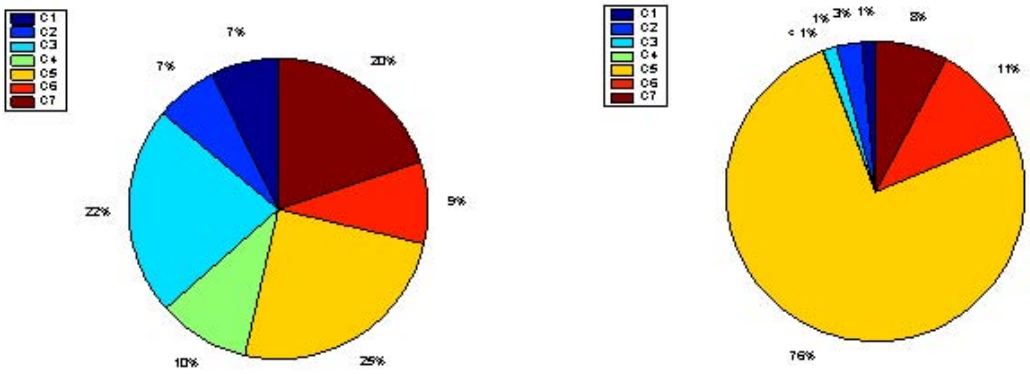
FIG. 1.3 – Carte décrivant les comportements par application et les 7 amas associés.

Les amas peuvent être interprétés en termes de comportements par application, c'est à dire, comme des combinaisons de forte et faible utilisation sur les douze applications identifiées au départ. Parmi tous les amas identifiés, deux amas de clients sont caractérisés par des fortes utilisations sur toutes les applications (amas 5 et 6 sur l'est de la carte), un amas avec un usage moyen de toutes les applications (amas 7), deux amas avec un faible ou très faible usage de toutes les applications (amas 3 et 4 dans le sud-ouest de la carte) et deux amas (1 et 2 dans le nord-ouest) avec des utilisations très fortes sur des applications spécifiques (mail, Web, ftp, streaming) et une activité faible sur les autres (en particulier le p2p).

L'ordonnancement des nœuds sur la carte est tel que les amas avec des comportements voisins sont proches sur la carte et il est alors possible de visualiser comment le comportement évolue progressivement quand on se déplace sur la carte. La carte est globalement organisée le long d'une ligne partant du coin sud-ouest (amas 4) vers le nord-est (amas 5) et décrivant une progression d'une activité faible sur toutes les applications vers une activité forte sur toutes les applications.

Les clients avec une activité très forte sur toutes les applications (25% des clients de l'amas 5) engendrent 76% du volume global par mois. Ces clients sont spécialement actifs sur les applications p2p et de jeux ; ils produisent plus de 85% du volume associé à ces applications. Les clients avec un faible usage des applications (32% des clients dans les amas 3 et 4 avec 2% du volume mensuel)

sont actifs sur le Web, les News et le courrier électronique. La distribution des clients dans les amas et la distribution des volumes associés sont donnés par la figure 1.4.

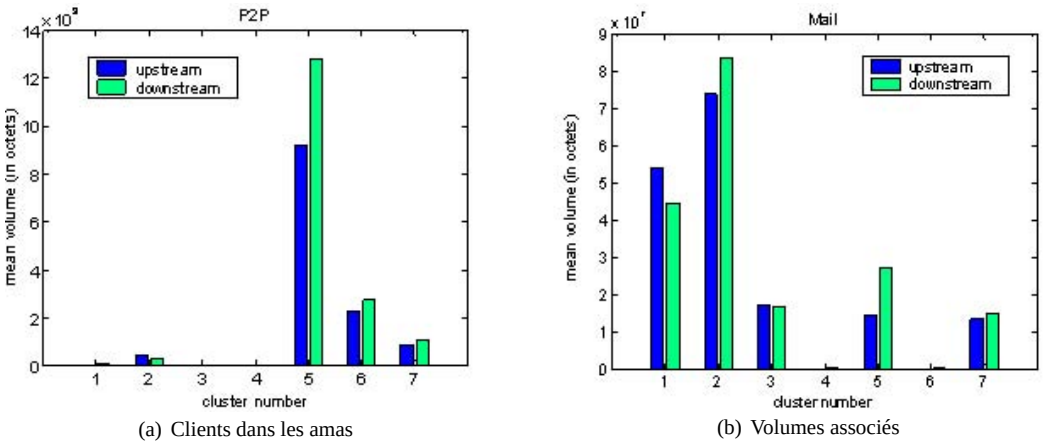


(a) Clients dans les amas

(b) Volumes associés

FIG. 1.4 – Distribution des clients dans les amas et répartition des volumes associés.

La figure 1.5 représente les volumes moyens associés avec chaque amas pour le courrier électronique et le p2p. Le volume moyen d'un amas est calculé en estimant la moyenne des volumes consommés sur les applications par les clients de l'amas correspondant. Chaque figure permet de comparer les amas du point de vue des applications et illustre les différents comportements et consommations en termes de volume. Les clients des amas 5 et 6 sont des "heavy users" pour le p2p (en termes de statistiques de rang) mais exhibent des volumes moyens fort différents (cf. la figure 1.4).



(a) Clients dans les amas

(b) Volumes associés

FIG. 1.5 – Volumes moyens de p2p et de courrier électronique.

Tous les clients, à l'exception des amas 4 et 6, ont un volume significatif en termes de courrier électronique. Les plus forts utilisateurs sont ceux des amas 1 et 2. En comparaison, les clients de l'amas 5, 6 et 7 ont en moyenne un faible volume de courrier électronique.

1.4 Lien entre usage et abonnement

A la date où les mesures ont été effectuées, les clients de France Télécom avaient le choix entre plusieurs types de contrat :

Net 0 64 kbit/s remontant et 128 kbit/s descendant.

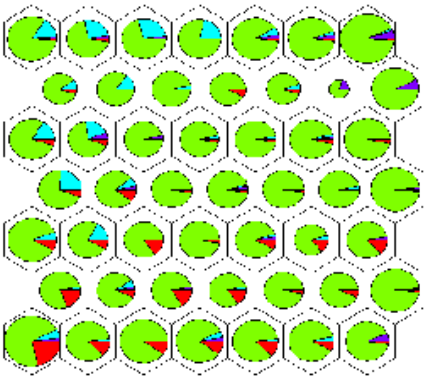
Net 1 128 kbit/s remontant et 512 kbit/s descendant.

Net 2 256 kbit/s remontant et 1024 kbit/s descendant.

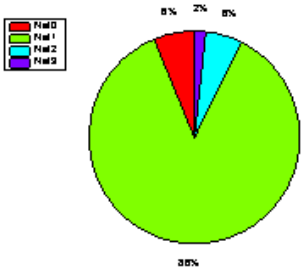
Net 3 128 kbit/s remontant et 1024 kbit/s descendant.

L'offre Net 2 a été conçue pour les utilisateurs professionnels.

En projetant l'information sur la carte des usages, on obtient la figure 1.6. Pour chaque nœud, nous avons indiqué la proportion de chaque abonnement. Il apparaît clairement que les types d'abonnement ne sont pas uniformément répartis sur la carte mais qu'il existe une forte corrélation entre activité et type d'abonnement. Les contrats de type Net 0 apparaissent principalement dans le coin sud ouest de la carte et correspondent à une faible activité. L'abonnement Net 3 se trouve dans quart nord-ouest et est relié à une forte activité ; on peut également vérifier qu'il existe peu d'abonnements de type Net 0 dans cette zone. Finalement, il existe une forte concentration de Net 2 dans le coin nord-ouest de la carte. Cette zone est liée aux deux amas caractérisés par un usage élevé de mail, ftp et Web. On peut supposer que les clients correspondants sont de type professionnel.



(a) Clients dans les amas



(b) Répartition de référence des contrats (Net0 en rouge, Net1 en vert, Net2 en cyan, Net3 en violet)

FIG. 1.6 – Projection des types de contrats sur la carte, la taille du cercle représente le log de la population.

Pour conclure cette section, on peut noter que les mêmes expérimentations ont été menées sur différents sites géographiquement éloignés. Les conclusions vis à vis de l'activité, ses relations avec

le type d'abonnement et l'apparition d'amas de comportements des clients sont similaires. Globalement, les clients ADSL se comportent de la même manière, quelle que soit leur situation géographique, Paris ou province.

1.5 Description détaillée de l'usage des applications p2p

Dans cette section, on décrit le comportement des utilisateurs d'applications p2p, qui engendrent la majeure partie du trafic. Par ailleurs, 99% des clients ont installé au moins un protocole p2p sur leur terminal.

1.5.1 Étude des volumes échangés

Dans un premier temps, nous nous intéressons à une description détaillée du trafic p2p. La figure 1.7 représente la distribution sur 6 minutes des volumes dans les sens montant et descendant sur le site de Fontenay en février 2003. Les périodes de temps avec des volumes montant et descendant nuls ne sont pas mentionnées sur cette figure ; ces périodes représentent 70 % des périodes du mois.

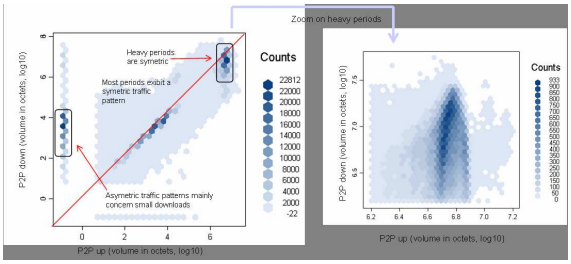


FIG. 1.7 – Volume du p2p dans les sens montant et descendant.

La plupart des périodes exhibe un usage symétrique, notamment pour les périodes de forte activité ("heavy periods"). Les configurations asymétriques concernent de très petits télé-chargements. En observant de plus près la distribution des périodes chargées, on constate que le débit remontant avoisine la limite de 128 kbit/s, qui est la limite des 88 % des clients sur le site considéré, seulement 6 % ayant un débit remontant limite de 256 Kbit/s. Les débits descendants sont plus éparpillés mais la plupart des valeurs tombe dans une fourchette de 128 kbit/s à 512 kbit/s. Ceci signifie que les clients ne sont pas très éloignés du débit descendant limite de 512 kbit/s, qui est la borne supérieure pour le débit descendant pour 92 % des clients. Seules quelques périodes atteignent 1024 kbit/s, i.e. la valeur limite pour Net 2 et Net 3.

Un autre point intéressant à observer réside dans la symétrie des échanges entre les utilisateurs. On peut supposer que les clients avec une forte activité en termes de p2p jouent à la fois le rôle de client et de serveur. Ceci s'observe par la quasi saturation des liens remontants.

1.5.2 Segmentation des clients suivant leur usage du p2p

On caractérise dans cette section les clients suivant leur activité en termes de p2p, évaluée en prenant en compte les volumes échangés dans les sens descendant et remontant. Un client est qualifié de nul, léger ou lourd suivant le volume de données échangées sur 6 minutes. Un client est "lourd" dans le sens remontant si son volume dépasse 15000 octets et dans le sens descendant si son volume est supérieur à 5000 octets.

Les clients sont alors segmentés avec une nouvelle carte auto-organisante. Le processus de création des amas conduit à la définition de 7 amas de clients caractérisés par des volumes descendant et remontant symétriques. Les amas se distinguent essentiellement par les niveaux des volumes échangés. Par exemple, 37 % des clients ont une activité très faible en termes de p2p avec 95 % de périodes nulles et 5 % de périodes légères. A l'autre extrême, 6.5 % des clients ont une activité élevée avec 70 % de périodes lourdes, 20 % de périodes nulles et 10 % de périodes légères. Seul un amas correspond à des volumes asymétriques avec des périodes nulles et légères. Les clients associés à ces périodes ont une activité globale faible ; ils représentent 15 % des clients et produisent environ 3 % du volume.

Seulement 14 % des clients peuvent être considérés comme actifs en termes de p2p ; ils produisent 75 % du volume mensuel. La plupart des client (73 %) sont inactifs et produisent 10 % du volume. Les distributions des volumes dans les amas sont du même ordre de grandeur dans les deux sens. Les clients actifs ont des abonnements Net 1 et Net 3 alors que Net 0 et Net 2 sont associés à une activité faible.

1.5.3 Évolution temporelle de l'usage des clients en termes de p2p

On procède de la même manière que dans les sections précédentes pour segmenter les clients d'un même site en novembre 2003. Nous obtenons 7 amas de clients. Le profil, la taille et les volumes échangés par les clients sont similaires à ceux de février 2003. La population des clients est globalement inchangée, 1724 clients se retrouvent dans les deux mois étudiés, 268 apparaissent et 568 disparaissent.

Les clients actifs sont plus nombreux en novembre qu'en février. 60 % des clients actifs en novembre étaient peu actifs en février. Ces clients se sont doute familiarisés avec l'usage de l'Internet et des protocoles p2p. Comme en Novembre, les nouveaux clients ont une activité moyenne en termes de p2p, on peut supposer que l'emploi de ces protocoles réclame une période d'adaptation.

1.6 Segmentation des clients suivant leur activité journalière

Le but de cette section est de mieux comprendre l'activité journalière des clients. Sur un mois, nous agrégeons les données en un ensemble de profils journaliers donnés par le volume horaire

(toutes applications confondues) pour chaque client et chaque jour. Un client est donc caractérisé par un vecteur multidimensionnel.

Pour la segmentation des clients, on procède en trois étapes :

1. On agrège les profils journaliers d'activité (tous clients confondus). On obtient ainsi des profils journaliers typiques, qui permettent de comprendre comment les clients utilisent leur accès ADSL sur une journée.
2. Ensuite, on considère les clients décrits par leur propre ensemble de profils journaliers. A chaque profil client, un jour typique est associé et on caractérise ensuite ce client par un profil décrivant la proportion des jours passés dans chaque jour typique.
3. Finalement, les clients décrits par leurs profils permettent de définir par agrégation des clients types. Cette seconde agrégation permet de lier le client à son activité journalière.

1.6.1 Résultats de l'agrégation

On considère le site de Fontenay en février 2003. Toutes les segmentations sont effectuées à l'aide de cartes auto-organisantes. La première étape conduit à la formation de 12 amas de profils journaliers types dont les comportement peuvent être résumés en jours actifs, jours avec une forte activité sur des périodes de temps limitées (debut ou fin de soirée ou midi), et jours de forte activité pendant des périodes de temps longues (journée, nuit par exemple). La dernière étape conduit à la formation de 11 amas de clients qui peuvent être caractérisés par la prépondérance d'un nombre limité de jours typiques.

La figure 1.8 illustre le comportement d'un client type. On affiche le profil moyen de l'amas (il s'agit d'une moyenne de logs), calculé à l'aide de la moyenne de tous les clients classés dans l'amas (en haut à gauche, en rouge). Pour comparaison, on donne également le profil moyen calculé sur toutes les observations (en bas à gauche, en bleu).

L'examen de la figure 1.8 montre que le client moyen associé à l'amas 5 est principalement actif sur le jour type 5 pour 23 % et le jour type 6 pour 28 % pour le mois. Les contributions sur les autres jours types sont inférieures à 19 % et sont proches de la moyenne globale, sauf pour le jour 8 avec une contribution bien plus petite que la valeur moyenne.

Le jour type n°8 correspond au jour inactif. Les profils moyens des jours types n° 5 et 6 sont donnés par la figure 1.8. Sur la droite est indiqué en rouge le profil horaire du jour type et en bleu le profil moyen global. Ces jours types sont caractérisés par une activité journalière élevée de 9h à 24h et une activité faible la nuit.

L'analyse ci-dessus permet d'inspecter en détail les cycles journaliers des clients dans un segment donné. Le segment étudié recouvre 9% des clients, qui ont une activité élevée pendant les heures de bureau et le soir et qui sont peu actifs la nuit pendant au moins la moitié du mois, ils ont également peu de jours de faible activité. Tous les autres segments admettent de la même manière une analyse qualitative. Par exemple, nous avons identifié un segment de clients qui sont très actifs pendant toute la journée, 7 jours sur 10 (8,5 % des clients) et un autre segment de clients sans activité 8 jours sur 10 (12 % des clients).

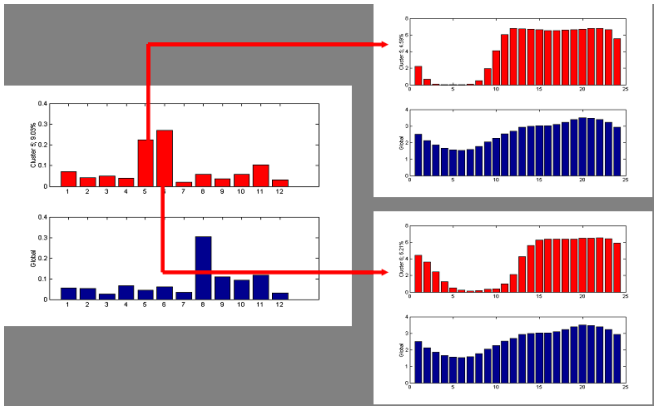


FIG. 1.8 – Profil d'un amas de clients (en haut) et profil moyen (en bas) et profils associés à un jour type à droite.

1.6.2 Relations entre amas et dynamique de trafic

La figure 1.9 représente l'évolution temporelle du trafic sur un mois avec la contribution de chaque segment. 70 % du trafic sont dus à deux segments (n° 8 et 9) qui ensemble regroupent 17 % des clients très actifs (avec une activité très forte pendant tout le jour, de 4 à 7 jours sur 10). Cependant, le cycle journalier ne peut pas être expliqué en ne prenant en compte que ces deux segments. En réalité, c'est le segment 5 (décrit ci-dessus) qui joue le rôle prépondérant. En effet, ce segment regroupe les clients avec une forte activité pendant les jours et les heures ouvrables. Il engendre seulement 15 % du volume mais donne la courbe générale de la dynamique du trafic.

Les groupes obtenus après segmentation des clients selon leur usage des applications et ceux obtenus d'après leur activité journalière sont consistants. En comparant les deux analyses, il est possible de connaître précisément le comportement des clients. Par exemple, les clients avec une forte activité journalière entre 9h et 24h sont aussi les forts utilisateurs de toutes les applications.

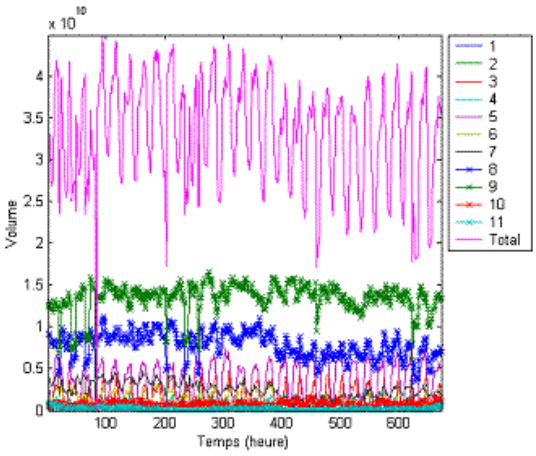


FIG. 1.9 – Évolution du trafic sur un mois et contribution des différents amas.

Chapitre 2

Analyse de surcharges en cas de panne

2.1 Introduction

Cette étude est basée sur des captures TCP sur un lien ADSL, que nous utilisons pour établir des caractéristiques générales des systèmes *Peer-to-Peer* (P2P). Nous ne prenons en compte que le trafic ADSL, ceci est important car les utilisateurs de cet accès sont très impliqués dans les échanges P2P. En effet, un accès forfaitaire ilimité en temps d'utilisation et en quantité est fourni à ces utilisateurs. Ainsi, comme nous allons le voir, le trafic TCP des applications P2P représente plus de 60% du trafic ADSL total.

L'originalité de nos mesures réside dans le fait que, premièrement, nous analysons tous les flux TCP d'un point de concentration régional sur ADSL, deuxièmement, nous observons uniquement le trafic ADSL (sans prendre en compte les modems 56k) ce qui est plus représentatif de l'utilisation du P2P, et troisièmement, les données collectées sont représentatives d'une utilisation générale de l'ADSL et non resteint à une classe spécifique d'utilisateurs ou de machines (*e.g.* une université ou un réseau privé). D'autre part, nos données regroupent plusieurs miliers d'utilisateurs.

Nous différencions systématiquement selon les utilisateurs P2P grâce à un identifiant ADSL unique. Comme remarqué dans [11], une analyse basée sur les adresses IP peut avoir une influence négative sur l'interprétation des traces à cause des NATs (*Network Address Translator*) ou des adresses IP dynamiques, par exemple. En effet, nous avons noté une différence significative entre les graphes basés sur les adresses IP et ceux basés sur les clients ADSL.

Dans cette étude, nous comparons quatre réseaux P2P (appelés aussi réseaux de pairs) populaires entre eux, à savoir : eDonkey [6], BitTorrent [5], FastTrack et WinMX [7].

La persistance de nos mesures nous permet de montrer que la popularité des réseaux P2P est très volatile au cours du temps.

L'analyse au niveau des flux TCP nous permet de décrire les propriétés volumétriques, les durées des connexions, les fluctuations du trafic à travers le temps, la connectivité et la localisation géo-

graphique des *peers*. Nous approximons aussi quelques distributions expérimentales avec des lois statistiques classiques. L'analyse au niveau paquet, quant à elle, nous offre la possibilité d'identifier clairement les débuts et fins de connexions conduisant à des remarques intéressantes. Un résultat important est que 40% des connexions ne sont que des tentatives de connexions et que cela concerne 30% des *peers*.

2.2 Méthode de capture et description générale du P2P

2.2.1 Détails des mesures

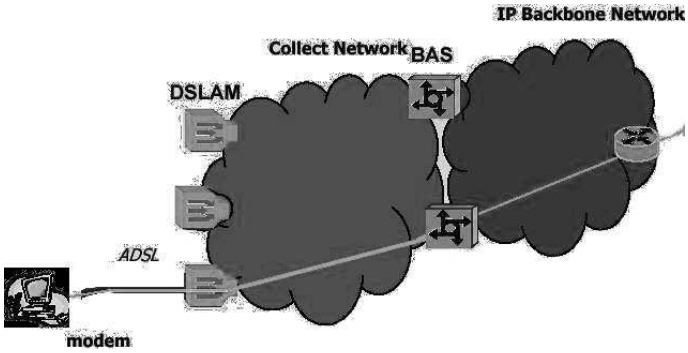


FIG. 2.1 – Architechture de l'ADSL

Tout d'abord, nous détaillons notre protocole d'expérimentation. Comme montré sur la Figure 2.1, le BAS[1] collecte le trafic provenant des DSLAMs[2] avant de le transmettre à travers le POP[3] au réseau backbone IP de France Télécom.

Notre sonde est localisée entre un BAS et le réseau backbone IP. Nous soulignons, ici, que nous capturons tous les paquets TCP sans échantillonnage ou perte. Notre analyse est réalisée sur le trafic de jours de semaine et week-end en septembre 2004, et nous comparons ces résultats avec les données enregistrées une année auparavant.

Nous désignerons par *peers* ou utilisateurs *locaux* les machines ADSL connectées directement au BAS observé, et par *peers distants* le reste des machines. Le trafic *montant* représente les transferts des *peers* locaux vers le backbone, alors que le trafic *descendant* les transferts du backbone vers les *peers* locaux.

[1] Broadband Access Server
[2] Digital Subscriber Line Access Multiplexer
[3] Point-Of-Presence

TAB. 2.1 – Répartition du trafic des protocoles P2P sur le trafic P2P total

Protocole	Juin 2003		Septembre 2004	
	Volume	# Connexions	Volume	# Connexions
eDonkey	84%	96%	91%	93%
BitTorrent	0.8%	0.009%	6%	2.7%
Gnutella	0.8%	0.9%	1%	3.6%
WinMX	1.3%	0.06%	1%	0.08%
FastTrack	12%	1.8%	1%	0.01%
autres protocoles	1.1%	1.2%	0%	0.6%

2.2.2 Description générale du P2P

Parmi nos données, environ 60% du trafic est transporté sur les ports P2P *officiels* en septembre 2004. Celà représente une petite baisse par rapport à la proportion du trafic sur les ports P2P en juin 2003 qui était d'environ 65%.

Dans le Tableau 2.1, nous indiquons la proportion des principaux protocoles P2P par rapport au trafic P2P total. En septembre 2004, eDonkey est de loin le plus populaire en termes de volume, BitTorrent est le deuxième plus populaire et tous les autres protocoles génèrent un volume quasi négligeable comparé à eDonkey.

La popularité de chaque système P2P de partage de fichiers est très variable selon le lieu et le temps. Selon [15] en octobre 2003, en Europe eDonkey était extrèmement dominant en volume, alors qu'aux Etats-Unis, FastTrack était le plus populaire suivi par WinMX. L'évolution dans le temps sur nos données montre que FastTrack a perdu sa popularité en France en un peu plus d'un an.

Dans le reste de cet article, nous ne considèrerons que les protocoles eDonkey, BitTorrent, FastTrack and WinMX, à cause de leur popularité et de la diversité de leur fonctionnement.

Comme mentionné par Karagiannis *et al.* dans [13] et [14], une partie du trafic P2P peut utiliser des ports non-standard de telle sorte que nous ne voyons pas tout le trafic P2P à partir de notre analyse par ports. Dans [16], Sen *et al.* montrent qu'une identification du trafic P2P en utilisant les signatures applicatives pourrait conduire à tripler le volume comptabilisé commme P2P. Mais, nous pouvons remarquer que d'une part, seulement Kazaa (utilisant le réseau FastTrack) a une grande part de trafic caché, et d'autre part, les *peers* eDonkey et BitTorrent utilisent principalement les ports standards. En effet, dans le réseau FastTrack, il n'y a pas de limitation basée sur les ports TCP utilisées, et certains utilisateurs (en fait la plupart) changent leur numéro de port. Mais dans le réseau eDonkey, les *peers* utilisant leur application avec un port non standard reçoivent une *Low ID* quand ils se connectent au serveur eDonkey, alors que les autres obtiennent une *High ID*. Les *High ID peers* n'ont pas de restrictions sur les téléchargements alors que les *Low ID peers* ne peuvent télécharger que des *High ID peers*, de telle sorte que les *peers* eDonkey essayent de ne pas changer le numéro de port de leur application.

Comme nous l'avons vu, en France, la principale part du trafic P2P repose sur les réseaux eDonkey ou BitTorrent, et notre analyse identifiant les applications P2P par leur numéro de port est donc valide dans ce cas.

2.3 Caractéristiques du trafic P2P

2.3.1 Trafic de signalisation

Le trafic P2P peut être séparé en deux parties :
- le trafic généré strictement pour le téléchargement de fichiers,
- le trafic généré pour maintenir le réseau et effectuer les requêtes, que nous appellerons *trafic de signalisation*.

Nous séparons ces deux types de trafic grâce à un seuil sur le volume transféré par chaque connexion. Nous avons choisit un seuil de 20 ko à partir de l'analyse de nos données.

Comme observé dans [17] et [10], la grande majorité des connexions (environ 90 %) est composée de connexions de signalisation, alors qu'elles ne représentent qu'une petite proportion du volume transféré (eDonkey a la plus grande proportion de trafic de signalisation avec 6%).

2.3.2 Volumes montants vis-à-vis des volumes descendants

Comparaison basée sur les utilisateurs

Dans nos données, le total du volume descendant est plus grand que le volume montant pour chaque protocole. Cela signifie que les *peers* locaux (*i.e.* des miliers d'utilisateurs) téléchargent plus vers leur machine qu'ils ne sont téléchargés par les autres utilisateurssur notre point d'observation.

En analysant les volumes descendant et montant par utilisateur, nous pouvons distinguer deux types d'utilisateurs :
- les *peers* générant de faibles volumes, téléchargent des fichiers, mais ont très peu de trafic montant, ils peuvent donc avoir un ratio volume montant sur volume descnedant jusqu'à 100 ;
- les *peers* générant des volumes importants cont des volumes descendants et montants comparables avec un ratio volume montant sur volume descendant d'environ 1.3.

Cette dichotomie est le cas des *peers* eDonkey, mais ils représentent la vaste majorité des *peers*, et les autres protocoles ont des tendances similaires.

Les utilisateurs de la première classe partagent peu de fichiers, ou se déconnectent après un téléchargement. Les utilisateurs de la seconde classe, quant à eux, restent connectés pour de longues périodes de sorte à obtenir de grands volumes descendants (beaucoup de téléchargements), et ainsi partagent leurs fichiers (au moins ceux qu'ils sont en train de télécharger). Nous pouvons voir dans la Figure 2.2 un nuage de points représentant pour chaque utilisateur d'eDonkey son volume descendant en fonction de son volume montant. Sur la figure, nous pouvons identifier les utilisateurs générant des volumes importants (dans le coin en haut à droite), ceux générant de faibles volumes (au dessus de la diagonle, au milieu), et un certain nombre d'utilisateurs qui n'ont pas de volume descendant alors qu'ils ont du volume montant (à expliquer dans la Section 2.3.6).

En rappelant que moins de 10% des utilisateurs contribuent à la plupart (98%) du trafic (voir aussi [17]), nous pouvons voir que le caractère non-coopératif des utilisateurs générant de faibles volumes ne perturbe pas l'équilibre des réseaux de pairs. Le ratio volume montant sur volume descendant pour tout le trafic P2P est d'environ 1.3 sur notre échantillon d'utilisateurs.

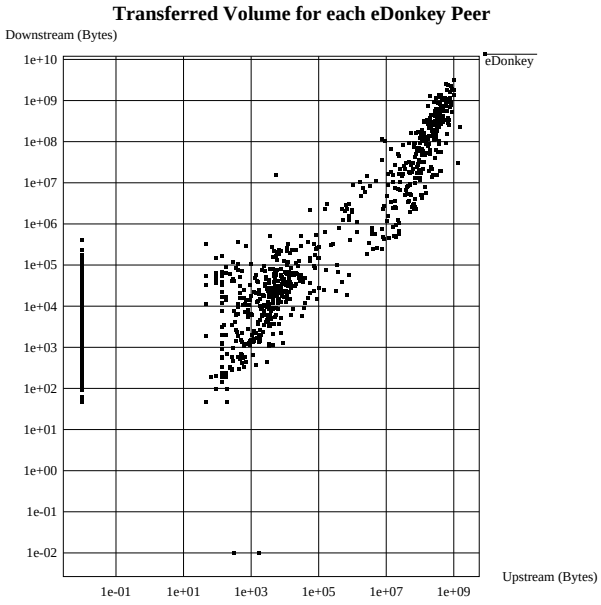


FIG. 2.2 – Volume descendant en fonction du volume montant pour chaque *peer* eDonkey

Au cours de notre analyse, nous identifions environ 20% des *peers* comme étant des *free-riders* (*i.e.* des *peers* qui ne partagent pas de fichiers). Dans [8] et [9], le nombre de *free-riders* dans le réseau Gnutella est approximé à 70% et 42% en 2000 et 2001 respectivement. Les mécanismes comme le *multi-part download* (*i.e.* téléchargement par parties, qui est maintenant utilisé par la plupart des applications P2P) permet aux *peers* de partager les morceaux de fichiers qu'ils ont déjà téléchargés. Ceci explique la réducction du nombre de *peers* qui ne partagent pas de données par rapport aux études précédentes.

Comparaison basée sur les connexions

La taille des connexions indique que la plupart des connexions ne transportent qu'une très faible proportion du volume de chaque réseau P2P. En effet, les connexions transferant moins de 100 ko génèrent environ 8% du volume total transféré, alors qu'elles représentent plus de 90% des connexions.

Nous expliquons ce nombre écrasant de petites connexions par les raisons suivantes :
– la signalisation génère beaucoup de petits transferts ;
– beaucoup de transferts sont interrompus ;
– beaucoup de *peers* tentent de se connecter à des *peers* déconnectés (voir Section 2.3.6).

Pour BitTorrent, la distribution des tailles de transferts est différente. En effet, BitTorrent génère une plus grande proportion de *grands* transferts.

Nous avons approximé la distribution expérimentale des tailles de transferts par des lois statistiques classiques minimisant la distance de Kolmogorov-Smirnov (K-S), et nous trouvons que :
– eDonkey peut être approximé par une distribution lognormale ;

– FastTrack par une loi lognormale, mais la queue de la distribution (les *grands* transferts) correspond mieux à une distribution de Pareto ;
– BitTorrent par une loi de Weibull.

Pour conclure cette section, nous pouvons mentionner que non seulement le volume médian par connexion est très faible (moins de 1 ko), mais le volume médian par utilisateur l'est aussi (10 ko). La grande proportion de trafic de signalisation induit un volume moyen par connexion de 10 ko. Au contraire, à cause des utilisateurs générant d'importants volumes, le volume moyen par utilisateur se porte à 70 Mo.

2.3.3 Durées des connexions

Les durées des connexions sont très longues au vu de leur taille. En effet, plus de 85% des connexions restent ouvertes plus de 10 secondes alors que plus de 85% des connexions comportent moins de 20 ko. Ceci indique la présence de périodes d'inactivité (*idle times*) dans les connexions.

Les connexions eDonkey et FastTrack sont plus courtes que celles de BitTorrent, qui voit aussi de plus grands débits.

La différence entre BitTorrent d'une part et eDonkey et FastTrack de l'autre est confirmée par l'interpolation des distribution des durées de connexions. Nous trouvons à nouveau qu'une loi lognormale approxime mieux les distributions de eDonkey et FastTrack, alors que le test K-S ne donne pas de résultat satisfaisant pour les connexions BitTorrent.

2.3.4 Fluctuations quotidiennes du trafic

Le volume de trafic eDonkey transféré par heure ne varie pas beaucoup au cours de la journée en semaine. Une petite baisse (environ 20%) est observée de minuit à 9h. Ceci correspond à une baisse du nombre de *peers* connectés.

Cette observation contredit le *time-of-day effect* observé dans [17], [11] et [12], nous avançons les raisons suivantes pour expliquer celà :
– la plupart du trafic est générée par une petite proportion des utilisateurs (10%), et ces utilisateurs sont connectés de façon quasi-permanente au réseau P2P ;
– les *peers* téléchargent des très grands fichiers qui nécéssitent de rester connecté longtemps au réseau de pair.

Pour FastTrack et WinMX, nous observons des fluctuations au cours de la journée : les utilisateurs des ces réseaux P2P se connectent principalement entre midi et minuit les jours de semaine, et du samedi au dimanche midi les week-ends. Nous expliquons ce phénomène comme ceci :
– les après-midi et soirées sont les périodes de plus grande activité sur l'ADSL (en semaine) ;
– les utilisateurs de ces réseaux téléchargent des fichiers plus courts (en général), et donc restent connectés moins longtemps.

Pour BitTorrent, le comportement est différent. Les connexions sont plus longues, mais le récepteur voit des périodes actives ou complètement inactives au cours d'un transfert, donc les fluctuations sont un peu désordonnées. Mais de manière générale, le trafic transféré avec BitTorrent est assez stable au cours de la journée du fait de ses nombreux utilisateurs.

Pour les week-end, le trafic suit une évolution différente : toute la journée du samedi et le dimanche matin sont des fortes périodes d'activité, mais cela s'arrête le dimanche midi !

2.3.5 Distribution géographique des *peers*

Nous localisons les destinations des transferts ainsi que la longueur des chemins pour les atteindre. Cette information est utilisée pour déterminer les resources réseau consommées.

La localisation des clients montre que la destination principale des transferts est la France, suivie des Etats-Unis pour eDonkey et BitTorrent. Sur les réseaux FastTrack et WinMX, les Etats-Unis sont la principale source et destination des transferts.

Nous devons mentionner que ces distributions sont très sensibles à l'activité du jour considéré, c'est pourquoi nous les avons établies sur une semaine.

2.3.6 Fins de connexions

Dans le but d'observer les fins de connexions, nous identifions les connexions avec une terminaison TCP normale, qui se caractérise par une double procédure de handshake, et pour les autres connexions, nous notons le TCP-Flag du dernier paquet.

Les quatre protocoles ont des tendances similaires : seulement peu de connexions se terminent normalement (environ 15% des connexions). Un client fermant son application P2P se déconnecte du réseau en envoyant un paquet RESET, ceci représente environ 15% des connexions. Nous observons aussi entre 30 et 40% des connexions qui terminent anormalement, par exemple avec un PUSH.

Mais la principale remarque, ici, est qu'environ 40% des connexions ne sont que des tentatives de connexions (le dernier paquet étant un paquet SYN). Les *peers* impliqués dans ces connexions (environ 30% des *peers*) reçoivent une demande de connexion alors qu'ils ne sont plus connectés au réseau. Nous pouvons observer ceci dans la Figure 2.2 avec les utilisateurs qui ont du trafic descendant alors qu'ils n'ont pas de trafic montant. Nous expliquons ceci par le délai dans la transmission de l'information sur la disponibilité des *peers* dans le réseau.

2.4 Connectivité des *peers*

2.4.1 Connectivité des *peers* locaux

Nous étudions à présent le nombre de machines distantes contactés par un *peer* local, *i.e.* la connectivité d'un *peer* local.

Premièrement, la connectivité des *peers* eDonkey est la plus grande dans nos données. En effet, ce protocole génère beaucoup de connexions entre les *peers*. Seulement 5% des *peers* eDonkey ne contactent qu'un seul autre *peer*, 70% des *peers* locaux se connectent à plus de 10 autres *peer*. Certains *peers* vont jusqu'à établir des connexions avec plus de 100 000 autres *peers*, ce qui montre que nous avons un petit nombre de serveurs d 'indexation eDonkey dans nos *peers* locaux. Nous notons

aussi que les *peers* locaux eDonkey communiquent avec plus de 10 autres *peers* simultanément 85% du temps.

Pour BitTorrent, le trafic de téléchargement présente une plus grande connectivité que le trafic de signalisation : ceci est dû à la gestion de l'information, qui est réalisée à partir d'un *tracker* unique pour chaque fichier alors que le téléchargement est réalisé depuis plusieurs sources.

2.4.2 Connectivité des *peers* distants

Maintenant, nous examinons la connectivité des *peers* distants, *i.e.* le nombre d'utilisateurs locaux contactés pour chaque *peer* distant. Ici, nous ne distinguons les *peers* distants que par leur adresse IP. Ces résultats sont établis à partir d'un pool de plus d'un milion de *peers* distants, principalement des *peers* eDonkey.

Seulement peu de *peers* locaux sont connectés au même *peer* distant. Il n'y a pas de point d'accumulation sur une adresse IP distante : le trafic et les requêtes sont bien distribuées sur les *peers* distants.

Les *peers* eDonkey ont la plus dense connectivité avec 25% des *peers* distants qui contactent plus de trois *peers* locaux. Pour les trois autres protocoles, environ 90% des *peers* distants ne contactent qu'une seule fois les *peers* locaux.

2.5 Conclusion

Dans cet article, nous comparons les performances et les caractéristiques de quatre applications P2P. Notre méthode de mesure nous permet d'analyser profondément un ensemble complet de captures de trafic provenant de tous les utilisateurs d'un point de concentration régional sur ADSL.

Notre étude indique que, premièrement, même pour le trafic P2P, la plupart des connexions sont très courtes et représentent un faible volume, et deuxièmement que très peu d'utilisateurs contribuent à la majeure partie du volume transféré. Les deux types de *peers* impliqués dans les réseaux P2P (*i.e.* ceux générant d'importants volumes et les autres) influencent fortement le fonctionnement des systèmes P2P d'échange de fichiers. Nous dévoilons que même au niveau d'un point de concentration régional les utilisateurs transfèrent plus dans le sens descendant que montant. Nous trouvons aussi que les *peers* locaux ont tendance à contacter plusieurs *peers* distants. En se concentrant sur les paquets envoyés en fin de connexion TCP, nous détectons que les tentatives d'établissement de connexion représentent énormément de connexions et concernent beaucoup d'utilisateurs.

La persistance de nos mesures nous a permis de voir un changement dans la popularité des applications P2P (FastTrack étant dépassé par BitTorrent) et quelques changements dans la localisation des sources à travers une année.

Chapitre 3

Analyse du trafic *Peer-to-Peer* sur l'ADSL

3.1 Introduction

Nous étudions dans ce chapitre l'évolution du trafic en cas de surcharges à partir de catpures de trafic effectuées sur un réseau de collecte ADSL grand public. Il s'agit d'un réseau arborescent, dont les liens sont sécurisés, i.e. dédoublés, à partir d'un certain niveau de concentration. Le trafic se réparti équitablement sur chacun des deux liens, et en cas de panne, tout le trafic se reporte sur le lien restant.

Ce réseau n'observe habituellement pas de congestion. Une surcharge a tout de même été observée lors de la panne d'un lien, tout le trafic s'est reporté sur le lien restant qui a donc vu sa charge augmenter. La charge initiale de chaque lien étant en temps normal supérieure à 50%, pour des raisons de coût, la panne a donc occasionné une surcharge sur le lien restant.

La panne observée a eu lieu entre le lien sur lequel étaient effectuées les captures et le coeur du réseau, donc sur un lien où le trafic que nous observons est multiplexé avec d'autres flux.

La figure 3.1 est issue de l'application Otarie surveillant le trafic écoulé sur le réseau de collecte ADSL. Cette figure représente le débit cumulé des applications, mesuré en moyenne toutes les 6 mn. Nous observons que le débit global chute juste avant 19h, donc probablement entre 18h48 et 18h54. Le débit écoulé est passé de 45 Mbits/s à 35 Mbits/s lors de la panne. La période de bas débit dure jusque vers 23h, donc bien au-delà des captures disponibles (de 18h à 21h).

Grenouille [3] est une association d'utilisateurs de l'ADSL comparant, entre autres, les performances des accès interndees divers founisseurs d'accès internet (FAI) et alimenté par les clients des différents FAI. La figure 3.2 représente l'historique du débit moyen des clients Wanadoo de Lille pour la même journée, rapporté par Grenouille. Les mesures sont apparemment effectuées toutes les 1/2h. Le débit observé est de 50 Ko/s (400 kbits/s) par client avant 19h, et de 10 Ko/s (80 kbits/s) par client de 19h à 23h, c'est-à-dire pendant la panne. Nous remarquons que le débit, tel qu'il est mesuré par Grenouille, est relativement stable, que ce soit avant, ou pendant la panne.

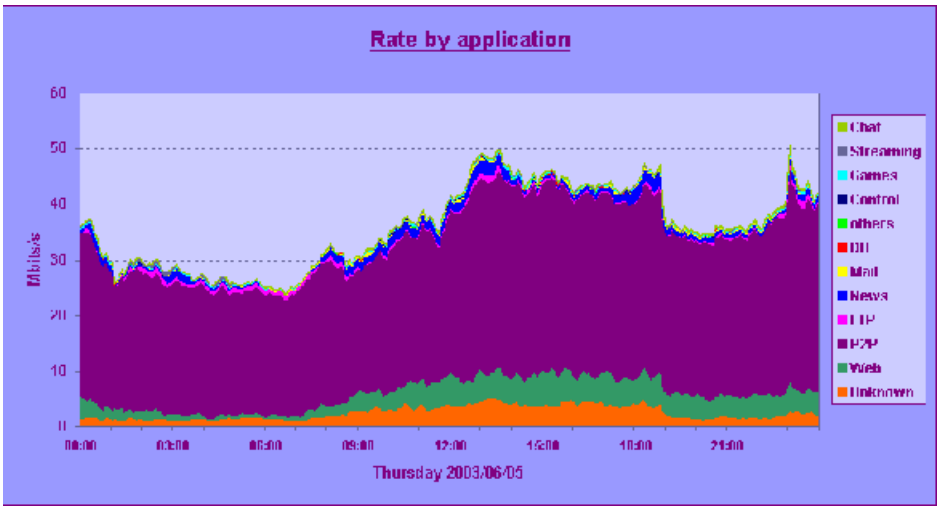


FIG. 3.1 – Débit par application (sonde France Télécom Otarie)

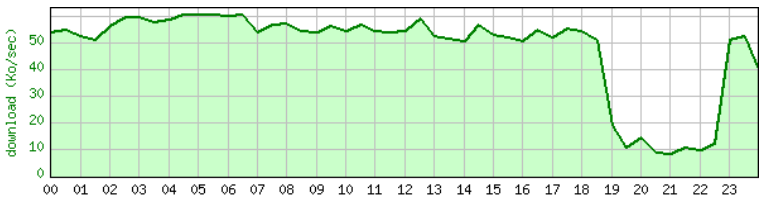


FIG. 3.2 – Débit par client Wanadoo

Il est par ailleurs à noter que les clients ADSL qui n'hésitent pas à se plaindre sur les forums de discussion en cas de problèmes de performance, ne se sont pas manifestés à l'occasion de cette panne.

Lors de la panne, une sonde France Télécom Otarie effectuait une capture de trafic par paquet en continu. Les informations enregistrées par cette sonde pour chaque paquet sont : la date, le VP associé au DSLAM et le VC associé au client, le protocole d'encapsulation, le protocole de transport, les adresses IP et les ports source et destination, la taille de l'en-tête et l'offset, la taille du paquet et les flags TCP. Les numéros de séquence et d'acquittement des paquets TCP n'ont cependant pas été conservé, ce qui ne nous a empêcher de pouvoir estimer le taux de perte et le RTT perçus par les connexions TCP et donc par les clients.

Trois heures de capture par paquet ont été conservés, de 18 h à 21h, dans chaque sens. Par la suite, le sens montant sera noté (1) et le sens descendant (0).

Dans l'analyse de trafic présentée au chapitre suivant nous avons étudié trois points qui pouvaient être impactés par la congestion : la charge du réseau (cf. §3.2.1), le comportement des clients (cf. §3.2.2), et les performances perçues par les clients (cf. §3.2.3). Afin d'étudier l'évolution de ces trois aspects au cours de la congestion, nous avons découpé les captures en période de 10 minutes.

3.2 Analyse de trafic

3.2.1 Charge du réseau

Nous avons représenté sur la figure 3.3 les trafics offerts et écoulés dans chaque sens, respectivement en noir pour le sens descendant, et en rouge pour le sens montant. Le trafic offert sur une période est défini comme la somme des tailles des connexions arrivées sur cette période.

La courbe noire (en trait plein) montre que c'est le lien descendant qui a été touché par la panne. Nous retrouvons la courbe du trafic cumulé de la figure 3.1 sur la période de capture, de 18h à 21h. Nous avons les mesures d'un seul des DSLAM d'un des BAS de la plaque ADSL, ce qui explique les petites variations du trafic pendant la période de surcharge. On ne peut donc pas vraiment déduire des captures la charge totale observée par les flux sur le lien restant. Tout au plus peut-on estimer, en supposant que tous les DSLAM de la plaque observent la même variation, qu la surcharge doit être de l'ordre de 50/35 1,4. Il est à noter que la sonde est donc placée après le goulet, donc elle ne voit pas les paquets retransmis.

Nous observons également une hausse légère mais continue sur toute la période de capture du trafic écoulé dans le sens montant. Nous n'avons pas trouvé d'interprétation claire de ce phénomène. Il s'agit peut-être simplement de l'évolution habituelle en soirée du trafic.

Nous avons représenté sur la figure 3.4 l'historique des nombres de connexions actives dans chaque sens, mesurés à chaque minute. Les dates sont indiquées en secondes depuis 18h, la panne a lieu entre 18h48 et 18h54 donc vers la date de 3000s.

La principale remarque est que, contrairement aux modèles [2], ces grandeurs évoluent peu malgré la surcharge estimée, de l'ordre de 10% sur les trois heures de mesure. En particulier, nous n'observons pas la croissance linéaire prévue par le modèle. Nous avons considéré deux hypothèses pour expliquer cette stabilité du trafic global : soit les clients, percevant la surcharge, ont modifié

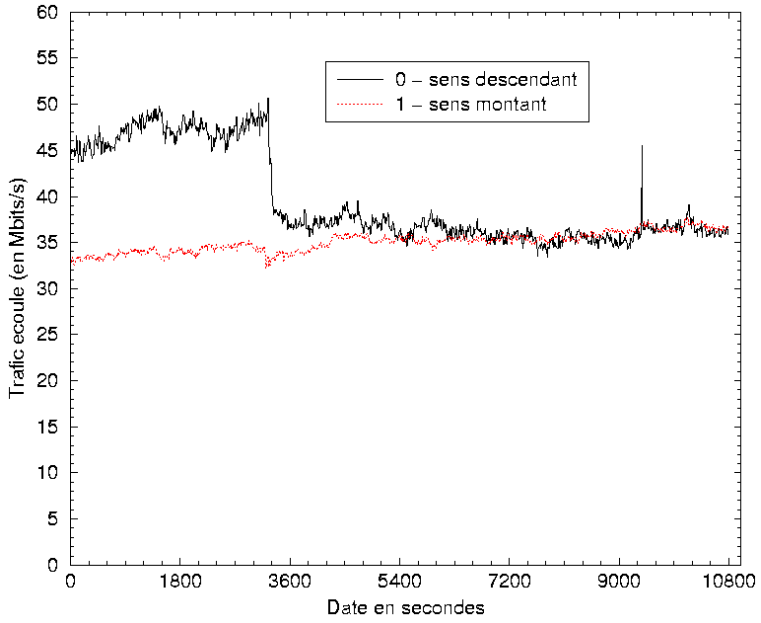


FIG. 3.3 – Trafic écoulé dans chaque sens

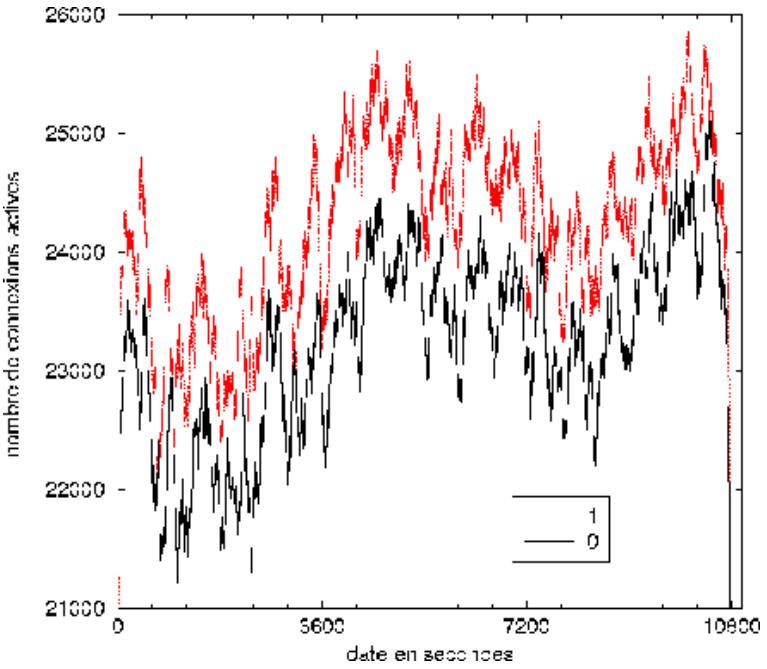


FIG. 3.4 – Historique du nombre de connexions actives dans chaque sens

leur comportement, générant moins de connexions, ou des connexions plus petites, soit les débits individuels des transferts ont été réduits.

3.2.2 Comportement des clients

Nous étudions dans cette section l'évolution du comportement des clients au cours de la panne, si par impatience, ils renouvellent des transferts, ou s'ils abandonnent certains transferts. Nous allons essayer de détecter ces deux cas en considérant l'évolution au cors de la panne de la répartition des transferts en fonction de leur taille [4].

Nous avons représenté sur les figures 3.5 et 3.6 la densité de probabilité des tailles de connexions par période de 10 mn. L'objectif étant de détecter des arrêts prématurés, abandons de transferts, ou des renouvellement de transferts, ce qui dans les deux cas se traduirait par une augmentation de la part des petits transferts par rapport aux longs. a) b)

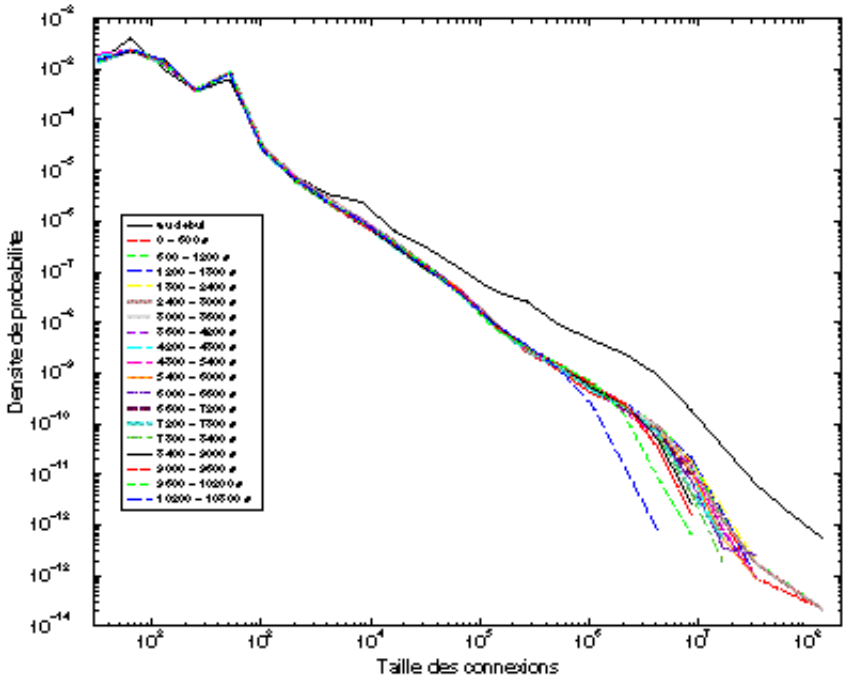


FIG. 3.5 – Densité de probabilité des tailles de connexions pour chaque période de 10 mn

Nous observons que tel n'est pas le cas, mis à part 2 biais de mesure près concernant les connexions présentes au début de la capture, et les courbes des deux dernières périodes. En effet, d'une part les connexions présentes en début de capture ont déjà été partiellement servies, donc les

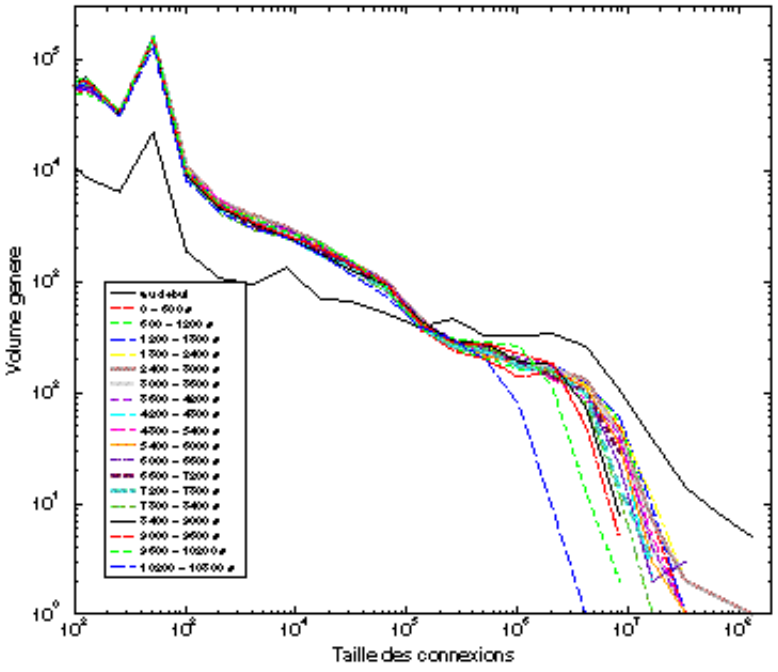


FIG. 3.6 – Volume généré en fonction de la taille pour chaque période de 10 mn

plus gros transferts sont sur-représentés du fait de la distribution en loi de Pareto des tailles. D'autre part, pour les deux dernières périodes, les plus grosses connexions n'ont pu être servies, elle l'ont seulement été partiellement, d'où un décalage vers la gauche des distributions. A ces deux biais près, les courbes sont confondues ; donc les clients n'ont pas modifié leur comportement malgré la panne.

3.2.3 Performances des connexions

Le dernier aspect que nous avons étudié à partir des captures de trafic est les performances perçues. Comme les numéros de séquence et d'acquittement n'étaient pas disponibles, nous n'avons pas pu estimer précisément les taux de perte et les délais de transmission. Nous allons donc considérer le débit des connexions TCP et leur durée de service.

Débit en fonction de la taille des transferts

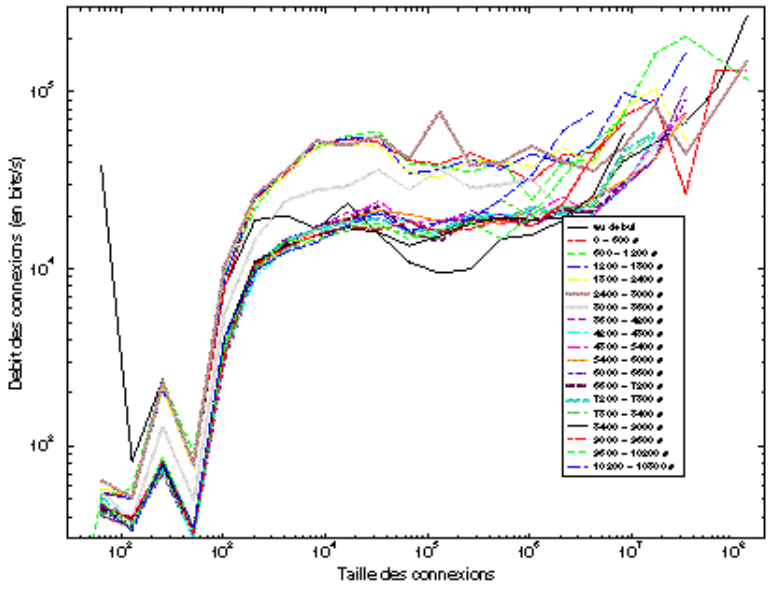


FIG. 3.7 – Débit des connexions ayant débuté dans la période en fonction de la taille

Nous avons représenté sur la figure 3.7 le débit des transferts en fonction de leur taille, pour les périodes de 10 minutes successives. Nous reconnaissons tout d'abord pour les petits transferts, inférieurs à quelques kilo-octets, l'influence du "Slow Start", mécanisme de TCP limitant le débit des connexions à leur démarrage. Nous observons également un biais de mesure du à la dispersion importante des débits, d'autant plus marqué que les connexions sont grosses et que la courbe cor-

respond aux dernières périodes, car dans ce cas, les connexions qui ont pu émettre des volumes importants sur la durée de la capture sont justement celles qui ont obtenu un débit élevé.

Finalement, pour les tailles de connexions intermédiaires, entre quelques kilo-octets et quelques méga-octets, nous distinguons deux groupes de courbes, correspondant justement aux deux phases de la période étudiée :

- débits élevés avant 2400 s, de l'ordre de 50 kbits/s par connexion ;
- débits plus faibles après 3000 s , de l'ordre de 20 kbits/s par connexion ;
- la courbe associée au début de la panne (2400 s - 3000 s) étant intermédiaire entre ces deux groupes.

Nous notons que ces débits des connexions TCP mesurés à partir des captures sont naturellement différents, et inférieurs aux débit des accès Netissimo mesurés par Grenouille (cf. figure 3.2). Ce qui est plus surprenant est que les rapports entre les débits avant et pendant la panne soient différents, de l'ordre de 2,5 (de 50 à 20 kbits/s) , alors que Grenouille donnait pour les débits par client sur la plaque ADSL un rapport de 5 (de 50 Ko/s à 10 Ko/s), alors que les nombres de connexions restent semblables.

Dispersion des durées de service

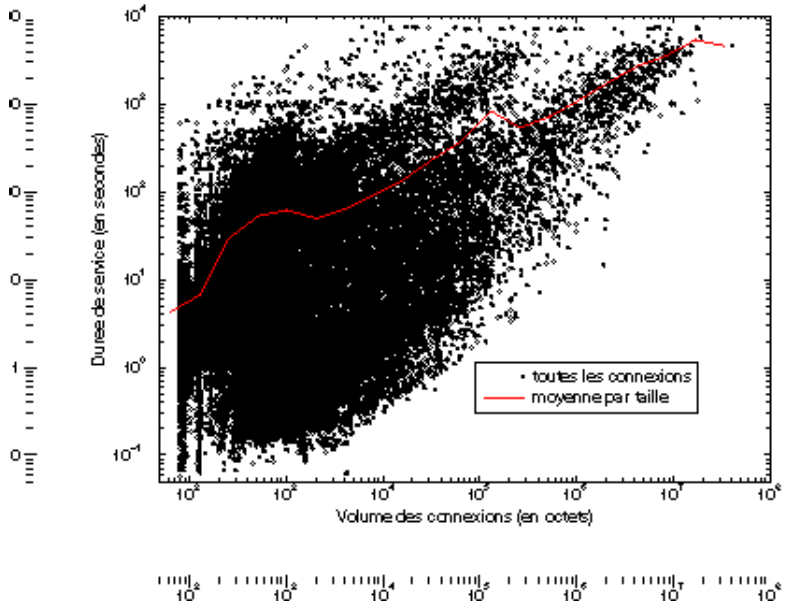


FIG. 3.8 – Durée de service des connexions ayant débuté entre 3000s et 3600s en fonction de la taille

Nous avons représenté sur la figure 3.8 les durées de service des connexions arrivées entre 3600s et 4200s. Chaque point représente une connexion avec en abscisse la taille et en ordonnée la durée de service. La courbe rouge (traits long-court alternés) représente la moyenne des durées de service pour chaque taille. Vue la dispersion élevée, nous avons également représenté la médiane (tirets en vert), représentant la durée de service pour 50% des transferts, et le 9ème décile (traits longs en bleu) représentant la durée maximale pour 90

La très grande dispersion des durées de service pour une taille donnée sur la figure 3.8 est remarquable, avec un rapport de près de 10000 pour une même taille, entre les connexions servies le plus rapidement et celles servies le plus lentement.

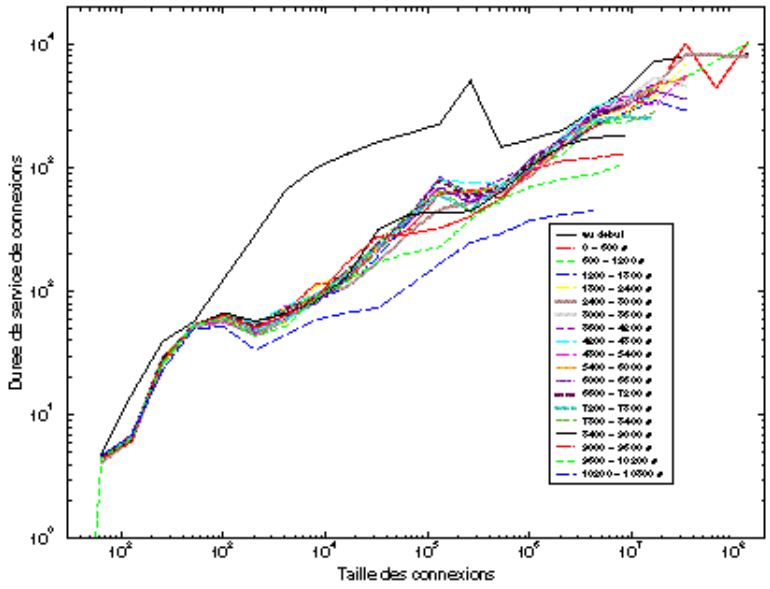


FIG. 3.9 – Durée de service des connexions ayant débuté dans la période en fonction de la taille

Les courbes de la figure 3.9 montrent l'évolution de la durée moyenne de service au cours de la panne. A part la première et la dernière courbes, biaisées, la première par le fait que ?, la deuxième par le fait que pour une taille donnée, nous n'avons observé que les transferts ayant eu un débit suffisant, les autres sont très proches.

3.3 Influence des transferts peer-to-peer

Suite à l'observation que la panne affecte finalement peu les performances des transferts, nous avons supposé que le modèle de partage de charge multiplexant arrivées suivant des processus de

Poisson et connexions permanentes était susceptible d'en être la raison. En effet, ce modèle de partage de charge ne sature que lorsque la charge des flux poissonniens est supérieure à 1. Nous avons pensé à ce modèle car les transferts de peer-to-peer sont connus pour représenter une part importante du trafic, et où le nombre de chargement (upload) simultanés en constitue la contrainte. Donc toute fin de transfert est immédiatement suivi d'un nouveau transfert ; le nombre de transfert est alors très peu variable.

Afin de valider ce modèle nous donc considérons d'une part l'évolution au cours du temps du nombre de gros transferts peer-to-peer, et d'autre part la charge relative des gros transferts peer-to-peer par rapport au trafic global.

Pour les analyses, nous avons identifiés comme "gros transferts peer-to-peer", les transferts de plus de 10 Ko ayant utilisé les ports applicatifs 1214 (Kazaa), 4661, 4662 ou 4672 (Edonkey).

La figure 3.10 montre le l'évolution du nombre de transferts au cours de la période de capture. Nous constatons effectivement, comme supposé précédemment, que le nombre de gros transferts peer-to-peer est quasiment constant.

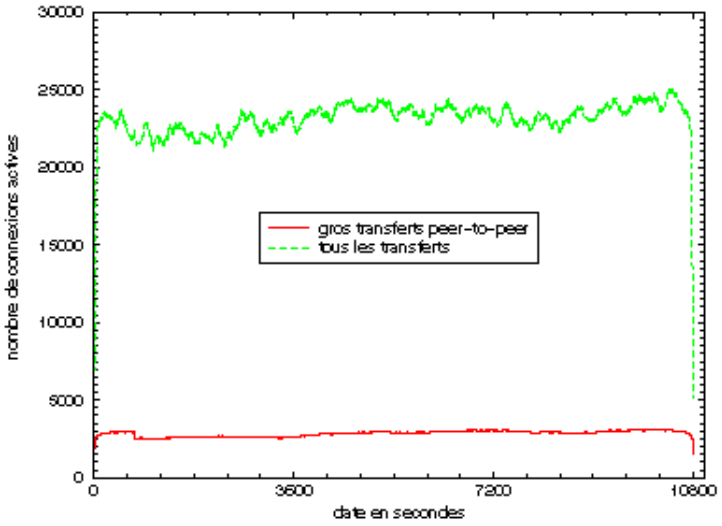


FIG. 3.10 – Evolution du nombre de transferts

Nous avons représenté sur la figure 3.11 la part de trafic des applications peer-to-peer, toutes tailles de transferts confondues car ce trafic est calculé à partir des traces de trafic par paquet donc l'identification des gros transferts est plus compliquée. A titre d'information et de validation, nous avons indiqué sur figure 3.12 la part des nouvelles petites connexions par rapport aux grosses.

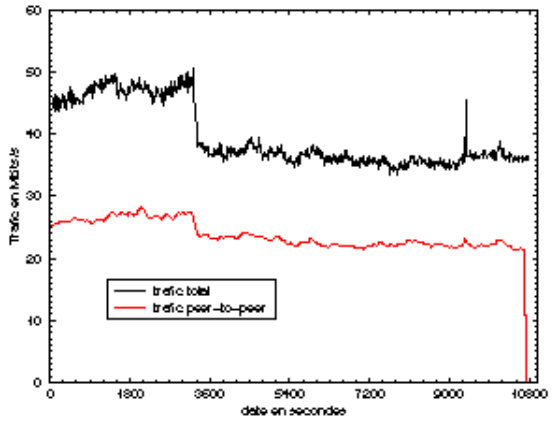


FIG. 3.11 – Trafic des applications peer-to-peer par rapport au trafic global

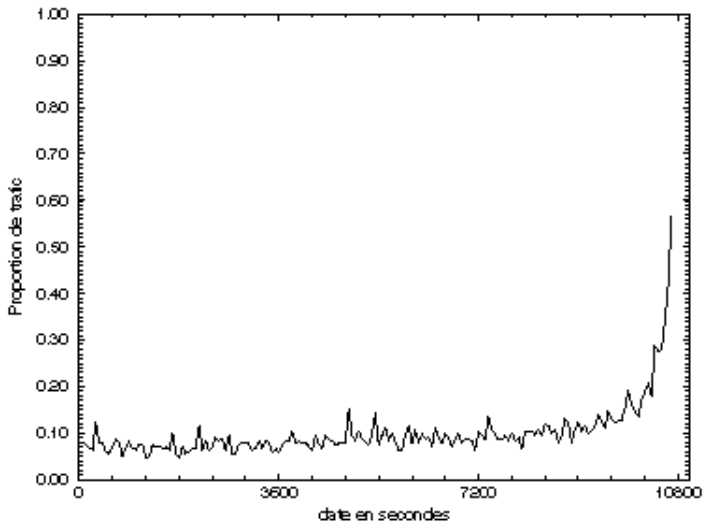


FIG. 3.12 – Part des petits transferts peer-to-peer par rapport aux gros transferts peer-to-peer

Nous vérifions donc tout d'abord sur la figure 3.12 que la part des petits transferts est inférieure à 10% du volume pour le peer-to-peer. Cette part augmente en fin de période toujours en raison du même biais qui fait que les derniers transferts arrivés n'ont pu faire passer que peu de paquets.

Nous constatons ensuite sur la figure 3.11 que le trafic peer-to-peer est certes impacté par la panne, son débit passe de 27 à 23 Mbits/s, mais donc beaucoup moins que le trafic global, qui passe d'environ 50 à 35 Mbits/s. Le trafic non peer-to-peer passe donc de 23 à 12 Mbits/s, il est donc divisé par environ 2.

3.4 Conclusions

Nous avons étudié dans ce chapitre l'impact sur le trafic, le comportement des clients et les performances des transferts de la panne d'un des liens de raccordement de la plaque ADSL de Lille le 5 juin 2003. Nous avons constaté que l'impact de cette panne, et de la surcharge de trafic attendue, n'a pas été très marqué, alors que, compte tenu du trafic avant la panne, nous aurions pu nous attendre à une surcharge de l'ordre de 40% :

- le nombre de connexions actives a peu évolué, tout comme le taux d'arrivée des nouvelles connexions ;
- les clients n'ont pas modifié leur comportement, il n'y a eu ni abandon ni renouvellement des transferts ;
- le débit moyen des transferts a donc été divisé par environ 2,5, du fait de la réduction par 2 de la capacité disponible.

L'objectif de cette étude était de valider les conclusions des modèles d'évolution de TCP en surcharge [2], qui montrent que le système évolue relativement lentement. Nous avons été surpris de voir que le système n'est au final quasiment pas impacté, encore moins que les modèles ne le prévoient. Comme une part important du trafic est constituée de longs transferts (peer-to-peer) qui se succèdent de manière quasi déterministe, nous avons envisagé la possibilité d'utiliser un modèle de partage de charge multiplexant connexions poissonniennes et connexions permanentes, les connexions permanentes correspondant aux longs transferts peer-to-peer.

Chapitre 4

Blind applicative flow recognition through behavioral classification

4.1 Introduction

Application recognition is a fundamental task for a large number of applications. Firewalls rely on it to decide if a particular flow is allowed to get inside (or outside) the company network. Charging mechanism use it to decide which rate applies to a flow. Quality of Service management systems base their access control decision on the application, *etc.* The straightforward approach is to use the well-known port number assignments. In recent work [24] the idea that this classification scheme is no longer appropriate seems to grow. This is mainly because applications as FTP, P2P, game *etc.* are now using dynamic port negotiation. Thus, even if ports can be used to identify the control flows, resulting flows, using ports negotiated in this first connection, are not going to be recognized. FTP, H323 or SIP are very good examples of this kind of behavior. Some applications may even use a well-known port for a different usage (in order to bypass firewall rules for instance). Besides, the recent Peer to Peer study by CAIDA [24] shows that if P2P traffic seemed to have decrease (according to port usage studies), packets payload analysis proves that it is not the case.

This problem shows up clearly by looking at the usual breakdown by application. As an example in August 2000 the application breakdown in the Sprint backbone as seen on the IPMON website (ipmon.sprintlabs.com) show a large amount of HTTP traffic. Using well-known port there was roughly 15% of packets representing 9% of the bytes classified as unknown. But in February 2005, on the same link 35% of packets representing 35% of the bytes are classified as unknown. Even if this is a particular example this increase in the unknown traffic is not only observed in Sprint backbone but appears to be a general trend.

In order to address this problem it seems that a recognition approach based on full packet payload is needed. However this approach raises several fundamental concerns. The first one may be the computing power needed to process full payload packets. Even if solutions can be found for low rate links, handling very fast ones remains a huge challenge. Besides, computation power is not the only issue to be faced when dealing with packets payloads. What can be achieved with encrypted packets for instance ? Or, is it legal to read the data inside packets without written approval by the final user ?

This paper tries to fit into the gap between the unreliable port number classification and the costly, barely legal, full packet analysis. In this paper we propose a framework to do application classification based on behavior recognition. Throughout this paper we will use the term application with the meaning of the protocol assigned by the IANA in the well-known port numbers. An example includes FTP, HTTP, POP, etc. We do believe that different applications exhibits fundamentally different characteristics in term of packet sizes, inter-packet time, as well as interaction with other flows. As an example, we can cite a simple HTTP transfer that consists of a small request from the client followed by a sequence of packets coming in the reverse direction containing the answer to the request. A chat application would behave in a very different way, generating a series of small packets in both directions. These two examples exhibit very different behaviors. In the first case the server tries to send its answer as fast as possible while in the second one the rate is relatively slow and the inter packet time is human driven.

Using a full packet trace collected at the edge of our university, we extract the application flows and the application tag using a full packet analyzer. Each application flow is then modeled using a discrete Hidden Markov Model. We extract some clusters from the set of HMM. Finally with the help of the application tag we describe each application as a mixture of clusters.

HMM have been successfully used in different areas such as voice, handwriting or even image recognition. An important issue is the number of clusters needed to achieve a good classification. We propose a decision process based on the *MDL* (Minimum Descritpion Length) method. This method chooses a number of clusters such that the tradeoff between the computational complexity and the gain of adding new cluster is optimal. This trade off is evaluated through measuring the size of the description of the observation knowing a model fitted to the data.

The results of the classification scheme proposed in this paper are not perfect. But the results are good enough to imagine a new hybrid method where a statistical tool is first applied to each flow. Then a more complex full payload analyze is applied to flows that do not fit perfectly one of the applications models.

The paper is organized as follows. The first section presents some previous works related to this subject. Then we describe the packet traces we collected and how to translate an application flow to a sequence of symbols on which the classification process can be applied. Classical port number classification will also be evaluated in this section. Section 4 will describe the proposed clustering mechanism for finding flows with similar behavior. The MDL criteria for choosing the number of clusters will be described in this section. An interpretation of the clustering in term of applications behaviors we be shown to conclude. The recognition algorithm will be presented with the results in section 5. We will conclude in section 6.

4.2 Background

This paper can be seen as an extension to previous works in flow classification [31, 26, 27]. However, it noteworthy that this work is applied to individual application flows where previous works have been applied at the BGP level. Moreover here we are able to recognize around 55 different flow behaviors where previous works where aimed toward having a small number of class (2 for elephant and mice separation as in [27, 26] and at most 10 as in [31].

More similar to this work, the authors of [19] proposed to represent flows as wavelet coefficients extracted from packet size distributions and interarrival times, and then compared the results of clustering techniques. The authors of [22] proposed a feature-based representation of the flows, and applied a hierarchical clustering technique to find groups of similar flows that can be recycled to form a classification process. However the complex feature-based representation of the flows in these works makes the classification task uneasy. In contrast, in this paper we do the classification directly on the flow sequence. Moreover, we present here a complete quantitative evaluation of our proposed recognition scheme where [22] is lacking such an evaluation.

4.3 Trace description and initial results

4.3.1 Traces description

Our main data set consists in several packet traces collected at the exit of the UPMC university network. We used an optical splitter and a DAG card to capture data transiting through the edge router connecting the university to the French academic network RENATER. The link we monitored was a Gigabit Ethernet Link. We captured three traces of one hour each. Since most of the traffic on this link consisted in a few applications (HTTP, FTP, NNTP) we needed long traces to be also able to classify less common applications. The DAG card is a very powerful tool which allowed us to capture every packets coming in and out the edge router. Besides DAG cards are able to capture more than just packet headers which was very helpful to find out what application was behind each flow. We tested thoroughly what payload size was needed to an accurate classification and it turned out that only 300 bytes per packets were enough. This is far better than full packet capture since a one hour traces shrunk down from a typical size of 50 GB to a more convenient 15GB. As an indication the last one our trace we collected was 16GB, for a total traffic of 58GB and contains 220,000+ flows (a flow being identified by the classical 5-tuple : source IP address, destination IP address, source port, destination port and IP protocol).

Once we had captured these traces we needed to analyze them and remove irrelevant flows for our study. In order to do so we used a tool based on the Coralreef suite developped by Caida (www.caida.org). First of all, we needed to remove non-TCP flows, which are easily filtered out using the IP protocol field. Besides, we had to keep only TCP-flows which started within the trace since we our classification method is based on the first packets. We achieved this by checking each flow. We decided to keep a flow only if it began by a full successful TCP handshake. On the trace mentioned before, after filtering out non TCP traffic, 76% of the flows remains (representing more than 98% of the total volume). After the full handshake filtering 62% remains (accounting for 76% of the total volume).

In order to achieve a proper classification we also filtered out TCP control packets and kept only those with an application payload. As we want to focus on the behavior of applications we tried to avoid the pollution of the trace by TCP control packet (TCP KeepAlive or TCP Ack with no data). Besides, in order to find out relevant patterns we needed flows which exchanged at least a certain number of packets with application data. Thus our final sequences were extracted from long enough flows and consisted in the sizes of the first data packets exchanged. We decided to keep only flows with more than seven packets containing actual application data. This decision was based on some initial testing and results in a trade-off between the quality of the clustering (the more packets kept,

the more efficient) and the complexity. Besides, using more packets decrease the numbers of flows that can be dealt with in our method (since in most of the flows only a few packets are exchanged).

On the trace previously mentioned, this last filtering left us with 22% of the original but they accounted for 74% of total volume. In a situation where we would be doing real time analysis, flows without handshake would no longer be an issue and our filtering would leave with 35% of the TCP flows, accounting for more than 98% of the traffic (for the example trace).

Finally, another TCP related behavior has to be taken into account. Indeed, if you look at packets belonging to a flow on a link, they may not appear in order or they may appear more than once. There can be several reasons for packets not appearing in the sequence they were sent as shown in [23]. Out of order packets will decrease the quality of the clustering by introducing new non-legitimate HMM states. We need to consider this behavior by removing any re-transmission and deliver the packets in order to our analysis tool.

4.3.2 Application labeling

To describe applications with a mixture of clusters in a semi-supervised way we needed to label each flow with the corresponding application. This labeling could have relied on TCP ports. However using IANA well-known ports is no longer reliable when studying traces as mentioned in the introduction. Since our mixtures was based on the application labels we needed it to be as reliable as possible.

That is why we decided to use an applicative classification tool : Traffic Designer (www.qosmos.fr). Traffic designer is able to label a flow with an application based on the TCP payload. A flow is said to belong to a specific application only if the syntax of the data exchanged between the two TCP peers matches to the application syntax. For instance if a flow runs on port 21 and contains something like "GET /index.html HTTP/1.1", Traffic Designer is going to classify it HTTP whereas a standard port classification would describe it as FTP. The classification engine of Traffic Designer is quite complex. The main idea behind it is to find out the protocol stack in each packet. In order to do so each layer starting at the data link level is thoroughly examined. Once Traffic designer has found out what protocol is running in a specific layer it tries to match its payload with a protocol than can be found on top of it. This allows Traffic Designer to successfully analyze complex protocol stacks such as those hidden inside in tunnels (which 5-tuple analysis can not find).

An example of the proporion of unknown traffic seen using well-know port is given in fig. 4.1. To compare these numbers with the application breakdown optained using the Traffic Designer see fig. 4.2.

As we can see on these figures the use of an applicative classification tool improves a lot the quality of the analysis. The main difference here is due to FTP data flows which can not be found based on TCP ports only since server ports are dynamically negotiated. We can also observe a certain amount of misclassified flows (applications not running on their official port). However less than a percent falls in this category for our specific trace, which would not make it an issue. But in the case of commercial traffic (ADSL user for instance) this proportion should increase a lot.

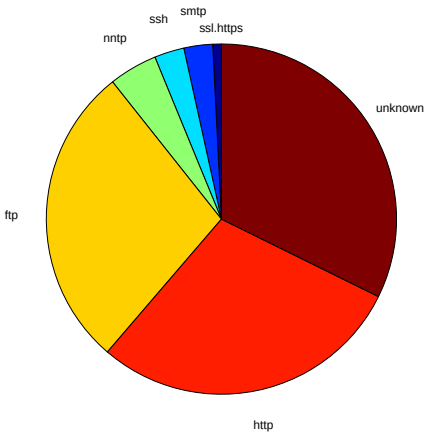


FIG. 4.1 – Breakdown according to TCP ports

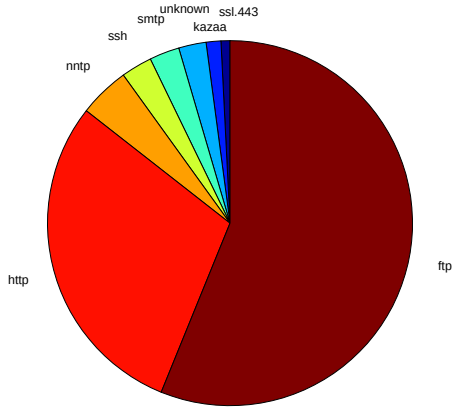


FIG. 4.2 – Breakdown according to Traffic Designer

4.3.3 Flow representation

In this work, we want to do application recognition using a minimal amount of information and especially without going into the packet payload. However, inter-packet time that might be easily extracted over a packet flows appears to be unusable for our purpose. This is related to the fact that inter-packet interval time is affected by by the network load, TCP acknowledgments and window based mechanisms, as well as application related properties. In practice it is very difficult to separate TCP and network induced delay characteristics from applicative properties. This motivates our decision of not using the inter-packet delay and to keep for each packet only two informations : the direction of the packet, and its size. The side that opens the connection will be later refered as the client. The host waiting for the connection will be named server. This denomination is often the intuitive one but in some case when the port are dynamically assigned this can be misleading. However this does not affect our classification scheme.

The precise size of packets is not very important for our metho. To facilitate later interpretation, we quantize packet size in four level $\{1, 2, 3, 4\}$ for small, small-mid,large-mid and large size packets. We choose the quantification steps by observing the multi-modal size distribution of packets (see Fig. 4.3). This distribution shows clearly three modality for packet size and we choose the discretization levels based on these modality as $\{1 = [0, 150], 2 = [150, 700], 3 = [700, 1300], 4 = [1300, 1500]\}$.

We represent the direction of the packet as a sign (client : + and server : -). This lead to the representation of a flow as a sequence of positive or negative number in $\Sigma = \{\pm 1, \pm 2, \pm 3, \pm 4\}$. Each flow is therefore represented as $f_i = (\sigma_1, \ldots, \sigma_{l_i})$ where l_i is the length of the sequence and σ_t represents the t^{th} packet of the sequence ans $\sigma_t \in \Sigma$. Throughout this paper the recognition horizon, i.e the maximal value for l_i will be noted $N_{\max}$. And finally let $\mathcal{F} = \{f_i\}$ be the set of all the sequences.

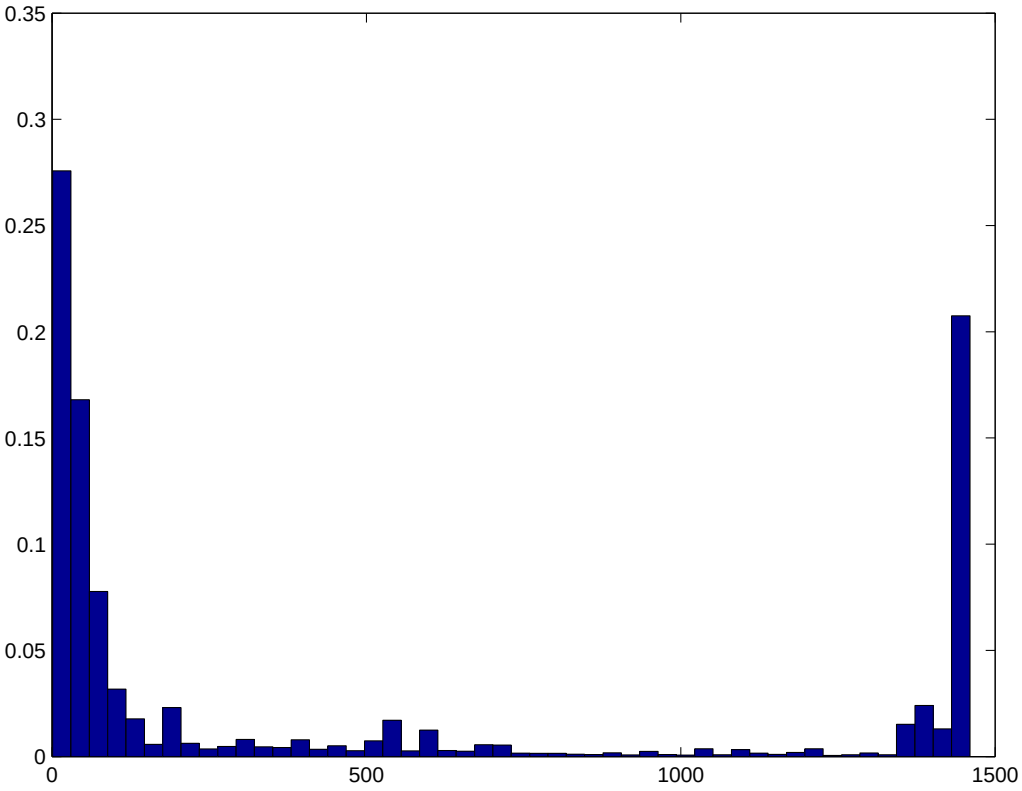


FIG. 4.3 – Empirical packet size distribution

4.4 Flow behavioral clustering

The proposed methodology in this paper is described in fig. 4.4. In this section we will describe the the learning phase. This phase validates the hypothesis that flows might be recognized through their behavior captured by a sequence as described in previous section. This validation is obtained in two steps : a first step where we apply an unsupervised clustering algorithm to the flows observed over a trace. The second step compare the cluster contents with the applications labe as assigned by the Traffic Designer box. Our learning set is composed of 1000 flows tagged by the Traffic Designer. We randomly choose 100 flows from each application : HTTP, FTP, Kazaa, POP3, POP3S, SMTP, HTTPS, Edonkey, NNTP and SSH. The random choice is introduced to deal with the overwhelming presence of HTTP as much as 90% of the flows in some of our traces. The learning set is only utilized to validate the application recognition. The clustering is using all the avaliable flows.

FIG. 4.4 – Methodology of blind application recognition

4.4.1 Capturing Flow behavior through HMM

In the previous sections we presented the prefiltering done to each flow crossing our measurement point. This section explain the algorithm to cluster flows with similar behavior. Hidden Markov Models (HMM) have been widely used as probabilistic model to describe the behavior of complex systems [29]. The idea of this class of model is that the observed system output at step i, $\sigma_i \in \Sigma$ can be explained as a random function of an unobserved internal state $q_i \in \mathcal{Q}$. The unobserved internal state q_i follows a Markov chain. In this paper we assume that the observed output σ_i comes from a finite set $\Sigma = \{\pm 1, \pm 2, \pm 3, \pm 4\}$. And the state space $\mathcal{Q}$ is also discrete. The random function relating the output symbol to the state is characterized through an emission probability distribution $\mathbb{P}\text{rob}\{\sigma_t = i | q_t = j\}$. As explained before, the sequence of states is supposed to be hidden for HMMs. We therefore need to use the observed outputs of the system to infer the state sequence $Q_i = \{q_1, \ldots, q_{l_i}\}$ for the sequence f_i of length l_i.

A k-state HMM with states $\{s_1; \ldots, s_k\}$ is formally defined as a 3-tuple (Π_0, A, B) such that :

- $\Pi_0 = \{\pi_1, \ldots, \pi_k\}$, $\pi_i = \mathbb{P}\text{rob}\{q_0 = i\}$ are the initial state probabilities
- $A = \{a_{ij}\}$, $1 \leq i, j \leq |\mathcal{Q}|$, $a_{ij} = \mathbb{P}\text{rob}\{q_{t+1} = i | q_t = j\}$ is the transition matrix between states.
- $B = \{b_{ij}\}$, $1 \leq i \leq |\mathcal{Q}|$, $1 \leq j \leq |\Sigma|$, $b_{ij} = \mathbb{P}\text{rob}\{\sigma_t = i | q_t = j\}$ are the emission probability in each of the states.

The simple structure of HMM enables an easy and efficient computation of the likelihood of the observation of a sequence f_i given the parameters of a particular HMM. This derivation is based on the well-known Baum-Welch "Forward-Backward" algorithm.

The idea of using a HMM for our behavioural analysis is that any packet send by an application depends on the overall applications context. Unfortunately this context is hidden and we only see it throught the packet flow. The behavior of flows (as sequences) might be captured in a HMM through several means, with varying number of states and different structures. Here we focus on a specific HMM structure proposed in [21, 28] where each state corresponds to an element of the sequence. This structure has been intensively applied in the context of sequence analysis and handwritten recognition. As explained before, input sequences have a length of at most $N_{\max}$, meaning that the Markov chain has at most $N_{\max}$ states. The state Markov chain is assumed to go only from left to right, *i.e.* $a_{ij} = 0$ if $j \neq i + 1$ and $a_{i,i+1} = 1$,which is more suitable for analyzing a sequential behavior.

The remaining parameters of the HMM (i.e. the emission probabilities) are estimated like in [21]. We first extract from the overall sequence set $\mathcal{F}$ and for each symbol $s \in \Sigma$ a probability measure $P_s(r) = \mathbb{P}\text{rob}\{\sigma_{t-1} = r | \sigma_t = s\}$. This represents the probability that the symbol r follows the symbol s. This probability measures is estimated by $P_s(r) = \frac{\#(\sigma_{t-1}=r, \sigma_t=s)}{\#(\sigma_t=s)}$, where $\#(.)$ means the number of occurrences of the given sequence. The similarity measure between two symbols s_1 and s_2 $s_1, s_2 \in \Sigma$ noted $c(s_1, s_2)$, is defined as the correlation between each probability law P_{s_1} and P_{s_2}.

We define $b_{t,j}$ as the emission probability of the symbol j at the state t of the HMM derived from the sequence $f_i = \{\sigma_1, \ldots, \sigma_{l_i}\}$. Then $b_{i,j} = \frac{c(\sigma_t, j)}{\sum_{k \in \Sigma} c(\sigma_t, k)}$, $j \in \Sigma$. This definition of emission probability relates the hidden context to affinities between the observed symbol σ_t to all the other symbols. This enables a highly generic relationship between context and observation. The emission

probabilities are assigned to states of the HMM representing each existing flow in $\mathcal{F}$, *i.e.* each flow is now represented by a HMM with the structure depicted in Fig. 4.5.

FIG. 4.5 – HMM Structure used throughout the paper.

Each sequence f_i has its properties summarized by its HMM. This new representation will allow us to quantify the similarity between the sequences. This similarity will be used in the clustering step.

4.4.2 Hidden Markov Model based Clustering

The method described in this paper consists of projecting the sequences representing the flows in a high dimensional space using Hidden Markov Models. The generic method for HMM-based clustering of sequences is given in [30]. Lets suppose that each sequence $f_i \in \mathcal{F}$ is associated with a HMM H_i as described in previous section. One might obtains for each HMM H_i, the log-likelihood of all other sequences f_j. This log-likelihood is easily obtained using the Baum-Welch forward-backward filter which is known to have a linear complexity in the number of symbols in the sequence.

We define a matrix of log-likelihood $\Lambda = (L_{ij})$ where L_{ij} is the log-likelihood that sequence f_j have been generated by HMM H_i related to sequence f_i *i.e.* $L_{ij} = -\log \mathbb{P}\text{rob}\{f_j|H_i\}$. Now based on this log-likelihood matrix we can define a distance matrix $D = (d_{ij})$ through the euclidean distance as,

$$d_{ij} = \sqrt{\sum_{k=1}^{N} \|L_{ik} - L_{jk}\|^2},\ i, j = 1, ..., N.$$

It is noteworthy that each distance vector (L_{i*}) (each line of the matrix Λ) defines a point in $\mathbb{R}^N$, representing the sequence f_i. Similar flows with similar behaviors should be intuitively close together. We therefore have to regroup close enough points in the N-dimensional (N being the number of initial applicative flows) into homogeneous clusters. The distance matrix D extracted from Λ expresses distances between these points.

The application of the presented derivation on our learning set lead to a distance matrix of dimension 1000, and the clustering step have to cluster 1000 points in a 1000-dimensional space. We show in Fig. 4.6 the obtained distance matrix over the evaluation learning set.

The complexity involved in the calculation of the distance matrix is $\mathcal{O}(N^3)$ in time.

The term "clustering" refers to the task of identifying disjoint concentrations of points (clusters) in a multidimensional space. It is an instance of unsupervised learning, meaning that these groups have to be inferred only using a distance measure between the points and not from a class membership as for supervised learning. Intuitively the clusters found must have a high intra-cluster and a low inter-cluster similarities, as the desirable result is to find well-separated and compact clusters. Several methods of clustering might be applied here (e.g. K-means, Hierarchical Clustering,

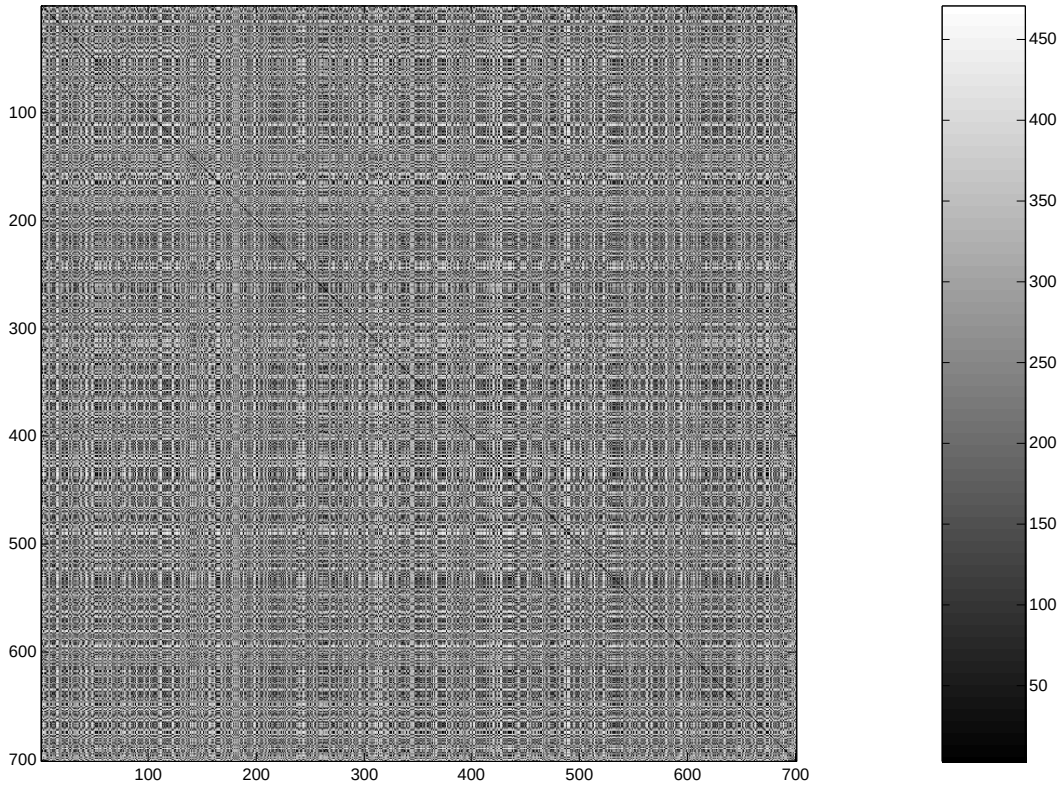


FIG. 4.6 – Distance Matrix obtained over the evaluation learning set

...). However, despite a very rich literature clustering is known to be a hard task. Classical clustering approaches as k-means or approaches based on Gaussian mixtures work well when clusters have a spherical or elliptic form. However in a lot of applications, clusters might have complex non-spherical structure, and even not being convex. Moreover, classical methods frequently fall into the pitfall of local minima of likelihood function. We have chosen to apply here the so-called Spectral Clustering method.

4.4.3 Spectral method for the distance matrix

Spectral clustering methods have become popular recently [25] because of their efficiency in a wide range of problems where clusters are not following a spherical structure. We show in Fig. 4.7 four projections of the points we have to cluster. As the points are defined over a 1000-dimensional space we can only illustrate them through projections. It can be seen that it is almost impossible to find out through eyes evaluation a trivial clustering of these points. Moreover the possible clusters,

do not seem to have a spherical structure, meaning that the application of the k-means method is likely to fail in this context.

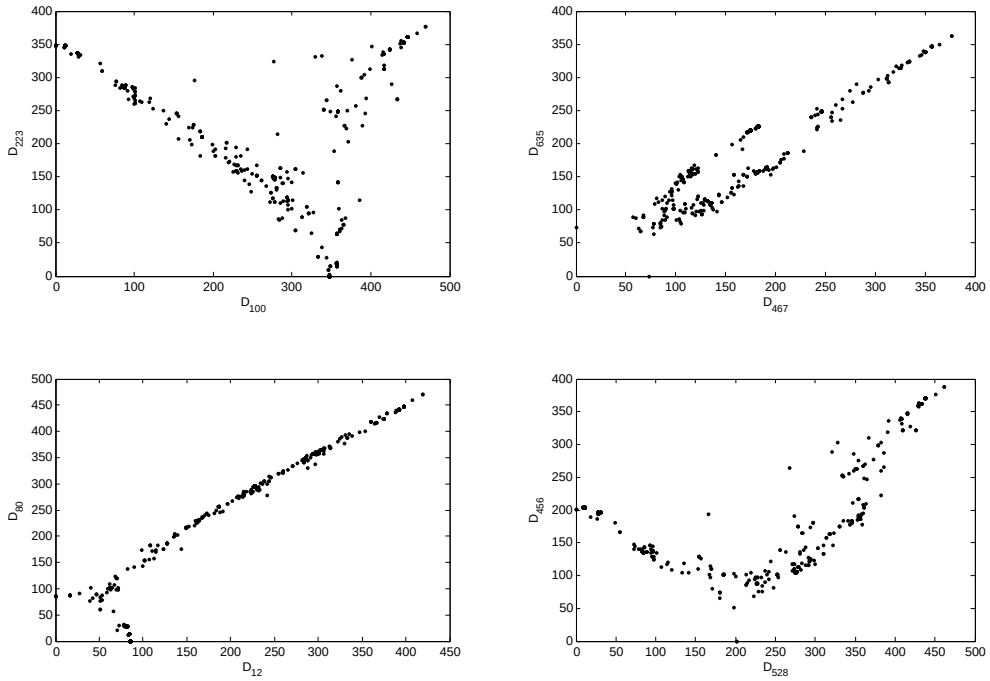


FIG. 4.7 – Four projection of the points coming from the distance matrix. The labels define the variable that where used for the projection

The spectral method consists of three main steps : a first step of preprocessing that transforms the distance matrix to an affinity matrix, followed by a second step of spectral analysis that might be assimilated to a Principal Component analysis and a last step of clustering *per se* that will do the regrouping operation over a transformed space. The first step of preprocessing is in fact divided in two sub-steps : the mapping of the distance matrix to an affinity matrix and a further transform of affinity to a conductivity matrix. These two preprocessing steps are meant to reinforce the structure of the distance matrix to enable an easy clustering at the end.

Affinity matrix

To explain how the spectral clustering method works, we will assume that the clustering is *a priori* known and that points have been ordered so that points in the same cluster are consecutive. Moreover the clusters have been ordered in size-increasing order. This hypothesis will be applied through the analysis but as we will see at the end it is not necessary for the analysis. The distance matrix after reordering is shown in Fig. 4.8.

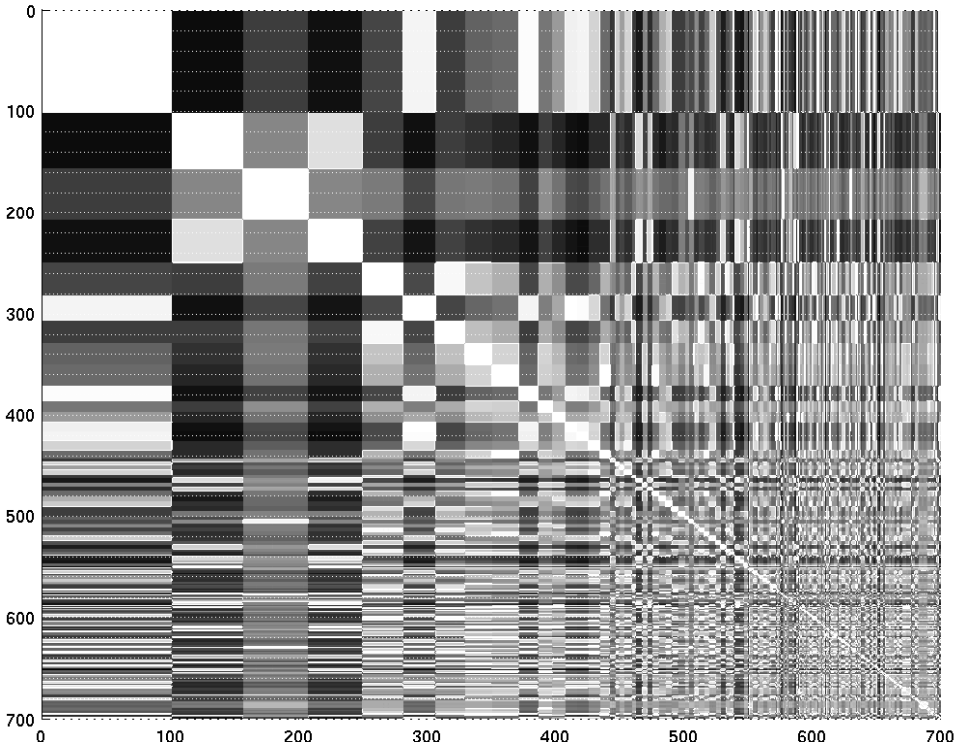


FIG. 4.8 – Distance Matrix after reordering following the clustering

The new distance matrix has now a block structure that was not visible in the initial matrix before clustering (see Fig. 4.6). However even doing the clustering using this new distance matrix is not straightforward as the block structure is spoiled by elements far from the diagonal. To deal with this problem a non-linear operation is frequently applied to the distance to derive the so-called *Affinity Matrix*, $A = (a_{ij})$. The Affinity matrix contains values between 0 and 1, and evaluates the similarity of points. The non-linearity of the mapping function between distance and affinity is meant to reinforce the block-diagonal structure of the distance matrix. It is usual to take a gaussian kernel as the mapping function between the affinity and distance, *i.e.* $a_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right)$. This choice of the mapping function leads to a positive definite affinity matrix. This property will show to be very useful when we will have to calculate eigenvalues and eigenvectors of a large matrix, as it enables the application of Cholesky decomposition and simplify the calculations.

The choice of σ, the radius of the gaussian kernel, is important for the efficiency of the clustering. We have followed in this paper the method proposed in [20] ; we have chosen for each point i, the radius σ_i as the value where $\sum_{j=1}^{N} \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right) = \tau$. This choice will be motivated through the interpretation of the affinity matrix in terms of graph. The affinity matrix might be interpreted as a

random adjacency matrix, *i.e.* an affinity a_{ij} can be interpreted as the existence of a link between i and j with a probability a_{ij} in a graph. Each particular clustering can be seen as a particular realization of this random graph where each cluster is a connected subgraph.

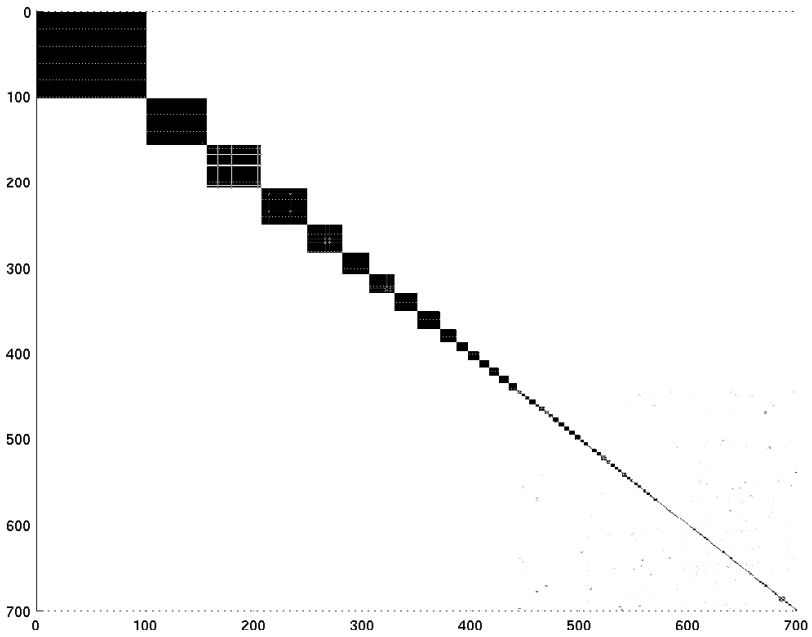


FIG. 4.9 – Affinity Matrix after reordering following the clustering

Moreover, the value $\sum_{j=1}^{N} a_{ij}$ represents the mean number of neighbors of the point i in this random graph. This motivates the choice of σ_i to fix the mean size of the neighborhood τ of each point. However this choice of σ_i lead to an assymetric affinity matrix as a_{ij} is controlled by σ_i, where a_{ji} depend on σ_j. We make the affinity matrix symmetric by taking $\alpha_{ij} = \min(a_{ij}, a_{ji})$.

The application of the described derivation of the affinity matrix to the reordered distance matrix leads to the affinity matrix shown in Fig. 4.9. In this derivation the number of neighbors has been fixed to $\tau = 5$. This figure shows clearly that the affinity matrix has a reinforced block diagonal structure, meaning that clustering is easier using the affinity matrix in place of the distance matrix.

However, affinity matrix might be insufficient to tackle with clusters that are not convex. Fig. 4.7 gives the intuition that this might be case for the learning set under study. Moreover it can be seen that the block structure of the affinity matrix is almost perfect up to line and column 750, but we observe some dispersion after these values. This motivates another processing step to deal with such clusters. This step makes use of the interpretation of the affinity matrix as an adjacency matrix.

Conductivity matrix

In order to handle cases where data do not form compact and convex clusters, we have to extend our definition of a cluster : two points belong to a cluster if they are close enough (or equivalently

they have enough affinity), or if they are well connected by paths of short "hops" over other points. The more such paths exists, the higher the chances are that the points belong to the same cluster. We are therefore defining a new affinity measure, based on the random graph view of the affinity matrix. Instead of considering two points similar if they are connected with a high probability by a link, we assign them high affinity if the overall graph conductivity between them is high. This new affinity matrix is called the *Conductivity Matrix*, $C = (c_{ij})$. This definition enables affinity to depend on the distance between points as well as to the neighborhood and the "weight" of a cluster helping to shape non-convex clusters. This definition is very similar to electrical circuit, where the conductivity between two nodes depends on all paths between them. The idea is to suppose that the graph is an electrical network with a resistance a_{ij} between each two nodes i and j.

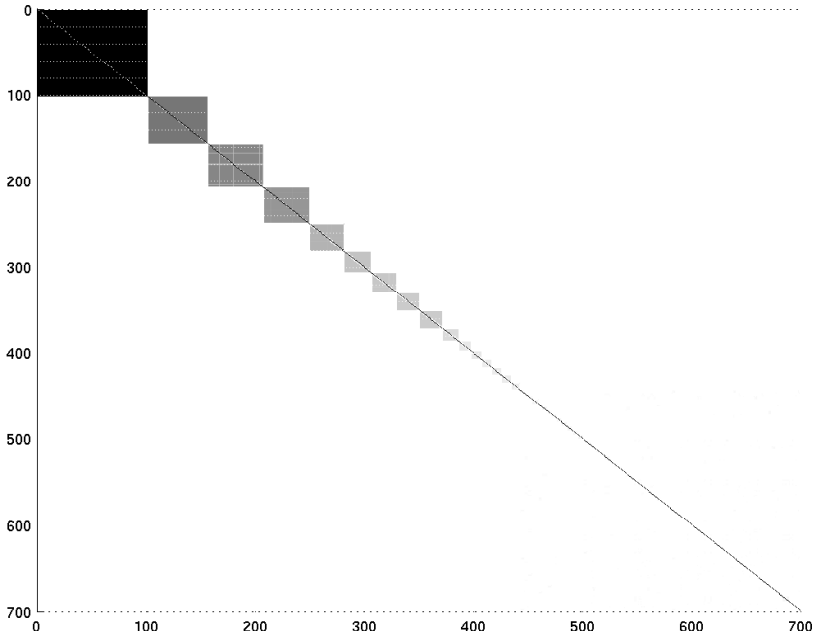


FIG. 4.10 – Conductivity matrix after reordering following the clustering

c_{ij} is computed by submitting the network to an entering current I at node i and an output current I and node j and deriving the voltage between these two points. The conductivity is found through Ohm law as the ratio between the current and the voltage. A look to an electrical circuit theory book will give the following formula for the conductivity matrix :

$$\begin{aligned} c_{ij} &= \frac{1}{G^{-1}(i,i) + G^{-1}(j,j) - G^{-1}(i,j) - G^{-1}(j,i)} \; i \neq j \\ c_{ii} &= \max\{c_{ij}\} \end{aligned}$$

where the matrix $G = (g_{ij})$, represents the matrix of voltage between point i and j. This matrix is defined through

$$\begin{aligned} G(1,1) &= 1, \\ G(1,j) &= 0, for\, j > 1 \\ G(i,j) &= \sum_{k \neq i} \alpha(i,k)\, for\, i \neq j \\ G(i,i) &= -\alpha(i,i), \end{aligned}$$

The complexity involved in the derivation of the conductivity matrix is $\mathcal{O}(N^3)$ in time leading to an overall complexity of $\mathcal{O}(N^3)$ for the preprocessing step.

The application of this ultimate preprocessing step to the reordered distance matrix leads to the conductivity matrix shown in Fig. 4.10. This figure shows that the conductivity matrix has now a perfect block diagonal structure and that the clustering might be done easily on it. In next section we will explain the next step which is the spectral clustering by itself. This clustering will be applied on the conductivity matrix.

Spectral decomposition and k-lines

At the end of the preprocessing step we have a set of points (each line of the conductivity matrix) that has to be clustered and the conductivity matrix is supposed to be highly block diagonal. The block diagonal structure means that the eigenvalues and eigenvectors of the matrix C can be obtained as a union of eigenvalues and eigenvectors of its blocks (the latter padded with zeros at appropriate position). Now stacking eigenvectors of C column-wise in a proper order we obtain a matrix X with the following form :

$$X = \begin{bmatrix} \vec{x}^1 & \vec{0} & \dots & \vec{0} \\ \vec{0} & \vec{x}^2 & \dots & \vec{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \dots & \vec{x}^L \end{bmatrix}$$

where each sub-matrix $\vec{x}^i$ denotes the eigenvectors of block i, and $\vec{0}$ is a null sub-matrix. The clustering is now done in the space spanned by the eigenvectors. Let's suppose that we want to regroup the data in K clusters, the clustering occurs in the K-dimensional space spanned by the first k eigenvectors relative to the K largest eigenvalues. Points belonging to each particular cluster will be along the subspace spanned by the eigenvectors. We are assuming here that the number of clusters K is known in advance. We will present in section 4.4.4 how to choose this number.

Spectral clustering is very flexible as it enables to choose the number of clusters, through choosing the number of eigenvectors and eigenvalues. Choosing a smaller number of eigenvectors is equivalent to doing the clustering in a space of lower dimensionality. Moreover, the spectral decomposition has an interpretation in terms of Principal Component Analysis (PCA) : the best clustering using K clusters is the one that is done using the eigenvectors relative to the K largest eigenvalues.

The spectral decomposition leads to the following algorithm for clustering : ***calculate the eigenvectors of the conductivity matrix and regroup points in the subspace spanned by the eigenvectors along the plane defined by the eigenvectors subspaces***.

However the previous derivation makes the hypothesis of a "**perfect**" block diagonal structure. In practice, even after the mapping to the affinity and the conductivity matrices, the obtained matrix is not perfectly block diagonal. This means that the clusters might not perfectly be aligned with the eigenvectors. This means that we might have to search linear (or planar) alignment in the eigenvector spanned space. This search can be achieved by using the k-lines algorithm. This algorithm is similar with the well known k-means algorithm but it find linear alignments in place of spherical clusters. It works through choosing a linear alignment and finding the set of closest points to this alignment. The alignment is readjusted at the end of each assignment round. This k-lines algorithm is described below.

1: Initialize vectors $m_1, \ldots, m_k$ as the k eigenvectors related to the largest eigenvalues.
2: **for** $j = 1 \ldots k$: **do**
3: Define P_j as the set of indices of all points y_i that are closest to the line define by m_j and create the matrix $M_j = [y_i]_{i \in P_j}$ whose columns are the vectors y_i.
4: **end for**
5: Compute the new value of every m_j as the first eigenvector of $M_j M_j^T$.
6: Repeat until m_j are stable.

Algorithm 1: k-lines algorithm

This algorithm can be proved to converge to a stable solution (similarly to the k-means algorithm).

In the previous descriptions we assumed that the clustering is *a priori* known and that the distance, affinity and conductivity matrix were reordered to follow the block diagonal structure. However, the eigenvectors are insensitive to the order of column of a matrix, meaning that even if the conductivity matrix was not reordered, the obtained eigenvectors are the same. This means that distance, affinity and conductivity matrix have not to be block diagonal for the eigenvectors being helpful for clustering. We illustrate this point in Fig. 4.11, by showing the full methodology on the non-ordered distance matrix.

As can be seen in this figure the eigenvector have a clear block structure even if the initial distance matrix have not been reordered to have a block-diagonal structure. The block structure of the eigenvector should ease the clustering using the k-lines algorithm.

Selecting Representative HMM for cluster At the end of the clustering in eigenvectors subspace, each point related to a particular flow through a sequence $f_i = \{o_1, ..., o_N\}$ and an HMM H_i will be assigned to one of the K clusters, $\mathcal{C}_k$. However we have now to choose for each cluster the "best" representing HMM, $H_k^*,\ k \in \{1, \ldots, K\}$. This choice might be done as usual through a Maximum likelihood criterion :

$$H_k^* = \arg\max_H \left(\sum_{i:\, f_i \in \mathcal{C}_k} \log \mathbb{P}\mathrm{rob}\{f_i | H\} \right)$$

This choice finishes the first step of the methodology consisting of unsupervised learning. We will discuss in section 4.4.5 the results of this clustering on the learning set. However before doing

this we need to decide how many clusters are needed to describe the learning set. This will be the subject of the next section.

4.4.4 Model selection and MDL criterion

Up to know we have assumed that the number of cluster K is a known value. However in real world the choice of this value is not straightforward and evaluating this number under general conditions is an open problem. Choosing the number of clusters can be seen as a *model selection* problem. We will describe here a model selection method based on the so-called Minimum Description Length (MDL) approach [18]. This method applies the Occam razor [1] to choose the number of needed clusters through a trade-off between model accuracy as evaluated by the likelihood of the model and model complexity measured by the number of needed parameters.

Our aim here is to find a good clustering with not too many clusters and a sufficient goodness-of-fit to the data (a correct likelihood). The idea of MDL is to translate this problem into a *description length* problem. Let's suppose that one wants to describe the obtained clustering in any specific language he wants. The description of the data will consist of two parts : description of the model, and description of the data assuming the model. The description length $\mathcal{L}$ is therefore,

$$\mathcal{L} = Length(Data|Model) + Length(Model)$$

In our context the model is the cluster structure itself, meaning that we have to first describe the clusters and after that to describe the points inside the clusters.

MDL assumes that these two descriptions are done in the most compressed way through an Information Theoretic compression mechanism. Now, if the likelihood of observing data x through the model $\mathcal{M}$ is given by by $\mathbb{P}\text{rob}\{x|\mathcal{M}\}$, the well-known Shannon compression theorem states that the length of the description of x will be larger than $-\log \mathbb{P}\text{rob}\{x|\mathcal{M}\}$. We will assume that we are using an optimal compression code and therefore length of the representation converges asymptotically with large N (number of observations) to the log-likelihood, $Length(Data|Model) = -\log \mathbb{P}\text{rob}\{x|\mathcal{M}\}$. Moreover the description size of the model will depend on the number of parameters required by the model.

In our context the model consists of the obtained clustering containing K clusters, represented each by an HMM H_k^* with $N_{\max}$ states. First of all we have to describe the value of K the number of clusters. As the number of bits needed for this description is not known *a priori* we have first to give this number through a sequence of $\log K$ zeros followed by a one used as a delimiter. After that the precise value of K, is provided using $\log K$ bits. All in all, $2\log K + 1$ are needed to describe K. After that, we have to describe each one of the K representative HMMs. Each HMM has $N_{\max}$ states and each state has $|\Sigma|$ emission probabilities associated. If we represent each emission probability using α (α is fixed and does not depend on K), each cluster would be represented by $\alpha \times N_{\max} \times |\Sigma|$ bits. This leads to a description size for the overall model equal to $Length(Model) = 2\log(K) + \alpha K N_{\max}|\Sigma| + 1$.

The likelihood of the learning set knowing the model and the clustering might be derived as :

[1] *"Pluralitas non est ponenda sine necessitate"* or in other terms, *entities should not be multiplied needlessly*

$$\mathbb{P}\text{rob}\{x|\mathcal{M}\} = \prod_{i=1}^{N} \mathbb{P}\text{rob}\{f_i|H_k^*, i \in \mathcal{C}_k\}$$

where $\mathbb{P}\text{rob}\{f_i|H_k^*, i \in \mathcal{C}_k\}$ is the probability that the sequence f_i is generated by the representative HMM H_k^* of the cluster that f_i belongs to. Mixing the model description length with the likelihood calculation we reach to the following description length $\mathcal{L}(K)$ as a function of cluster number :

$$\begin{aligned}\mathcal{L}(K) = \quad & -\sum_{i=1}^{N} \log(\mathbb{P}\text{rob}\{f_i|H_k^*, i \in \mathcal{C}_k\}) \\ & +2\log(K) + \alpha K N_{max}|\Sigma| + 1\end{aligned}$$

The MDL criterion chooses the value of K that minimizes the description size, *i.e.* $K^* = \arg\min_K \mathcal{L}(K)$. We are showing in Fig. 4.12 the description length curve obtained over the learning set as function of cluster number K.

The curve shows a minimum of the description length at $K = 55$ meaning that 55 clusters generate the smallest description of the training set and is the MDL best estimate of cluster numbers. Doing clustering with a larger number of clusters yields some empty clusters, or too small clusters.

The fact that we have 55 clusters means that we have find over the trace 55 different type of behaviors. This observation have to be compared with the fact that we had in the training set 10 type of applications. This means that flow from the same application might exhibit different behaviors. Fig. 4.13 shows the Distribution of different behavior among different applications. Some application exhibits different behavior (as for example edonkey or FTP). The different behaviors are captured in different clusters. Some other applications appear remarkably homogeneous in term of behavior. NNTP is a very good example where 100% are in the same cluster. The dispersion of applicationw between different behaviours lead to an applicative model that will be as a mixture of HMM. This point will be developped in section **??**. In the next section we will analyze in more details the results of the application of the clustering method to the training set.

4.4.5 Clustering results

Analyzing the clusters in some details can be very interesting for several reasons. First of all, each cluster describe one type of behavior, and the relative weight of each cluster shows how common this behavior is in the studied training set. Moreover, looking at the applications appearing in each cluster (each flow is assigned a cluster and each flow is labeled with an application) shows that applications might exhibit different behaviors and that some applications have similar behaviors with others. The clustering step gives us 55 clusters and as an example we will analysis two of them.

Figure 4.14 represents the quantized packets size and the direction of clusters 1,2, 3 and 8 (in clockwise order). The first cluster contains 20.6% of the flows and the sequences for this cluster is shown in figure 4.14. table 4.1.. Table 4.1 show the proportion of applications that fall in this cluster..

As we can see from 4.14 the first cluster gathers flows containing small packets sent alternatively by the client (i.e. the machine that started the TCP connection) and by the server (i.e. the machine

TAB. 4.1 – Applications in cluster 1

Application	% in this cluster
NNTP	100%
POP3	86%
SMTP	11%
FTP	9%

that accepted the connection) starting with the server. If we give a closer look at the application seen in this cluster we can see that all of them, except SMTP, have a similar identification process (see Tab. 4.2 which make them having similar behavior. Thus, the first packet is a welcome packet sent

TAB. 4.2 – Packets Exchanged for cluster 1.(s→c) describes a packet sent from the server to the client and (c→s) a packet sent by the client to the server.

Packet	NNTP	POP3	SMTP	FTP
1 (s → c)	Banner	Banner	Banner	Banner
2 (c → s)	User	User	EHLO	User
3 (s → c)	Pass ?	User Ok	250 Ok	Pass ?
4 (c → s)	Pass	Pass	MAIL FROM	Pass
5 (s → c)	Limit	Ok	Ok	Login OK
6 (c → s)	Quit	Stat	RCPT TO	Type
7 (s → c)	Quit	Ok	Ok	Ok

by the server, while second, third, fourth and fifth are login exchanges and the sixth is a command sent from the client to the server. It is interesting to note that all the NNTP flows found in our data probably end with a "Daily Limit Exceeded" message. Some SMTP flows ends up in this cluster simply because the first packets exchanged to send an email can fit the identification behavior of the other applications (according to the way we build our sequences).

The second cluster is much simpler. It gathers 5.5% of the flows, all of which are HTTP ones. Besides 54% of the http flows fall in this cluster. The packets sequence is represented in figure 4.14. Basically, the first packet is a medium sized one sent by the client, and the following packets are big ones sent back by the server. The first thing to say about this sequence is straightforward : the first packet is an "HTTP GET" and the following ones are the answer from the Http server. However there are a few interesting things to note. First of all the size of the answer vary and this is the reason why the sizes of packets 6 and 7 are not all quantized to the same value. Besides, some flows even end up with a packet being sent by the client. This can be interpreted as follow : some HTTP 1.1 flows can be found in this cluster and the next packet sent by the client is the following "HTTP GET".

TAB. 4.3 – Classification results

	edonkey	ftp	http	kazaa	nntp	pop3	smtp	ssh	https	pop3s
edonkey	*84.2*	4.2	1	0	0	0	2.2	0	5.6	2.8
ftp	0	*87*	2.4	0.2	6.2	0	4.2	0	0	0
http	0	0	*99*	0	0	0	0	0	1	0
kazaa	0	0	4.76	*95.24*	0	0	0	0	0	0
nntp	0	0	0	0	*99.6*	0	0.4	0	0	0
pop3	0	0.6	0	0	86.8	*0*	12.6	0	0	0
smtp	0	1.4	0	0	14.2	0	*84.4*	0	0	0
ssh	0	1.54	0	0	0	0	1.54	*96.92*	0	0
https	3.2	0	15	0	0	0	0	0	*81.8*	0
pop3s	2	0	0.4	0	0	0	0	0	7.8	*89.8*

4.5 Behavioral Flow recognition

Up to now we described the unsupervised learning step of the methodology presented in Fig. 4.4. We will now analyze the second phase that is the recognition phase. In this phase, we use the results of clustering to derive a set of matched filter able to detect the applicative label of a flow using only the discretized packet size and the packet direction. We assume that the recognition have to be done based on only a small number of initial packets of a flow. Previous analysis and in particular Fig. 4.13 showed that a single application might exhibit different behaviors, and that a particular behavior is not specific to an application. This lead us to develop a recognition based on a mixture of HMM. This recognition mechanism is described in next section and it will be evaluated in section 4.5.2.

4.5.1 Mixtures of HMMs

As seen previously application are characterized by the fact that they might different behaviors. For dealing we this point we have modeled each application through a mixture of the representative HMMs of all clusters where this application have been observed. For a given application app, let α_i^{app}, be the proportion of flows of this application that are in cluster i. The mixture HMM for application app, denoted H^{app} is a HMM of the same structure as the the HMM for each cluster but with an emission probability at each state that is a mixture of emission probability of all cluster representative HMMs, *i.e.* $b_{ij}^{app} = \sum_{k=1}^{K} \alpha_k^{app} b_i^{*k} j$, where $b_i^{*k} j$ is the emission probability of symbol j in state i for the cluster representative of cluster k, H_k^* .

By using this mixture model, the likelihood that a sequence f is generated by a specific mixture can be easily derived as a mixture of the likelihood of the individual clusters.

$$\mathbb{P}\text{rob}\{f|H^{app}\} = \sum_{k=1}^{K} \alpha_k^{app} \mathbb{P}\text{rob}\{f|H_k^*\}$$

The assignment of a particular flow to an applicative mixture is done through a simple Maximum *a posteriori* rule.

In practice, the likelihood of a given sequence have to be obtained toward each one of the K clusters and these likelihood are mixed to derive the likelihood that this sequence belongs to a particular application. As the derivation of the likelihood that a sequence belongs to a particular cluster is obtained through a forward-backward linear filtering operation the complexity involved into this operation is mainly governed by the number of needed clusters which was in our case 55. The filtering structure of the likelihood derivation make it suitable for an implementation inside a DSP like architecture. We will discuss this point in the perspectives part of conclusion.

4.5.2 Classification results

The mixtures of HMMs can be used to determine whether a flow should be labeled with an application. In order to classify a flow, we first build the corresponding sequence of quantized packets and then compute the probability that it belongs to each mixture. In order to compute this probability we need first to evaluate the log-likelihoods that the sequence was generated by each HMM present in the mixtures. The final probability that the considered flow belongs to a mixture is the weighted sum of these log-likelihoods. In the end, a flow is labeled with the application corresponding to the mixture for which the computed probability is the highest.

In order to evaluate the quality of our classification method we use a test set consisting in 5000 flows and check whether the application label found using our mixtures is right or not. Table 4.3 show the quality of our classification method. Real applications appear on the lines while applications found using our method appear in columns. Each cell represents the proportion of actual applications flows labeled with each possible application. This first line thus reads : among edonkey flows, 84.2% were classified edonkey, 4.2% ftp, 2.2% smtp, 5.6% https and 2.8% pop3s.

A few conclusions can be derived from these results. First of all the overall classification performs quite well. If we take into account the relative weight of each application in the full initial trace, more than 89% of the flows would have been classified correctly. Besides, Peer To Peer applications which are known to be difficult to classify are labeled correctly 84.2% of the time for edonkey and 95% of the time for kazaa.

Another phenomenon needs to be explain : POP3 flows are labeled 100% of the time. This issue can be looked at from two different perspectives. From a networking point of view, the first seven packets of POP3 flows are identification packets, which once the quantization step done look a lot like smtp and nntp first packets (in term of size and direction). This accounts for the confusion between the three applications. From a classification point of view, the reason why all POP3 flows fall either in the NNTP mixture or in the SMTP one (and never the other way around) can be explained by looking at figure 4.13. If a POP3 flows behaves exactly like the HMM describing cluster1, the computed probabilty of this flow being a POP3 one is going to be the log-likelihood multiplied by 86% (since 86% of the POP3 traffic belongs to cluster 1, which is going to affect our mixture composition accordingly). At the same time, the probability of this flow belonging to the NNTP mixture is going to be the same log-likelihood multiplied by 100%, since the mixture for NNTP consists only in the HMM describing cluster 1. Thus the NNTP probability is always going to be higher. The same could be said for SMTP if we consider clusters 13 and 14.

In the end our classification shows some limitation when quantized information about the first seven packets can not differentiate between two behaviors. However with a classification right more than 90% of the time except for POP3, it performs fairly well.

4.6 Conclusion

We presented in this paper a Blind applicative flow recognition through behavioral classification. The approach was based on very simple sequences of quantified packet size and packet direction. These sequences were clustered through a powerful spectral clustering algorithm. We developed thereafter a recognition algorithm based on a mixture of HMM representative of the obtained clusters. The presented method appear to be very powerful as it reach recognition performance of 90% with only observing seven packets of a flow !.

This work is a first step toward an operational flow recognition system that will be robust toward flow morphing (tunelling flow in other protocol) and payload encryption. It indeed remains some future to be done. We list here some of the extension we work on.

- We have used in this work only seven packet to recognize an application and we got good recognition rate. However, this number of packet might be too less or too much for specific applications. We are working on extension of this method to variable horizon recognition mechanism that will adapt the number of needed packet to the suspected type of applications.
- We aimed toward a real-time running of the recognition algorithm on a Gigabit/sec link. For this purpose we are investigating how to implement the mechanism on a DSP or better on a Network processor architecture. The filter based structure of the algorithm should be very helpful in this context.
- In meanwhile we are developing hybrid recognition mechanism that will mix the blind recognition mechanism with the full packet payload system.The idea is to make use of the 99 % recognition rate of HTTP traffic of the proposed blind algorithm to reduce the load of a full packet payload analyzer but not forwarding these flow to the analyzer. As 90% of flows are HTTP, this will lead to a very important leverage of the full packet analyzer enabling it to cross the Gigabit/sec barrier.

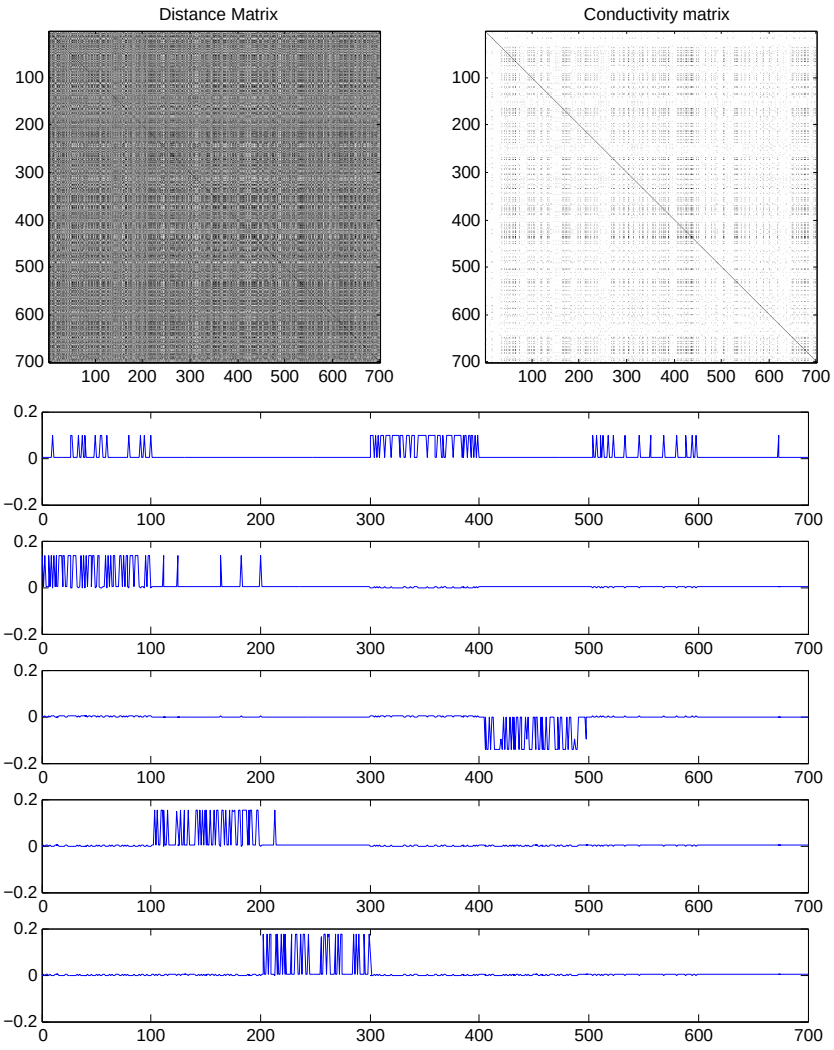


FIG. 4.11 – Real distance and conductivity matrix coming from the evaluation learning set followed by the first five eigenvectors

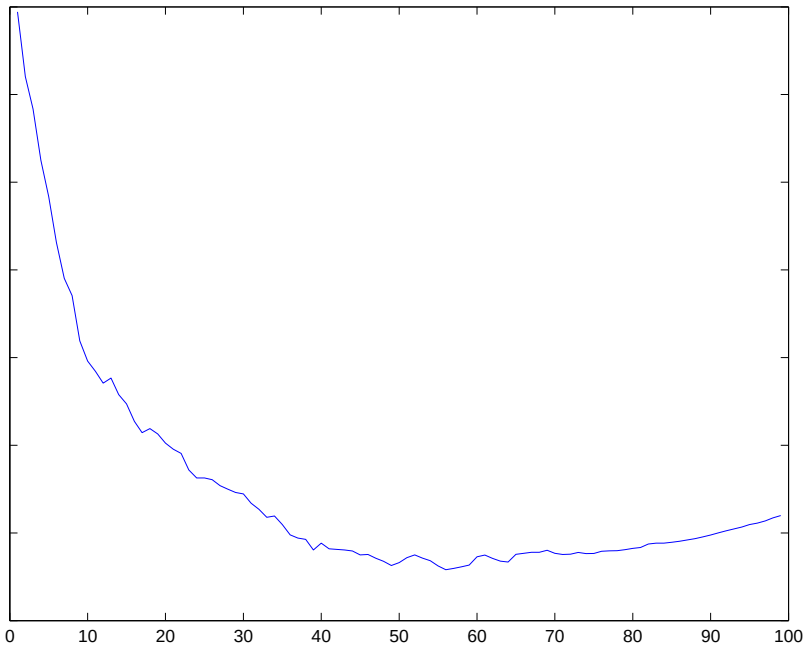


FIG. 4.12 – Description length as a function of number of clusters K

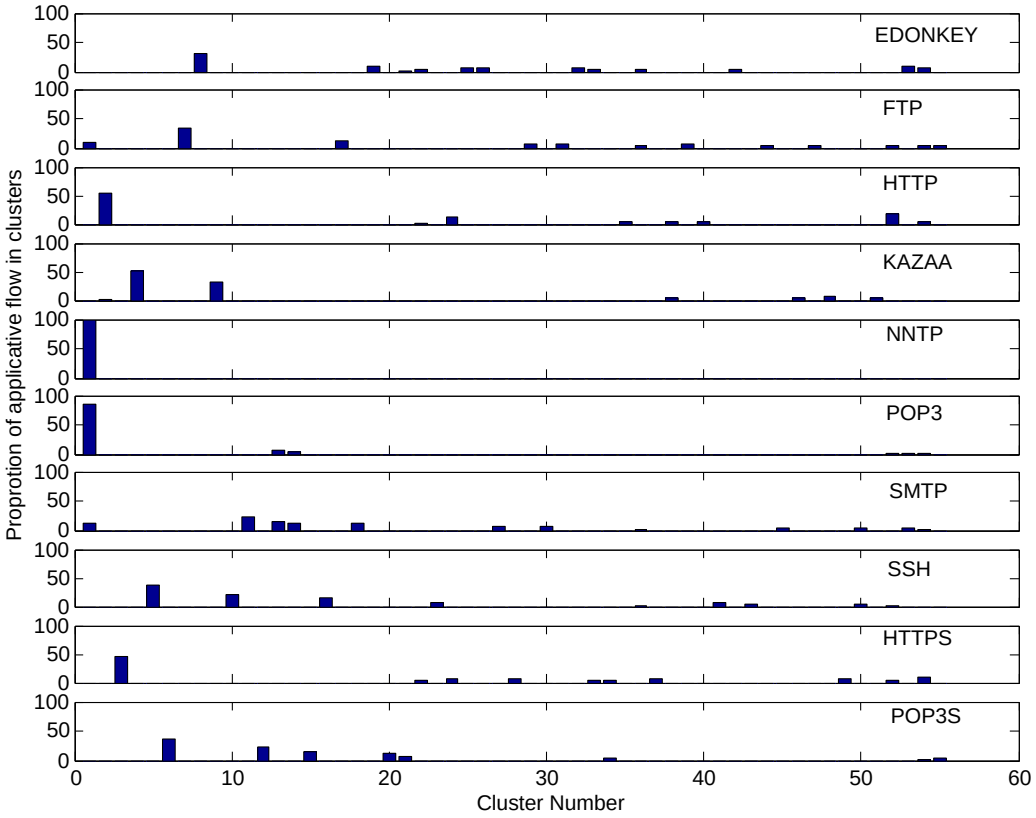


FIG. 4.13 – Distribution of different behavior among the 1O type of application used in the training set

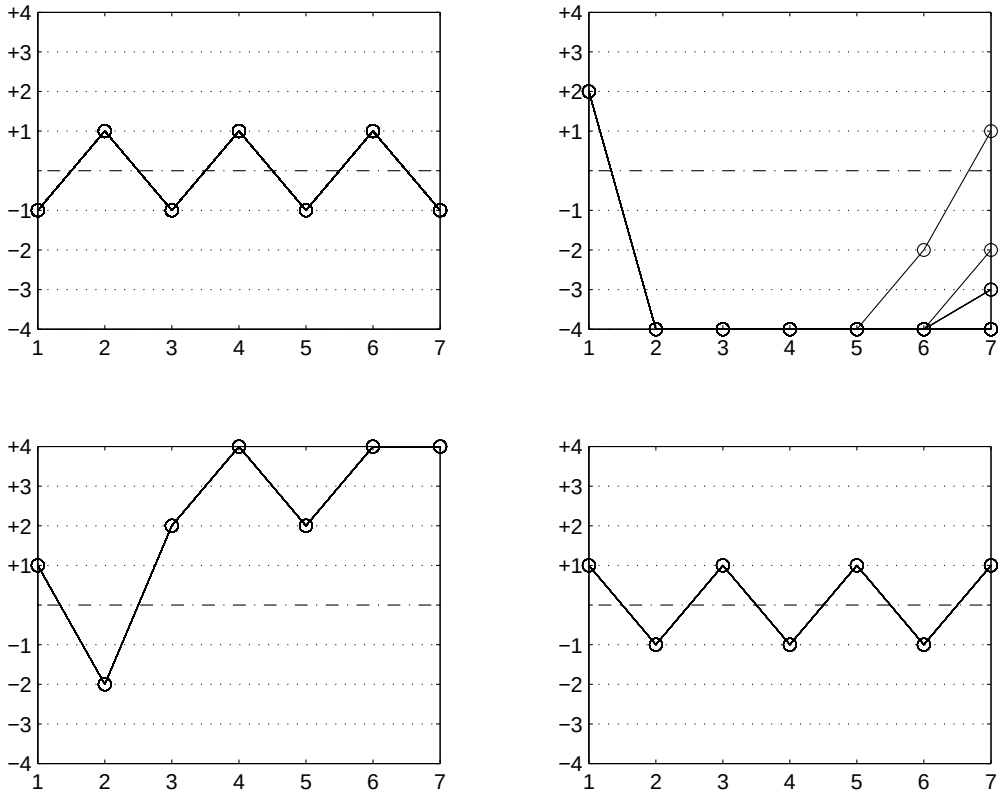


FIG. 4.14 – Packets Sequence for cluster 1,2,3,8

Bibliographie

[1] FT/BD.FTRD/DMI/ISE/02.124, “Performances de TCP en cas de surcharges”, D. Collange,09/02

[2] FT/NT/8555, ”Performance of TCP in overload”, P.Brown, D.Collange

[3] “Grenouille.com”, http ://www.grenouille.com/

[4] “Performance modeling of elastic traffic in overload”, T. Bonald, J. Roberts

[5] BitTorrent. www.bittorrent.com.

[6] eDonkey. www.edonkey2000.com.

[7] WinMX. www.winmx.com.

[8] E. Adar and B. Huberman. Free riding on Gnutella. *First Monday*, 5(10), October 2000.

[9] A. Asvanund, K. Clay, R. Krishnan, and M. Smith. An Empirical Analysis of Network Externalities in P2P Music-Sharing Networks. In *The 23rd Annual International Conference on InformationSystems (ICIS'02)*, Barcelona, Spain, December 15–19 2002.

[10] N. Ben Azzouna and F. Guillemin. Analysis of ADSL traffic on an IP backbone link. In *Proceedings of Globecom 2003*, San Fransisco, CA, USA, December 2003.

[11] R. Bhagwan, S. Savage, and G. Voelker. Understanding Availability. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*, Berkeley, CA, USA, February 2003.

[12] J. Chu, K. Labonte, and B. Levine. Availability and Locality Measurements of Peer-to-Peer File Systems. In *Proceedings of ITCom : Scalability and Traffic Control in IP Networks*, July 2002.

[13] Thomas Karagiannis, Andre Broido, Nevil Brownlee, K. C. Claffy, and Michalis Faloutsos. Is P2P dying or just hiding ? In *IEEE Globecom 2004 - Global Internet and Next Generation Networks*, Dallas, Texas, USA, December 2004.

[14] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and K. C. Claffy. Transport Layer Identification of P2P Traffic. In *Internet Measurement Conference (IMC)*, Taormina, Sicily, Italy, October 2004.

[15] Sandvine Corporation. Regional characteristics of P2P, October 2003. www.sandvine.com.

[16] Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang. Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. In *13th International World Wide Web Conference*, New York City, 17–22 May 2004.

[17] Subhabrata Sen and J. Wang. Analyzing Peer-to-Peer Traffic Across Large Networks. *IEEE/ACM Transactions on Networking*, 2004.

[18] A.Barron, J.Rissanen, and B.Yu. The minimum description length principle in coding and modeling. . *IEEE Trans. Information Theory*, 44, 1998.

[19] C.Trivedi, H. Trussell, A. Nilsson, and M.-Y. Chow. Implicit traffic classification for service differentiation. In *15th ITC Specialist Seminar, Internet Traffic Engineering and Traffic Management*, 2002.

[20] I. Fischer and J. Poland. New methods for spectral clustering. June 2004.

[21] H.Binsztok, T. Artières, and P. Gallinari. A model-based approach to sequence clustering. In *ECAI*, Madrid, 2004.

[22] F. Hernandez-Campos, A. Nobel, F. Smith, and K. Jeffay. Statistical clustering of internet communication patterns. In *Proceedings of the 35th Symposium on the Interface of Computing Science and Statistics*, 2003.

[23] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone. In *IEEE Infocom*, San Francisco, 2003.

[24] T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos. Is p2p dying or just hiding ? In *Globecom*, 2004.

[25] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering : analysis and an algorithm. In *NIPS*, 2001.

[26] A. Oveissian, K. Salamatian, A. Soule, and N. Taft. Fast flow classification over internet. In *CNSR*, 2004.

[27] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot. A pragmatic definition of elephants in internet backbone traffic. In *Internet Measurement Workshop*, 2002.

[28] F. Porikli. Trajectory distance metric using hidden markov model based representation. In *IEEE European Conference on Computer Vision, PETS Workshop*, 2004.

[29] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, 1989.

[30] P. Smyth. Clustering sequences with hidden markov models. In *Advances in Neural Information Processing*, 1997.

[31] A. Soule, K. Salamatian, R. Emilion, N. Taft, and K. Papagiannaki. Flow classification by histograms or how to go on safari in the internet. In *ACM Sigmetrics*, 2004.

Deuxième partie

Sous Projet 4

Méthodologie et Echantillonnage

Chapitre 5

Rapport technique de la troisième année, Introduction

5.1 Introduction et Administrativia

Cette partie technique présente les travaux réalisés dans le contexte du SP4, "Echantillonnage et Mesures Actives" du projet RNRT METROPOLIS, correspondant à la dernière période du projet.

5.1.1 L'échantillonnage temporel passif

L'échantillonnage temporel passif déterministe $1/N$ est analysé dans le chapitre 6. Ce type d'échantillonnage consiste simplement à capturer un paquet tous les N paquets passant sur le lien observé. Ainsi, seule une fraction $1/N$ du trafic global en termes de paquets est analysée. Les auteurs étudient l'information obtenue en effectuant un échantillonnage en $1/N$ déterministe sur du trafic ADSL et en particulier du trafic TCP transitant sur un lien du backbone IP de France Telecom (RBCI).

Ce type d'échantillonnage est utilisé dans la pratique pour réduire la quantité d'information à examiner, les vitesses des liens de transmission étant tellement élevées à l'heure actuelle que l'observation du trafic sur un lien pendant quelques minutes se traduit rapidement par une masse d'informations très importante, qui demande des capacités de stockage et des temps d'analyse prohibitifs. L'échantillonnage en $1/N$ permet de réduire notablement les quantités d'information. En général, N varie de quelques centaines à quelques milliers ; dans l'étude menée ici, N a été fixé à 1000, ce qui permet de passer du Giga au Mégaoctet d'information à analyser. Le procédé d'échantillonnage en $1/N$ déterministe est notamment utilisé par les sondes NetFlow embarquées dans les routeurs, qui servent, en autres, à facturer les ISP. Ceci permet aux routeurs de réduire le nombre des rapports d'observation (NetFlow records) qui sont envoyés aux collecteurs de données.

5.1.2 La Plateforme de mesure active Saturnev6

Ensuite, dans le chapitre 8, les évolutions de la plate-forme Saturnev6 sont présentées, notamment, l'évolution de l'outil de métrologie active Saturne vers IPv6 et vers la mesure passive. IPv6 permet de placer l'estampille entre l'en-tête IPv6 et la partie donnée du paquet, dans une extension. La définition d'une extension IPv6 pour la métrologie permet à l'outil de devenir plus proche des méthodes passives. Ainsi, en ajoutant quelques octets au paquet de façon totalement transparente, on peut estampiller un paquet applicatifs de façon passive. Ce chapitre présente les principes des extensions d'IPv6, l'extension de métrologie définie et mise en œuvre dans le cadre de l'outil Saturne ainsi que les premiers tests d'exploitation.

5.1.3 Techniques d'Estimation de Bande Passante : Etat de l'Art

Un état de l'art autour des mécanismes d'estimation de bande passante est donné au chapitre 9. Plus d'une trentaine de techniques d'estimation de bande passante (en concret, des métriques comme la bande passante nominale, bande passante disponible, throughtput) one été analysées. La classification traditionnelle en techniques actives et passives a été étendue en prenant en compte les techniques dites "hybrides", et une taxonimie plus raffinée est proposée.

5.1.4 Evaluation des métodes d'estimation de bande passante dans les environements réalistes

Finalement, dans le chapitre 10[1], et en relation avec la section précédente, un certain nombre de techniques utilisées pour la mesure et estimation de bande passante ont été évaluées. Les expérimentations réalisées et les résultats obtenus mettent en cause l'adéquation de ces techniques et évaluent les répercussions de la mesure en les utilisant.

[1] ce chapitre est rédigé en anglais

Chapitre 6

Etude de l'échantillonnage en 1/N déterministe sur du trafic ADSL

[1]

6.1 Introduction

6.1.1 Positionnement du problème

Dans ce rapport, nous étudions l'information obtenue en effectuant un échantillonnage en $1/N$ déterministe sur du trafic ADSL et en particulier du trafic TCP transitant sur un lien du backbone IP de France Telecom (RBCI). Ce type d'échantillonnage consiste simplement à capturer un paquet tous les N paquets passant sur le lien observé. Ainsi, seule une fraction $1/N$ du trafic global en termes de paquets est analysée.

Ce type d'échantillonnage est utilisé dans la pratique pour réduire la quantité d'information à examiner, les vitesses des liens de transmission étant tellement élevées à l'heure actuelle que l'observation du trafic sur un lien pendant quelques minutes se traduit rapidement par une masse d'informations très importante, qui demande des capacités de stockage et des temps d'analyse prohibitifs. L'échantillonnage en $1/N$ permet de réduire notablement les quantités d'information. En général, N varie de quelques centaines à quelques milliers ; dans l'étude menée ici, N a été fixé à 1000, ce qui permet de passer du Giga au Mégaoctet d'information à analyser. Le procédé d'échantillonnage en $1/N$ déterministe est notamment utilisé par les sondes NetFlow embarquées dans les routeurs, qui servent, en autres, à facturer les ISP. Ceci permet aux routeurs de réduire le nombre des rapports d'observation (NetFlow records) qui sont envoyés aux collecteurs de données.

Prélever un paquet tous les N se traduit nécessairement pas une perte d'information. L'objet de cette étude est précisément d'évaluer la perte d'information provoquée par cet échantillonnage. On s'intéresse principalement à répondre aux questions suivantes :

[1] par Stéphanie Poisson et Fabrice Guillemin

1. Est-il possible de reconstruire les caractéristiques statistiques du débit sur un lien à partir du trafic échantillonné ?
2. Est-il possible de capturer des phénomènes sporadiques (par exemple, une avalanche de SYN vers une destination donnée dans le cas d'une attaque DoS) ?
3. En termes de volumes transmis, dans quelle mesure les volumes de données observés reflètent-ils les volumes transitant réellement sur le lien ?

Ces questions sont capitales pour juger de l'efficacité de l'échantillonnage NetFlow dans l'optique sécurité : il faut que les phénomènes d'avalanche, de messages SYN par exemple, puissent être détectés par NetFlow.

Pour donner des éléments de réponse aux questions précédentes, nous considérons des traces de trafic capturées en Octobre 2003 entre 21h et 23h sur lien à 1 Gbit/s desservant plusieurs plaques ADSL ; seul le trafic TCP descendant (i.e. en provenance du RBCI) est observé. La plage horaire retenue correspond habituellement à une activité soutenue de la part des utilisateurs.

6.1.2 Un modèle de référence pour le trafic ADSL

Pour évaluer la perte d'information due à l'échantillonnage en termes statistiques, il est essentiel de disposer d'un modèle de référence. Pour le trafic ADSL transitant sur un lien du RBCI ce modèle de référence a été mis au point dans le cadre de la thèse de Nadia Ben Azzouna [**?**, **?**].

Pour caractériser le trafic, on adopte une description par flots avec les définitions suivantes pour les souris et les éléphants, en se rappelant qu'un flot est un ensemble de paquets IP avec les mêmes adresses IP source et destination, les même numéros de port et le même protocole (TCP en l'occurrence).

Définition 1 (Souris). *Une souris correspond à un flot TCP comprenant un nombre de paquets inférieur ou égal à 20 ; une souris est considérée comme terminée lorsque aucun paquet du flot n'a été observé pendant une période de 5 secondes.*

Définition 2 (Eléphant). *Un éléphant est composé d'un flot de plus de 20 paquets.*

Les définitions ci-dessus peuvent paraître sommaires, mais sont amplement suffisantes pour décrire le débit sur un lien du RBCI. Le fait le plus marquant pour le trafic observé est la prépondérance du trafic peer-to-peer (p2p). Près de 80% du trafic sont engendrés par des applications p2p. Par ailleurs, les souris qui représentent environ 95% des flots n'engendrent que 6% du trafic global.

Pour décrire le débit des souris, on est amené à agréger les souris suivant certains critères (cf. le rapport [**?**] pour les détails). Cette agrégation donnent naissance à des macro-souris regroupant plusieurs souris arrivant proches les unes des autres et partageant certaines caractéristiques. Pour que le procédé d'agrégration soit efficace, il est nécessaire de distinguer les souris p2p et non p2p, ce qui conduit à introduire les macro-souris p2p et non p2p. Certains paquets de souris ne peuvent pas être agrégés aux macro-souris et créent un débit assimilable à un bruit blanc.

En ce qui concerne les éléphants, il faut d'abord distinguer les éléphants essentiellement composés de messages d'acquittement de ceux qui transportent de l'information. Les premiers sont composés de paquets de petite taille, en général inférieure à 80 octets. Ces éléphants sont engendrés par des terminaux rapatriant des données d'un serveur. Etant donné que nous observons dans cette étude le

trafic descendant, ceci indique que des terminaux de clients ADSL jouent le rôle de serveurs. Cette situation est de plus en plus fréquente avec l'émergence des protocoles p2p, qui favorisent un usage symétrique du réseau, le paradigme pair à pair remplaçant peu à peu celui du client/serveur. On ne constate cependant pas encore une symétrisation des flux en termes de débit à cause des limitations des débits remontants des lignes ADSL.

Pour distinguer les éléphants essentiellement composés de messages d'acquittement, appelés éléphants ACK, des autres éléphants, on fixe un seuil pour la taille moyenne des paquets d'un éléphant : si celle-ci est inférieure à 80 octets, alors l'éléphant est considéré comme éléphant ACK. Ce type d'éléphant donne naissance à un bruit blanc non centré, de faible moyenne (en général, il y a deux ordres de grandeur entre le débit moyen des éléphants ACK et les autres).

En ce qui concerne les éléphants qui transportent des données, le point majeur à noter est que les flots correspondants (flots longs) ne sont pas toujours actifs, mais sont plutôt composés de rafales espacées de période de moindre activité. Ceci conduit à découper les éléphants en mini-éléphants (correspondant aux rafales) et souris (périodes de moindre activité). Par ailleurs, on constate empiriquement que le débit des gros mini-éléphants peut être très bien approximé par une constante, de l'ordre de 30 Kbit/s. Ce phénomène est typiquement relié aux protocoles p2p. La propriété d'indépendance du débit aux longues durées est également partagée par les macro-souris, qu'elles soient p2p ou non.

Le trafic global est ainsi composé de plusieurs types d'objets : macro-souris p2p et non p2p, mini-éléphants et souris d'éléphants. Le point important du point de vue de la modélisation est que ces différents objets arrivent suivant des processus de Poisson et que leurs durées peuvent être approchées par des lois de Weibull à deux paramètres. Cette propriété pour les mini-éléphants est capitale et explique l'absence de dépendance à long terme dans le trafic ADSL (cf. l'article [**?**] pour plus de détails).

En supposant que les arrivées de paquets donnent naissance à un bruit blanc, le débit global sur le lien, évalué sur des intervalles de $\Delta = 100$ millisecondes et représenté par la série chronologique $\{X_n\}$ supposée stationnaire au second ordre, peut se décomposer comme

$$X_n = \sum_{i=1}^{4} \Lambda_n^i + \sigma\varepsilon_n + m,$$

où σ est l'amplitude du bruit, $\{\varepsilon_n\}$ est un bruit blanc standard, m est une constante et les processus $\{\Lambda_n^i\}$, $i = 1, ..., 4$ représentent les débits fluides des macro-souris p2p, des macro-souris non p2p, des souris d'éléphants et des mini-éléphants.

Tous les processus ci-dessus peuvent être considérés comme des signaux que l'on caractérise à l'aide de leur spectre de puissance (ou densité spectrale). Le point remarquable est que le spectre des mini-éléphants est dominant dans les basses fréquence et celui des souris non p2p l'est dans les hautes fréquences.

Les caractéristiques des différents flots sont données par le tableau 6.1. Les paramètres η et β font référence à la distribution de Weibull à deux paramètres : une variable aléatoire S suit une loi de Weibull à deux paramètres η (facteur d'échelle) et β (facteur de forme) si

$$\mathbb{P}(S > x) = \exp\left(-\left(\frac{x}{\eta}\right)^{\beta}\right).$$

	mini-éléphants	souris (éléphants)	m-souris non p2p	m-souris p2p
taux d'arrivée	$\lambda_h = 40.01\ s^{-1}$	$\lambda_w = 44.349\ s^{-1}$	$\lambda_m = 326.47\ s^{-1}$	$\lambda_\mu = 903.3\ s^{-1}$
β	$\beta_h = 0.399$	$\beta_w = 0.842$	$\beta_m = 0.873$	$\beta_\mu = 1.207$
η	$\eta_h = 64.043$	$\eta_w = 15.568$	$\eta_m = 3.172$	$\eta_\mu = 6.36$
$\mathbb{E}[S]$	$192.95s$	$14.288s$	$3.249s$	$5.695s$
$\mathbb{E}[Y^2 \mid S]$	$\kappa_h = 4e9$	$\kappa_w = 5e7$	$\kappa_m = 3.5e8$	$\kappa_\mu = 1.5e6$
Y	6.32e4	7.07e3	1.87e4	1.22e3
variance σ^2	$\sigma_h^2 = 20.25e12$	$\sigma_w^2 = 4.11e10$	$\sigma_m^2 = 9.6e11$	$\sigma_\mu^2 = 9.35e10$
$\lambda\kappa$	$1.6004e11$	$2.22e9$	$1.14e11$	$1.35e9$

TAB. 6.1 – Caractéristiques des mini-éléphants, des souris (associées aux éléphants) et des souris habituelles pour la trace d'Octobre 2003.

6.2 Caractéristiques du trafic échantillonné

6.2.1 Composition en termes de flots

Dans un premier temps, on remarque que l'échantillonnage préserve bien les quantités macroscopiques d'information et la durée moyenne entre deux paquets consécutifs, comme le montre le tableau 6.2 obtenu pour $N = 1000$. Nous remarquons qu'il y a un rapport 1000, soit N, entre la trace réelle et la trace échantillonnée pour ces trois critères.

	trace réelle	trace échantillonnée
nombre de paquets au total	271 455 718	272 746
temps moyen entre 2 paquets	0.053 ms	52.7 ms
nombre d'octets au total	156 080 024 149	156 627 654

TAB. 6.2 – Effets macroscopiques de l'échantillonnage pour $N = 1000$.

On s'intéresse à présent à la composition du trafic échantillonné en termes de flots. Etant donné que le nombre de paquets dans une souris et même dans une macro-souris est faible (quelques dizaines de paquets), on peut s'attendre à ce que, quand une souris est vue par échantillonnage, elle ne le soit qu'une seule fois. De plus, voir une souris est équivalent à voir une macro-souris. En première approximation, si on néglige les éléphants ACK qui ont un débit relativement faible, la probabilité de voir une souris peut être évaluée par

$$\mathbb{E}\left[\frac{N_s}{N_s + N_e}\right] = \frac{\lambda_s \mathbb{E}[S_s]}{\lambda_s \mathbb{E}[S_s] + \lambda_e \mathbb{E}[S_e]}$$

où N_s et N_e sont respectivement le nombre de macro-souris (p2p et non p2p) et le nombre d'éléphants actifs dans l'état stationnaire et les paramètres λ_s, λ_e, $\mathbb{E}[S_s]$, $\mathbb{E}[S_e]$, Y_s et Y_e sont les taux d'arrivée, les durées moyennes et les débits des éléphants et des souris donnés par le tableau 6.1.

Le tableau 6.3 donne le pourcentage en nombre de flots et en volume de données des flots vus une seule fois et les compare avec le trafic réel des souris.

Nous remarquons d'emblée que les flots vus une seule fois ne peuvent pas être assimilés aux souris, ni en volume ni en pourcentage. Ce phénomène est dû au fait que les éléphants de petite taille (ils existent et sont assez nombreux) ne sont vus qu'une seule fois. Cette constatation est importante à prendre en compte pour le reste de l'analyse : l'échantillonnage a tendance à casser la structure de corrélation des flots ; seuls les flots assez longs qui ont une bonne chance d'être vus plusieurs fois apparaissent comme des éléphants dans la trace échantillonnée. En d'autres termes, l'échantillonnage a tendance à blanchir le signal dans les fréquence intermédiaires ; seules les basses fréquences sont épargnées. Ce phénomène sera étudié en détails dans la suite.

	souris dans la trace réelle	flots vus une seule fois dans la trace échantillonnée
volume	6.68%	23.617%
nombre de flots	96.24%	80.957%

TAB. 6.3 – Comparaison entre les flots vus une seule fois et les souris ($N = 1000$).

Pour corriger l'estimation des souris, on considère les flots vus une seule fois et ayant moins de 1000 octets en partant du principe qu'une bonne partie des souris contient des paquets de petite taille et que les éléphants sont formés de paquets de grande taille (typiquement 1500 octets). Avec cette définition, les résultats obtenus sont donnés par le tableau 6.4. Avec ce nouveau principe de ségrégation, le volume des souris est mieux estimé mais ce qui est gagné en volume est perdu en nombre.

	souris dans la trace réelle	flots vus une seule fois dans la trace échantillonnée
volume	6.68%	5.97%
nombre de flots	96.24%	68.95%

TAB. 6.4 – Comparaison entre les flots vus une seule fois avec moins de 1000 octets et les souris ($N = 1000$).

Une dernière tentative consiste à enlever les éléphants ACK comme dans le cas du modèle pour la trace entière. Dans la trace échantillonnée, un éléphant ACK est un flot vu plus d'une fois et dont le volume moyen par paquet est inférieur à 80 octets. Les changements dans les estimations sont donnés par le tableau 6.5. Cette prise en considération des éléphants ACK n'apporte pas de changement significatif et ne sera donc pas conservée dans la suite.

	souris dans la trace réelle	flots de moins de 1000 octets vus une seule fois dans la trace échantillonnée sans les éléphants ACK
volume	6.68%	6.03%
nombre de flots	96.24%	73.52%

TAB. 6.5 – Comparaison entre les flots vus une seule fois avec moins de 1000 octets et les souris quand les éléphants ACK sont retirés ($N = 1000$).

Pour confirmer les phénomènes observés ci-dessus, le taux d'échantillonnage a été changé en $N = 100$ et $N = 10000$. Les résultats pour la conservation des paramètres macroscopiques sont donnés par le tableau 6.6.

	trace réelle	trace échantillonnée
$N = 100$		
nombre de paquets au total	271 455 718	2 727 453
temps moyen entre 2 paquets	0.053 ms	5.4ms
nombre d'octets au total	156 080 024 149	1 564 126 848
$N = 10000$		
nombre de paquets au total	271 455 718	27 275
temps moyen entre 2 paquets	0.053 ms	526ms
nombre d'octets au total	156 080 024 149	15 588 830

TAB. 6.6 – Préservation des caractéristiques macroscopiques quand le taux d'échantillonnage change.

Le tableau 6.7 montre que même pour un taux d'échantillonnage élevé ($N = 100$) les flots vus une seule fois ne peuvent pas être assimilés facilement aux souris. Cette situation est encore plus accentuée quand N devient très grand, par exemple $N = 10000$. On constate dans ce cas qu'un grand nombre d'éléphants passent dans la catégorie des flots vus une seule fois. Ceci correspond à un blanchiment du signal.

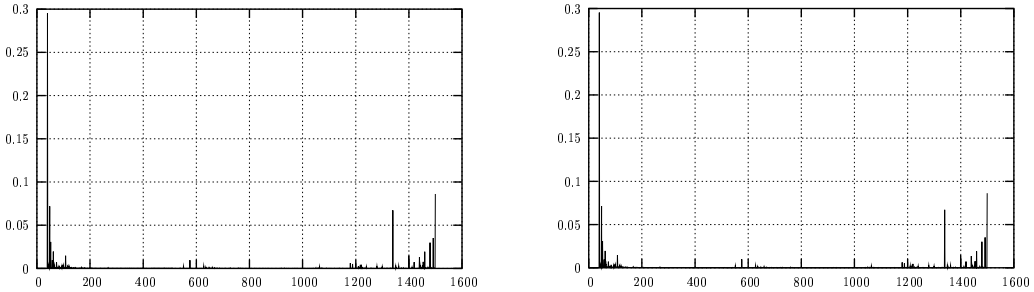
flots vus une seule fois	trace réelle	trace échantillonnée $N = 100$	trace échantillonnée $N = 10000$
volume	6.68%	10.426%	64.891%
nombre	96.24%	87.343%	90.0.95%
volume avec moins de 1000 octets	6.68%	4.507%	9.432%
nombre avec moins de 1000 octets	96.24%	81.689%	63.939%
volume avec moins de 1000 octets sans les éléphants ACK	6.68%	4.568%	9.466%
nombre avec moins de 1000 octet sans les éléphants ACK	96.24%	86.262%	65.433%

TAB. 6.7 – Comparaison entre les flots vus une seule fois et les souris réelles.

6.2.2 Comparaison des volumes.

Nous avons une moyenne de 575 octets par paquet pour la trace entière et une de 581 octets par paquet pour la trace échantillonnée avec $N = 1000$, une de 573 octets par paquet pour la trace échantillonnée avec $N = 100$ et une de 571 octets par paquet pour la trace échantillonnée avec N=10 000. Ainsi, le volume moyen par paquet estimé est proche du vrai volume moyen et cela pour les trois valeurs de N qui ont été prises.

La figure 6.1(a) représente la densité de probabilité de la taille des paquets en octets pour la trace entière. La densité correspondante pour la trace échantillonnée est illustrée par la figure 6.1(b).



(a) Trace entière (b) Trace échantillonnée

FIG. 6.1 – Densité de probabilité de la taille des paquets en octets.

Comme nous le voyons, ces deux graphiques sont très proches l'un de l'autre. Nous en déduisons que la répartition de la taille des paquets est conservée entre la trace entière et la trace échantillonnée. Nous nous intéressons à présent au nombre d'éléphants dont le volume total est dans une certaine fourchette, ceci pour la trace entière et la trace échantillonnée (cf. les figures 6.2- 6.3). La trace entière a un premier pic à 18 000 octets (premier graphique), puis un second à 48 000 octets (second graphique), enfin la probabilité que la taille des éléphants soit supérieure à 95000 octets reste inférieure à 7.10^{-5}. Pour la trace échantillonnée, la densité de probabilité de la taille des éléphants a un premier pic à 80 octets (premier graphique), puis un second à 21 000 octets (second graphique), enfin à partir de 55 000 octets la courbe reste inférieure à 7.10^{-5}.

On déduit de ces figures qu'il n'y a pas une relation évidente entre les volumes observés sur la trace échantillonnée et ceux de la trace entière. En particulier, il n'y a pas une simple relation de proportionnalité par un facteur N entre les volumes échantillonné et réel. Ce point doit être pris en compte dans la facturation où une règle de trois ne suffit pas à déterminer les volumes de trafic qui sont à imputer à des ISP. En réalité, l'échantillonneur est très injuste vis à vis des volumes. Les très gros volumes ont plus de chance de se faire repérer par l'échantillonneur que les volumes plus modestes. Ce phénomène se voit sur les pics significatifs des volumes des éléphants : le pic à 18000 octets est ramené à un pic à 80 octets (seulement un fraction de 80/18000 = 0.4% est repérée par échantillonnage) alors que le pic à 48000 est ramené à 21000 (soit 43%). Si un ISP ne transporte que du Web, les volumes correspondants sont assez faibles et passeront assez inaperçus par échantillonneur, alors qu'un ISP qui transporte du p2p verra son volume sur-évalué.

Pour conclure l'examen des volumes, on s'intéresse à présent à la correspondance entre les volumes des éléphants échantillonnés par rapport aux éléphants initiaux. Les figures 6.4 et 6.5 donnent à partir des éléphants de la trace entière qui ont un volume dans une certaine fourchette, la densité de probabilité du volume des éléphants échantillonnés correspondants. On constate que plus les éléphants sont volumineux, plus les éléphants échantillonnés le sont également. Cependant, les pics à 1500 octets indiquent que les éléphants échantillonnés ne sont vus qu'une seule fois. Seuls un petit nombre d'éléphants sont vus plus d'une fois.

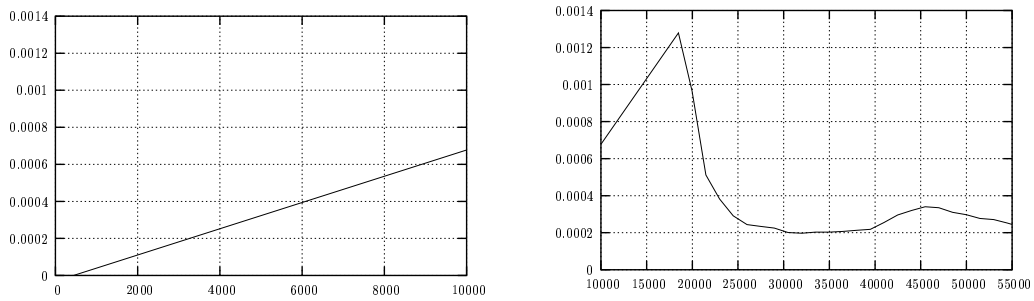


FIG. 6.2 – Densité de probabilité de la taille des éléphants en octets dans la trace entière.

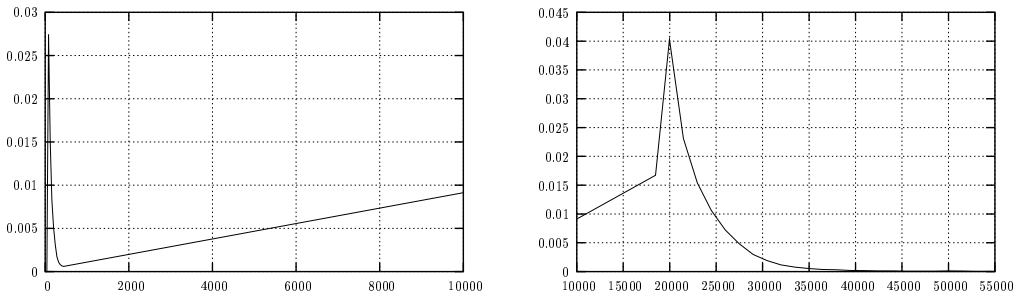


FIG. 6.3 – Densité de probabilité de la taille des éléphants en octets dans la trace échantillonnée.

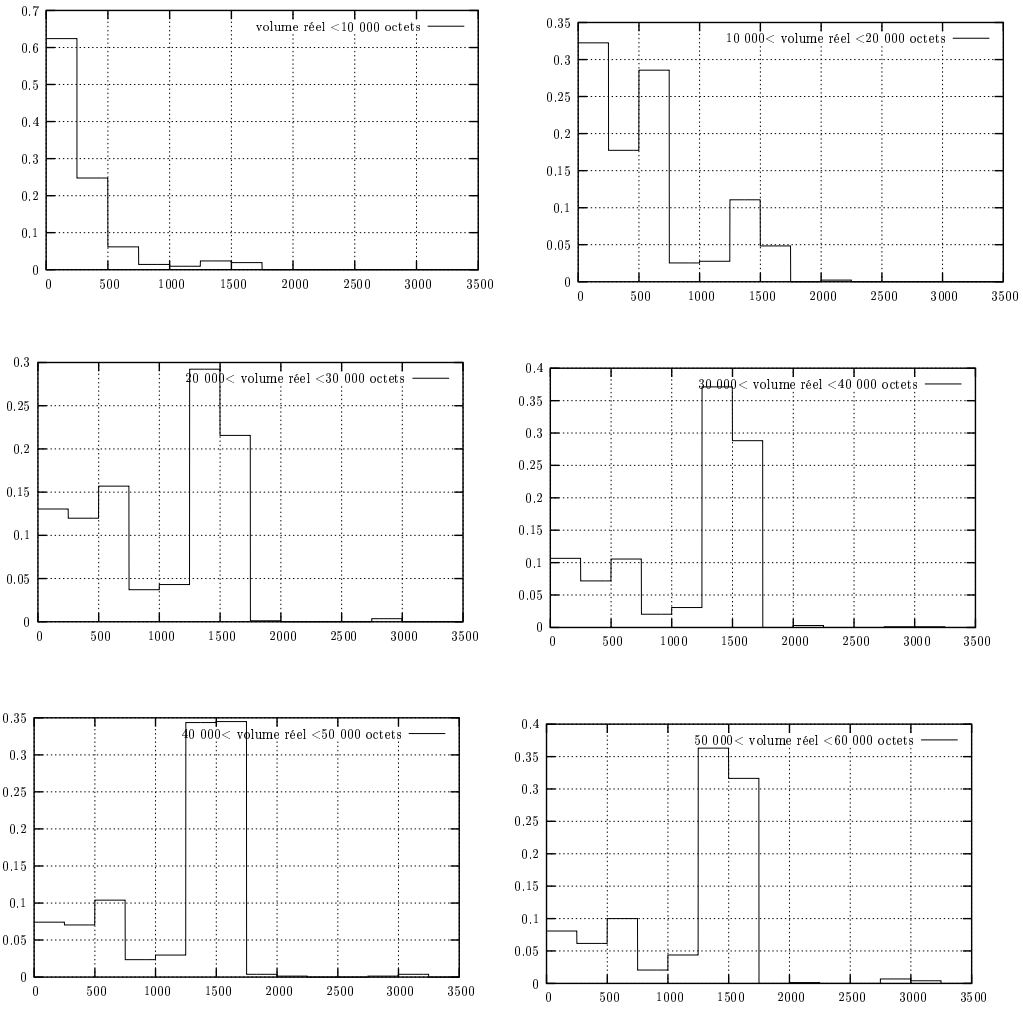


FIG. 6.4 – Correspondance entre les volumes des éléphants échantillonnés et les éléphants initiaux

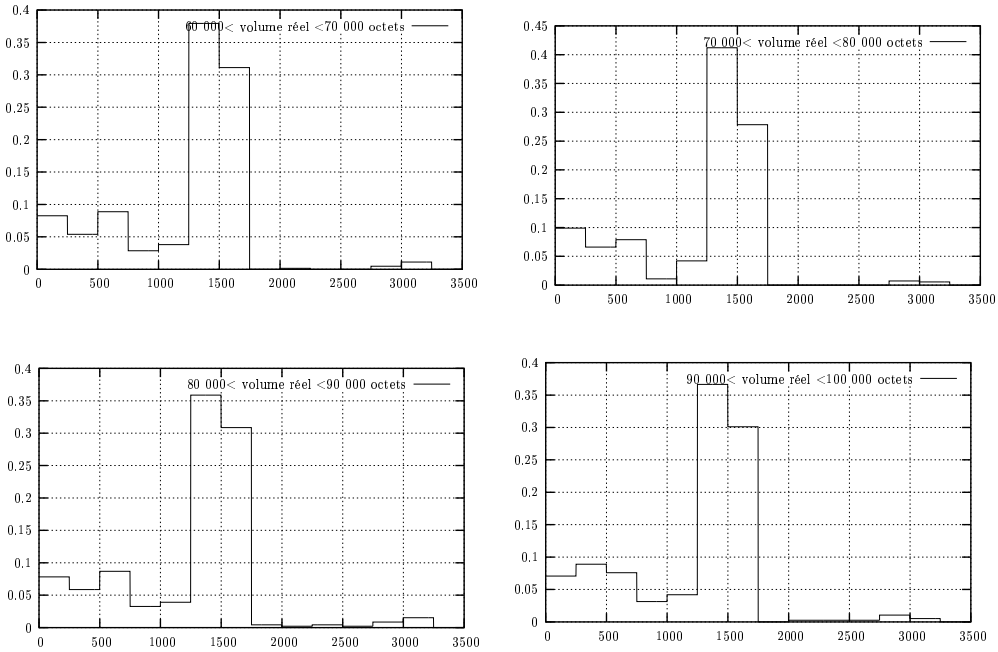


FIG. 6.5 – Correspondance entre les volumes des éléphants échantillonnés et les éléphants initiaux

6.2.3 Nombre de flots actifs

Dans cette section, nous nous intéressons au nombre de flots actifs au cours du temps. Le problème majeur dans le calcul de ce nombre réside dans la délimitation des flots. Le nombre d'éléphants actifs au cours du temps dans la trace entière est illustré par la figure 6.6. On constate sur cette figure que le processus correspondant est fortement non stationnaire. Ce phénomène est dû au fait que les éléphants ont tendance à être longs et que leur date de début ou de fin peuvent se situer en dehors de l'intervalle d'observation (effets de bord) ; ceci explique la montée du nombre d'éléphants actifs en début et de la décroissance en fin de plage d'observation. La figure 6.6 décrit également le nombre de mini-éléphants actifs au cours du temps, qui semble beaucoup plus stable et moins sensible aux effets de bord. Ceci s'explique par le fait que les mini-éléphants sont beaucoup plus petits que les éléphants entiers.

Pour comparer des quantités similaires, on est amené à réduire l'intervalle d'observation afin d'obtenir un processus qui semble stationnaire, en l'occurrence de 1000 à 5000 secondes. On obtient ainsi un processus $\{\nu_n\}$, qui représente le nombre d'éléphants actifs aux instants $n\Delta$ (avec $\Delta = 100$ ms). On calcule ensuite la densité spectrale de ce processus et on la compare avec les densités spectrales des mini-éléphants dans la trace entière et des éléphants dans la trace échantillonnée (cf. la figure 6.8(a)). Dans tous les cas, les éléphants ACK (où estimés comme tels dans la trace échantillonnée) ne sont pas considérés.

A partir de la trace échantillonnée, il est possible de reconstruire la notion d'éléphants, c'est à dire les flots vus plus d'une fois avec une taille de paquet moyenne supérieure à 80 octets. Ces

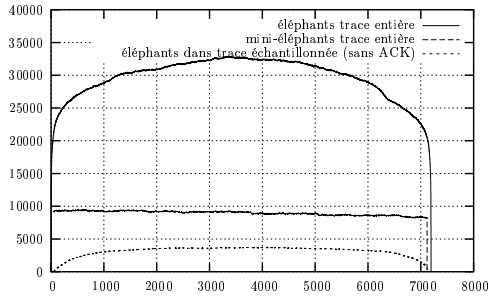


FIG. 6.6 – Nombre de flots actifs au cours du temps.

éléphants “reconstruits” ont une durée Weibull presque exponentielles ($\eta = 120$ et $\beta = 1,05$), comme indiqué par la figure 6.7. Par ailleurs, on peut supposer que ces éléphants échantillonnés arrivent suivant un processus de Poisson.

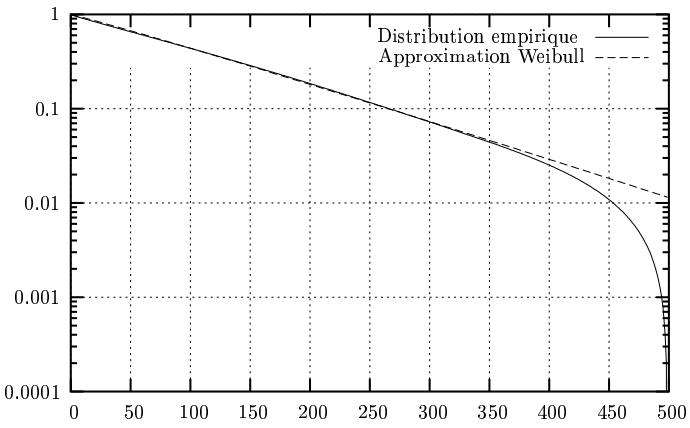


FIG. 6.7 – Durée des éléphants échantillonnés.

Il est important de noter que les éléphants échantillonnés ne peuvent pas être rapprochés facilement des mini-éléphants. En effet, l’apparition d’un éléphant échantillonné est dû au fait qu’un éléphant original est vu plusieurs fois par l’échantillonneur, mais les paquets correspondants peuvent appartenir à plusieurs mini-éléphants. Dans l’expérience rapportée ici, le temps moyen entre deux paquets d’un éléphant échantillonné est d’environ 300 secondes, comparable à la durée moyenne d’un mini-éléphant, et seulement 12% des éléphants originaux sont vus par l’échantillonneur. En quelque sorte, celui-ci casse la structure de corrélation dans le processus initial et recrée un autre processus avec de nouvelles corrélations.

En ce qui concerne les densités spectrales, elles exhibent toutes la même forme, en accord avec la formule théorique en c/x^2 liée au fait que les éléphants échantillonnés et les mini-éléphants ont des durée Weibull et arrivent suivant un processus de Poisson. Ceci se reflète par le fait que les densités

spectrales semblent proportionnelles entre elles ; seul le coefficient c change d'une densité à l'autre. Cependant, il n'y a pas une relation simple entre ces différentes densités spectrales pour plusieurs raisons :

1. seule une partie des éléphants dans la trace entière est vue dans la trace échantillonnée, ceci explique pourquoi le spectre des éléphants dans la trace échantillonnée est inférieur à celui des éléphants dans la trace entière, cette proportion n'est pas simple à déterminer en fonction des paramètres (cf. la figure 6.8(b)) ;
2. les éléphants dans la trace entière contiennent certes plusieurs mini-éléphants, mais certains éléphants sont eux-mêmes composés principalement de souris (éléphants) qui ne sont pas comptabilisées comme mini-éléphants ; ce phénomène se produit de manière assez significative pour que le spectre des éléphants soit inférieur à celui des mini-éléphants dans la trace entière alors que l'on aurait pu s'attendre à l'inverse.

En guise de conclusion pour cette section, on doit noter que l'exploitation pratique du nombre de flots actifs semble très délicate. Cette estimation est biaisée par plusieurs facteurs :

1. le mélange des paquets entre les souris et les éléphants n'est pas homogène, les éléphants soumettent plus de paquets par seconde que les souris ; ceci conjugué avec le fait que les éléphants sont beaucoup plus longs que les souris, les éléphants ont plus de chance d'être vus par échantillonnage, ce qui défavorise d'autant plus la détection des souris ;
2. le périodogramme du nombre de flots actifs a bien la forme prédite par le modèle théorique mais il n'y a pas une règle de conservation simple sur les nombres de flots : quand on prélève 1 paquet sur N, on peut s'attendre à obtenir une fraction $1/N$ du débit global, mais on ne peut pas en dire autant pour les flots. L'information utile dans le périodogramme se situe dans les fréquences intermédiaire, là où l'approximation en c/x^2 peut être utilisée avec une certaine confiance.

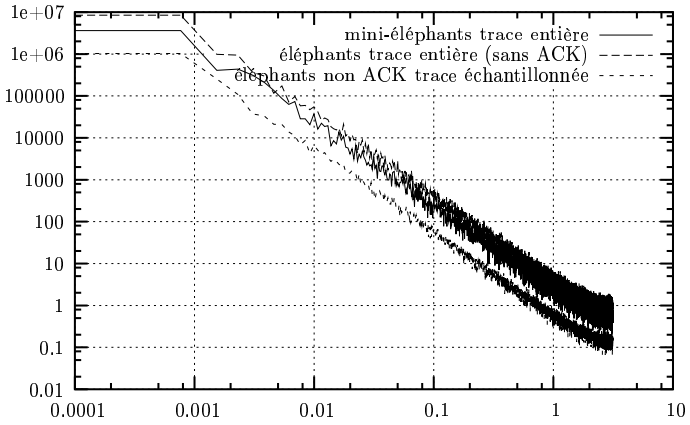
6.2.4 Etude du débit échantillonné

Dans cette section, on s'intéresse aux propriétés du débit évalué sur les intervalles de temps successifs de $\Delta = 100$ ms dans la trace échantillonnée. Ce débit est dénoté par $\{\chi_n\}$; le débit dans la trace entière est noté $\{X_n\}$. Dans un premier temps, on peut vérifier que le débit moyen est globalement conservé, comme le montre le tableau 6.8. Nous avons calculé les débits moyens pour la trace entière et la trace échantillonnée pour trois valeurs du paramètre d'échantillonnage N.

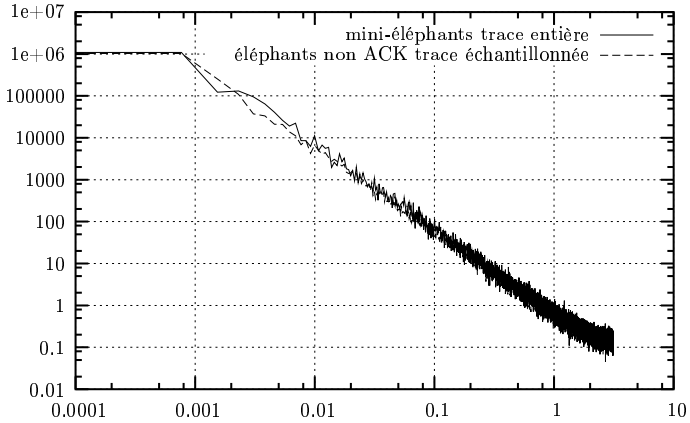
	débit moyen en bit/s
trace entière	$\approx$ 180 000 000
trace échantillonnée	
$N = 1000$	174 160
$N = 100$	1 743 600
$N = 10000$	17 280

TAB. 6.8 – Comparaison des débits moyens dans la trace entière et dans la trace échantillonnée pour différentes valeurs de N.

Pour étudier de manière plus approfondie les propriétés statistiques du débit dans la trace échantillonnée, on calcule la densité spectrale du signal $\{\chi_n\}$, qui est représentée par le figure 6.9(a) où la



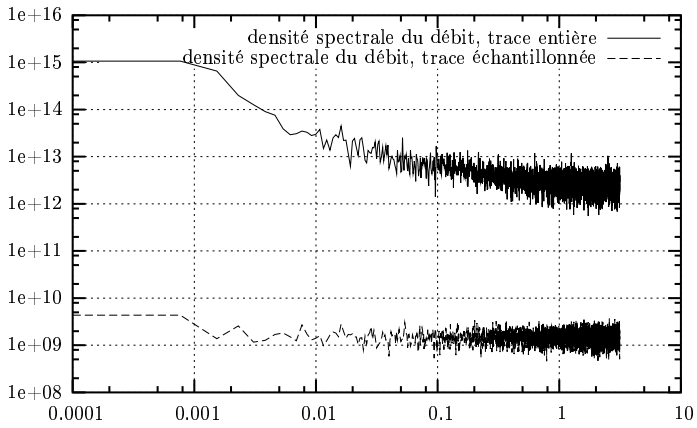
(a) Comparaison



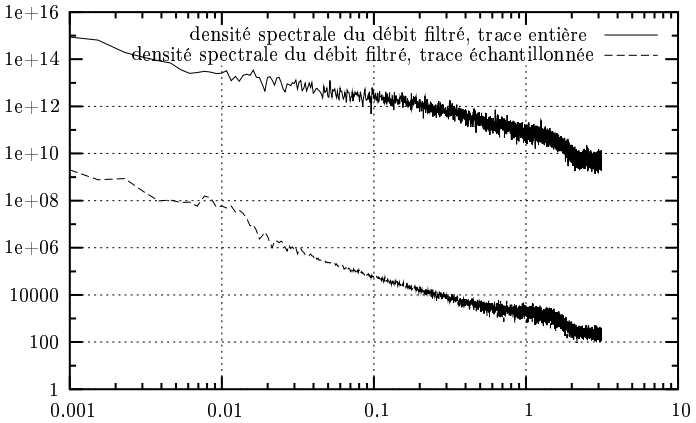
(b) Remise à l'échelle par un facteur 0.25

FIG. 6.8 – Densité de probabilité de la taille des paquets en octets.

densité spectrale du débit dans la trace entière est également reportée. Comme on peut le constater, l'échantillonnage affaiblit le signal et augmente le niveau de bruit. Les mini-éléphants qui sont trop petits par rapport à une certaine constante de temps, qui dépend de N, de l'inter-paquet et du nombre de flots simultanément actifs , ne sont vus qu'une seule fois et ont tendance à être assimilés à du bruit (phénomène de blanchiment du signal), d'où l'accroissement du niveau de bruit dans le signal et la diminution du rapport signal/bruit.



(a) Global



(b) Filtré

FIG. 6.9 – Périodogrammes des débits dans la trace entière et dans la trace échantillonnée.

Cependant, on observe que dans les basses fréquences, le signal existe toujours, bien que très attenué. Cette composante du périodogramme est due aux mini-éléphants qui sont suffisamment longs pour être vus plusieurs fois par l'échantillonneur. En filtrant les débits dans les traces entière et échantillonnée avec un filtre par ondelettes, on peut vérifier que les deux densités spectrales sont

proportionnelles d'un facteur $1/N^2$ dans les basses fréquences. Ce rapport s'explique par le fait que le signal $\{\chi_n\}$ est à peu près égal à $\{X_n/N\}$ dans les composantes en fréquences qui ont un impact sur le débit moyen (cf. le tableau 6.8). Dans les hautes fréquences, les signaux échantillonné et filtré présentent un déficit qui est dû au phénomène de blanchiment du signal.

6.2.5 Conclusions sur les caractéristiques du trafic échantillonné

Des résultats présentés dans les deux section précédentes, on peut noter deux points fondamentaux concernant l'échantillonnage en $1/N$:

1. **Phénomène de blanchiment des signaux (nombre de flots actifs et débit).** L'échantillonneur induit une constante de temps que l'on peut estimer être égale à $\mathbb{E}[N]I$ où $\mathbb{E}[N]$ est le nombre de flots actifs dans la trace entière et I est la durée interpaquet dans la trace échantillonnée. Dans la trace entière, on observe empiriquement que le débit des éléphants en paquet/s (en non pas en bit/s) est 2 fois supérieur à celui des souris. La probabilité de voir un flot particulier est $1/\mathbb{E}[N]$. En moyenne un flot long sera visité tous les $2\mathbb{E}[N]I/3$ (= 500 secondes). Dès lors, les mini-éléphants (dont la moyenne est égale à 500 secondes) qui ont une durée plus petite que $2\mathbb{E}[N]I/3$ ont peu de chance d'être vus plus d'une fois et donnent naissance à un signal faiblement corrélé assimilable à un bruit blanc. Ce qui explique l'accroissement du bruit dans le signal du débit.
2. **Bruit dû au souris.** Lorsqu'un paquet est sélectionné par l'échantillonneur dans la trace entière, il a une chance d'environ 2/3 d'appartenir à un mini-éléphant. Mais dans ces 53%, seuls les mini-éléphants avec des durées supérieures à $2\mathbb{E}[N]I/3$ ont un chance d'être vus plusieurs fois par échantillonnage.

Il ressort de cette analyse qu'il est très difficile d'estimer les paramètres des mini-éléphants. Seuls les vrais éléphants ont une chance d'être vus plusieurs fois après échantillonnage. Par ailleurs, étant donné les niveaux de bruit dans les signaux observés, il semble très délicat d'utiliser des méthodes de traitement du signal. C'est pourquoi nous développons dans la section suivante une approche probabiliste.

6.3 Inférence des paramètres de trafic

Comme nous l'avons vu dans la section précédente, à un taux d'échantillonnage de $1/1000$, seul un nombre très faible de flots sont vus plus d'une fois par l'échantillonneur. Pour mener à bien une étude probabiliste, nous avons besoin d'un nombre assez grand d'échantillons. C'est la raison pour laquelle nous ramenons dans cette section le taux d'échantillonnage à $1/100$.

Nous nous intéressons ici aux caractéristiques des éléphants réguliers, c'est à dire à ceux qui ne sont pas formés principalement d'acquittements. Nous utilisons la définition suivante afin de distinguer les éléphants réguliers dans la trace échantillonnée.

Définition 3. *Un éléphant est un flot vu au moins deux fois dans la trace échantillonnée. Il est dit* ***régulier*** *lorsque la taille moyenne de ses paquets est supérieure à 80 octets.*

Comme dans le cas des mini-éléphants, le processus d'arrivée des éléphants est approché dans toute la suite par un processus de Poisson de paramètre λ. Le calcul de la distribution empirique de

la durée des éléphants montre que celle-ci peut être approchée par une loi de Weibull de paramètres d'échelle $\eta = 116\,s$ et de forme $\beta = 0,39$ (voir figure 6.10). La forme weibullienne de la durée et les arrivées poissonnienne sont les deux caractéristiques du trafic des éléphants originaux qui sont utilisés dans la suite pour retrouver les caractéristiques des éléphants après échantillonnage.

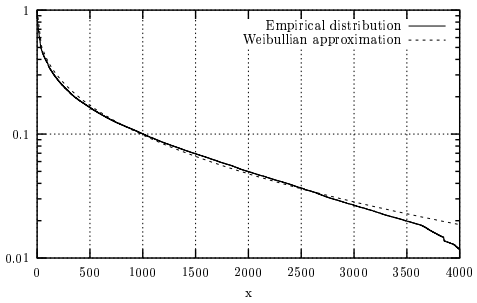


FIG. 6.10 – Distribution empirique de la durée des éléphants (en secondes) et l'approximation de type Weibull

De plus, nous faisons l'hypothèse que tous les éléphants ont la même probabilité d'être vus dans la trace échantillonnée, c'est à dire qu'ils ont tous le même débit en paquet/s. Cette hypothèse n'est pas tout à fait exacte à cause de la présence des périodes de forte activité et de faible activité pour certains éléphants. Néanmoins, elle permet de simplifier le calcul et donne une bonne indication sur le taux d'arrivée des éléphants comme nous allons le voir par la suite.

La probabilité de selectionner un éléphant particulier par échantillonnage à l'instant t est égale à la probabilité de tirer un paquet appartenant à l'éléphant, égale à $p = 1/N_t$, où N_t est le nombre d'éléphants actifs à t. Etant donné que le nombre d'éléphants actifs est grand et change peu au cours du temps,

$$p \approx 1/\mathbb{E}[N_t] = \frac{1}{\lambda \mathbb{E}[S]} = \frac{1}{\lambda \eta \Gamma(1+1/\beta)}, \tag{6.1}$$

où $\mathbb{E}[S]$ est la durée moyenne des éléphants (Γ désignant la fonction d'Euler).

Soit τ le temps d'interarrivée moyen des paquets des éléphants dans la trace échantillonnée. L'échantillonnage d'un éléphant est approximativement poissonnien de taux $\lambda_p = p/\tau$. Ainsi, la probabilité de voir un éléphant plus de n fois dans la trace échantillonnée est donnée par le résultat suivant.

Proposition 1. *En supposant que la durée des éléphants est Weibull, de paralmètres η et β, et que le processus d'échantillonnage d'un éléphant est poissonnien de taux λ_p, le nombre de fois ν qu'un éléphant est vu par échantillonnage est donné par*

$$\mathbb{P}(\nu > n) = \int_0^{+\infty} \frac{x^n}{n!} e^{-(x/\lambda_p \eta)^\beta}\, dx. \tag{6.2}$$

Démonstration. Soient S une loi de Weibull de paramètres d'échelle η et de forme β et N un processus de Poisson de paramètre λ_p ; la fonction de répartition de la loi de Weibull est notée $F(x)$ et

les dates auxquelles surviennent les événements du processus de Poisson sont notées T_n pour $n \geq 0$ avec $T_0 \leq 0 < T_1 < T_2 < \dots$ Par définition, on a

$$\mathbb{P}(\nu > n) = \mathbb{P}(\mathcal{N}[0, S] > n) = \int_0^\infty \mathbb{P}(T_{n+1} \leq x) F(dx)$$

et en effectuant une intégration par partie, on obtient l'équation (6.2). □

Quand n est grand, on a l'approximation suivante pour $\mathbb{P}(\nu > n)$.

Proposition 2. *Pour n grand et $\beta < 1$, on a*

$$\mathbb{P}(\nu > n) \sim \exp\left(-\left(\frac{n}{\lambda_p \eta}\right)^\beta\right). \tag{6.3}$$

Démonstration. Si y_n est la racine de $f_n'(x) = 0$, avec

$$f_n(x) = n \log(\lambda_p x) - \lambda_p x - (\frac{x}{\eta})^\beta - \log n!,$$

alors (y_n) tend vers l'infini quand $n \to \infty$, puisque

$$n = \lambda_p y_n + \beta(\frac{y_n}{\eta})^\beta. \tag{6.4}$$

On pose $y_n = \frac{n}{\lambda_p} - z_n$ avec $\frac{z_n}{n} \to 0$. En introduisant cette expression dans l'équation (6.4), on obtient $\lambda_p z_n = \beta\left(\frac{\frac{n}{\lambda_p} - z_n}{\eta}\right)^\beta$, et on a alors $z_n/n^\beta \sim \frac{\beta}{\eta^\beta}\lambda_p^{-(\beta+1)}$.
Ainsi,

$$y_n \sim \frac{n}{\lambda_p} - \frac{\beta}{\lambda_p}\left(\frac{n}{\lambda_p \eta}\right)^\beta. \tag{6.5}$$

En utilisant l'équivalent ci-dessus, il est facile de montrer que $\delta_n \stackrel{\text{def.}}{=} \sqrt{|f^{(2)}(y_n)|}$ est équivalent à $1/\sqrt{n}$. Pour $x > -y_n\delta_n$,

$$f_n(x/\delta_n + y_n) - f_n(y_n) = \lambda_p y_n \left(\log\left(1 + \frac{x}{y_n\delta_n}\right) - \frac{x}{y_n\delta_n}\right) + \left(\frac{y_n}{\eta}\right)^\beta \left(\beta \log\left(1 + \frac{x}{y_n\delta_n}\right) - \left(1 + \frac{x}{y_n\delta_n}\right)^\beta + 1\right).$$

Ainsi, $f_n(x/\delta_n + y_n) - f_n(y_n) \to -x^2/2$ quand $n \to \infty$. Et puisque

$$\delta_n \int_0^{+\infty} \exp\left(f_n(x) - f_n(y_n)\right) dx = \int_{-y_n\delta_n}^{+\infty} \exp\left(f_n(x/\delta_n + y_n) - f_n(y_n)\right) dx,$$

les inégalités

$$f_n(x/\delta_n + y_n) - f_n(y_n) \leq y_n \left(\log\left(1 + \frac{x}{y_n\delta_n}\right) - \frac{x}{y_n\delta_n}\right)$$

$$\log(1+x) - x \leq -\frac{1}{3}x, \qquad x \geq 2$$

$$\log(1+x) - x + \frac{x^2}{6} \leq 0, \qquad -1 < x \leq 2$$

et le théorème de Lebesgue donnent la relation de convergence suivante

$$\lim_{n\to+\infty} \delta_n \int_0^{+\infty} \exp\left(f_n(x) - f_n(y_n)\right)\, dx = \int_{-\infty}^{+\infty} \exp\left(-\frac{x^2}{2}\right)\, dx = \sqrt{2\pi}. \tag{6.6}$$

A partir de la formule de Stirling $n! \sim n^{n+1/2}\sqrt{2\pi}e^{-n}$, on déduit que

$$\sqrt{2\pi}e^{f_n(y_n)} \sim \frac{1}{\sqrt{n}} \exp\left(n \log\left(1 - \frac{n-\lambda_p y_n}{n}\right) - (n-\lambda_p y_n)\frac{1-\beta}{\beta}\right).$$

A partir de la limite (6.6) et de la relation d'équivalence (6.5), on obtient le développement asymptotique de $\mathbb{P}(\nu > n)$ en fonction de n,

$$\mathbb{P}(\nu > n) \sim \exp\left(-\left(\frac{n}{\lambda_p \eta}\right)^{\beta}\right) \quad \text{quand} \quad n \to \infty,$$

qui est valable pour tout $\beta < 1$, $\eta > 0$ et $\lambda_p > 0$. □

Soit D la durée d'un éléphant échantillonné et soit M le nombre de paquets qu'il contient.

Proposition 3. *La durée D d'un éléphant échantillonné et le nombre M de paquets qu'il contient vérifient la relation en termes de valeurs moyennes*

$$\mathbb{E}[D] = \frac{1}{\lambda_p}\mathbb{E}[M-2]. \tag{6.7}$$

Démonstration. Dans la trace échantillonnée, un éléphant est un flot comprenant au moins 2 paquets. Soit y la durée originale d'un éléphant, sa durée D dans la trace échantillonnée est supérieure à x si $y > x$ et si on tire, par échantillonnage, un paquet à un instant $u \in [0, y-x]$ et au moins un autre paquet dans l'intervalle $[u+x, y]$. Il s'ensuit que

$$\mathbb{P}(D > x | y \text{ et } 2 \text{ paquets} \in [0,y]) = \frac{1}{\mathbb{P}(\nu > 1)} \int_0^{y-x} \lambda_p e^{-\lambda_p u}(1 - e^{-\lambda_p(y-x-u)})du$$

Ensuite, en intégrant sur y, on obtient

$$\begin{aligned}
\mathbb{P}(D > x) &= \int_x^{\infty} \mathbb{P}(D > x | y \text{ et } 2 \text{ paquets} \in [0,y]) F(dy) \\
&= \frac{1}{\mathbb{P}(\nu > 1)} \int_x^{\infty} \int_0^{y-x} \lambda_p e^{-\lambda_p u}(1 - e^{-\lambda_p(y-x-u)})du\, F(dy) \\
&= \frac{1}{\mathbb{P}(\nu > 1)} \int_x^{\infty} (1 - e^{-\lambda_p(y-x)} - \lambda_p(y-x)e^{-\lambda_p(y-x)})\, F(dy),
\end{aligned}$$

et via une intégration par partie, on trouve

$$\mathbb{P}(D > x) = \frac{1}{\mathbb{P}(\nu > 1)} \int_x^{\infty} \lambda_p^2 (y-x) e^{-\lambda_p(y-x)} e^{-(y/\eta)^{\beta}} dy. \tag{6.8}$$

Par définition,

$$\mathbb{P}(M = n) = \frac{\mathbb{P}(\nu = n)}{\mathbb{P}(\nu > 1)}, \quad n \geq 2.$$

Or,

$$\mathbb{P}(\nu > 1)\mathbb{E}(D) = \mathbb{P}(\nu > 1) \int_0^\infty \mathbb{P}(D > x)dx = \int_0^\infty \left(1 - e^{-\lambda_p y} - \lambda_p y e^{-\lambda_p y}\right) e^{-(y/\eta)^\beta} dy.$$

Ensuite, en utilisant l'équation (6.2) et le fait que $\mathbb{E}(\nu) = \lambda_p \mathbb{E}(S)$, on obtient

$$\mathbb{E}(D) = \frac{\mathbb{E}(\nu) - \mathbb{P}(\nu > 0) - \mathbb{P}(\nu > 1)}{\lambda_p \mathbb{P}(\nu > 1)} = \frac{1}{\lambda_p}\mathbb{E}(M - 2),$$

ce qui achève la démonstration. □

Lorsque x est grand, en utilisant l'équation (6.8) et le théorème de convergence dominée, on a

$$\mathbb{P}(D > x) = \frac{e^{-(x/\eta)^\beta}}{\mathbb{P}(\nu > 1)} \int_0^{+\infty} \lambda_p^2 z e^{-\lambda_p z} e^{-\left(\frac{x}{\eta}\right)^\beta \left[\left(1+\frac{z}{x}\right)^\beta - 1\right]} dz \sim \frac{1}{\mathbb{P}(\nu > 1)} e^{-(x/\eta)^\beta},$$

et donc le corollaire suivant

Corollaire 1. *Pour x grand, on a*

$$\mathbb{P}(D > x) \sim \frac{1}{\mathbb{P}(\nu > 1)} e^{-(x/\eta)^\beta}. \tag{6.9}$$

L'approximation de la distribution empirique de ν par une distribution de Weibull lorsque n devient grand permet d'estimer la valeur de $\lambda_p \eta$ et de β. La fonction $n \rightarrow \log(-\log(\mathbb{P}(\nu > n)))/\log(n)$ empirique et l'approximation par une fonction ayant la forme $f(x) = a + b/\log(x)$ sont représentées dans la figure 6.11. Ensuite, en utilisant l'équation (6.7), il est possible d'estimer la valeur de λ_p à partir des valeurs empiriques de $\mathbb{E}(D)$ et $\mathbb{E}(M)$ qui sont égales à 673 s et 12,1 paquets respectivement. La valeur de η est calculée en utilisant l'estimation de $\lambda_p \eta$ faite précédemment.

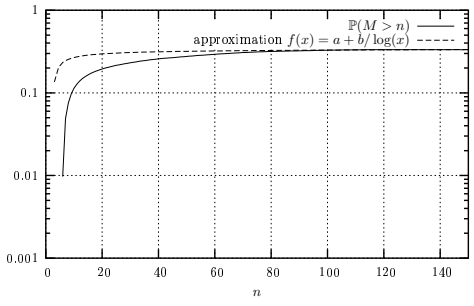


FIG. 6.11 – La fonction $n \rightarrow \log(-\log(\mathbb{P}(\nu > n)))/log(n)$ empirique et son approximation théorique

Le taux d'arrivée des éléphants λ est calculé en utilisant l'équation (6.1), la valeur approchée de λ_p et la valeur de τ sont calculées empiriquement à partir de la trace échantillonnée (égale à 4,1 ms.)

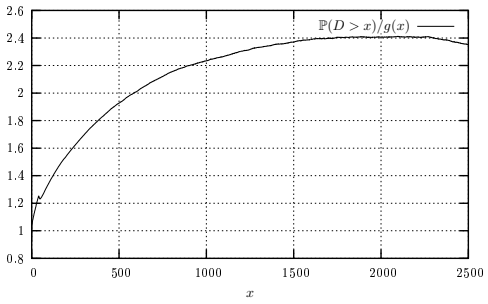


FIG. 6.12 – La fonction $x \rightarrow \mathbb{P}(D > x)/g(x)$ avec $g(x) = \exp\left(-(x/\eta)^{\beta}\right)$; x est en secondes.

En utilisant l'équation (6.9), la figure 6.12 représente le rapport $\mathbb{P}(D > x)/\exp\left(-(x/\eta)^{\beta}\right)$ et montre clairement qu'il tend vers la valeur 2,4 lorsque la durée x devient élevée. Les valeurs considérées de η et β sont celles estimées précédemment. Ainsi, il est possible d'estimer la proportion des éléphants vus dans la trace échantillonnée $\mathbb{P}(\nu > 1)$ en calculant l'inverse de la valeur de ce rapport.

Le tableau 6.9 représente un récapitulatif des valeurs numériques réelles et estimées des différents paramètres. Il est remarquable que les valeurs estimées fournissent une bonne indication sur l'ordre de grandeur de chaque paramètre.

Paramètre	Valeur estimée	Valeur empirique
λ (s^{-1})	33,3	35,2
η (s)	114,66	116
β	0,39	0,39
$\mathbb{P}(\nu > 1)$	0,41	0,47

TAB. 6.9 – Comparaison des valeurs estimées à partir du trafic échantillonné à $N = 100$ avec les valeurs réelles.

sectionRobustesse du modèle Dans la section précédente, nous avons pris une trace exhaustive sur un lien du RBCI (appelée trace entière) et nous avons effectué un échantillonnage en $1/N$. Le trafic échantillonné exhibe deux caractéristiques fondamentales :

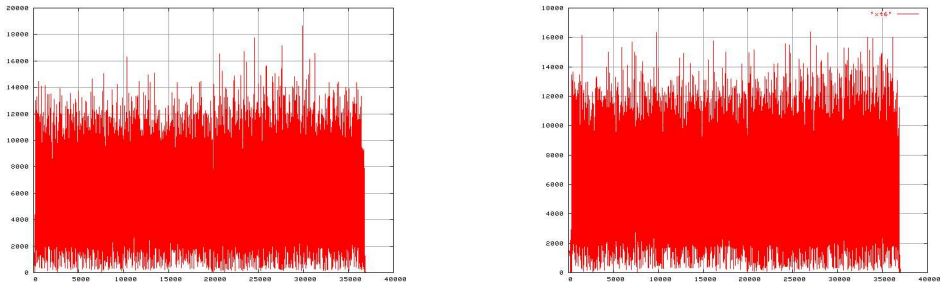
– les éléphants échantillonnés arrivent suivant un processus de Poisson ;
– la durée des éléphants échantillonnés suit une loi de Weibull à deux paramètres.

Ces observations sont en accord avec les prédictions du modèle théorique mis au point dans [**?**], aussi bien en termes de densités spectrales que de caractéristiques probabilistes (durée des éléphants échantillonnés, nombre de paquets dans un éléphant, etc.).

Dans cette section, nous validons un peu plus la robustesse du modèle en considérant une trace issue d'un routeur NC du RBCI (ncidf302), qui met en oeuvre la sonde NetFlow. Deux "engines" au sens NetFlow sont examinés. La trace a été capturée début 2004.

6.3.1 Débit instantané

Dans un premier temps, nous évaluons le débit “instantané” sur le trafic des engines 4 et 6. Ce débit instantané est calculé en comptant le nombre de bits sur des intervalles successifs de $\Delta = 100$ ms. Nous obtenons ainsi une suite chronologique $\{X_n\}$. Les suites correspondantes pour les engines 4 et 6 sont représentées par la figure 6.13. Nous nous apercevons d'emblée que les débits instantanés présentent une forte non-stationarité, due à un accroissement de débit aux environs de 50 secondes. Dans la suite, nous ne nous intéressons qu'à la partie “stationnaire” du débit (entre 50 et 50000 secondes).



(a) Id engine 4 (b) Id engine 6

FIG. 6.13 – Débit instantané du trafic.

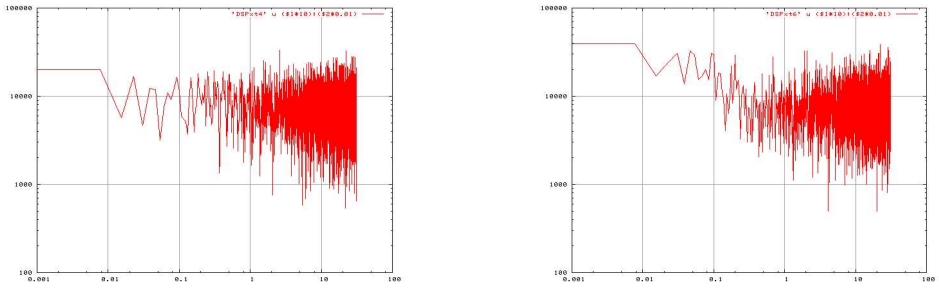
Le tableau 6.10 donne le nombre de flots ainsi que le débit moyen pour l'indice n de X_n variant entre t_1 et t_2 (en millisecondes dans la référence de temps du routeur, depuis sa mise en fonction).

Engine	t_1	t_2	# de flots	$\mathbb{E}[X_n]$ (octet/s)	inter-paquet (en ms)
4	1961627672	1965127672	195935	5745	18.57
6	97664992	101309992	203227	5842	18.45

TAB. 6.10 – Quelques caractéristiques

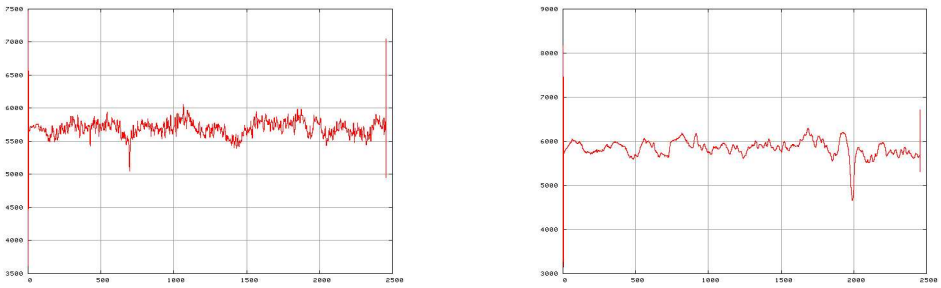
Dans sa partie stationnaire, la suite chronologique $\{X_n\}$ sera caractérisée par son spectre de puissance. La figure 6.14 donne les densités spectrales des débits pour les engines 4 et 6. Comme dans la section précédente, nous observons que les spectres de puissance sont fortement bruités et contiennent peu d'information.

Pour éliminer la composante de bruit, nous utilisons un filtre en ondelette. Nous choisissons l'ondelette de Daubechies d'ordre 6 comme ondelette mère pour la suppression du bruit dans la trace NetFlow. Pour les trafics sortant des engines 4 et 6, le nombre d'échantillons considérés est $N = 24576 = N_0 2_0^J = 12 \times 2^{11}$. La figure 6.15 réprésente le débit instantané filtré du trafic qui sort des engines 4 et 6. La comparaison de la densité spectrale du débit filtré $\{X_t\}$ et la fonction de $f(x) = \frac{c}{x^2}$ (formule théorique en supposant que les flots ont une durée distribuée suivant une loi de Weibull) est illustrée dans la figure 6.16.



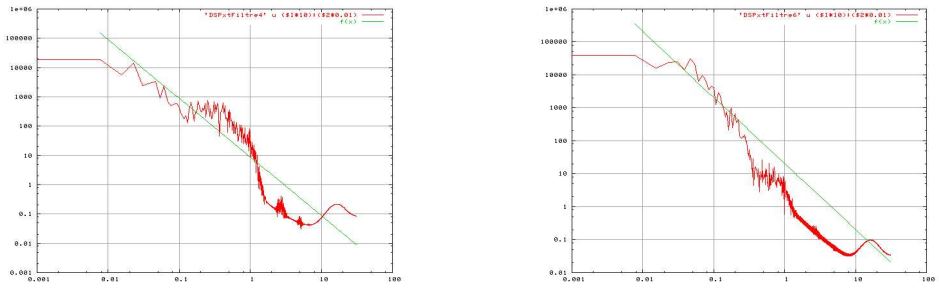
(a) Id engine 4 (b) Id engine 6

FIG. 6.14 – Densité spectrale du débit instantané.



(a) Id engine 4 (b) Id engine 6

FIG. 6.15 – Débit instantané filtré



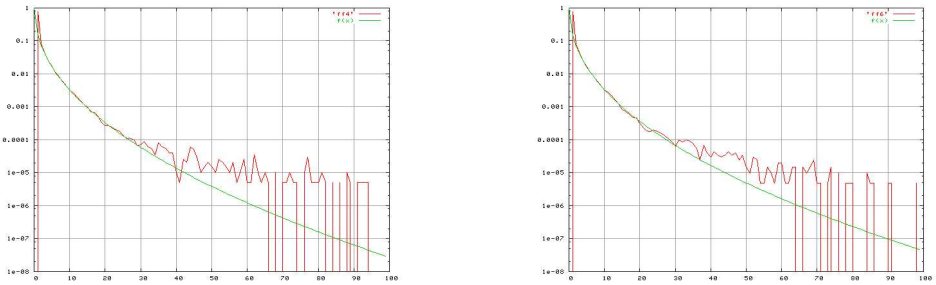
(a) Id engine 4 ($c = 2.4551$) (b) Id engine 6 ($c = 7.44173$)

FIG. 6.16 – Densité spectrale du débit instantané du trafic filtré

Comme dans la section précédente, nous constatons sur une trace NetFlow réelle que l'échantillonnage dégrade énormément les caractéristiques spectrales du débit. Même si les résultats sont conformes aux prédictions théoriques, il semble difficile de déduire quoique ce soit sur le débit initial, la longueur des flots ou leur volume en analysant le débit échantillonné.

6.3.2 Caractéristiques des flots

Dans cette section, nous examinons les caractéristiques des flots en utilisant les mêmes définitions que dans la section précédente. La figure 6.17 représente le nombre de fois qu'un flot est vu. La distribution correspondante peut être approchée par une distribution de Weibull à deux paramètres (paramètre d'échelle η et paramètre de forme β). La faiblesse du paramètre d'échelle s'explique par la forte probabilité de voir un flot une seule fois quand nous ne trions pas les flots supposés éléphants suivant la définition 3.



(a) Id engine 4 ($\beta = 0.483, \eta = 0.269$)

(b) Id engine 6 ($\beta = 0.473, \eta = 0.252$)

FIG. 6.17 – Densité du nombre de flots vus plus de n fois.

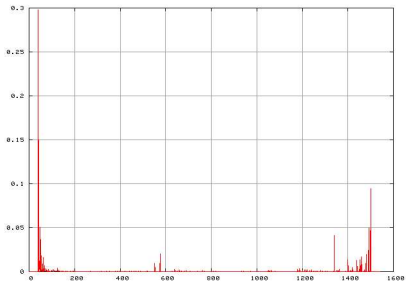
La densité de probabilité de la taille des paquets est illustrée dans la figure 6.18 pour les engines 4 et 6. Les densités ont la même allure que dans les figures 6.1(a) et 6.1(b), ceci indiquant une certaine invariance sur la composition du trafic en termes de taille de paquets.

La densité de probabilité du volume des flots (en octets) est représentée sur la figure 6.19. Nous observons clairement un pic à 40 octets (correspondant le plus souvent à des souris), les flots à un paquet de 40 octets étant sur-représentés dans les statistiques.

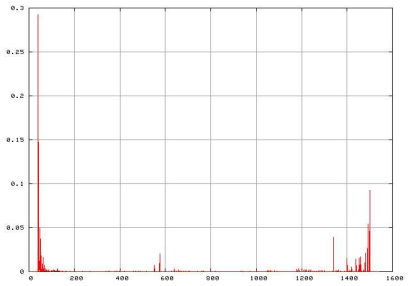
En utilisant la définition 3, la distribution complémentaire d'un éléphant régulier est représentée par la figure 6.20. Nous vérifions que la durée peut être approchée par une loi de Weibull à deux paramètres. Nous pouvons également vérifier que le paramètre de forme β est à peu près le même que celui associé au nombre de paquets dans un flot (cf. la figure 6.17).

6.3.3 Conclusion

A partir d'une trace NetFlow réelle, nous observons que les caractéristiques fondamentales des flots sont retrouvées. En particulier, la loi de Weibull joue un rôle fondamental dans l'interpolation

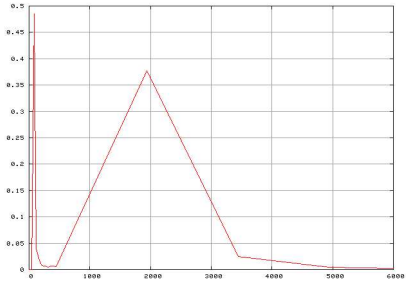


(a) Id engine 4

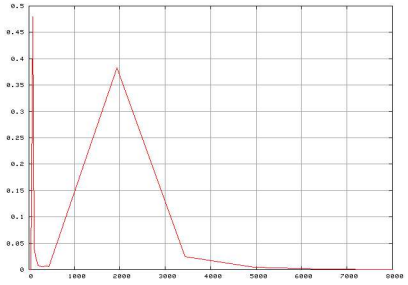


(b) Id engine 6

FIG. 6.18 – Densité de la taille des paquets.

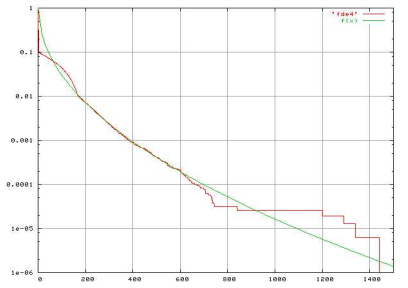


(a) Id engine 4

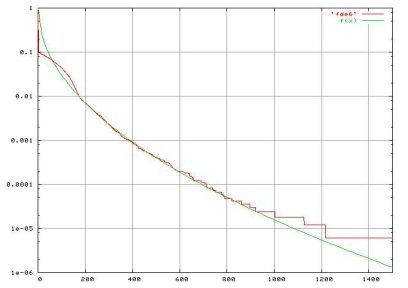


(b) Id engine 6

FIG. 6.19 – Densité du volume des flots.



(a) Id engine 4 ($a = 0.499, b = 8.139$)



(b) Id engine 6 ($a = 0.499, b = 8.123$)

FIG. 6.20 – Durée des éléphants (vus plusieurs fois et >80 octets) - unité de temps 100 millisecondes.

des durées des flots et du nombre de paquets qu'ils contiennent. La même méthode que celle présentée dans la section 6.3 peut alors être utilisée pour retrouver les caractéristiques des éléphants réguliers initiaux, qui engendrent près de 90% du trafic. Le modèle mathématique proposé dans [**?**] semble suffisamment robuste pour analyser des traces de trafic ADSL. Il a été mis au point sur des traces capturées en décembre 2002, puis validé sur des traces d'octobre 2003 puis sur la trace NetFlow étudiée dans cette section (capturée début 2004).

Ce modèle n'est qu'une approximation de la réalité et repose sur un certain nombre d'hypothèses assez fortes. En particulier, la limitation des périodes d'observation peut entraîner un biais important sur les statistiques des flots individuels, surtout que certains sont réputés pour durer pendant plusieurs heures. Néanmoins, il permet d'expliquer de manière cohérente les différentes observations faites jusqu'à ce jour sur du trafic ADSL.

sectionAgrégation et applications

Comme nous l'avons remarqué, nous n'arrivons pas à retrouver les volumes originaux de manière simple. Ce problème provient surtout du fait que beaucoup de flots ne sont jamais vus. Ainsi notre idée consiste à agréger, de façon à avoir plus de paquets d'un même type, ayant alors plus de chance d'être vu. Nous avons décidé de regrouper non plus les flots, mais tous les paquets qui vont à une même adresse IP destination. Nous allons dans ce chapitre étudier les résultats obtenus et les comparer aux résultats précédemment trouvés.

6.3.4 Nombre de paquets

Nous avons pris une fourchette du nombre de paquets associés à une adresse IP destination dans la trace entière et nous avons regardé ce que sont devenus exactement ces adresses après échantillonnage. Ce que nous avons obtenu est dans le tableau 6.11.

L'erreur est obtenue en faisant le volume moyen dans la trace entière moins le volume moyen dans la trace échantillonnée multiplié par mille. Comme nous pouvons le constater, l'erreur reste inférieure à 6% dès que nous avons plus de 5 000 paquets pour une même adresse destination. En dessous, l'erreur commise est importante.
Nous avons voulu savoir si cela revenait au même, ou pas, de partir du nombre de paquets retrouvés dans la trace échantillonnée. Nous avons donc pris une fourchette du nombre de paquets associés à une adresse IP destination dans la trace échantillonnée et nous avons regardé ce qu'ils étaient à l'origine dans la trace entière. Ce que nous avons obtenu est dans le tableau 6.12.

L'erreur est obtenue en faisant le volume moyen dans la trace entière moins le volume moyen dans la trace échantillonnée multiplié par mille. Comme nous pouvons le constater, l'erreur reste inférieure à 3.25% dès que nous avons plus de 5 paquets pour une même adresse destination.
Ainsi, les résultats obtenus en partant de la trace entière ou de la trace échantillonnée sont du même ordre de grandeur. En particulier, dès que nous avons plus de 5000 paquets dans la trace entière ou plus de 5 paquets dans la trace échantillonnée, les résultats obtenus sont satisfaisants.
Nous avons voulu savoir l'erreur commise en partant des adresses destination dans la trace échantillonnée appartenant à une certaine fourchette en ce qui concerne le nombre de paquets, en les assimilants aux adresses destination dans la trace entière mais appartenant à la même fourchette fois $N = 1000$.
Pour cela, nous avons le tableau 6.13.

nombre de paquets dans la trace entière	trace réelle	trace échantillonnée $N = 1000$
$\geq 100\ 000$	nombre d'adresse IP : 905 volume moyen : 116 406 725 octets écart type : 79 571 532	nombre d'adresse IP : 905 volume moyen : 116 986 octets écart type : 81 154 $\Rightarrow$ erreur de +0.5%
$\in [50000, 100000]$	nombre d'adresse IP : 551 volume moyen : 39 037 913 octets écart type : 26 302 285	nombre d'adresse IP : 551 volume moyen : 38 990 octets écart type : 26 948 $\Rightarrow$ erreur de -0.12%
$\in [10000, 50000]$	nombre d'adresse IP : 1 595 volume moyen : 13 560 167 octets écart type : 12 531 101	nombre d'adresse IP : 1 595 volume moyen : 13 298 octets écart type : 12 946 $\Rightarrow$ erreur de -1.9%
$\in [5000, 10000]$	nombre d'adresse IP : 1 497 volume moyen : 3 640 367 octets écart type : 3 640 367	nombre d'adresse IP : 1 492 volume moyen : 3 861 octets écart type : 3 861 $\Rightarrow$ erreur de +6%
$\in [1000, 5000]$	nombre d'adresse IP : 2 083 volume moyen : 938 706 octets écart type : 1 012 957	nombre d'adresse IP : 1 778 volume moyen : 1 084 octets écart type : 1 501 $\Rightarrow$ erreur de +15%
≤ 1000	nombre d'adresse IP : 14 025 volume moyen : 16 985 octets écart type : 55 576	nombre d'adresse IP : 1 065 volume moyen : 215 octets écart type : 478 $\Rightarrow$ erreur de +1165%

TAB. 6.11 – Comparaison à partir du nombre de paquets dans la trace entière

nombre de paquets dans la trace échantillonnée	trace réelle	trace échantillonnée $N = 1000$
≥ 300	nombre d'adresse IP : 127 volume moyen : 217 957 228 octets écart type : 95 448 086	nombre d'adresse IP : 127 volume moyen : 222 793 octets écart type : 98 658 $\Rightarrow$ erreur de +2.2%
$\in [100, 300]$	nombre d'adresse IP : 774 volume moyen : 99 418 379 octets écart type : 63 016 260	nombre d'adresse IP : 774 volume moyen : 100 167 octets écart type : 62 816 $\Rightarrow$ erreur de +0.75%
$\in [50, 100]$	nombre d'adresse IP : 558 volume moyen : 39 778 582 octets écart type : 27 318 768	nombre d'adresse IP : 558 volume moyen : 39 134 octets écart type : 26 607 $\Rightarrow$ erreur de -1.6%
$\in [10, 50]$	nombre d'adresse IP :1 718 volume moyen : 13 205 249 octets écart type : 13 124 252	nombre d'adresse IP : 1 718 volume moyen : 13 315 octets écart type : 12 823 $\Rightarrow$ erreur de +0.83%
$\in [5, 10]$	nombre d'adresse IP : 1 358 volume moyen : 3 962 174 octets écart type : 3 485 527	nombre d'adresse IP : 1 358 volume moyen : 4 091 octets écart type : 3 396 $\Rightarrow$ erreur de +3.25%
≤ 5	nombre d'adresse IP : 3 359 volume moyen : 1 055 204 octets écart type : 1 603 115	nombre d'adresse IP : 3 359 volume moyen : 903 octets écart type : 1 371 $\Rightarrow$ erreur de -14.42%

TAB. 6.12 – Comparaison à partir du nombre de paquets dans la trace échantillonnée

nombre de paquets dans la trace échantillonnée	trace réelle intervalle*N	trace réelle adresses originales retrouvées	estimateur du volume $V_{estime} \simeq N * \tilde{V}$
≥ 300	nb paquets $\geq 300\,000$ nombre d'adresse IP : 122 volume moyen : 219 548 676 octets	nombre d'adresse IP : 127 volume moyen : 217 957 228 octets	V_{estime} : 222 793 000 octets
$\in [100, 300]$	nb paquets $\in [100000, 300000]$ nombre d'adresse IP : 783 volume moyen : 100 336 076 octets	nombre d'adresse IP : 774 volume moyen : 99 418 379 octets	V_{estime} : 100 167 000 octets
$\in [50, 100]$	nb paquets $\in [50000, 100000]$ nombre d'adresse IP : 551 volume moyen : 39 037 914 octets	nombre d'adresse IP : 558 volume moyen : 39 778 582 octets	V_{estime} : 39 134 000 octets
$\in [10, 50]$	nb paquets $\in [10000, 50000]$ nombre d'adresse IP : 1 595 volume moyen : 13 560 168 octets	nombre d'adresse IP : 1 718 volume moyen : 13 205 249 octets	V_{estime} : 13 315 000 octets
$\in [5, 10]$	nb paquets $\in [5000, 10000]$ nombre d'adresse IP : 1 133 volume moyen : 4 068 624 octets	nombre d'adresse IP : 1 358 volume moyen : 3 962 174 octets	V_{estime} : 4 091 000 octets
≤ 5	nb paquets $\in [1000, 5000]$ nombre d'adresse IP : 2 447 volume moyen : 1 142 702 octets	nombre d'adresse IP : 3 359 volume moyen : 1 055 204 octets	V_{estime} : 903 000 octets

TAB. 6.13 – Reconstruction à partir du nombre de paquets

Nous désirons retrouver les volumes dans la trace réelle à partir des volumes retrouvés après échantillonnage. D'après le tableau 6.13, nous pouvons reconstruire le volume original à partir de la trace échantillonnée : en partant d'adresses destination qui ont un nombre de paquets appartenant à une fourchette, nous pouvons reconstruire le volume original des adresses destination qui avaient un nombre de paquets appartenant à cette même fourchette multipliée par $N = 1000$. L'estimateur choisi pour le volume est : $V \simeq N * \tilde{V}$ où V est le volume initial estimé, $\tilde{V}$ est le volume observé et N la fréquence d'échantillonnage.
Nous remarquons qu'il y a une erreur maximale de 2.69% entre les adresses destiantion retrouvées dans la trace réelle et la trace réelle multipliée par N, dès que le nombre de paquets est supérieur à 5 dans la trace échantillonnée.

6.3.5 Volumes

Nous avons réalisé la même chose mais avec les volumes.
Nous sommes partis des adresses destination dans la trace entière qui ont un volume appartenant à une certaine fourchette et nous avons récupéré ces adresses destination dans la trace échantillonnée, les résultats obtenus sont dans le tableau 6.14. L'erreur est obtenue en faisant le volume moyen dans

volume dans la trace entière	trace réelle	trace échantillonnée, $N = 1000$
$\geq 100.10^6$ octets	nombre d'adresse IP : 455 volume moyen : 175 212 277 octets écart type : 71 228 723	nombre d'adresse IP : 455 volume moyen : 176 275 octets écart type : 73 487 $\Rightarrow$ erreur de +0.6%
$\in [10.10^6, 100.10^6]$	nombre d'adresse IP : 1 754 volume moyen : 37 071 168 octets écart type : 24 432 001	nombre d'adresse IP : 1 754 volume moyen : 36 884 octets écart type : 25 338 $\Rightarrow$ erreur de -0.5%
$\in [1.10^6, 10.10^6]$	nombre d'adresse IP : 2 687 volume moyen : 3 888 681 octets écart type : 2 462 793	nombre d'adresse IP : 2 609 volume moyen : 4 079 octets écart type : 3 313 $\Rightarrow$ erreur de +4.9%
$\in [1.10^5, 1.10^6]$	nombre d'adresse IP : 1 972 volume moyen : 405 453 octets écart type : 292 178	nombre d'adresse IP : 1 524 volume moyen : 532 octets écart type : 704 $\Rightarrow$ erreur de +31%
$\leq 1.10^5$ octets	nombre d'adresse IP : 13 785 volume moyen : 6 323 octets écart type : 12 372	nombre d'adresse IP :1042 volume moyen : 82 octets écart type : 105 $\Rightarrow$ erreur de +1196%

TAB. 6.14 – Comparaison à partir des volumes dans la trace entière

la trace entière moins le volume moyen dans la trace échantillonnée multiplié par mille. Comme nous pouvons le constater, l'erreur reste inférieure à 4.9% dès que nous avons plus de 10.10^6 octets pour une même adresse destination. En dessous, l'erreur commise est importante.

Ici également, nous avons voulu savoir si cela revenait au même, ou pas, de partir des volumes retrouvés dans la trace échantillonnée.
Nous sommes donc partis des adresses destination dans la trace échantillonnée qui ont un volume appartenant à une certaine fourchette et nous avons récupéré les adresses destination correspondantes dans la trace entière, les résultats obtenus sont dans le tableau 6.15.

volume dans la trace échantillonnée	trace réelle	trace échantillonnée, $N = 1000$
$\geq 100.10^3$	nombre d'adresse IP : 458 volume moyen : 173 826 382 octets écart type : 72 182 149	nombre d'adresse IP :458 volume moyen : 176 469 octets écart type : 72 795 $\Rightarrow$ erreur de +1.52%
$\in [10.10^3, 100.10^3]$	nombre d'adresse IP : 1 788 volume moyen : 36 259 944 octets écart type : 25 347 347	nombre d'adresse IP :1 788 volume moyen : 36 414 octets écart type : 24 402 $\Rightarrow$ erreur de +0.42%
$\in [1000, 10.10^3]$	nombre d'adresse IP : 2 497 volume moyen : 4 034 484 octets écart type : 3 037 115	nombre d'adresse IP : 2 497 volume moyen : 4 001 octets écart type : 2 480 $\Rightarrow$ erreur de -0.83%
$\in [100, 1000]$	nombre d'adresse IP : 1 326 volume moyen : 685 123 octets écart type : 862 121	nombre d'adresse IP : 1 326 volume moyen : 335 octets écart type : 222 $\Rightarrow$ erreur de -51.1%
$\leq$ 100octets	nombre d'adresse IP : 1 329 volume moyen : 230 434 octets écart type : 498 240	nombre d'adresse IP : 1 329 volume moyen : 57 octets écart type : 19 $\Rightarrow$ erreur de -75.26%

TAB. 6.15 – Comparaison à partir des volumes de la trace échantillonnée

L'erreur est obtenue en calculant la différence entre le volume moyen dans la trace entière et le volume moyen dans la trace échantillonnée multiplié par mille. Si nous ne regardons que les volumes de plus de 1 000 octets, l'erreur reste inférieure à 1.52%.
Ainsi, les résultats obtenus en partant de la trace entière ou de la trace échantillonnée sont du même ordre de grandeur. En particulier, dès que nous avons plus de 1.10^6 octets dans la trace entière ou plus de 1 000 octets dans la trace échantillonnée, les résultats obtenus sont satisfaisants.

Nous avons voulu savoir l'erreur commise en partant des adresses destination dans la trace échantillonnée appartenant à une certaine fourchette en ce qui concerne les volumes, en les assimilants aux adresses destination dans la trace entière mais appartenant à la même fourchette fois $N = 1000$.
Pour cela, nous avons le tableau 6.16.

Nous pouvons remarquer que les moyennes sont conservées (différence maximale de 3.61% entre les adresses destination retrouvées dans la trace réelle et la trace réelle avec l'intervalle des volumes fois N, dès que le volume est supérieur à 1 000 octets dans la trace échantillonnée) mais également la distribution des volumes (écart type dans la trace entière $\simeq$ écart type dans la trace échantillonnée multipliée par $N = 1000$.)

volume dans la trace échantillonnée	trace réelle intervalle*N	trace réelle adresses originales retrouvées	estimateur du volume $V_{estime} \simeq N * \tilde{V}$
$\geq 100.10^3$	volume $\geq 100.10^6$ nombre d'adresse IP : 455 volume moyen : 175 212 277 octets écart type : 71 228 723	nombre d'adresse IP : 458 volume moyen : 173 826 382 octets écart type : 72 182 149	V_{estime} : 176 469 000 octets écart type*N : 72 795 000
$\in [10.10^3, 100.10^3]$	volume $\in [10.10^6, 100.10^6]$ nombre d'adresse IP : 1 754 volume moyen : 37 071 168 octets écart type : 24 432 001	nombre d'adresse IP : 1 788 volume moyen : 36 259 944 octets écart type : 25 347 347	V_{estime} : 36 414 000octets écart type*N : 24 402 000
$\in [1.10^3, 10.10^3]$	volume$\in [1.10^6, 10.10^6]$ nombre d'adresse IP : 2 687 volume moyen : 3 888 681 octets écart type : 2 403 054	nombre d'adresse IP : 2 497 volume moyen : 4 034 484 octets écart type : 3 307 115	V_{estime} : 4 001 000 octets écart type*N : 2 480 000
$\in [1.10^2, 1.10^3]$	volume $\in [1.10^5, 1.10^6]$ nombre d'adresse IP : 1 972 volume moyen : 405 453 octets écart type : 257 305	nombre d'adresse IP : 1 326 volume moyen : 685 123 octets écart type : 862 121	V_{estime} : 335 000 octets écart type*N : 335 000
$\leq 1.10^2$	volume$\leq 1.10^5$ nombre d'adresse IP : 13 785 volume moyen : 6 324 octets écart type : 15 739	nombre d'adresse IP : 1 329 volume moyen : 230 434 octets écart type : 498 240	V_{estime} : 57 000 octets écart type*N : 19 000

TAB. 6.16 – Reconstruction à partir des volumes

Ici encore, nous désirons retrouver les volumes dans la trace réelle à partir des volumes retrouvés après échantillonnage. D'après le tableau 6.16, nous pouvons reconstruire le volume original à partir de la trace échantillonnée : en partant d'adresses destination qui ont un volume appartenant à une fourchette, nous pouvons reconstruire le volume original des adresses destination qui avaient un volume appartenant à cette même fourchette multipliée par $N = 1000$. L'estimateur choisi pour le volume est le même que précédemment.

6.3.6 Comparaisons sur les volumes avec N=100

Nous sommes partis des adresses destination dans la trace entière qui ont un volume appartenant à une certaine fourchette et nous avons récupéré ces adresses destination dans la trace échantillonnée pour N=100, les résultats obtenus sont dans le tableau 6.17. L'erreur est obtenue en faisant le volume

volume dans la trace entière	trace réelle	trace échantillonnée, $N = 100$
$\geq 100.10^6$ octets	nombre d'adresse IP : 455 volume moyen : 175 212 277 octets écart type : 71 228 723	nombre d'adresse IP : 455 volume moyen : 1 754 085 octets écart type : 713 260 $\Rightarrow$ erreur de +0.11%
$\in [10.10^6, 100.10^6]$	nombre d'adresse IP : 1 754 volume moyen : 37 071 168 octets écart type : 24 432 001	nombre d'adresse IP : 1 753 volume moyen : 370 247 octets écart type : 244 655 $\Rightarrow$ erreur de -0.13%
$\in [1.10^6, 10.10^6]$	nombre d'adresse IP : 2 687 volume moyen : 3 888 681 octets écart type : 2 462 793	nombre d'adresse IP : 2 669 volume moyen : 38 998 octets écart type : 25 047 $\Rightarrow$ erreur de +0.28%
$\in [1.10^5, 1.10^6]$	nombre d'adresse IP : 1 972 volume moyen : 405 453 octets écart type : 292 178	nombre d'adresse IP : 1 939 volume moyen : 4 113 octets écart type : 3 194 $\Rightarrow$ erreur de +1.44%
$\leq 1.10^5$ octets	nombre d'adresse IP : 13 785 volume moyen : 6 323 octets écart type : 12 372	nombre d'adresse IP : 3 806 volume moyen : 226 octets écart type : 326 $\Rightarrow$ erreur de +257%

TAB. 6.17 – Comparaison à partir des volumes dans la trace entière

moyen dans la trace entière moins le volume moyen dans la trace échantillonnée multiplié par cent. Comme nous pouvons le constater, l'erreur reste inférieure à 1.44% dès que nous avons plus de 1.10^5 octets pour une même adresse destination. En dessous, l'erreur commise est importante.
Ici également, nous avons voulu savoir si cela revenait au même, ou pas, de partir des volumes retrouvés dans la trace échantillonnée.
Nous sommes donc partis des adresses destination dans la trace échantillonnée qui ont un volume appartenant à une certaine fourchette et nous avons récupéré les adresses destination correspondantes dans la trace entière, les résultats obtenus sont dans le tableau 6.18.

volume dans la trace échantillonnée	trace réelle	trace échantillonnée, $N = 100$
$\geq 100.10^3$	nombre d'adresse IP : 2 203 volume moyen : 65 650 729 octets écart type : 68 169 215	nombre d'adresse IP : 2 203 volume moyen : 656 945 octets écart type : 682 430 $\Rightarrow$ erreur de +0.07%
$\in [10.10^3, 100.10^3]$	nombre d'adresse IP : 2 654 volume moyen : 3 924 953 octets écart type : 2 486 572	nombre d'adresse IP : 2 654 volume moyen : 39 317 octets écart type : 24 344 $\Rightarrow$ erreur de +0.17%
$\in [1000, 10.10^3]$	nombre d'adresse IP : 1 895 volume moyen : 440 308 octets écart type : 323 441	nombre d'adresse IP : 1 895 volume moyen : 4 058 octets écart type : 2 538 $\Rightarrow$ erreur de -7.8%
$\in [100, 1000]$	nombre d'adresse IP : 1 798 volume moyen : 47 789 octets écart type : 47 789	nombre d'adresse IP : 1 798 volume moyen : 368 octets écart type : 242 $\Rightarrow$ erreur de -23%
$\leq$ 100octets	nombre d'adresse IP : 2 098 volume moyen : 6 447 octets écart type : 12 629	nombre d'adresse IP : 2 098 volume moyen : 58 octets écart type : 20 $\Rightarrow$ erreur de -10%

TAB. 6.18 – Comparaison à partir des volumes de la trace échantillonnée

L'erreur est obtenue en calculant la différence entre le volume moyen dans la trace entière et le volume moyen dans la trace échantillonnée multiplié par cent. Si nous ne regardons que les volumes de plus de 1 000 octets, l'erreur reste inférieure à 7.8%.
Ainsi, les résultats obtenus en partant de la trace entière ou de la trace échantillonnée sont du même ordre de grandeur. En particulier, dès que nous avons plus de 1.10^5 octets dans la trace entière ou plus de 1 000 octets dans la trace échantillonnée, les résultats obtenus sont satisfaisants.

Nous avons voulu savoir l'erreur commise en partant des adresses destination dans la trace échantillonnée appartenant à une certaine fourchette en ce qui concerne les volumes, en les assimilants aux adresses destination dans la trace entière mais appartenant à la même fourchette fois $N = 100$.
Pour cela, nous avons le tableau 6.19.

volume dans la trace échantillonnée	trace réelle intervalle*N	trace réelle adresses originales retrouvées	estimateur du volume $V_{estime} \simeq N * \tilde{V}$
$\geq 100.10^3$	volume $\geq 10.10^6$ nombre d'adresse IP : 2 209 volume moyen : 65 524 860 octets écart type : 68 524 860	nombre d'adresse IP : 2 203 volume moyen : 65 650 729 octets écart type : 68 169 215	V_{estime} : 65 694 500 octets écart type*N : 68 243 000
$\in [10.10^3, 100.10^3]$	volume $\in [1.10^6, 10.10^6]$ nombre d'adresse IP : 2 687 volume moyen : 3 888 681 octets écart type : 2 403 054	nombre d'adresse IP : 2 654 volume moyen : 3 924 953 octets écart type : 2 486 572	V_{estime} : 3 931 700 octets écart type*N : 2 434 400
$\in [1.10^3, 10.10^3]$	volume$\in [1.10^5, 1.10^6]$ nombre d'adresse IP : 1 972 volume moyen : 405 453 octets écart type : 257 305	nombre d'adresse IP : 1 895 volume moyen : 440 308 octets écart type : 323 441	V_{estime} : 405 800 octets écart type*N : 253 800
$\in [1.10^2, 1.10^3]$	volume $\in [1.10^4, 1.10^5]$ nombre d'adresse IP : 1 929 volume moyen : 37 638 octets écart type : 24 648	nombre d'adresse IP : 1 798 volume moyen : 47 789 octets écart type : 54 220	V_{estime} : 36 800 octets écart type*N : 24 200
$\leq 1.10^2$	volume$\leq 1.10^4$ nombre d'adresse IP : 11 856 volume moyen : 1 229 octets écart type : 1 913	nombre d'adresse IP : 2 098 volume moyen : 6 447 octets écart type : 12 629	V_{estime} : 5 800 octets écart type*N : 2 000

TAB. 6.19 – Reconstruction à partir des volumes

Ainsi dès que nous avons plus de 1 000 octets dans la trace échantillonnée, nous pouvons prendre comme estimateur du volume initial : le volume échantillonné fois N=100.

6.3.7 Conclusion

Il y a une bijectivité entre travailler à partir des volumes ou paquets de la trace entière et ceux de la trace échantillonnée (en divisant dans le premier cas et en multipliant dans le second par le facteur N).
Rappellons que nous voulons reconstruire le trafic réel à partir du trafic échantillonné avec par exemple un estimateur pour le volume. Dès que nous avons suffisament de paquets observés (plus de 5 dans la trace échantillonnée) ou dès que nous avons suffisament d'octets observés (plus de 1 000 dans la trace échantillonnée), un estimateur du volume initial peut être défini comme : $V \simeq N * \tilde{V}$ où V est le volume initial estimé, $\tilde{V}$ est le volume observé et N la fréquence d'échantillonnage. Cela s'explique par le fait que nous retrouvons presque toutes les adresses destination lorsque nous regardons les adresses vues plus de 5 fois : nous nous retrouvons alors dans une situation équivalente à faire un échantillonnage au sein d'une même adresse destination. Tout ce passe comme si faire un échantillonnage en $\frac{1}{N}$ avec $N = 1000$ sur des adresses destination de plus de 5 000 paquets revenait à voir au moins une fois cette adresse après échantillonnage.

Chapitre 7

Etude de l'échantillonnage NetFlow

[1]

7.1 Introduction

Le but de cette observation est de déterminer si l'outil NetFlow[2] configuré en mode échantillonnage permet d'obtenir des mesures fiables afin de maintenir les systèmes de métrologie éxistants (calcul de consommation, détection des incidents de sécurité, etc).

7.2 Principe de l'échantillonnage NetFlow

Avec l'émergence de liaisons ayant un débit de plus en plus important, les mechanismes de mesures présents dans les équipements demandent de plus en plus de ressources pour traiter le flot d'information circulant sur le réseau. Le méchanisme NetFlow permet de construire ces flux à partir de l'observation de l'entête des paquets et de la table de routage. Créé pour accélérer le routage à partir de l'activation du cache CEF[3], le NetFlow n'est plus utilisé dans ce but aujourd'hui. Il est quand même maintenu sur les équipements CISCO[4] à des fins de supervision. Mais pour ne pas léser les autres fonctionnalités du routeurs lors de l'explosion des débits, la fonctionnalité d'échantillonnage est apparue. Cet échantillonnage fut d'abord déterministe et c'est ce mode qui est étudié ici. Le routeur construit les flux en fonction du X^e paquet qui traverse l'équipement (7.1). Sur les équipements RENATER (routeur CISCO 124xx) la granularité la plus fine que l'on puisse configurer est de 1 paquet sur 10.

[1] par François Xavier Andreu
[2] NetFlow est une technologie de CISCO SYSTEMS : http ://www.cisco.com/warp/public/732/Tech/nmp/netflow/
[3] CISCO Express Forwarding
[4] Les autres constructeurs ont également implémenté des méchanismes similaires

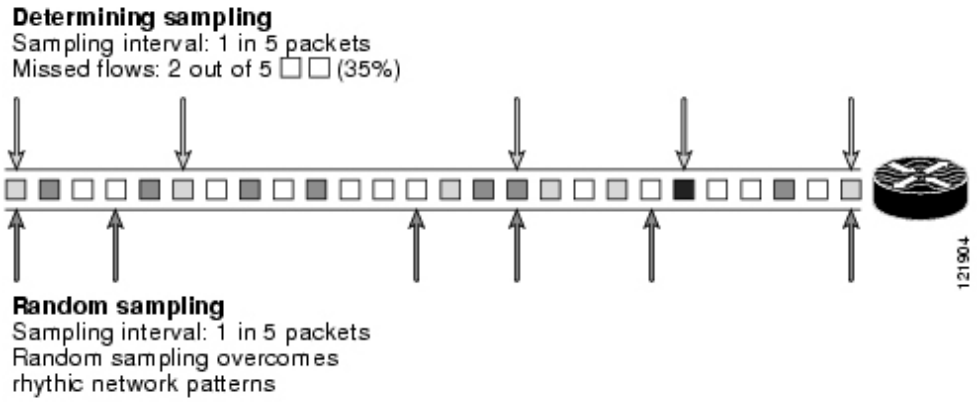


FIG. 7.1 – Construction d'un flux en mode échantillonné, source site web CISCO

Il éxiste sur les autres routeurs CISCO un mode d'échantillonnage aléatoire. Malheuresement nous n'avons pas pu tester cette fonctionnalité car non présente sur les GSR 124xx qui équipent les points de présence RENATER en métropole.

7.3 Capture des flux

Le principe de cette étude est de comparer le mode "sampled" aux autres méthodes de mesures à notre disposition : SNMP (qui permet seulement de comparer le débit) et le mode NetFlow "full"[5]. Pour cette dernière observation il est nécessaire de capturer le même trafic en deux points différents avec à chaque fois une des deux configuration sur le routeur. Nous avons pu utiliser cette configuration qu'en un seul point du réseau : cette architecture (cf figure 7.2) imposait pratiquement le choix du point de présence. La première observation (cf 7.4.1) compare les statistiques SNMP sur un lien avec le calcul de débit d'après le NetFlow en mode échantillonné.

7.4 Les résultats

7.4.1 Statistiques du lien

Cette étude NetFlow fut restreinte à la capture des flux d'un seul préfixe (/16). Ce préfixe correspond à un réseau de campus qui gère lui même l'adressage des sites. Ce réseau est raccordé directement au point de présence RENATER, nous pouvons ainsi considérer que les mesures réalisées avec SNMP représentent "exclusivement" le débit du trafic IPv4 du préfixe (on ne prend pas en compte ici l'existence sur le lien d'un trafic multicast et/ou IPv6 qui est négligeable).
Les mesures de débits sont donc réalisées d'un côté avec un polling SNMP et avec le Netflow.

[5]mode originel, tous les paquets sont analysés par le routeur

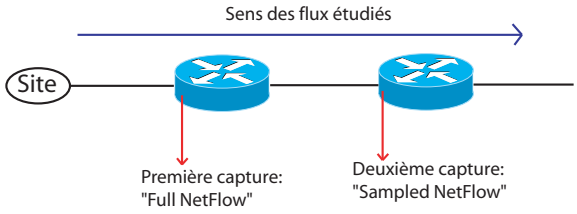


FIG. 7.2 – Architecture lors de la capture des flux NetFlow

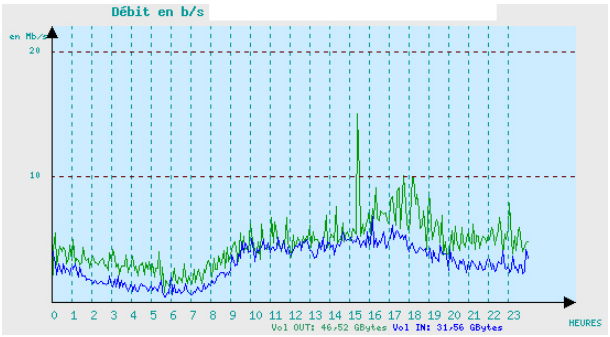


FIG. 7.3 – Débit et volumétrie du préfixe sur une journée d'après le NetFlow

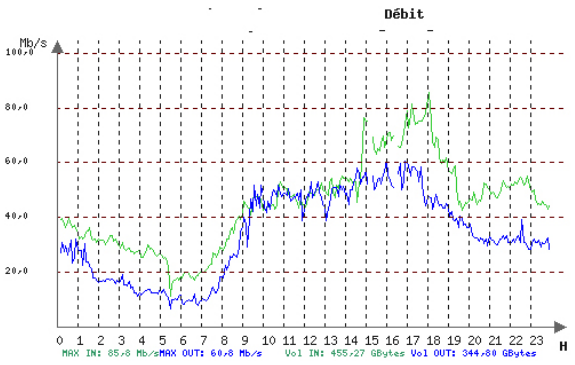


FIG. 7.4 – Débit et volumétrie du préfixe sur une journée d'après SNMP

Les valeurs observées sur les histogrammes montrent que l'on peut déjà sur une journée multiplier les valeurs de volumétrie (en bas sur le graphique 7.3) mesurées par le NetFlow "sampled" par 10 pour obtenir un résultat proche de la réalité (celle-ci étant représentée par les statistiques SNMP[6]

[6]technique considérée comme fiable ;)

- graphique 7.4).
Les pics de trafic sur le graphe construit d'après les valeurs NetFlow sont dus à la configuration du collecteur. Ce dernier comptabilisait les octets d'un flux sur 5 minutes et non pas sur la durée de vie du flux. Dans cette configuration les débits maximum observés avec le NetFlow "sampled" ne doivent pas être extrapolés : les paquets choisis ne sont pas représentatifs de l'ensemble du trafic réel sur un tel intervalle. Cela ce traduit par un effet en "dent de scie" sur le graphe "sampled". Cet effet est accentué par la prise en compte des longs flux qui ne sont comptabilisés qu'au bout d'un certain temps : un transfert de 50Mo sur plus de 5 minutes n'est pris en compte qu'à la fin du flux, et n'est comptabilisé que dans une tranche de 5 minutes au lieu de plusieurs. Mais dès que l'on moyenne sur deux heures, ce phénomène disparait et l'on peut utiliser ce mode de mesure pour des calculs de volumétrie pour des sites clients (même avec un collecteur "mal réglé").

7.4.2 Evaluation de la perte d'information

Il est normal d'obtenir une perte d'information lors de l'utilisation du NetFlow en mode échantillonnage[7]. Cette perte dépend entièrement du type de trafic qui circule sur le réseau. Si tous les flux étaient composés d'un seul paquet, nous ne pourions observer que 10% des flux ; à l'inverse si tous les flux étaient composés d'une multitude de paquets la perte serait minime.
Afin de déterminer ou plutôt quantifier cette perte en fonction du type de trafic sur le réseau RENATER, nous avons capturé les flux de deux préfixes sur deux routeurs comme expliqué au paragraphe 7.3, nous avons estimé que le trafic de ces deux préfixes était représentatif de l'ensemble du trafic des sites RENATER.

TAB. 7.1 – *Observation quantitative*

	Full	Sampled	Pertes
Nombre de flux	3416043	1522338	55%
Nombre d'octets	19344774677	2078190250	90%
Nombre de paquets	48802075	5303552	90%

On observe une perte de plus de la moitié des flux en mode échantillonné (tableau 7.1). Le nombre d'octets peut quant à lui être multiplié par dix pour obtenir la quantité réelle du trafic (ce qui confirme l'observation du paragraphe 7.4.1 pour le calcul volumétrique).
Nous allons maintenant essayé de connaître le type des flux qui ne sont pas capturés par le méchanisme NetFlow.

La perte des flux UDP est très importante (tableau 7.2), et c'est en fait les flux TCP (qui représentent la majeur partie du trafic) qui déterminent la perte global de flux en mode échantillonné : 55%.

L'observation de la répartition des flux en fonction de leur taille (nombre de paquets -figure 7.6- ou quantité d'octets dans un flux -figure7.5-) fait bien apparaître des courbes similaires où le facteur 10 est déterminant. On peut remarquer que le nombre de flux composés d'un seul paquet (ou de

[7] le mode déterministe est peut-être moins performant que le mode aléatoire

TAB. 7.2 – *Étude sur les protocoles de la couche transport*

	Full	Sampled	Pertes
Nombre de flux	3416043	1522338	55%
Nombre de flux ICMP	241	103	58%
Nombre de flux TCP	3204712	1487522	54%
Nombre de flux UDP	211089	34713	84%

deux) est supérieur en mode “sampled” alors que nous observons au total une perte de flux. Cela s’explique encore par la technique utilisée : les flux de petites tailles, même s’ils ont peu de chance d’être analysés, voient leur taille diminuée lorsqu’ils sont capturés et augmentent ainsi le nombre réel de flux de petite taille.

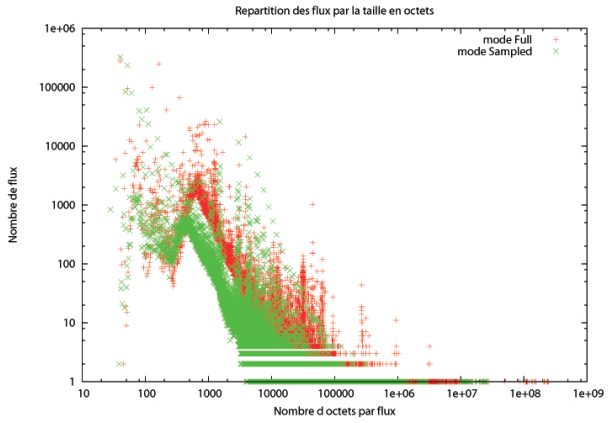


FIG. 7.5 – Répartition des flux en fonction de leur taille en octets

L’étude sur les protocoles de la couche applicative (histogramme 7.7) est basée sur les numéros de ports (sources en l’occurrence). Certains protocoles disparaissent comme le ssh et les flux des ports 137 et 139. Malgré l’echelle logarithmique la perte de flux est toujours présente. Le protocole HTTP est de loin le protocol le plus représenté avec une faible perte d’information. Le protocole DNS (port 53) subit une forte perte qui s’explique par l’emploie du protocole de transport UDP.

7.5 Conclusion

Nous pouvons conclure par cette observation que l’utilisation du mode échantillonné du NetFlow conserve la possibilité d’effectuer des calcul de consommation et d’observer les problèmes de sécurité sur le réseau. Il est quand même nécessaire d’ajuster les outils de traitement des flux (par exemple en diminuant la valeur des seuils de détection des “Deny of Service”). Bien sûr, le problème est aujourd’hui juste repoussé, et il faudra un jour construire les flux à partir de 1 paquet sur

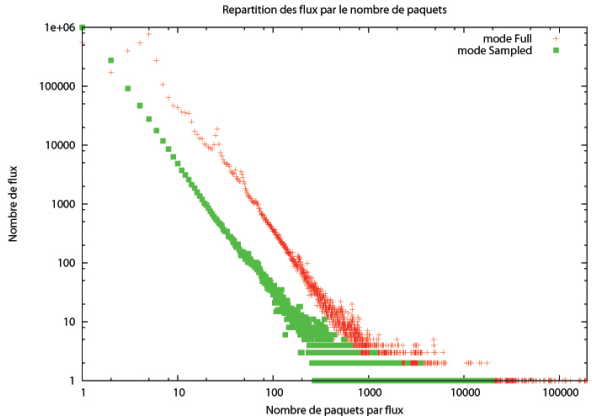


FIG. 7.6 – Répartition des flux en fonction de leur taille (nombre de paquets)

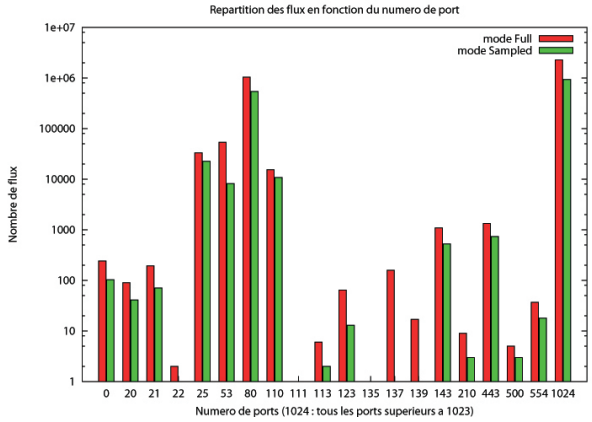


FIG. 7.7 – Répartition des flux en fonction du port applicatif

100 ou 1000 au vu des débits atteints sur les backbones des opérateurs (même si le traitement est actuellement fait en hardware sur la plupart des cartes), augmentant ainsi la perte d'informations. La volumétrie sera sans doute toujours réalisable avec le NetFlow mais les signatures définissant les problèmes de sécurité seront de plus en plus difficile à définir à ce niveau, repoussant ces types de mesures aux bords des réseaux voir à l'extérieur (site client, réseaux régionaux).

Chapitre 8

La plate-forme Saturnev6

[1]

8.1 Saturnev6 : Une architecture de mesure passive et active

Ce chapitre, issu de [**?**], présente l'évolution de l'outil de métrologie active Saturne vers IPv6 et vers la mesure passive. Grâce aux particularités du protocole IPv6, nous disposons de nouveaux mécanismes pour mettre en œuvre l'estampillage des paquets. Alors que les méthodes classiques insèrent l'estampille dans la partie données des paquets de sonde, IPv6 permet de placer l'estampille entre l'en-tête IPv6 et la partie donnée du paquet, dans une extension. La définition d'une extension IPv6 pour la métrologie permet à l'outil de devenir plus proche des méthodes passives. Ainsi, en ajoutant quelques octets au paquet de façon totalement transparente, on peut estampiller un paquet applicatifs de façon passive. Ce chapitre présente les principes des extensions d'IPv6, l'extension de métrologie définie et mise en œuvre dans le cadre de l'outil Saturne ainsi que les premiers tests d'exploitation.

8.1.1 Les changements apportés par le protocole IPv6

IPv6 représente non seulement une solution aux problèmes d'épuisement des adresses IPv4 disponibles et d'explosion des tables de routage, mais il offre également un véritable modèle de bout en bout permettant à chaque équipement du réseau d'obtenir une adresse IP globale et unique. Les problèmes d'épuisement des adresses disponibles, le recours à la traduction d'adresses (NAT : Network Address Translator) et l'explosion de tables de routage sont les motivations initiales pour définir le protocole IPv6. Les capacités du protocole IPv6 ne se limitent pas à la modification de la taille des adresses. Le protocole est simplifié et permet plusieurs autres avantages tels que la configuration automatique, la sécurité et la mobilité. Au cours de sa définition, le format des en-têtes IPv6 a été simplifié et conçu pour réduire leur traitement, permettant aux routeurs intermédiaires de meilleures performances. La Figure 8.1 compare le format des en-têtes IPv4 et IPv6.

[1] par Joel Corral et Géraldine Texier

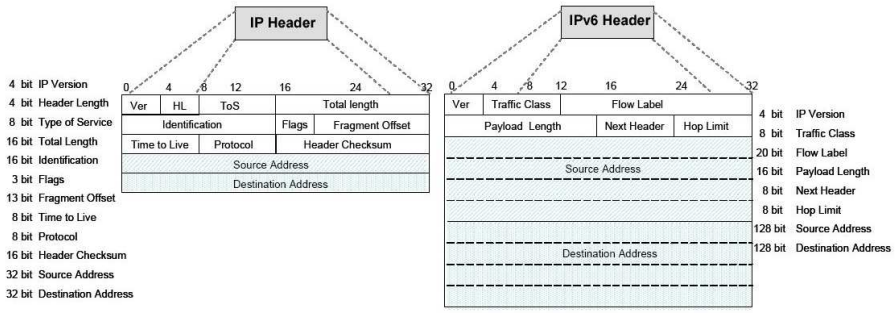


FIG. 8.1 – Format des en-têtes IPv4/IPv6

Dans IPv4, les options doivent être recalculées par chaque routeur. Ces options ont été retirées de l'en-tête IPv6 et remplacées par de nouveaux en-têtes appelés extensions qui peuvent être facilement ignorés par les routeurs intermédiaires. De plus, le champ de somme des bits de contrôle (checksum) a disparu de l'en-tête IPv6. Tous les protocoles de niveau supérieur doivent mettre en œuvre un mécanisme de checksum de bout en bout. Le checksum d'UDP, facultatif pour IPv4, devient ainsi obligatoire.

Les extensions d'IPv6 offrent plus de flexibilité pour intégrer de nouvelles fonctionnalités. Elles peuvent être vue comme un prolongement de l'encapsulation d'IP dans IP et fournit une information complémentaire de façon efficace. Les options ayant été retirées de l'en-tête et remplacées par des extensions facultatives placées entre l'en-tête IPv6 et l'en-tête de la couche supérieure, la taille de l'en-tête IPv6 est fixe. Le protocole IPv6 peut être amélioré et de nouvelles fonctionnalités peuvent être ajoutées en utilisant ces en-têtes d'extension.

À la différence de l'en-tête IPv4, qui ne peut avoir que 40 octets d'options, une extension a une longueur multiple de 8 octets (cf. figure 8.2). Les extensions IPv6 sont transparentes pour les protocoles de la couche supérieure. Sept types d'extensions ont été définis dont : Proche-en-proche, destination ou routage (cette extension permet d'imposer à un paquet une route différente de celle offerte par les politiques de routage présentes sur le réseau). La liste complète peut être trouvée dans [**?**]. L'extension de proche-en-proche est traitée par tous les routeurs intermédiaires, les autres extensions ne sont prises en compte que par les équipements destinataires du paquet.

8.1.2 L'extensions destination

Les caractéristiques de cette extension sont intéressantes, en particulier dans une perspective de mise en œuvre d'un mécanisme de bout en bout dans le réseau. Cette extension, dont le format est identique à l'extension de proche-en-proche, contient des informations qui ne sont traitées que par l'équipement destinataire et n'introduit donc pas de traitement dans les équipements intermédiaires du réseau. La figure 8.3 montre le format de l'extension destination. Une extension commence par un champ d'un octet donnant l'en-tête suivant qui définit le type de données qui suit l'extension (un protocole de niveau 4 ou bien une autre extension). Cette extension est identifiée par la valeur 60 (0x3c) dans le champ de en-tête suivant.

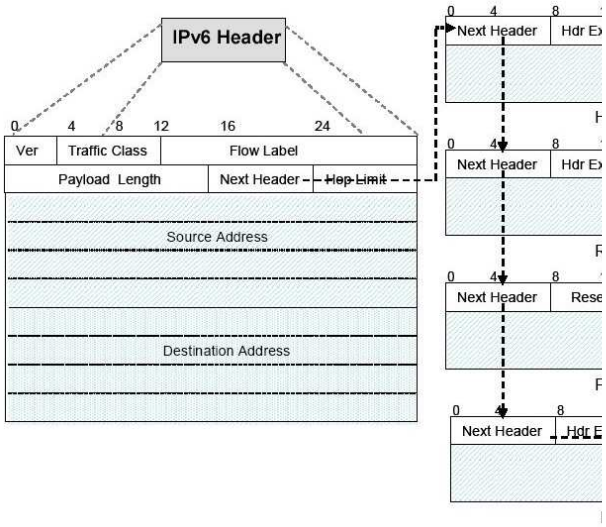


FIG. 8.2 – En-têtes d'extension IPv6

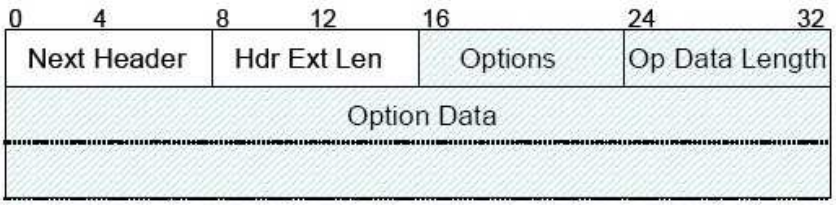


FIG. 8.3 – Format de l'extension destination

– **En-tête suivant** (Sélecteur sur 8-bits). Il permet de identifier l'en-tête suivant immédiatement l'entête de l'extension de destination. Il utilise les mêmes valeurs que le champ en-tête suivant d'IPv6.
– **Longueur de l'en-tête** (Entier non signé de 8 bits). Ce champ permet d'identifier la longueur de l'en-tête des extensions de destination en mots de 8 octets, sans compter les 8 premiers octets.
– **Options** Champ de longueur variable, telle que l'en-tête des options de destination complet soit un entier multiple de 8 octets. Contient au moins une option encodée TLV (Type, Longueur, Valeur) avec la structure suivante :
  Type d'option : Identificateur sur 8 bits du type d'option
  Longueur données d'option (entier non signé de 8 bits). Il représente la longueur du champ donnée, en octets

La séquence d'options à l'intérieur d'un en-tête est traitée strictement dans l'ordre où apparaissent ces options dans l'en-tête, un nœud ne doit pas parcourir l'en-tête à la recherche d'un type particulier d'option et traiter cette option avant d'avoir traité toutes celles qui la précèdent [**?**].

8.1.3 IPv6 et Saturne

Dans un premier temps, la mesure IPv6 a été mise en œuvre dans l'architecture Saturne [**?**] en utilisant le même mécanisme que pour IPv4 [**?**] (figure 8.4. La valeur d'une estampille temporelle d'émission est insérée dans les paquets appartenant à un flux de sonde UDP avant leur envoi sur le réseau.

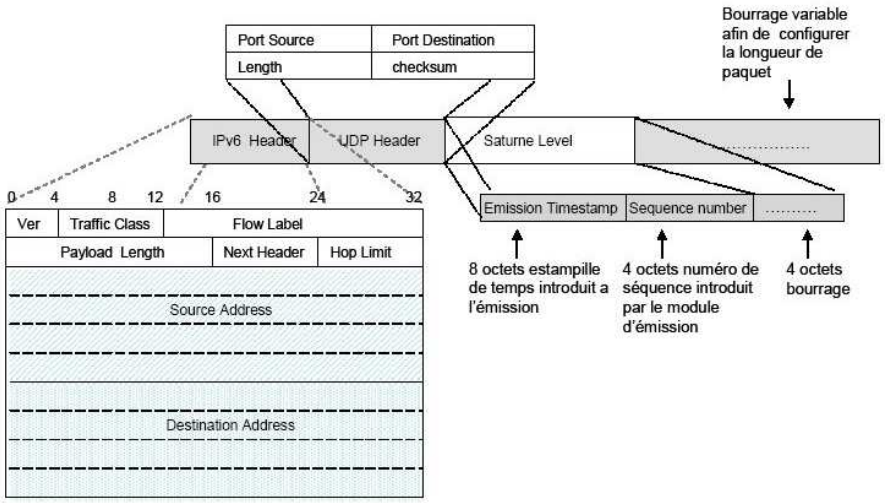


FIG. 8.4 – Format des paquets de sonde IPv6 utilisant le même mécanisme que pour IPv4

Le mécanisme assure la compatibilité de l'architecture originale Saturne avec IPv6, néanmoins il se limite aux scénarios traditionnels de mesure active. Pour profiter des fonctionnalités offertes par

IPv6, en particulier du mécanisme d'extension l'architecture Saturne a été améliorée pour effectuer la mesure active et passive des flux applicatifs.

8.1.4 Définition d'une extension IPv6 pour la métrologie

Nous proposons la définition d'une extension IPv6 de type destination qui permettra le transport d'information de métrologie dans l'en-tête du paquet IPv6, permettant une nouvelle méthodologie de mesure à la frontière des mesures actives et passives.

L'extension de métrologie porte le numéro de séquence du paquet et l'estampille temporelle de départ du paquet tels que définis par l'IETF. L'extension offre la possibilité d'insérer des données de métrologie à la suite de l'en-tête des paquets IPv6, permettant d'estampiller un paquets d'un flux de sonde mais aussi un paquet appartenant à un flux applicatifs de façon transparente pour les équipements intermédiaires et les applications. L'avantage principal de cette méthode est que l'information de métrologie est également transparente pour les protocoles de niveaux supérieurs. En revanche, le mécanisme modifie légèrement la longueur du paquet applicatif puisqu'il ajoute les quelques octets de l'extension de métrologie.

L'insertion de l'estampille de temps dans l'en-tête IPv4 au niveau de la couche physique exigerait le recalcul de tout l'en-tête avant retransmission. Cette limitation imposait alors d'insérer cette estampille soit au niveau applicatif, ce qui interdisait une mesure au niveau filaire des performances du réseau, soit dans la partie données du paquet, rendant ce paquet inexploitable pour l'application. L'approche utilisant l'extension de métrologie offre la possibilité de mesurer n'importe quel flux d'application dans une perspective de bout en bout réduisant au minimum le trafic intrusif.

Format de l'extension de métrologie

L'extension IPv6 pour la métrologie est du type destination, la section précédente fait une brève description de ce type d'extension. La figure 8.5 montre la description de l'extension pour la métrologie.

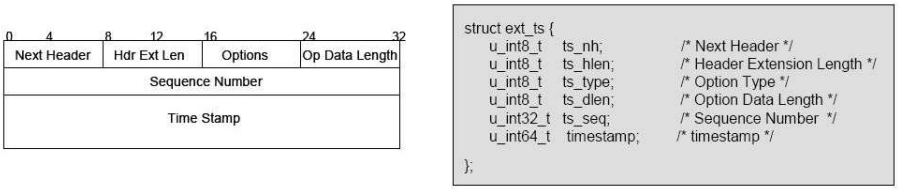


FIG. 8.5 – En-tête d'extension pour la métrologie

- **Type d'option** Identificateur sur 8 bits du type extension. Nous avons arbitrairement choisi le type le numéro 60 décimal (0x3c) pour l'identification de l'extension de métrologie. En accord avec la norme IPv6 [**?**], cette valeur signifie que cette option sera ignorée par un nœud différent de la destination, le troisième bit indique que les données de l'extension peuvent changer en cours de route vers la destination finale.
- **ts_seq** (32-bits). Indique le numéro de séquence du paquet appartenant au flux mesuré.

– **Estampille** (64-bits). Contient l'estampille temporelle d'envoi du paquet. La valeur de cette estampille est donnée par le système, synchronisé sur le temps global d'un GPS.

Le temps d'envoi du paquet étant transporté par l'extension, le temps de traversée unidirectionnel du réseau est calculé de bout en bout en le comparant avec l'horloge globale à la réception du paquet.

Mise en œuvre pour FreeBSD

L'architecture Saturne se divise en quatre modules.

– **Le module de génération du flux :** Ce module est utilisé pour envoyer des paquets de sonde active dans le réseau. Il offre plusieurs méthodes de génération du flux. La distribution des inter-arrivées des paquets peut ainsi être poissonnienne, régulière, etc.

– **Le module d'estampillage** est un point central du module d'émission dans l'architecture Saturne. La plate-forme met en œuvre un mécanisme d'étiquetage pour ajouter des estampilles temporelles dans les paquets au moment de leur émission ou de leur réception dans le réseau. Le but ici est de s'assurer de la précision de la métrique employée. Le délai simple calculé à l'arrivée du paquet ne doit prendre en compte que le temps de trajet du paquet dans le réseau. Afin d'éviter les perturbations liées au gestionnaire de tâches du système et au temps passé dans les files d'attente des machines émettrices et réceptrices, l'estampillage se fait le plus tard possible chez l'émetteur et le plus tôt possible chez le récepteur.

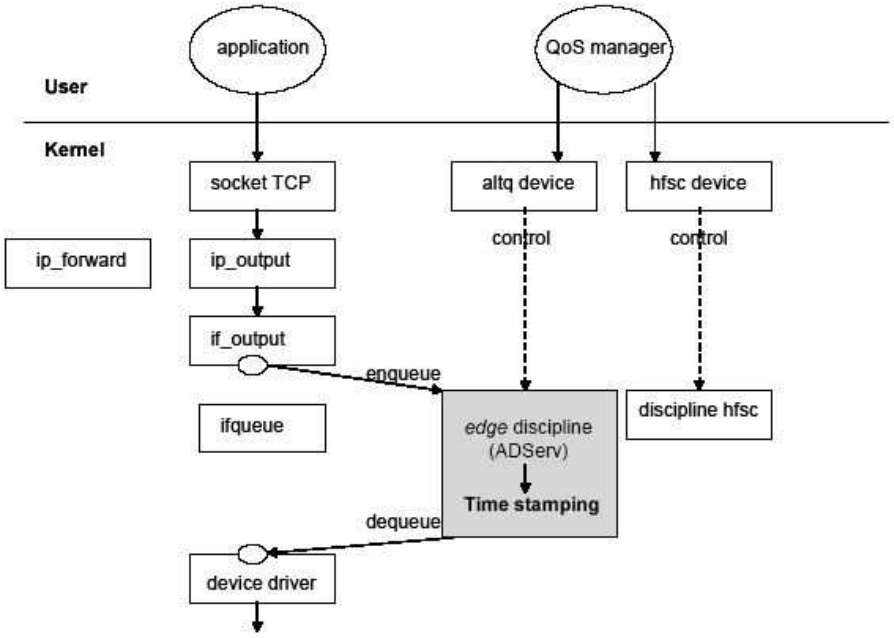


FIG. 8.6 – En-tête d'extension pour la métrologie

Les mécanismes d'étiquetage ont été développés sur des machines FreeBSD (cf. figure 8.6) et ont dans un premier temps été ajoutés aux logiciels ADServ [**?**] et ALTQ [**?**]. ALTQ (Alternative Queueing), développé par Kenjiro Cho , ajoute à FreeBSD la capacité de définir de

nouvelles techniques d'ordonnancement en modifiant les fonctions de file d'attente dans un grand nombre de pilotes d'interface réseau. ALTQ est particulièrement évolutif, chaque nouvelle politique d'ordonnancement est un module indépendant qui ne nécessite pas la modification des autres fonctions. Un grand nombre de techniques sont déjà disponibles dont CBQ, WFQ, RED ou RIO. ADServ [**?**], un module indépendant de la souche ALTQ, est la mise en œuvre du modèle de différentiation de service DiffServ. Ce module a été conçu pour faciliter la configuration des différents mécanismes composant l'architecture DiffServ. ADServ a été adapté pour pouvoir offrir la possibilité d'ajouter une étiquette dans des paquets sélectionnés. Dans le cadre de Saturne, cette fonctionnalité est utilisée pour ajouter une estampille de temps dans les paquets appartenant à un flux de sonde avant leur envoi sur le réseau. La valeur de cette estampille temporelle est donnée par le système, celui-ci étant synchronisé sur le temps global du GPS.

– **Le module de réception :** la récupération des estampilles temporelles est effectuée chez le récepteur du flux de sonde à l'aide de BPF (le Berkeley Packet Filter), un mécanisme destiné à spécifier aux couches réseau du noyau les paquets possédant des caractéristiques intéressantes. BPF est mis en œuvre sous la forme d'un interpréteur de langage à pile spécialisée construit dans un niveau bas du code réseau. Le noyau exécute le programme de filtre sur chaque paquet entrant. Si le noyau possède plusieurs applications BPF, l'ensemble des filtres sont exécutés pour chaque paquet.
– **Le module de traitement des données :** les données collectées sont stockées dans une base de donnée centralisée mysql. Elles sont également stockées dans des bases circulaires RRDTool ce qui permet d'afficher sur la page web http ://saturne.ipv6.rennes.enst-bretagne.fr l'évolution des mesures à la volée.

Des considérations spéciales doivent être prises en compte afin de mettre en œuvre l'extension de métrologie dans FreeBSD. La gestion des mbuf (*Memory Buffers*, structures de données utilisées dans le noyau au niveau des couches réseau pour gérer des informations diverses) est plus compliquée parce qu'il faut, après l'identification du paquet, insérer de l'espace pour l'extension afin de l'intégrer à la suite des définitions d'extension existantes sans perdre les données de niveau supérieur. La figure 8.7 donne un exemple de l'insertion de l'extension de métrologie. Tout d'abord l'en-tête UDP est replacé dans un nouveau mbuf avant que l'extension ne soit insérée.

La mise en œuvre de l'extension de métrologie IPv6 utilisant les mécanismes d'étiquetage développés sur des machines FreeBSD et ajoutés aux logiciels ADServ/ALTQ a été validée avec succès. Néanmoins, afin d'améliorer la flexibilité de l'architecture les fonctions d'étiquetage ont subies dans un second temps une amélioration et sont maintenant exécutées par le module Netgraph [**?**]. Netgraph est un sous système de gestion des couches réseau du noyau FreeBSD qui permet le traitement des paquets de façon indépendante. Il offre des routines permettant à l'utilisateur d'insérer ses propres fonctions. Ainsi, la classification de paquet et les fonctions d'estampillage ont pu être ajoutées dans le noyau. Des équipements de mesure peuvent maintenant installer le mécanisme d'estampillage de Saturne pour IPv6 en le chargeant comme un module indépendant au moment de l'exécution, évitant la recompilation de noyau exigée par l'environnement ADServ/ALTQ.

8.1.5 Expérimentations avec l'extension de métrologie

Cette section présente les expérimentations réalisées avec l'architecture Saturne et l'extension de métrologie. Les résultats obtenus montrent que le mécanisme de l'extension de métrologie IPv6 présent de meilleures performances en réduisant l'impact des méthodes de mesure actives tradition-

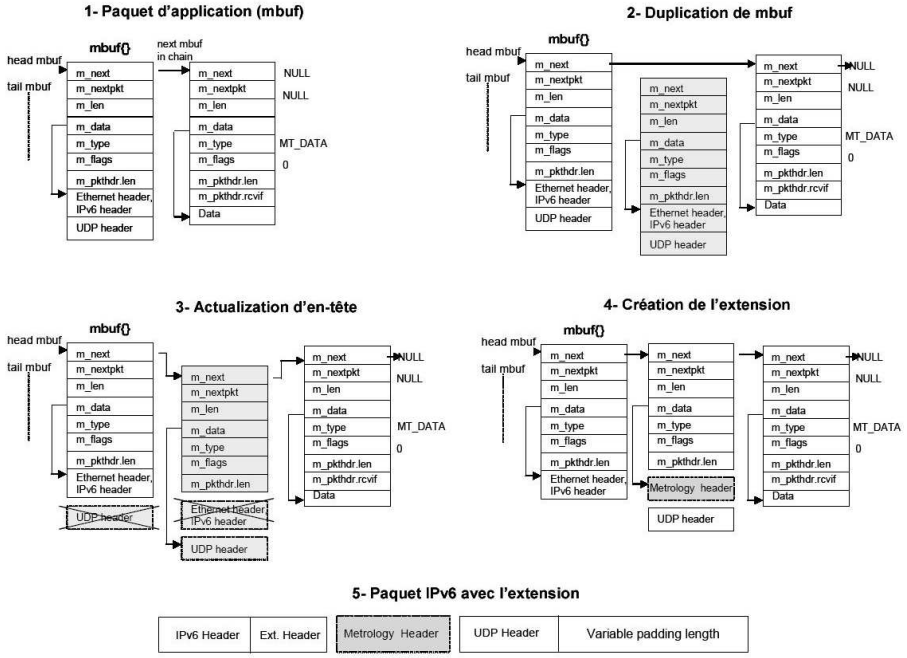


FIG. 8.7 – Insertion d'extension de métrologie

nelles. Nous avons mis en œuvre une plate-forme de test permettant d'expérimenter deux types de médias : Bluetooth [**?**] et GPRS [**?**]. Les premiers résultats obtenus dans le cadre de ces expérimentations sont présentés dans les paragraphes suivants. Dans les réseaux sans fil, on peut observer des variations importantes de l'OWD et de la gigue. Ces variations peuvent être liées à la politique d'allocation de ressource ou à l'utilisation des canaux radio. Les variations perturbent les mécanismes de contrôle de flux utilisés par TCP et sont à l'origine de la détection fausse de pertes ayant pour conséquence des retransmissions inutiles. Ce comportement pénalise les utilisateurs en particulier dans le contexte des réseaux mobiles puisque la facturation est basée sur la quantité de données transférées. L'étude et la compréhension des comportements de la transmission radio, bien qu'elle permette à des opérateurs et à des fabricants d'améliorer leurs équipements, ne sont pas essentielles pour les développeurs de logiciels souhaitant adapter leurs logiciels et protocoles de transport pour un environnement sans fil (par exemple GPRS). En revanche, les mesures de bout en bout peuvent fournir une évaluation du service obtenu par l'utilisateur.

8.1.6 Expérimentations sur Bluetooth

La technologie sans fil Bluetooth, développée dans les années 90, s'est rapidement imposée comme standard industriel pour doter les ordinateurs et dispositifs mobiles de capacité de connexion sans fil à courte portée. Elle permet la communication d'appareils compatibles Bluetooth (PDA, imprimeurs, téléphone mobile, etc.) dans un rayon de quelques mètres [**?**]. Nous avons testé les

principaux protocoles basés sur IP (ICMP, UDP et TCP) au-dessus de Bluetooth afin d'analyser son comportement.

Description de la plate-forme

La plate-forme de mesure utilisée est décrite dans la figure 8.8. Le transport des paquets IPv6 dans des paquets IPv4 est décrit dans [**?**]. Il existe deux modes de tunnels, point à point qui établit une liaison fixe entre deux machines et automatique qui permet à une machine isolée d'accéder à toutes les autres machines à l'aide des adresses IPv4 compatibles. L'utilisation de tunnel est justifiée par le support Bluetooth pour IPv4.

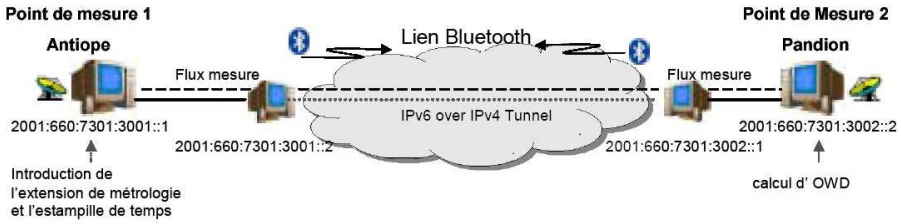


FIG. 8.8 – Plate-forme de mesure IPv6 (Bluetooth)

ICMP : Mesure de ping

Le principe de cette expérimentation est de mesurer l'OWD d'un flux applicatif, comme le ping. Les deux points de mesure sont interconnectés par un lien Bluetooth (cf. figure 8.8). L'expérimentation cherche à mesurer l'OWD d'un flux de ping pour s'assurer que les résultats de nos mesures sont similaires à ceux obtenus pour le flux de ping. La mesure d'OWD a été effectuée dans le trajet aller/retour. Cette étude a pour objectif de vérifier la précision des mesures de RTT et l'impact de la symétrie des chemins dans les liens bas débit. Cette expérimentation se compose de cinq séries de mesure de cent demandes de ping, les paquets de taille fixe commençant à 128 octets et augmentant progressivement (128, 256, 384, 512, 640, 768, 896, 1024, 1152 et 1280) (cf. tableau 8.9)

Les mesures du ping permettent une surveillance peu intrusive de l'état du réseau en donnant les temps de traversées aller/retour de celui-ci. Néanmoins elles offrent une perspective limitée de la performance de réseau car la symétrie de chemins entre la source et la destination ne peut pas être assurée avec le ping. Le tableau 3 démontre ce comportement. La somme des valeurs d'OWD est proche du RTT comme nous pouvons l'observer dans la figure 8.10.

UDP : Mesure d'un flux VoIP

Ce test a été conçu pour mesurer le trafic émulant une application de voix sur IP (VoIP), un flux UDP est émis à débit constant (CBR) avec une taille de paquet fixe de 100 bytes, il ne comprend pas la taille des entêtes (40 octets pour IPv6, 16 octets pour l'extension de métrologie, et 8 octets pour

Taille du Paquet octets	OWD requête microsecondes	OWD répond microsecondes	Total OWD	RTT	RTT – Total OWD
128	65.210913	9.62369	74.834603	75.164	0.329397
256	96.953712	10.92644	107.880152	108.246	0.365848
384	121.62508	12.227758	133.852838	134.169	0.316162
512	157.298302	13.570377	170.868679	171.204	0.335321
640	186.903188	15.883643	202.786831	203.115	0.328169
768	224.046786	19.247286	243.294072	243.624	0.329928
896	246.266895	22.223412	268.490307	268.809	0.318693
1024	274.212753	21.47178	295.684533	296.007	0.322467
1152	314.946675	24.978729	339.925404	340.292	0.366596
1280	341.294857	27.04387	368.338727	368.716	0.377273

FIG. 8.9 – Tableau 3. Résultats des mesures d'OWD et ping (Bluetooth)

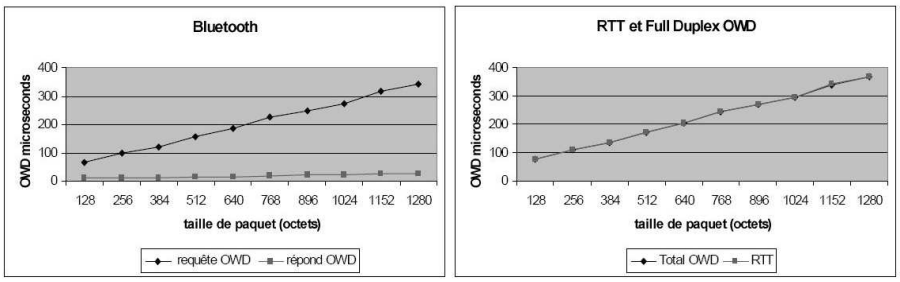
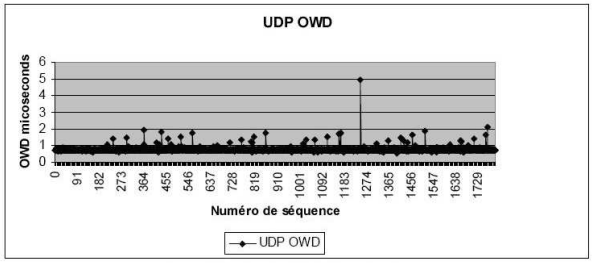


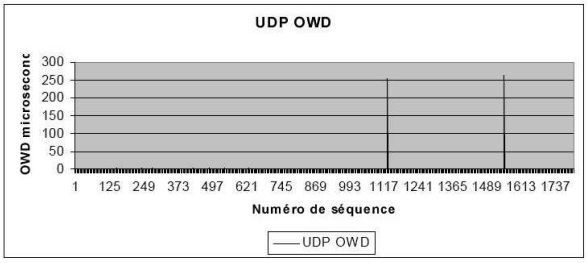
FIG. 8.10 – Résultats des mesures d'OWD du ping (Bluetooth)

UDP). Nous pouvons observer l'importance des en-têtes des protocoles par rapport à la quantité d'information utile transportée, ce qui justifie les efforts considérables pour normaliser les techniques de compression d'entête [**?**][**?**]. La mesure d'OWD a été effectuée pour des trajets aller/retour pendant un intervalle de 180 secondes (3 minutes). Le comportement de l'OWD présente différentes caractéristiques selon le trajet mesuré. Dans le premier cas (cf. figure 8.11), la valeur d'OWD est fortement variable mais aucune perte n'est rapportée. La valeur d'OWD est plus stable mais le pourcentage de perte est 19.44dispositifs Bluetooth et leur différences techniques (l'adapteur USB et le l'interface PCI).



Moyenne	0.734381458
Ecart-type	0.139873093
Intervalle de confiance 95%	0.00073664
Perte %	0

Point de mesure 1 -> Point de mesure 2



Moyenne	0.783545
Ecart-type	8.55297307
Intervalle de confiance 95%	0.0008247
Perte%	19.44

Point de mesure 2 -> Point de mesure 1

FIG. 8.11 – Résultats des mesures d'OWD de VoIP (Bluetooth)

TCP : Mesure d'un flux FTP

Ce test mesure le flux d'une connexion ftp pour le transfert d'un fichier de 1Mo. Le but de cette expérimentation est d'analyser le comportement d'un flux TCP au-dessus d'un lien bas débit et les mécanismes de contrôle de flux pour éviter une saturation des ressource du réseau [**?**][**?**]. La mesure a été effectuée dans une seule direction. La figure 8.12 montre la courbe d'OWD des paquets applicatifs TCP. On peut facilement constater que les valeurs des premiers 1600 paquets présentent une forme de vague. La première constatation est que le lien entre ce comportement est trouvée dans la performance du routeur, car la file d'attente est saturée, ceci est la cause de la perte de paquet, appréciée dans la deuxième courbe de la figure. Le mécanisme de control de flux de TCP s'active déclenchent le mécanisme de slow-start (paquet numéro 1600) La seconde constatation est l'adaptation de la performance de TCP aux restrictions de la bande passante disponible. Après la phase d'adaptation de TCP, la performance sa meilleure et la perte de paquet est réduite considérablement.

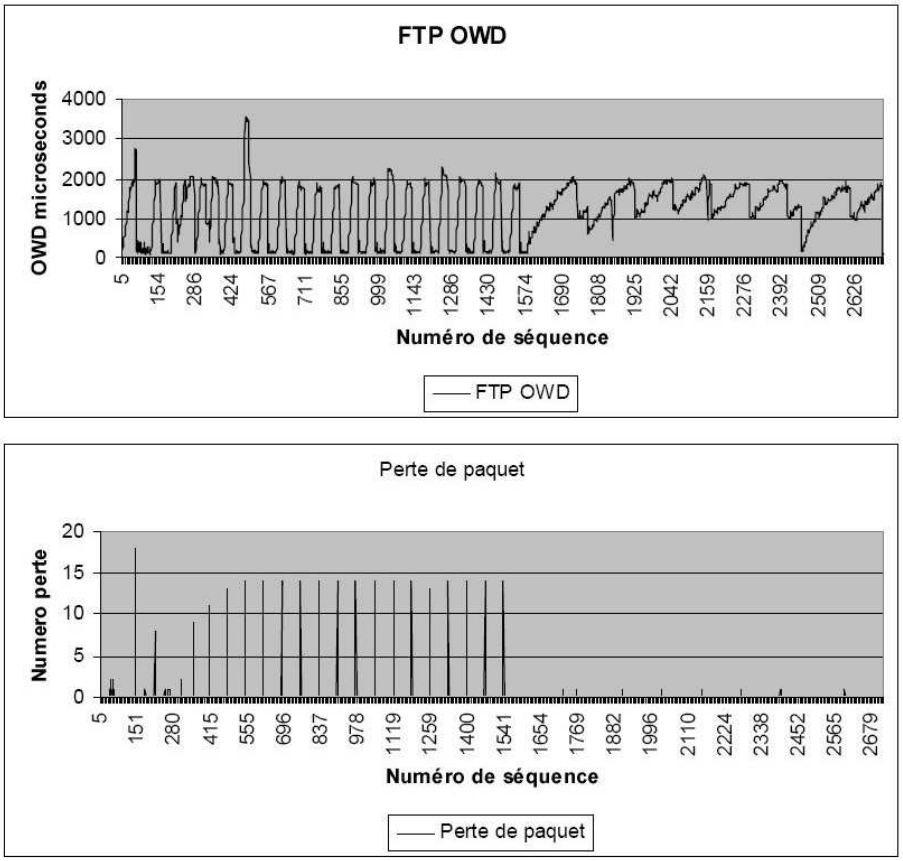


FIG. 8.12 – Résultat des mesures d'OWD d'un flux FTP (Bluetooth)

8.1.7 Expérimentations sur GPRS

Le service général de radiocommunication en mode paquet (General Packet Radio Service) est un service de transmission de données par radio, utilisant la commutation de paquets, offert par un réseau de téléphonie mobile de type GSM. Le service offre l'accès Internet (IPv4) qui disposera d'une bande passante pouvant théoriquement atteindre 115 Kbit/s, limitée dans le système actuel à 9,6 Kbit/s. Mais les opérateurs vont proposer des débits assez proches de ceux disponibles sur le réseau téléphonique filaire (environ 50 Kbit/s). Nous avons répété les tests présentés précédemment en remplaçant le lien Bluetooth par une connexion GPRS. En raison du coût du transfert de données, la durée du test et la taille des paquets ont été réduits.

Description de la plate-forme

La figure 8.13 présente l'architecture de mesure utilisée. Un lien PPP est créé entre le nœud mobile et la passerelle GPRS qui donne accès à l'Internet. L'adresse IPv4 attribuée au nœud mobile est privée, l'accès Internet est donc souvent restreint.

Afin d'obtenir connectivité IPv6 de bout en bout, un mécanisme de tunnel appelé L2TP (Layer 2 Tunelling) [**?**] est utilisé. Le but de ce protocole est de prolonger la connectivité offerte par PPP de la couche 2 vers le fournisseur d'accès Internet. Le tunnel L2TP permet aux paquets de niveau 3 (IPv4 ou IPv6) de voyager d'une manière transparente du nœud mobile vers l'Internet au-dessus du service GPRS et du réseau IPv4. Notez que le MTU sur le chemin de GPRS est considérablement réduit à cause l'encapsulation IPv6 au-dessus d'IPv4.

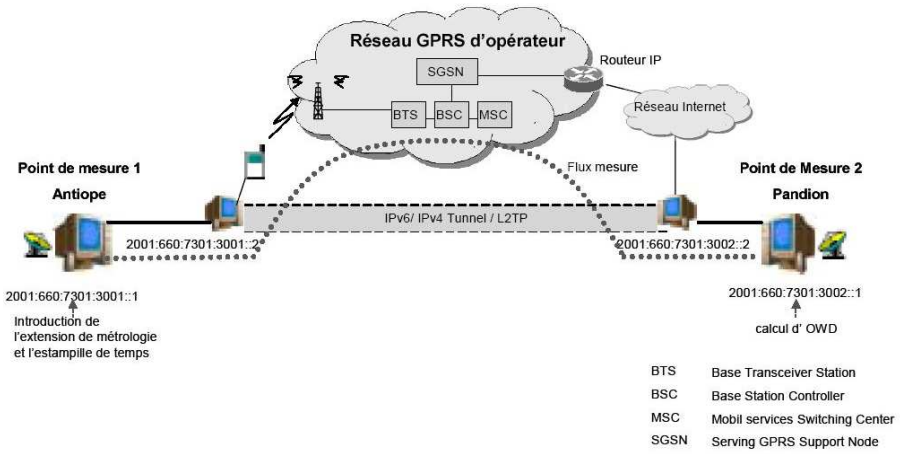


FIG. 8.13 – Plate-forme de mesure IPv6 (GPRS)

ICMP : Mesure de ping

Le principe de cette expérimentation est de mesurer l'OWD d'un flux applicative comme le ping. Les deux point de mesure sont interconnectés à travers de un lien GPRS. L'expérimentation consiste

en mesure l'OWD d'un flux ping et s'assurer que les résultats des mesures et ces résultats du ping sont similaires. Nous avons comparé le RTT calculé par le ping à nos mesures. Les paquets avec une taille fixe commençant à 64 bytes et puis augmente leur taille progressivement (64, 96, 128, 160 et 192) des bytes (cf. tableau 8.14)

Taille du Paquet octets	OWD requête microsecondes	OWD répond microsecondes	Total OWD	RTT	RTT – Total OWD
64	652.815379	295.620725	948.436104	948.707	0.270896
96	657.145108	317.657086	974.802194	975.098	0.295806
128	725.791769	324.000174	1049.791943	1050.083	0.291057
160	726.575237	351.769508	1078.344745	1078.643	0.298255
192	743.738251	310.099669	1053.83792	1054.136	0.29808

FIG. 8.14 – Résultats des mesures d'OWD du ping (GPRS)

L'asymétrie de chemin est confirmée encore, nous observons que la différence entre la demande d'OWD et la réponse d'OWD ne préserve pas une relation linéaire La somme des valeurs d'OWD est proche du RTT comme nous pouvons le constater dans la figure 8.15.

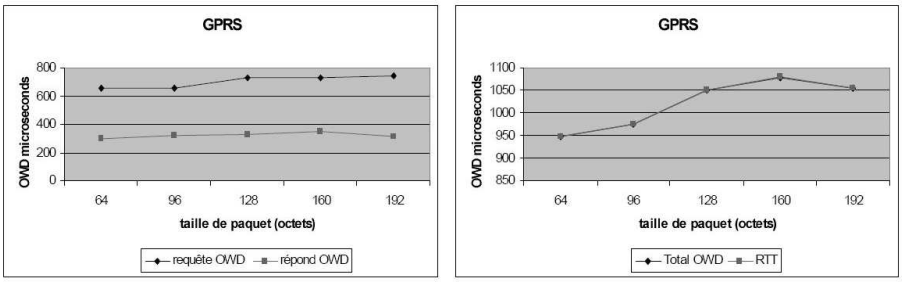


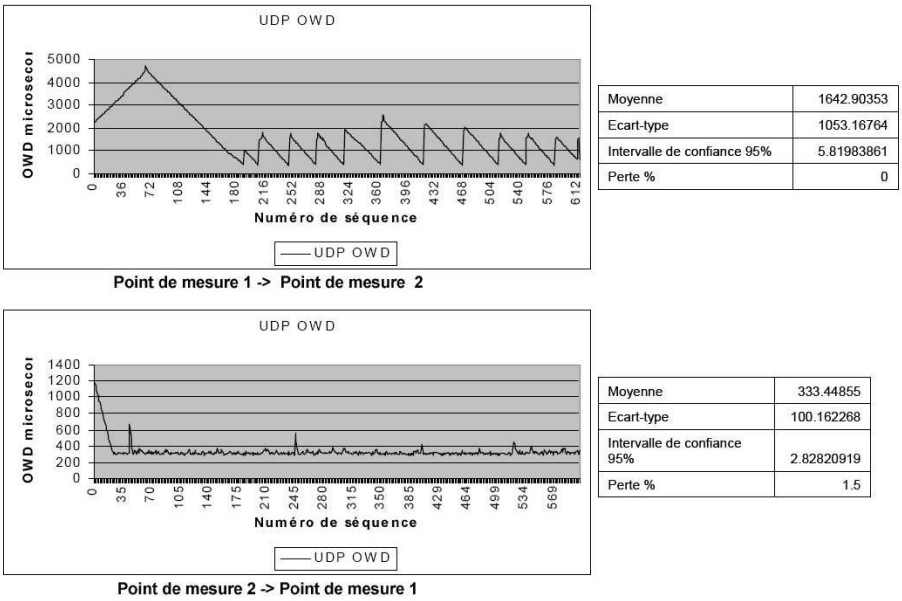
FIG. 8.15 – Résultats des mesures d'OWD du ping (GPRS)

UDP : Mesure d'un flux VoIP

La figure 8.16 montre le comportement de l'OWD d'un flux UDP de VoIP décrit précédemment à travers le réseau GPRS. Dans ce test le débit fixe (1ko/s) et la taille des paquets (100 octets) sont les mêmes que ceux utilisés dans le test Bluetooth. Le comportement de l'OWD présente des différentes caractéristiques selon le sens du trajet mesuré. Il est intéressant également de noter que l'OWD préserve l'asymétrie constatée dans le test Bluetooth.

TCP : Mesure d'un flux FTP

Pour réaliser cette expérimentation, nous employons la méthodologie décrite dans la section 8.1.6. Ce test mesure le flux d'une connexion ftp pour un transfert d'un fichier de 100k, la valeur maximale de MTU étant de 256.



Moyenne	1642.90353
Ecart-type	1053.16764
Intervalle de confiance 95%	5.81983861
Perte %	0

Moyenne	333.44855
Ecart-type	100.162268
Intervalle de confiance 95%	2.82820919
Perte %	1.5

FIG. 8.16 – Résultats des mesures d'OWD de la VoIP (GPRS)

La figure 8.17 montre la courbe d'OWD des paquets TCP. La première constatation est le lien entre la perte de paquet et les mécanismes de control de flux TCP. L'efficacité qu'on observe est moins performante que celle observée dans le cadre du test Bluetooth, et le mécanisme de retransmission prend plus de temps à s'activer.

8.1.8 Architecture de mesure

Avec l'introduction de l'extension de métrologie, la mesure active ou passive peut être effectuée entre presque n'importe quelle paire de points de mesure, sous réserve d'une bonne synchronisation des machines. L'utilisation d'équipements ou de matériels dédiés à la mesure active ou passive peut être remplacée par la mesure au niveau logiciel. L'utilisation de l'extension de métrologie apporte un éventail de possibilités pour effectuer la mesure active ou passive. Cette section présente quelques architectures pouvant être mises en œuvre, chacune offrant une alternative à l'évaluation des performances de réseau.

Mesure de bout en bout par des applications

L'évaluation active de bout en bout des performances d'un réseau est l'approche la plus populaire pour évaluer la qualité de service (QoS) perceptible par l'utilisateur. Son efficacité a été démontrée et en fait une alternative sérieuse aux méthodes de mesure passive, plus sophistiqués et coûteuses. La figure 8.18 montre une représentation de cette architecture. L'avantage de cette architecture de mesure réside dans l'évaluation des performances de QoS observée par l'utilisateur final. Par contre,

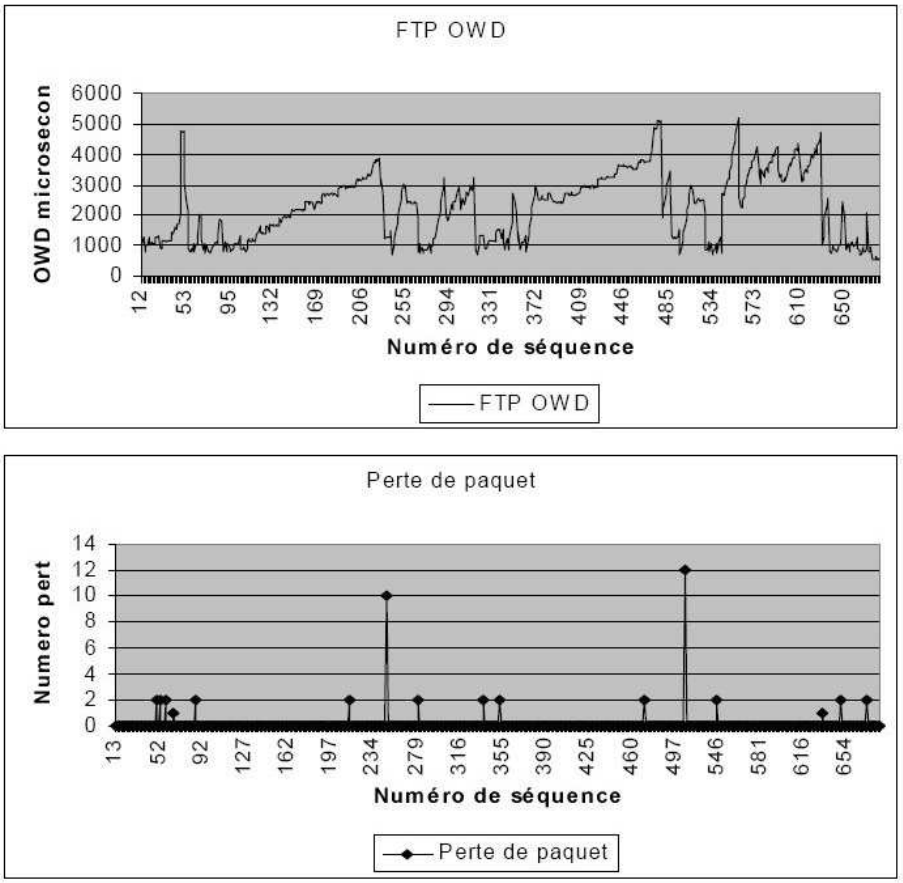


FIG. 8.17 – Résultat des mesures d'OWD d'un flux FTP (GPRS)

puisque la synchronisation d'horloge est une des restrictions les plus importantes dans l'évaluation d'OWD, l'utilisation obligatoire des systèmes de synchronisation (GPS) peut être un handicap important dans l'utilisation de cette approche.

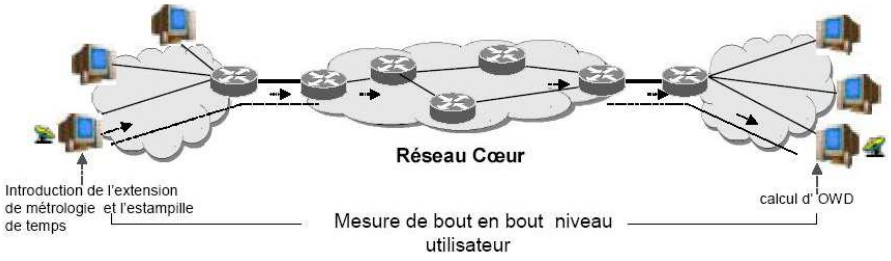


FIG. 8.18 – Evaluation de performance de bout en bout

.

Mesure de bout en bout assistée

Cette architecture peut être employée comme une alternative quand les mécanismes efficaces de synchronisation sont limités ou bien ne sont disponibles que dans quelques endroits placés à la frontière ou en extrémité du réseau. Dans cette architecture :

- Les nœuds finaux n'ont pas de source directe pour la synchronisation de temps (GPS ou radio) ;
- L'application mesurée peut générer elle-même l'en-tête d'extension de métrologie ;
- L'extension de métrologie insérée par le nœud de mesure est initialement vide ;
- Le nœud suivant dans le chemin est équipé d'une source précise de temps et insère l'estampille dans l'extension initialement vide en pleuvant ajouter plus d'information de mesure.

L'application envoie son flux avec une extension vide. Le nœud suivant qui dispose d'un mécanisme pour la synchronisation détecte l'extension de métrologie vide et met à jour l'estampille de temps. Le calcul d'OWD est exécuté dans le nœud de mesure de destination (cf. figure 8.19). Le résultat d'OWD est alors inséré dans l'extension de métrologie en remplaçant l'estampille d'émission avec la valeur d'OWD. Le paquet arrive au nœud final où l'information de métrologie (l'OWD) peut être employée par l'application pour améliorer ses performances.

Mesure edge to edge transparente intra-domaine

L'estampillage peut également être exécuté par les routeurs d'entrée d'un domaine, afin d'effectuer la mesure à l'intérieur d'un domaine réseau. A l'arrivée du trafic applicatif dans le domaine de réseau, le routeur d'entrée insère l'extension de métrologie pour dans ce flux. Le routeur de sortie du domaine capture l'estampille de temps et calcule l'OWD avant d'enlever l'extension et de laisser le paquet continuer son chemin.

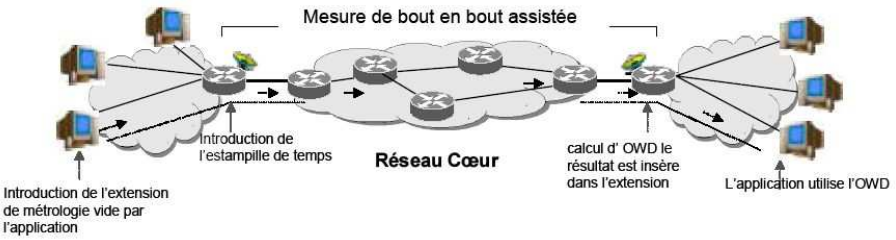


FIG. 8.19 – Mesures de bout en bout assistées

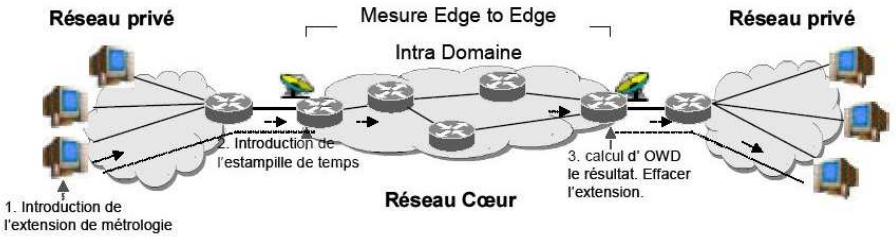


FIG. 8.20 – Mesure transparente intra-domaine

Cette architecture permet aux fournisseurs d'accès et aux ISPs de mesurer la performance de leur réseau d'une manière indépendante et transparente pour les applications et les utilisateurs, émulant l'approche passive de mesure (cf. figure 8.20).

Mesure inter-domaine

L'extension de métrologie peut être attachée à n'importe quel paquet IPv6, dans n'importe quel nœud du réseau mettant en œuvre ce mécanisme, la mesure peut ainsi être exécutée entre deux nœuds IPv6 quelconques. Si les fournisseurs de réseau veulent améliorer leurs services réseau en offrant une qualité de service globale, il est nécessaire d'avoir une vue globale de la performance de la mise en œuvre de cette qualité de service. La figure 8.21 montre comment une architecture de mesure peut être naturellement adaptée pour mesurer des paramètres de performance de réseau dans une approche d'inter-domaine.

Cette architecture exige des accords entre les différents domaines et les ISPs doivent établir les conditions de base afin d'effectuer la mesure.

Conclusion

Les mesures passives ont pour but de mesurer les performances réelles des flux applicatifs en observant le trafic qui traverse et d'étudier ses propriétés en un ou plusieurs points de réseau mais nécessitent en général de lourds investissements. Le principe des mesures actives est de générer un

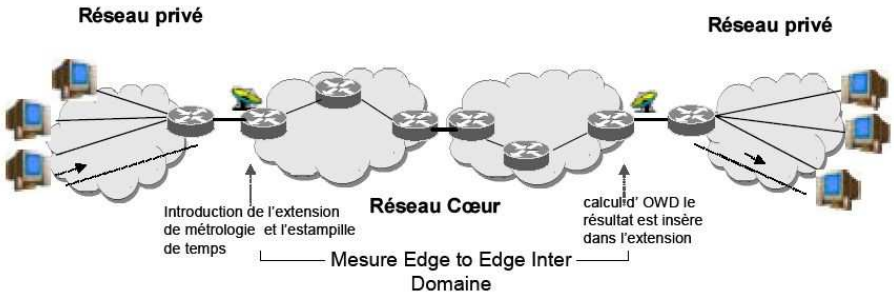


FIG. 8.21 – Mesures inter-domaine

flux de test qui se mêlant au trafic transitant sur le réseau. Cependant, la mesure active présente l'inconvénient de pouvoir perturber le comportement du réseau par l'introduction d'un trafic de sonde.

L'architecture Saturne représente une alternative intéressante aux équipements de mesure passive coûteux et aux outils commerciaux de mesure active. Outre une architecture de mesure active classique, Saturne s'est adapté à IPv6 et a su en tirer parti pour définir une extension de métrologie permettant la mesure de n'importe quel paquet applicatif de façon simple et transparente dans le réseau.

La série d'expérimentations a été menée et les premiers résultats ont été présentés, validant l'efficacité du mécanisme développé et sa pertinence. Ce nouveau mécanisme permet aussi d'envisager des scénarios de mesures différentes des architectures classiques de mesure de bout en bout et constitue le trait d'union entre les deux grandes méthodologie divisant le domaine de la mesure dans les réseaux : les méthodologies actives et passives. Le développement de l'architecture Saturne s'oriente maintenant plus sur le perfectionnement des techniques d'échantillonnage des paquets mesurés (en génération pour le principe actif et en sélection pour le principe passif) et l'exploitation des mesures effectuées.

Chapitre 9

Echantillonnage actif : Techniques d'Estimation de Bande Passante :État de l'Art

[1]

9.1 INTRODUCTION

Les réseaux Internet de nouvelle génération (New Generation Internet ou NGI) verront comme la norme la convergence de services, la mobilité et la convergence fixe/mobile, la qualité de service et la connectivité variable sur différents types de terminaux et de réseaux d'accès. L'explosion de nouveaux services et la disponibilité de réseaux d'accès innovants fixes et mobiles fournissant une connectivité à haut débit rendent très difficile la prédiction du trafic et les performances des réseaux Internet nouvelle génération.

Des nouvelles approches de conception, dimensionnement et optimalisation plus adaptative sont nécessaires pour mieux contrôler et gérer ces nouveaux réseaux. Vu que le trafic sera de plus en plus difficile à modéliser l'approche d'ingénierie de trafic basée sur mesures semblent être le plus adéquat et donc des techniques d'échantillonnage et estimation des ressources disponibles doivent être développées. En particulier pour les métriques liées à la bande passante.

Même si les bandes passantes dans les réseaux de cœur et d'accès sont augmentées, l'optimalisation et l'allocation dynamique de la bande passante reste encore un problème très intéressant de résoudre pour continuer à assurer la qualité de service requise par les nouvelles applications supportées par ces nouveaux réseaux.

La bande passante est un des ressources le plus important des réseaux à paquets. Une fois le réseau est en état opérationnel, elle est parfois la plus chère. En autre, le dimensionnement d'un réseau consiste aussi à déterminer la quantité de bande passante ou capacité nécessaire pour fournir

[1] par César Cardenas et Ramon Casellas

les services demandés avec la qualité de service requise par ses utilisateurs. Pour dimensionner un réseau il est donc très important de faire du monitoring de métriques liées à la bande passante (capacité du chemin réseau, bande passante disponible du chemin réseau, et débit) a fin de connaître son utilisation, sa dynamique, pour ensuite les maîtriser et les optimiser.

Le défi de l'approche basé sur mesures est d'estimer les métriques liées à la bande passante d'une manière précise, rapide, robuste et presque sans obstruction. Robuste veut dire qu'il est applicable dans la plus part d'environnements dans l'Internet : peut ou plusieurs liens de la source à la destination, des canaux vides ou saturés, un ou plusieurs canaux par lien, différentes technologies fixes et sans fils, différentes disciplines d'attente, et différentes implémentations de routage. Non obstructive veut dire qui ne demande une charge additionnelle minimale au réseau.

Ce rapport présente l'état de l'art des techniques d'estimation des métriques liées à la bande passante. La structure générale est la suivante : nous allons commencer par l'aspet normalisation et définitions ensuite, les classification des techniques puis, nous allons continuer en décrivant les techniques actives, les techniques passives, les techniques hybrides. La derni‘ere partie présentera la bibliographie de référence.

9.2 NORMALISATION ET DÉFINITIONS

En ce qui concerne à la normalisation, le travail de l'IETF (Internet Engineering Task Force) lié à ce rapport se trouve aux groupes : IPPM (Internet Protocol Performance Metrics), PSAMP (Packet Sampling) et IPFIX (Internet Protocol Fow Information Export).

Nous allons décrire d'une manière générale le travail de chacun de ces groupes, en particulière depuis le point de vue des définitions et architectures. Ce qui nous donnera un contexte formel pour ce rapport. Ensuite nous allons complementer cette partie avec d'autres définitions utilisées dans ce rapport.

9.2.1 IPPM

Le groupe de travail IPPM [**?**] définit des métriques spécifiques et leur procedures pour les mesurer d'une manière précise et les documenter. Les métriques peuvent être appliquées à la qualité, performance, et disponibilité des réseaux Internet. Les métriques peuvent être executées par les opérateurs réseaux ou les utilisateurs. Les métriques n'indiquant pas un jugement au contraire elles produisent des mesures quantitatives impartielles sur le performance. Quelques métriques définies sont : la connectivité, le délai et les pertes dans un sens, le délai et les pertes aller-retour, la variation du délai, les modèles de pertes, le re-organisation de paquets, la capacité de transfert et la bande passante du lien.

Équipement Hôte (Host)

Un ordinateur capable de se communiquer en utilisant les protocoles d'Internet ; y compris les routeurs.

Lien (Link)

Connexion entre deux ou plus hôtes au niveau de la couche liaison ; y compris les lignes louées, réseaux ethernet, nuages "Frame Relay", etc.

Routeur (Router)

Un hôte que facilite la communication au niveau de la couche réseau entre hôtes en acheminant paquets IP.

Chemin Réseau / Route (Path)

Une séquence $\{h_0, l_1, l_2, \ldots, l_n, h_n\}$, d'où $n \geqslant 0$, chaque h_i est un hôte, chaque l_i est un lien entre h_{i-1} et h_i, chaque $h_1, \ldots, h_{n-1}$ est un routeur. Une paire $< l_i, h_i >$ est défini comme un saut (hop). Les liens et le routeurs dans le chemin réseau facilitent la communication de paquets au niveau de la couche réseau de h_0 à h_n. Notez que le concept de chemin réseau est dans une direction.

Sous-Chemin Réseau (Subpath) :

Étant donné un chemin réseau, un sous-chemin réseau est une sous-séquence d'un chemin réseau donné laquelle est elle-même un chemin réseau et pour laquelle le premier et dernier élément d'un sous-chemin réseau est un hôte.

Nuage/Aire (Cloud)

Graphe non dirigé (possiblement cyclique) d'où les routeurs sont sommets et les liens sont les bords que connectaient paires de routeurs. Formalement, les réseaux ethernet, les nuages "Frame Relay", et autres liens que connectaient plus de deux routeurs sont modelés comme maillages de bords des graphes. Se connecter à un nuage signifie se connecter à un routeur du nuage sur un lien ; ce lien n'appartient pas au nuage.

Sous-chemin Nuage (Cloud Subpath) :

Un sous-chemin d'un chemin d'où les hôtes sont routeurs d'un nuage donné.

Chemin Simplifié (Path Digest)

Une séquence $\{h_0, e_1, C_1, \ldots, e_n, h_n\}$, d'où $n \geqslant 0$, h_0 et h_n sont hôtes, chaque $e_1 \ldots e_n$ est un échange, et chaque $C_1 \ldots C_{n-1}$ est un sous-chemin nuage.

Lien de type Échange (Exchange)

Un cas spécial d'un lien, un échange connecte un hôte à un nuage et/ou un nuage à un autre nuage.

Métrique

En état opérationelle, ils existaient plusieurs quantités liées aux performances et disponibilité des réseaux l'Internet dont nous voudrions connaître les valeurs. Si cette quantité est clairement spécifie, alors nous l'appelons métrique.

La difficulté de mesurer une métrique est permit mais l'ambiguité en signification non. Chaque métrique doit être définie en unités standards de mesure. Dans ce cas le système métrique international est utilisé.

Types de métriques :

- Métrique Singleton : Ensemble de métriques de taille égale à l'unité.
- Métrique Échantillon : Métriques dérivées à partir des métriques singletons en prennant un ensemble de résultats.
- Métriques Statistiques : Métriques dérivées à partir des métriques échantillons en calculant une statistique des valeurs définies par les métriques singleton de l'échantillon.

Il est fortement récommandé d'eviter les définitions en termes stochastiques car quand les définitions sont faites par rapport aux probabilités ils existaient des considerations cachés pour la métrique à mesurer.

Métriques Analytiques :

Au cas une métrique est comprise en utilisant des concepts analytiques ; la métrique sera nommée "Métrique Analytique". Examples :

- Temps de Propagation d'un Lien : Temps, en sécondes, nécessaire pour qu'une unite binaire (bit) traverse un lien depuis le port de sortie d'un hôte vers un autre hôte au réseau Internet.
- Bande Passante d'un Lien pour Paquets de Taille k : Capacité, en unités binaires par séconde, pour paquets IP de taille k.
- Route : Chémin réseau à un instant donné.
- Compteur du lien appartenant à une route : La valeur "n" d'un chemin réseau.

Métriques Empiriques :

Métriques que ne peuvent pas être expresées dans une forme analytique. Elles ont les propriétés suivantes :

- Définition claire en termes de composants de réseau Internet.
- Il doit exister au moins une façon pour les mesurer.
- Il doit avoir une comprehension analytique (non complete) de la métrique d'une manière qu'il est possible d'utiliser les mesures pour raisoner sur la performance et confiabilité des composants analytiques et les agrégations des composants analytiques.

Méthodologie de Mesure

Ils existaient plusieurs méthodologies de mesures pour un ensemble de metriques bien définies, par example :

- Mesure directe d'une métrique en injectant du trafic test.
- Contruction d'une métrique à partir de mesures de niveaux inférieurs.

– Estimation d'une métrique composée à partir d'un ensemble de plusieurs mesures agrégrées.
– Estimation d'une métrique donnée en certain temps à partir d'un ensemble de métriques reliées en autres temps.

Propriétés des méthodologies

– Répétibilité (Consistence) : Si la méthodologie est utilisée plusieurs fois sous les mêmes conditions, les résultats doivent être consistents.
– Continuité : Une méthodologie de mesure est continue si, pour petites variations dans les conditions, les mesures montrent aussi petites variations. Une métrique a la propriété de continuité si au moins une de leur méthodologies l'a aussi.
– Conservativité : Le fait de mesurer ne modifie pas ou modifie un petit peu la valeur de la métrique de performance.

Objectifs des conceptualiseurs de méthodologies

– Minimiser les incertitudes/erreurs des méthodologies.
– Comprendre et rapporter les sources des incertidumbres/erreurs des méthodologies.
– Quantifier les incertidumbres/erreurs des méthodologies.

Méthodes d'Échantillonnage

Le but d'échantillonnage est d'observer les variations et consistences de la métrique mesurée .

– Échantillonnage Périodique : Soufre des problèmes de synchronisation et périodicité.
– Échantillonnage Aléatoire : Les échantillons sont séparés par intervals indépendants générés d'une manière aléatoire avec la même distribution statistique. Il ne soufre pas du problème de synchronisation mais au contraire complique l'analyse fréquentielle et si la distribution est exponentielle l'échantillonnage est prévisible.
– Échantillonnage Poissonien : Échantillonnage aléatoire avec distribution exponentielle.
– Échantillonnage Géométrique : Les événements sont mesurés avec une probabilité fixe (p).

Les méthodes d'échantillonnage seront aussi décrites plus bas.

Test de Qualité (Goodness of Fit)

La nature de ce test consiste en choisir un niveau de signification, lequel est la probabilité que le test déclare d'une manière érronée que la fonction de distribution émpirique ou EDF (Empirical Distribution Function) d'un ensemble de mesures suivent une certaine distribution de probabilité.

9.2.2 IPFIX

Le groupe de travail de l'IETF IPFIX [**?**] définit une architecture pour la surveillance (monitoring), mesure et exportation de l'information des flux de paquets IP d'une manière temporelle, depuis un exportateur vers une station ou collection de stations. La figure 9.1 montre la séquence de processus exécutés sur un dispositif IPFIX.

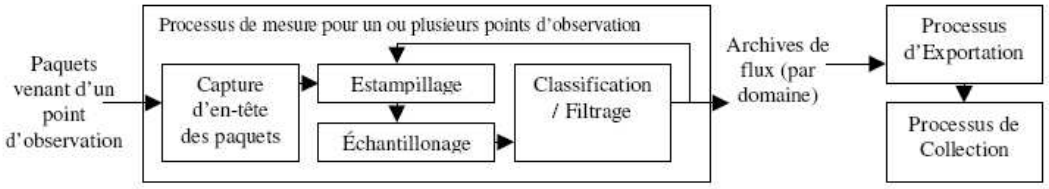


FIG. 9.1 – Architecture IPFIX

Point d'Observation

Éndroit du réseau d'où les paquets IP peuvent être observés.

Domaine d'Observation :

Un domaine d'observation est le plus grand ensemble de points d'observation pour lesquels l'information de flux peut être agrégrée avec un processus de mesure.

Flux ou Flux de Trafic IP

Ensemble de paquets traversant un point d'observation sur le réseau pendant un interval de temps. Tous les paquets appartenant à un flux particulier ont propriétés similaires. Chaque propriété est le résultat d'appliquer une fonction aux valeurs suivantes :

- Une ou plusieurs en-têtes des paquets, en-têtes au niveau transport ou en-têtes au niveau application.
- Une ou plusieurs caracteristiques du paquet lui-même.
- Une ou plusieurs champs dérivés du traitement des paquets.

Un paquet appartient à un flux si remplit toutes les propriétés définies pour le flux.

Archive de flux (Flow Record) :

Un archive de flux contient l'information d'un flux observé sur le point d'observation. Il contient propriétés mesurées du flux et normalement propriétés caracteristiques du flux. Les archives de flux sont générées par un ou plusieurs processus de mesure.

Processus de Mesure

Le processus de mesures génère archives de flux. Il consiste en un ensemble de fonctions que comprenne la capture de l'en-tête du paquet, l'estampillage, l'échantillonnage, la classification et le mantient des archives de flux.

Processus d'Exportation

Le processus d'éxportation envoi les archives de flux vers un ou plusieurs processus de collection.

Exportateur :

Dispositif accuiellant un ou plusieurs processus d'exportation.

Dispositif IPFIX

Un dispositif IPFIX a au moins un point d'observation, un processus de mesure et un processus d'exportation.

Processus de Collection

Un processus de collection reçoit archives de flux depuis un ou plusieurs processus d'exportation.

Collecteur :

Dispositif avec un o plusieurs processus de collection.

"Template"

Séquence ordonnée de paires <type, longueur> utilisée pour identifier la structure et la semantique d'un ensemble d'information nécessitant se communiquer depuis un dispositif IPFIX vers un collecteur.

Message IPFIX

Un message IPFIX est originé au processus d'exportation qu'emporte les archives IPFIX vers le processus de collection.

Critères de Sélection de Paquets

Le processus de mesure défine regles de tal manière que seulement certains paquets dans l'écoulement de paquets sot choisis d'être mesurés dans un point d'observation : fonctions d'échantillonnage et fonctions de filtrage.

Fonctions d'Échantillonnage :

Les fonctions d'échantillonnage déterminent les paquets à sélectionner d'écoulement des paquets.

Fonctions de Filtrage :

Les fonctions de filtrage sélectionnent les paquets rentrants qui satisfaites la fonction de filtrage dans les champs définis par l'en-tête des paquets, les champs obtenus en traitant le paquet ou les propriétés du paquet lui-même.

9.2.3 PSAMP

Le groupe de travail PSAMP [**?**, **?**] définit un ensemble de capacités standard des dispositifs réseau pour échantillonner sous-ensembles de paquets en utilisant la statistique ou d'autres méthodes. Les capacités deoivent être assez simples pour être implementées d'une manière ubiqua à vitesse de ligne maximale.

PSAMP signifie échantillonnage de paquets (Packet Sampling). La figure 9.2 montre, la séquence de processus exécutés sur un dispositif PSAMP.

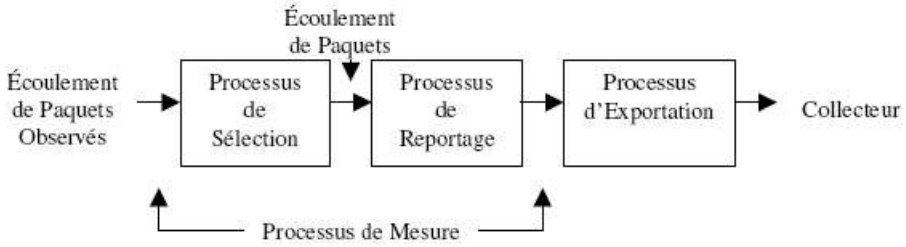


FIG. 9.2 – Architecture du protocole PSAMP

Le mot écoulement de paquets (packet stream) à était préferé car le mot flux de paquets implique que les paquets avec paramètres similaires appartient au même flux (addrese IP source et destination, port source et destination) .

Point(s) d'observation

Éndroit(s) sur le réseau d'où les paquets peuvent être observés.

Écoulement de paquets observés

Ensemble de tous les paquets observés au point d'observation.

Écoulement de paquets

Sub-ensemble d'un écoulement de paquets observés.

Processus de sélection

Un processus de sélection prend l'écoulement de paquets observés et sélectionne un subensemble de cet écoulement.

État du processus de sélection :

Le processus de sélection peut mantenir information à utiliser par le processus de sélection et/ou par le processus de "reporting". L'état du processus de sélection peut dépendre du paquet en proces-

sus, des paquets observés ulterieurment et d'autres variables. Le processus de sélection peut modifier l'état du processus de sélection comme résultat du processus de processer un paquet.

Sélection approximative :

La sélection de paquets peut être approchée par opérations de la même ou d'autre catégorie de sélection.

Sélecteur :

Un sélecteur défine l'action du processus de sélection pour un paquet. Le sélecteur peut utiliser comme information : le contenu du paquet, l'information dérivée du traitement du paquet au point d'observation ou l'état du processus de sélection mantenu par le processus de sélection.

Sélecteur composé :

Composision ordonné de deux ou plus sélecteurs d'où l'écoulement de paquets de sortie d'un sélecteur est l'écoulement d'entrée du sélecteur suivant.

Sélecteur primitif :

Sélecteur non composé.

Processus de "reporting"

Un processus de "reporting" cré un rapport à partir des paquets sélectionnnés par le processus de sélection.

Processus de mesure

Il est composé par le processus de sélection suivi par le processus de "reporting". Ce processus est similaire au processus de métrologie du IPFIX [**?**]

Processus d'exportation

Un processus d'exportation envoi, en forme de paquet exporté, la sortie d'un ou plusieurs processus de mesure vers un ou plusieurs collecteurs. Un processus de mesure peut alimenter un ou plusieurs processus d'exportation.

Dispositif PSAMP

Dispositif hebergeant au moins un point d'observation, un processus de mesure et un processus d'exportation.

Collecteur

Le collecteur réçoit un écoulement de rapport exporté par un ou plusieurs processus d'exportation.

Propriétés des Processus PSAMP

- Sélection : ubiquité, applicabilité, extensibilité, flexibilité, sélection robuste, mesures parallèles, causalité, paquets cryptés.
- Reportage : auto-défini, indicatif de pertes d'information, précision, fidelité, intimité.
- Exportation : opportunité, éviter la congestion, exportation assuré.
- Configuration : facile et assurée.

Filtrage

L'enlevement des paquets que ne nous intéressant pas pour l'éstimation motive le filtrage. Le filtrage est un sélecteur déterministe basé dans le contenu du paquet, son traitement ou leurs fonctionnalités.

Échantillonnage

La sélection d'un sous-ensemble répresentatif de paquets permetant d'estimations précises des propriétés du trafic non échantilloné motive l'échantillonnage. L'échantillonnage est un sélecteur sans catégorie de filtre. Ils existaient deux types : indépendant du contenu et dépendant du contenu.

- Échantillonnage indépendant du contenu : Opération d'échantillonnage non-basée dans le contenu du paquet pour affectuer l'opération de sélection : échantillonnage systèmatique, échantillonnage uniformement pséudo-aléatoire. Dans cette catégorie il n'est pas nécessaire d'accèder au contenu du paquet pour effectuer une decision de sélection.
- Échantillonnage dépendant du contenu : Opération d'échantillonnage basée dans le contenu du paquet pour effectuer l'opération de sélection : sélection pséudo-aléatoire par rapport à une probabilité dépendant du contenu du paquet. Cette opération n'est pas un filtre car la sélection n'est pas déterministe.

L'echantillonnage systèmatique décrits le processus de sélection ; leur points de début et leur durée de sélection par rapport à une fonction déterministe. Même si le processus de sélection ne suit pas une fonction périodique la sélection reste déterministe. Si la systèmique du processus d'échantillonnage ressemble la systèmatique du processus estocastique il y a une haute probabilité que l'éstimation soit décentrée.

Dans l'échantillonnage aléatoire la sélection des points de début des intervals d'échantillonnage est par rapport à un processus aléatoire. La sélection des éléments sont expériences indépendentes.

Dans l'échantillonnage probabiliste la decision d'une sélection est fait par rapport à une probabilité pre-sélectionnée. La probabilité de sélection n'est pas la même pour chaque paquet.

Hash définitions

- Domaine Hash : Sous-ensemble du contenu du paquet et du traitement du paquet, vu comme un mot de N unités binaires pour $N \geqslant 0$.
- Rang Hash : Ensemble de mots de M unités binaires pour $M \geqslant 0$.
- Fonction Hash : Une table deterministe du domaine Hash dans le rang Hash.

– Rang de Sélection Hash : Sous-ensemble du rang Hash. Le paquet est sélectionné si l'application de la fonction Hash dans le domaine Hash produit un résultat dans le rang de sélection Hash.
– Selection basée au Hash : Filtrage spécifie par un domaine Hash, une fonction Hash, un rang Hash et une sélection de rang Hash.

Population

Une population est l'écoulement de paquets ou un sous-ensemble du même. Une population est l'ensemble de base à parti duquel les paquets sont sélectionnés.

Taille de la population

Nombre de paquets dans la population.

Taille d'échantillon

Nombre de paquets sélectionnés d'une population avec un sélecteur.

Catégories des Techniques de Sélection de Paquets

La table 1 fournisse une vue des techniques de sélection de paquets et leur catégorie.

Schèmas de Sélection de Paquets	Déterministe	Dépendant du contenu	Catégorie
Comptage systèmatique	X	–	Échantillonnage
Systèmatiquement temporel (synchronisé)	X	–	Échantillonnage
Aléatoire à n de N	–	–	Échantillonnage
Uniformement probabiliste (aléatoire)	–	–	Échantillonnage
Non uniformement probabiliste (aléatoire)	–	(X)	Échantillonnage
Aléatoire non uniforme à état de flux	–	(X)	Échantillonnage
Filtrage à champ ressemblable	X	X	Filtrage
Fonction Hash	X	X	Filtrage
Filtrage à état du routeur	X	(X)	Filtrage

TAB. 9.1 – Catégories des techniques de sélection de paquets

Les catégories présentées ci dessus s'appliquaient aux sélecteurs primitifs. Dès techniques de sélection plus complexes peuvent être décrites avec la composition d'opérations de filtrage et échan-

tillonnage en cascade. Le symbole (X) implique l'existence des schèmas avec variantes à contenu indépendant.

- Comptage systèmatique : les déclenchements de début et la fin d'interval d'échantillonnage sont définis par rapport au comptage des paquets.
- Systèmatiquement temporel : les déclenchements de début et la fin d'interval d'échantillonnage sert à définir l'interval d'échantillonnage.
- Áleatoire à n de N : n éléments sont sélectionnés d'une population de N éléments.
- Uniformement probabiliste : Les paquets sont sélectionnés indépendement avec une probabilité uniforme de valeur p. Ce échantillonnage puet être basé au comptage, il est suivant référencié comme échantillonnage géométrique aléatoire.
- Non-uniformement probabiliste : Considéré comme une variante d'échantillonnage probabiliste d'où les probabilités d'échantillonnage peuvent dépendre de l'entrée du processus de sélection.
- Aléatoire non uniforme à état de flux : Les paquets sélectionnés en se basant dans l'état de la sélection. L'état de la sélection depends de l'état du flux du paquet et/ou l'état d'autres flux observés par la même fonction de monitoring.
- Filtrage à champ ressemblable : Le paquet est sélectionné si un champ espécifique du paquet égales une valeur prédéfinie.
- Fonction Hash : Une fonction Hash (h) rélie le contenu d'un paquet (c) ou une partie dans un rang Hash (R). Le paquet est sélectionné si h(c) appartient à S, d'où S est un sous-ensemble de R appelé rang de sélection Hash. Cette sélection est un cas particulier de filtrage. Le paquet est sélectionné si C appartient à l'inverse de h(S).
- Filtrage à état du routeur : Les paquets sont sélectionnés en se basant dans l'état du routeur.

9.2.4 Comparaison entre PSAMP et IPFIX

D'après [**?**], il existent similitudes et différences entre IPFIX et PSAMP. Le but de cette section et d'expliquer le travail en cours pour intégrer ces deux cadres/architectures pour eviter la duplication du travail, enrichir les deux propositions et harmoniser les standards dévéloppés.

Différence Architecturale 1

- PSAMP a pour but spécifique les procédures pour configurer la sélection de paquets, le processus d'échantillonnage et le processus d'exportation. La base de données de gestion est définie.
- IPFIX mentione la configuration des processus de mesure et exportation dans la section de conditions, il n'y a pas des plans pour standariser cette configuration.

Différence Architecturale 2

- IPFIX exporte et génère archives de flux contenant information par flux.
- PSAMP exporte et génère information par paquet y compris les interfaces source et destination, le numéro de séquence entre autres (la notion de flux n'existe pas).

Différence Architecturale 3

- IPFIX doit démarrer sur un protocole conscient de la congestion approuvé par l'IETF (TCP ou SCTP).
- PSAMP utilise la notion d'eviter la congestion par l'intermediaire d'une couche d'application ou d'étranglement par l'intermediaire de la configuration d'un paramètre.

Différence Conceptuel 1

- Le processus de mesure de IPFIX et le processus de sélection de PSAMP peuvent sélectionner les paquets observés en se basant dans le contenu de l'en-tête ou leur traitement. Mais le processus de sélection de PSAMP peut calculer quelques valeurs des paquets observés.

Différence Conceptuel 2

- PSAMP rapports information sur le séquencement d'octets du paquet et l'encapsulation des en-têtes si présentée.
- IPFIX rapports information seulement de l'en-tête du paquet.

Harmonisation entre IPFIX et PSAMP

L'harmonisation comprenne l'utilisation de IPFIX comme un protocole de "reporting" pour PSAMP ou bien l'utilsiation de PSAMP comme un composant IPFIX.

9.2.5 Autres Définitions

Le but de cette séction est de complementer les définitions utilisées dans ce rapport.

Technique d'Estimation

Une technique d'estimation est une procédure, algorithme ou méthode afin d'estimer une métrique.

Estimateur

Une règle qu'indique comment déterminer une estimation basée sur mesures contenues dans un ensemble d'échantillons [**?**].

Une fonction d'une donnée connue, utilisée pour estimer un paramètre inconnu. Une application réelle de la fonction à un ensemble de donnés particuliers. Plusieurs différents estimateurs sont possibles pour un paramètre donné. Il doit y avoir un critère de sélection pour choisir entre plusieurs estimateurs, ce critère n'est pas toujours très clair [**?**].

Heuristique

Une heuristique est l'utilisation de règles empiriques :

- Pratiques, simples et rapides,
- Facilitant la recherche des faits et l'analyse des situations,
- Dans l'objectif de résoudre problèmes et de prendre décisions,
- Dans un domaine particulier.

Bande Passante

Selon [**?**], on définit la bande passante comme la vitesse avec laquelle un composant réseau peut acheminer le trafic de paquets. Il existent deux types : Physique et Disponible. Ils sont indépendants d'équipements hôtes et du type de protocole. La bande passante physique est appelée aussi capacité du lien.

Capacité d'un lien ou Bande Passante Physique d'un lien

La capacité d'un lien est le débit maximal de transfert de paquets à la couche trois (IP). À la couche trois le lien libère paquets avec un débit inférieur au débit nominal ; cela est dû aux "overheads" des encapsulations et le cadrement (framing) de paquets. Le temps de transmission d'un paquet IP de taille L_{L3} est :

$$\Delta_{L3} = \frac{L_{L3} + H_{L2}}{C_{L2}}$$

La capacité à la couche trois par rapport à la capacité à la couche deux est de :

$$C_{L3} = \frac{L_{L3}}{\Delta_{L3}} = C_{L2}\left[\frac{L_{L3}}{L_{L3} + H_{L2}}\right]$$

D'où :

Δ_{L3} : Changement de la capacité à la couche trois par rapport à la capacité de la couche deux.

L_{L3} : Taille du paquet à la couche trois.

H_{L2} : Taille de l'entête ajouté aux paquets de la couche deux.

C_{L2} : Capacité à la couche deux.

Dans la couche deux le débit de transmission d'un segment est lié à la vitesse de l'horloge de l'équipement et il est limité par la bande passante physique du milieu de propagation ainsi comme par l'équipement transmetteur/récepteur électronique/optique.

Plusieurs technologies de la couche deux, en particulier les réseaux sans fil, ne travaillent pas toujours avec le même débit de transmission (par exemple les technologies 802.11b @ 11, 5.5, 2 ou 1 Mbps) et donc elles modifient les mesures de la capacité. Au cas où les techniques seront appliquées à ceux technologies, la définition présentée ci-dessus peut être utilisée pendant le temps que le débit de transmission de ces technologies reste constant.

Les mesures de la capacité peuvent aussi être affectées par les façonneurs de trafic (traffic shapers) ou les limiteurs de débit (rate limiters).

Capacité de bout en bout (End-to-End Capacity or Bottleneck Capacity)

D'après [**?**], c'est la capacité minimale d'un chemin réseau de bout en bout ou d'extrême à extrême ; il correspond à la capacité du lien avec moindre bande passante (narrow link). Elle est aussi le débit maximal qu'un lien de la couche IP (trois) puisse fournir à un flux quand il n'y a pas du trafic croisé compétitif. Dans ce cas elle est nommée bande passante de base à la place de bande passante au goulot d'étranglement.

$$C_{beb} = \min_{i=0...H}\{C_i\}$$

D'où :

C_{beb} : Capacité de bout en bout.

C_i : Capacité du lien i dans un chemin à H liens.

Utilisation instantanée et moyenne d'un lien

D'après [**?**], si l'on définit u(x) $\in$ {0,1} comme l'utilisation instantanée d'un lien (voir figure 9.3) lien pendant une période T,

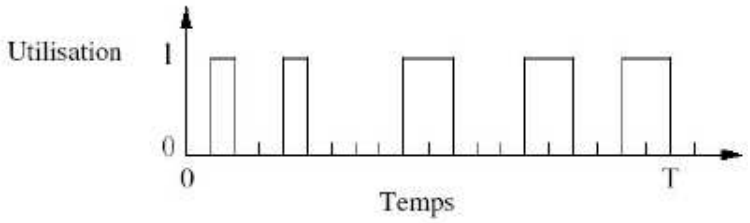


FIG. 9.3 – Fonction instantanée d'utilisation d'un lien pendant la période T

alors l'utilisation moyenne ($u_{i,\tau}$) du lien i dans l'intervalle τ avec $0 \leqslant u_{i,\tau} \leqslant 1$ est :

$$\mu_{i,\tau}(t) = \frac{1}{\tau}\int_{t}^{t+\tau} \mu(x)dx$$

Bande passante disponible (Available Bandwidth)

Toujours selon [**?**],la bande passante disponible d'une route est le débit minimal non-utilisé de la route, étant donné qu'il existe du trafic croisé ; Il correspond à la capacité du lien avec moindre capacité disponible (tight link). La figure 9.4 montre un chemin à trois liens avec le lien de moindre

capacité (C1) et le lien avec moindre capacité disponible (A3). La bande passante disponible est aussi nommée bande passante résiduelle ou bande passante offerte.

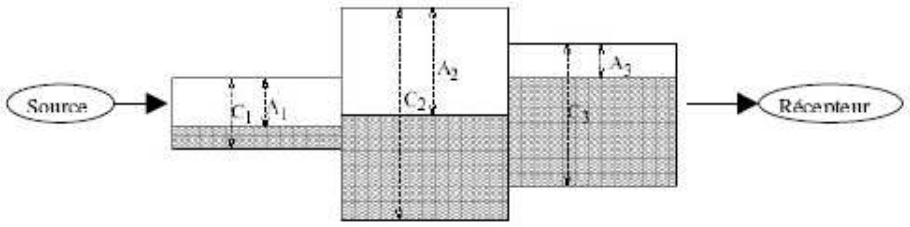


FIG. 9.4 – Lien avec moindre capacité (C1) et lien avec moindre capacité disponible (A3)

$$A_{i,\tau} = C_i(1 - \mu_{i,\tau})$$

$$A_\tau = \min_{i=0\ldots H} A_{i,\tau} = \min_{i=0\ldots H} \{C_i(1 - \mu_{i,\tau})\}$$

D'où :

$A_{i,\tau}$: Bande passante disponible du lien i.

Ci : Capacité du lien i.

$u_{i,\tau}$: Utilisation moyenne du lien i.

A_τ : Bande passante disponible de bout en bout.

Cette définition requiert que le trafic soit stationnaire aux échelles de temps larges pour que le terme d'utilisation u_i soit presque constant.

Goulots d'étranglement

Ils existent deux types de goulots d'étranglement [**?**] : Statiques et dynamiques, les goulots d'étranglements statiques sont les équipements réseaux avec la bande passante minimale tout au long du chemin (liens, routeurs, commutateurs, etc.). Les goulots d'étranglements dynamiques sont les équipements réseaux avec la bande passante disponible minimale pendant une durée déterminée. Les goulots d'étranglements dynamiques dévoilaient la bande passante minimale du chemin.

Volume de capacité de transfert (Bulk Transfert Capacity ou BTC)

Le volume de capacité de transfert [**?**] (aussi connu comme volume de transfert ou capacité de transfert) est particulier au protocole TCP. Le BTC est le débit moyen à long terme pour une connexioni, donc le débit maximal qu'un réseau peut concéder à une connexion capable de s'adapter aux états de congestion (souvent nommées "tcp-friendly" ou "congestion aware").

Le BTC dépend de plusieurs facteurs, à dire : le partage de la bande passante d'un flux avec d'autres flux TCP, l'implémentation du protocole TCP à la source et au récepteur, la bande passante

disponible de bout en bout, la taille des files d'attente des liens et l'élasticité due au trafic croisé entre autres.

Le BTC peut être dérivé mathématiquement par :

$$BTC = \frac{C * MSS}{RTT * \sqrt{Loss\_Rate}}$$

D'où :

MSS : taille maximal d'un segment (maximum segment size) = taille maximale de la charge utile d'un MTU, MTU est la taille maximale d'unité de transfert (Maximum Transfer Unit).

C : Capacité

RTT : Durée aller-retour.

Ce modèle ne peut pas être utilisé pour la prédiction du BTC, car le RTT et les taux de pertes peuvent s'incrémenter dû aux nouvelles connexions TCP. Il est très difficile de définir l'éspérance du débit (prédire le débit) d'une connexion TCP.

Throughput

Le throughput (débit) [**?**] est la quantité de données transferées avec succès entre deux équipements. Il peut être limité par chaque composant tout au long du chemin depuis source au récepteur, en incluant le matériel et logiciel. Ils existaient deux types : Atteignable et Maximal.

Débit Maximal

Le débit maximal [**?**] est le meilleur taux de transfert que peut être fait avec succès entre deux équipements connectés de bout en bout.

Débit atteignable (Achievable Throughput)

Débit maximal qu'une application peut obtenir (application de la couche cinq dans le sens du modèle ISO/OSI) ; il dépend du mécanisme de transmission utilisé (UDP, TCP, ATM, SONET, etc.).

Il est aussi connu comme le débit entre deux points sous conditions completement déterminées comme le protocole de transmission, le matériel d'équipement hôte, le système d'exploitation, la méthode d'ajustement et leurs paramètres, etc. Il est aussi connu comme le performance qu'une application peut atteindre [**?**].

Bande passante effective (Effective Bandwidth)

Soit X[0,t] la quantité de travail arrivant d'une source dans l'intervalle [0,t]. En assumant que X[0,t] ai incréments stationnaires. La bande passante effective de la source est définie :

$$\alpha(s,t) = \lim_{t\to\infty} \frac{1}{st} \log E\left[e^{sX[0,t]}\right] \forall 0 < s,t < \infty$$

Pour toute valeur de s et de t, la bande passante effective d'un flux a une valeur entre le débit moyen et le débit crête du flux.

Débit dynamique asymptotique (Asymptotic Dynamic Rate)

D'après [**?**], il est défini comme l'estimateur de la bande passante moyenne au récepteur. Normalement plus petit que la capacité de la route. Il est indépendant du nombre de paquets test dans les trains de paquets :

$$R = \frac{C_H}{\prod_{i=1}^{H}(1+\frac{r_i}{C_{i-1}})}$$

D'où :

R : Débit dynamique asymptotique.

C_i : Capacité du lien i, i=0..H.

r_i : Débit moyen du trafic croisé rentrant au lien i.

Bande Passante Surplus (Surplus Bandwidth)

Elle est définie [**?**] comme le taux de transfert maximum d'un nouvel utilisateur sans affecter le trafic croisé dans la route : sous l'hypothèse que le trafic croisé dans le lien i a un débit de transmission soutenu de m_i, alors, la bande passante surplus dans le lien i est :

$$s_i = l_i - m_i$$
$$l_i \geqslant s_i \forall i$$

La bande passante surplus au goulot d'étranglement est :

$$s_b = \min\{s_1, s_2, ..., s_n\}$$

La bande passante surplus est la borne inférieure de la bande passante disponible.

Bande Passante Disponible Dépendant du Protocole (Protocol Dependent Available Bandwidth)

Taux de transfert qu'une application peut atteindre[**?**]. Ce taux dépend ne seulement des protocoles et applications mais du comportement de trafic croisé. Quand le débit de transmission soutenue

de la source atteint le goulot d'étranglement surplus et plus loin, le trafic croisé sera affecté et peut réagir en basant leur débit de transmission ou rien faire.

La bande passante disponible dépendant du protocole est très difficile à prédire à partir de tests sauf si les tests se comportent exactement comme l'application. Une session de test peut générer une quantité considérable de trafic pour obtenir des estimations fiables.

Trains de paquets

Un train de paquets consiste en une série de paquets. Chaque paquet contient un nombre fixe d'unités de transmission. Le paquet définе une unité de mesure, les paquets dans le train normalement sont très proches l'uns d'autres.

Lien Différentiel (Hop-Differential ou HD)

Dans un chemin à deux sauts (avec trois nœuds Na, Nb, et Nc), le HD [**?**] est la différence entre le temps pour envoyer un paquet de taille S du nœud Na au nœud Nb et le temps pour envoyer la même taille de paquet du nœud Na au nœud Nc.

Taille Différentielle (Size Differential ou SD)

Dans un chemin avec deux nœuds aux extrêmes Na et Nb, le SD [**?**] est la différence entre le temps pour envoyer un paquet de taille S du nœud Na au nœud Nb, et le temps pour envoyer un paquet de taille S+Δs du nœud Na au nœud Nb.

Taille Maximale de Rafale (Maximum Burst Size ou MBS)

La MBS [**?**, **?**] est le nombre maximal d'octets qu'un routeur peut absorber sans jeter les paquets. Il est déterminé par la taille de la file d'attente au routeur, et par le trafic croisé au routeur.

En autres mots, le MBS est la nombre maximal d'octets que peuvent être énvoyés d'une source vers un récepteur en forme continue à travers le réseau, pendant une période de temps et sans jeter des paquets. Elle est aussi appelée taille effective de la file d'attente.

9.3 CLASSIFICATION

Ils en existent plusieurs classifications des techniques d'estimation de la bande passante, sélon le critère et l'intêret :

1. Par type de technique : Active, Passive, Hybride, Combinée, Multi-Métrique
2. Estimation par route ou par lien : Bout en bout (end-to-end), lien par lien (hop-by-hop).
3. Par type de métrique estimée : capacité, bande passante disponible, débit atteignable, bande passante effective, etc.

4. Par nombre de métriques estimées : mono-métrique ou multi-métrique.
5. Activée par SNMP (SNMP-Gathered), testée activement.
6. Intrusives, non-Intrusives (amicales).
7. Basée à la source (Sender-Based), basée seulement au récepteur (Receiver-Based Only), basée à la source et le récepteur (Sender-Receiver Based).
8. Déterministe, stochastique, probabiliste.
9. Basées aux délais, basées aux dispersions temporelles, basées aux variations des délais, etc.
10. Par politique de service considérée aux routeurs (PAPS, WFQ, etc.).
11. Par type d'évaluation des performances : quantitative, simulation, mesures, combinaison.
12. Par nombre de paquets : à un paquet, à deux paquets, à quatre paquets, à multiples paquets ou par nombre de paires : à une paire, à deux, à multiples paires.
13. Basée dans un sens (aller ou retour), basée sur aller-retour.

La figure 9.5 montre une classification possible des techniques d'estimation de la bande passante. Celle que nous allons prendre dans la structure de cette section.

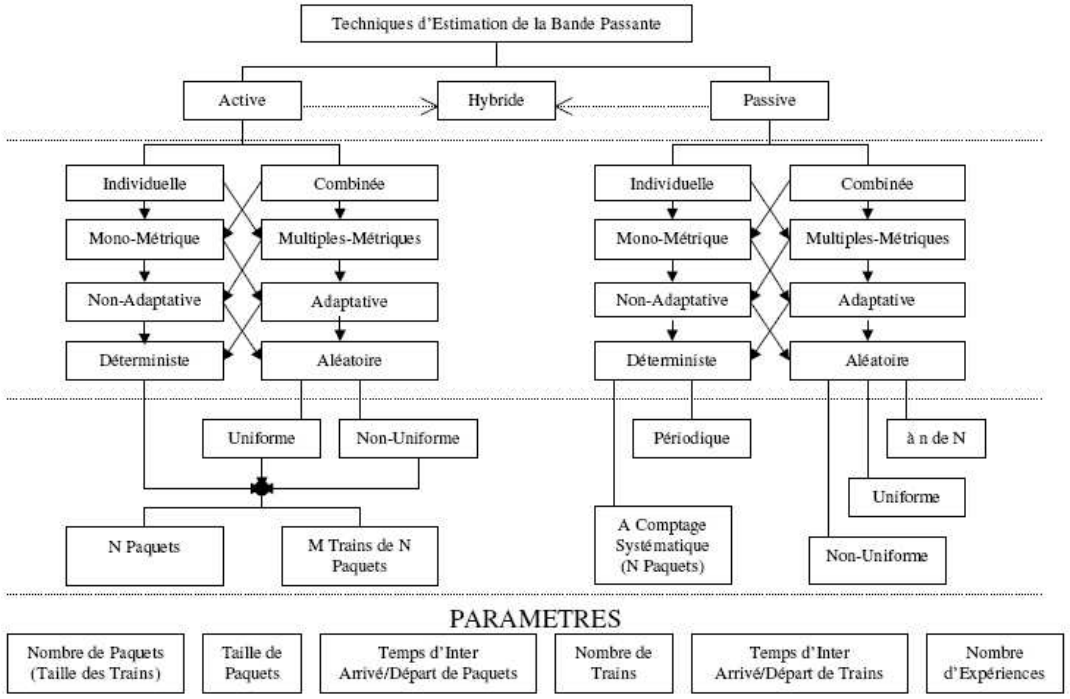


FIG. 9.5 – Classification des techniques d'estimation de bande passante

Les techniques dites actives injectaient du trafic test pour estimer les métriques d'intérêt tandis que les techniques passives utilisaient le trafic du réseau en activité normale pour estimer les métriques d'intérêt (approche inférencielle). Les techniques actives sont dites intrusives car les paquets qu'elles introduisaient sont mélangés avec le trafic normal.

Pour les techniques passives nous considérons la classification proposée au groupe de travail PSAMP a l'IETF : déterministe (comptage systématique et périodique) et aléatoire (uniforme, non-uniforme et à n de N).

Les techniques hybrides sont combinaisons de techniques actives et passives tandis que les techniques combinées sont combinaisons de techniques de la même catégorie (actives ou passives mais non les deux).

Les techniques que puissent effectuer mesures passives et/ou actives ou que mesurent plusieurs metriques seront nommées techniques multifonctionelles.

9.3.1 Phases des Techniques

Toute technique d'estimation de bande passante est une procedure divisée en trois grandes phases, par odre séquentiel : la phase de test, la phase de récollection et la phase d'estimation.

1. La phase de test consiste en envoyer des paquets test. Si la technique est de type active l'équipement source (le générateur) envoi paquets test au lien ou chemin réseau à tester. Pour les techniques passives cette phase n'applique pas. Une fonction de test est appliquée pour envoyer les paquets.
2. La phase de récollection consiste à récolter les paquets test si la technique est active ou bien à récolter les paquets de trafic normal si la technique est passive. Une fonction d'échantillonnage est appliquée pour récolter les paquets.
3. La phase d'estimation consiste à analyser les paramètres des paquets récoltés pendant la deuxième phase pour estimer les métriques d'intêret. Plusieurs phénomènes sont pris en compte comme la dispersion temporelle, la variation des délais, les variations d'inter arrivées, etc.

9.3.2 Échantillonnage Actif et Passif

L'échantillonnage actif consiste en bien définir la phase de test et la phase de récollection tandis que l'échantillonnage passif consiste seulement en bien définir la phase de récollection. Dans ce rapport nous allons ressembler les phases de test et de récollection en une seule phase que nous appelerons phase d'échantillonnage.

Nous définirons par la suite un algorithme d'estimation de bande passante comme une procédure qui établisse d'une manière très précise les phases d'échantillonnage et d'estimation.

Dans ce rapport nous allons décrire les techniques d'estimation par : leur principe de fonctionnement, leur modèle de base, leur phase d'échantillonnage et leur phase d'estimation.

9.3.3 Délais Impliqués dans les Mesures

Toutes les techniques de mesures basées sur le temps doivent considérer les délais suivants :

– Le délai de sérialisation d'un paquet de taille L dans un lien avec vitesse de transmission (ou capacité) C est le temps de transmission d'un paquet dans le lien, et est égal à L/C.

– Le délai de propagation dans un lien est le temps que prenne chaque bit des paquets pour traverser ce lien. Il est indépendant de la taille du paquet et il dépend plutôt des caractéristiques physiques du lien.
– Le délai dans la file d'attente des routeurs ou des commutateurs qui disposent de la contention (files d'attentes ou mémoires tampon) aux ports d'entrée ou de sortie.

9.3.4 Affectations Liées au Système d'Exploitation des Équipements

Les estimations de la bande passante disponible sont affectées pour la capacité et les ressources des systèmes d'exploitation installés aux équipements, à dire :

1. Résolution du temps au système.
2. Le temps pour effectuer un appel au système.
3. Le délai d'interruption (coalescence).
4. La bande passante d'entrée-sortie du système.

Les ressources a, b et c affectaient les algorithmes basés sous la dispersion plus que les algorithmes basés aux trains de paquets. Le ressource "d" affecte tous les algorithmes d'une manière similaire.

Il est trés important de comprendre les différents types d'erreurs et incertitudes introduites par les horloges imparfaites.

9.4 TECHNIQUES ACTIVES

Dans la plus part de réseaux seuls leurs administrateurs peuvent les accéder ou les tester afin de pouvoir déterminer les performances fournies à leurs clients. Les techniques actives sont conçues pour tester les réseaux en ayant pas la catégorie d'un administrateur et donc de pouvoir calculer des mesures de performance comme ceux liées à la bande passante.

9.4.1 Technique par Taille de Paquet Variable

Cette technique est connue en anglais comme Variable Packet Size (VPS)[**?**]. VPS estime la capacité d'une route et celle de ses liens en se basant dans l'estimation de la capacité par lien (per-hop capacity). Elle est utilisée dans l'outil : "pathchar", "netchar" et avec quelques modifications dans la phase d'estimation dans les outils "clink" et "pchar".

Principe

Le principe de fonctionnement de cette technique consiste à mesurer le Temps d'Aller-Retour (Round Trip Time ou RTT) des paquets test par rapport à leur taille. La figure 9.6 montre ce principe.

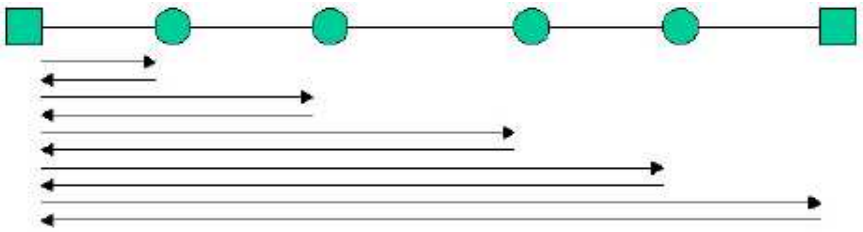


FIG. 9.6 – Principe de mesure des RTT avec les réponses ICMP

Modèle

Le modèle de cette technique est généralisé comme modèle à un paquet (One-Paket Model). Cette technique appartient aussi aux modèles déterministes.

Cette technique prend l'hypothèse, entre autres, que si on réalise un nombre considérable d'expérimentations pour un chemin réseau donné, au moins un des paquets test, y compris la réponse ICMP, ne trouveront pas délais d'attente. Cette considération nous donne le RTT minimal lequel est mesuré pour chaque taille de paquet et comprenne seulement deux composants :

- Un composant qui est indépendant de la taille de paquet et majoritairement dû aux délais de propagation aller-retour (ce délai comprenne aussi les délais de sérialisation des paquets ICMP de taille fixe) [α].
- Un composant proportionnel à la taille de paquet dû aux délais de sérialisation dans chaque lien tout au long du chemin dans un seul sens (aller) [L/C_i].

C'est à dire que le RTT minimal, T_i(L), d'un paquet de taille L jusqu'au lien i est de :

$$T_i(L) = \alpha + \sum_{k=1}^{i} \frac{L}{C_k} = \alpha + \beta_i L$$

$$\beta_i = \frac{[T_i(L) - \alpha]}{L} = \sum_{k=1}^{i} \frac{1}{C_k}$$

D'où :

α : Somme des délais de propagation aller-retour jusqu'au lien i et les délais de sérialisation des paquets ICMP de retour.

C_k : Capacité du lien k.

β_i : Pente du RTT minimal jusqu'au lien i par rapport à la taille (L) des paquets test. Mesures du RTT minimal pour chaque taille de paquet jusqu'au lien i déterminant la valeur de β_i (voir figure 9.7).

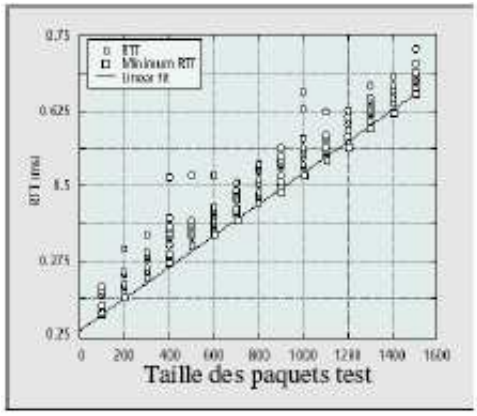


FIG. 9.7 – Modèle à un paquet

Phase d'Échantillonnage

- L'équipement source envoi plusieurs fois plusieurs paquets test UDP de taille fixe vers chaque équipement de la couche trois tout au long du chemin et jusqu'à l'équipement récepteur. La source change la taille des paquets test UDP à chaque fois. Le nombre de paquets par défaut pour l'outil "pathchar" est de 45 selon [**?**].
- Le champ de Durée De Vie (Time To Live ou TTL) de l'entête des paquets IP est utilisé pour forcer l'expiration des paquets test tout au long du chemin réseau jusqu'au récepteur.
- L'équipement récepteur réponds avec paquets d'erreurs ICMP (Internet Control Message Protocol) du type expiré par TTL (tous les paquets ICMP ont la même taille).
- L'équipement source utilise l'information contenue dans les paquets ICMP pour mesurer le RTT.

Phase d'Estimation

Une fois nous avons toutes les valeurs minimales de β_i il faut exécuter une régression linéaire pour obtenir la pente de toutes les mesures par rapport à toutes les tailles de paquets test. L'inverse de la pente de cette régression linéaire nous donne la capacité estimée du lien i.

En répétant les mesures du RTT minimal pour chaque lien i=1,..,H, d'où H est le nombre de liens dans le chemin testé. La capacité estimée à chaque lien tout au long du chemin réseau montant est de :

$$C_i = \frac{1}{\beta_i - \beta_{i-1}} = \frac{L}{[T_i(L) - T_{i-1}(L)]}$$

La statistique utilisée est donc celle de la valeur minimale des RTT's à chaque taille de paquet test et la régression linéaire. La figure 9.8 montre un exemple d'estimation de bande passante par lien dans un chemin réseau à huit liens.

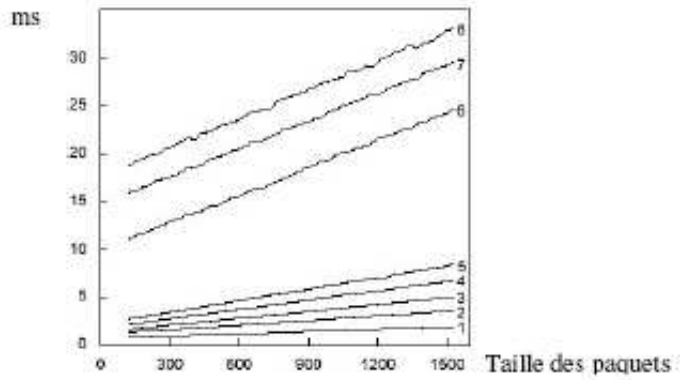


FIG. 9.8 – Exemple d'estimation de bande passante par lien dans un chemin réseau à huit liens

Dans la figure, les paramètres du chemin réseau comprennent la somme des paramètres des liens. C'est-à-dire, étant donné les estimations des paramètres cumulés du chemin jusqu'au lien i, nous pouvons trouver les paramètres des liens inférieurs et égaux à i par soustraction.

Commentaires

Cette approche a l'avantage qu'il n'est pas besoin de logiciel spécial intallé aux routeurs pour obtenir de l'information temporelle.

Cette technique puisse produire d'erreurs significatives de sous-estimations si le chemin mesuré comprenne des commutateurs du type Store-and-Forward travaillant dans la couche deux ou de tout autre équipement dans le chemin travaillant dans la couche deux car ils introduisent des délais de sérialisation, mais ils ne génèrent pas des réponses ICMP expirées par TTL [**?**, **?**].

Plusieurs problèmes pratiques limitaient cette technique : la régression linéaire est très coûteuse car elle doit être calculée par chaque lien et plusieurs paquets sont nécessaires pour calculer une régression fiable, les routeurs ne sont pas construits pour envoyer les acquittements d'une manière temporelle et le chemin de retour ajoute du bruit.

À cause de la mauvaise utilisation de paquets ICMP en situations de sécurité, quelques routeurs et equipements hôtes les limitaient ou les filtraient, en retardant ou excluant les paquets test [**?**].

La différence des RTT minimaux pour les tailles de paquets minimal et maximal est plus petite si la valeur de la bande passante à mesurer est plus grande. Si la valeur de la bande passante à mesurer est plus grande, la procédure est plus complexe pour aboutir les mesures de bande passante. Pour cela il faut prendre en compte que plusieurs tailles de paquets est meilleur que plusieurs paquets test par taille. En générale les estimations d'une bande passante basse sont plus précises et plus consistantes.

Le modèle de l'outil "pathchar" négligea plusieurs détails du monde réel :

1. La taille des paquets d'erreur est négligeable. Cela produise que l'estimation de la latence soit un petit peut haute. Pour l'ajuster, il faut diminuer le taux "taille_paquets_erreur/bande_passante_estimée" du terme de latence dans la formule. Dans la plus part de cas cet ajustement est insignifiant.

2. Le temps de renvoi aux routeurs est négligeable. Vu que le temps de renvoi aux routeurs est le même pour tous les routeurs or on ne le prend pas en compte.
3. Le chemin d'aller n'est pas le même que celui de retour.
4. L'existence de liens composés par chemins virtuels parallèles.
5. Des paquets test avec une taille supérieur à l'unité maximale de transport (MTU).
6. La vitesse de changements de routes.

Les facteurs liés à la latence (a, b et c) n'affectaient pas l'outil "pathchar" car ils ne distinguent entre paquets test petits et grands.

Le logiciel "clink"

Plusieurs améliorations de cette technique sont présentées dans [**?**] et implémentées dans l'outil "clink" : La première est appelée récolte de donnés adaptative (Adaptive Data Collection), elle utilise des méthodes statistiques non-paramétriques afin de diminuer le trafic des tests, elle prenne en compte qu'une fois que la tendance d'une bande passante est détectée évite d'envoyer plus des paquets test et avance au calcul du prochain lien en effaçant les mesures restantes. Le nombre de paquets par défaut pour l'outil "clink" est supérieur à l'outil "pathchar", il est de 93 paquets test [**?**].

Le critère de convergence pour cette approche est d'une grandeur de quatre estimations divisées par la plus petite estimation, en autres mots, la différence entre la plus petite estimation et la plus grande estimation, exprimé en pourcentage de la plus petite. Il est considéré qu'un lien converge si ce critère est plus bas que 10%. Dans la plus part de cas, deux tests par taille de paquet sont suffisants pour atteindre la convergence.

Le deuxième approche est appelée récolte de données rétroactives (Retroactive Data Collection). Pour chacun de quatre échantillons du lien actuel on prend un autre échantillon du lien antérieur. Cette approche requiert un nombre inférieur de paquets test que dans la version originale de l'outil "Pathchar".

Le troisième approche est appelée récolte adaptative dirigée (Directed Adaptive Collection), laquelle utilise le remplissage à residus de la courbe dans la régression linéaire pour commander la nouvelle récolte de données. Si le RTT minimal observé pour un paquet donné est supérieur à la courbe remplit, il est assumé que le minimum n'est pas encore arrivé et il faut prendre plus de paquets test de la même taille. Inversement si le RTT minimal observé pour un paquet donné est inférieur à la courbe remplit, il ne faut pas continuer à perdre le temps avec les mesures. Cette technique est efficace puisqu'elle réduit le nombre de mesures sans perdre la précision.

D'autre part, le filtrage à valeur minimale amplifie l'erreur de mesure car il sélectionne une valeur extrême. Pour résoudre ce problème [**?**] proposa plusieurs techniques sans résultats positifs :

1. En utilisant d'autres statistiques et en incluant le deuxième percentile.
2. Pour le remplissage de la courbe, la technique IWLS (Iteratively-Weighted Least Squares) était utilisée pour diluer les effets des valeurs extrêmes.
3. Modéliser les distributions des temps d'attentes et estimer le RTT minimal comme un paramètre du modèle de la distribution.

L'outil "clink" estime aussi la latence, elle differe de l'outil "Pathchar" car elle utilise une technique paire-impaire (even-odd) pour générer un intervalle d'estimation de la capacité. La technique est non-paramétrique et consiste en diviser un échantillon en sous-échantillons, et observer la variation des paramètres estimés dans les sous-échantillons. Les sous-échantillons ont était obtenus en prenant :

1. La différence entre les échantillons pairs.
2. La différence entre les échantillons impairs.
3. La différence entre les paires de (a) et les impaires de (b) et vice-versa.

En plus, quand le routage devient instable, l'outil "clink" récolte des données pour tous les chemins qui trouve jusqu'au un chemin génère suffisamment de donnés pour obtenir une estimation statistiquement signifiant.

L'outil "pchar"

L'outil "pchar" utilise la librairie libpcap pour obtenir temps d'échantillonnage au niveau noyau. Il estime aussi la capacité d'une route, la latence et les pertes. Il fournisse trois différentes algorithmes de régression linéaire pour obtenir la pente des RTT minimales par rapport à la taille de leurs paquets test (Least Sum of Squares, Nonparametric Method and Least Median of Squares). En plus, il a la possibilité de fournir différent tailles de paquets [**?**].

Le principe de l'outil "pchar" est basé sur la variation du délai dans un sens par rapport à la taille des paquets incrémentale.

L'outil "netchar"

En [**?**], l'auteur proposa le nom de "netchar", ce nom est lié à l'opération de démarrer l'outil "Pathchar" d'une manière périodique dans (n) hôtes et caractériser les chemins réseau de (n-1) hôtes. Cela est une solution pour résoudre le problème de changements de routes (au moins pour le routage à court terme), pour améliorer à l'estimation des métriques (plusieurs vues du même lien) et peut aider à mieux comprendre les asymétries du chemin d'Aller-retour.

9.4.2 Technique à Paire de Paquets Court-et-Long

Cette technique [**?**] est connue en anglais comme Packet Tailgating. Elle sert à estimer la bande passante du chemin réseau ainsi que de chacun de ses liens. Elle est une technique combinée.

Principe

Utilisation de la technique VPS par chaque lien du chemin réseau. L'équipement source envoie un paquet long suivit par un paquet court mis en attente derrière le paquet long jusqu'à ce dernier est jeté en se bansant sur leur champ de durée de vie (TTL). Puis le paquet court continue son chemin jusqu'au récepteur.

Modèle

Cette technique peut être dérivée à partir d'un modèle déterministe du délai des paquets [**?**]. Le modèle utilisé est appelé modèle à multiples paquets, avec lequel on peut déduire les modèles à un paquet et à paires de paquets ou à deux paquets. Elle prend l'hpothèse qu'on peut envoyer les paquets longs sans délais d'attente et les paquets courts mis en attente derrière les paquets longs. Ce modèle prend en compte tous les paquets d'un seul flux, leur latence et leur temps d'attente.

$$t_\ell^k = t_0^k + \sum_{i=0}^{\ell-1} (\frac{s^k}{b_i} + d_i + \max(0, t_{i+1}^{k-1} - d_i - t_i^k))$$

D'où :

t_0^k : Temps de transmission du paquet k.

t_ℓ^k : Temps d'arrivée au lien ℓ du paquet k.

$\frac{s^k}{b_i}$: Délai de transmission du paquet k au lien i.

d_i : Latence du lien i.

q_i^k : Délai d'attente du paquet k au lien i : $\max(0, t_{i+1}^{k-1} - d_i - t_i^k)$

La figure 9.9 illustre cette équation.

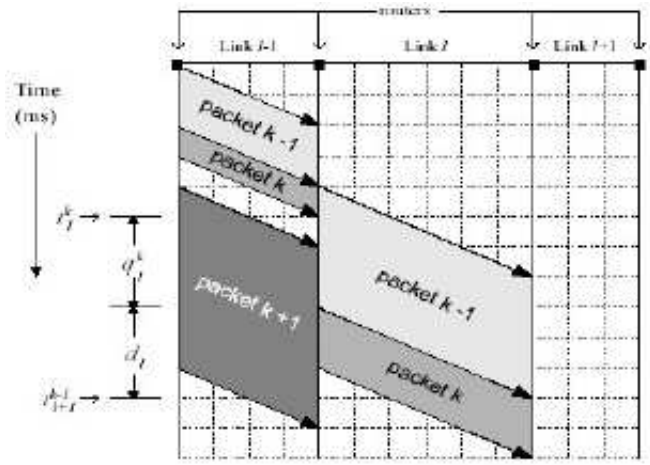


FIG. 9.9 – Illustration de la dernière équation

Ce modèle est défini sous l'hypothèse que les paquets d'autres flux ne produisent pas d'attentes au flux modélisé, et que le délai de transmission est linéaire par rapport à la taille des paquets et que les routeurs sont de type Store-and-Forward.

Les paquets courts ont des délais de transmissions inférieures à ceux des paquets longs. Cela produise qu'un paquet court (tailgater) soit mis en attente d'une manière continue après le paquet le

plus long (tailgated). Le paquet long (tailgated) sera jeté dans le lien ℓ_q, alors le "tailgater" continuera sans être mis en attente jusqu'au récepteur. La bande passante b_{l_q} au lien ℓ_q

est de :

$$b_{\ell_q} = \frac{s^{k-1}}{(t_n^k + \frac{s^k - s^{k-1}}{b^\ell q^{-1}} - \frac{s^k}{b^{n-1}} - t_0^{k-1} - d^{n-1})}$$

Cela veut dire que nous pouvons déterminer la bande passante d'un lien d'où l'attente se présente b_{l_q} à partir des tailles des deux paquets (s^{k-1}, s^k), le temps d'arrivée du deuxième paquet t_n^k, le temps de transmission du premier paquet t_0^{k-1}, la bande passante de tous les liens antérieures $b^\ell q^{-1}$, et les délais de tous les liens antérieures (d^{n-1}).

Pour utiliser cette équation nous devons résoudre certains problèmes comme le calcul du temps d'inter-transmission entre paquets, travailler avec la gigue des horloges et utiliser les mesures d'aller-retour entre autres.

Le modèle à délai de multiples paquets de cette technique ést utilisé pour dériver la propriété à paires de paquets des réseaux d'attentes de type PAPS. Laquelle a était utilisée pour mesurer la bande passante du goulot d'étranglement d'un lien, sous l'hypothèse que le délai de transmission est linéaire par rapport à la taille des paquets, les routeurs sont de type Store-and-Forward, les liens sont à monocanal et qu'il n'existe pas d'autre trafic dans le chemin que produise des attentes aux paquets de test.

Phase d'Échantillonnage et d'Estimation.

Cette technique est divisée en deux phases : la phase sigma qui mesure les caractéristiques du chemin, et la phase à déclenchement de queue (tailgating) qui mesure les caractéristiques de chaque lien.

- Phase sigma : consiste à envoyer paquets de différentes tailles et à utiliser la régression linéaire pour déterminer le délai minimal à chaque taille de paquet. Cette phase continue à envoyer des paquets test jusqu'à on obtient 99% d'intervalle de confiance. Cette phase demande beaucoup de paquets. Cette technique le fait seulement depuis l'équipement source à l'équipement récepteur tandis que les modèles à un paquet et à paires de paquets d'autres techniques le font à chaque lien.
- Phase "tailgating" : Consiste à envoyer paires de paquets court-et-long et calculer d'autres variables pour déterminer la bande passante. La valeur minimale du délai du paquet court parmi plusieurs échantillons est prise.

Pour l'ensemble de délais d'un lien l'échantillonnage est aléatoire avec remplacement jusqu'à on obtient un nouveau ensemble d'échantillons de 25% de la taille de l'original. Un nouveau délai minimal est calculé avec ce nouveau ensemble. Cette procédure est répétée 20 fois et après la variance de toutes les mesures de ces nouveaux délais est calculée. L'erreur de cette variance est divisée par la valeur actuelle du délai minimal de l'ensemble. Ces valeurs on était choisit car la surcharge des unités centrales de processus dans les ordinateurs ne soit pas assez grande avec des latences typiques.

Commentaires

Cette technique ne consume pas beaucoup de bande passante du réseau, elle n'est pas liée aux comportements consistants des routeurs qui manipulaient des paquets ICMP, et elle n'est pas liée avec le temps des livraisons d'acquittements.

Cette technique envoie peu de paquets avec la même précision. Néanmoins sa précision est basse pour chemins à haut débit de quelques liens.

Cette technique peut théoriquement détecter les liens à multiple canaux et peut aussi bien démarrer en arbres à multiple diffusions (multicast).

Cette technique effectue la régression linéaire une fois et non à chaque lien. Comme elle n'est pas liée à la livraison des acquittements des routeurs, elle est robuste contre les routeurs qui généraient des paquets ICMP sans consistance. Elle peut incrémenter la précision et réduire le nombre des paquets à envoyer en mesurant le délai à un sens à la place du délai d'aller-retour, sans nécessité d'un nouveau logiciel aux routeurs. Cette technique souffre aussi du phénomène des noeuds fantômes (nœuds de la couche deux).

Cette technique a les complexités suivantes :

1. Déterminer le temps de transmission inter-paquet : Il peut exister des variations d'ordonnancement dans le système d'exploitation de la source alors la date limite ne peut pas être atteignit, alors il est nécessaire de déterminer quant on auras besoin de filtrer ces mesures. Cela signifie que nous aurions quelques problèmes pour envoyer le paquet court avec vitesse suffisante si des liens tous prés ont bande passante à haut débit.
2. Mouvements d'horloge : tout mouvement d'horloge est cancellé dans les calculs.
3. Utiliser des mesures aller-retour : on peut transformer l'équation de ce modèle en utilisant seulement des temps d'envoi. L'idée est de demander au récepteur un acquittement pour chaque paquet court qu'arrive au récepteur. Si nous pouvons corréler les paquets courts et leurs acquittements, alors nous pouvons déterminer les propriétés d'aller-retour du chemin et les utiliser pour calculer les bandes passantes du chemin de retour. En assumant que les bandes passantes sont symétriques. Pour mesurer les liens asymétriques, nous devons être capables de mesurer le délai à un sens.
4. Forcer des acquittements à être envoyés : Nous devons produire des réponses consistantes aux arrivées de paquets au récepteur.
5. Paquets perdus : Un paquet court perdu ne représente pas de problème car cela signifie que nous aurions une taille d'ensemble plus petit. Néanmoins, une source peut seulement admettre un échantillon de paquet court si seulement a reçu le message ICMP de type TTL-expiré pour ce paquet.
6. Nœuds invisibles (couche deux) : Elle prend l'information du temps directement du noyau en utilisant la librairie "libpcap", à la place de la prendre du couche application. Celui enlève le nœud du noyau d'application du chemin à mesurer. Les ponts restent invisibles à l'outil "nettimer".

Les avantages de cette technique sont : vitesse, non-obstructive, robuste. Cette technique peut détecter liens à multiples canaux, elle n'est pas basée aux livraisons temporelles des paquets ICMP et peut démarrer sans acquittements. Cette technique n'utilise pas des paquets ICMP expirés par TTL des nœuds intermédiaires.

Cette technique peut travailler sans acquittements. En déployant l'outil dans l'équipement récepteur il est possible de mesurer les délais dans un sens des paquets. La source peut continuer à envoyer après les paquets courts sans la connaissance des délais antérieurs. Si un délai antérieur peut être envoyé à la source alors la source peut s'adapter pour décider le moment d'arrêter les deux phases. Dans autre situation, la source peut envoyer un nombre fixe de paquets à chaque étage. Eventuellement la source et la destination doivent se communiquer alors la source peut spécifier les paquets perdus d'une manière primature.

En mesurant sans acquittements évite les attentes dans le chemin de retour, habilitant la technique d'être deux fois plus précise que les techniques à un paquet. Les mesures sans acquittements évitaient l'implosion des acquittements sur arbres à multiple diffusion, habilitant cette technique de mesurer la bande passante dans plusieurs liens au même temps dans un arbre à multiple diffusion.

Les désavantages sont : nécessité d'envoyer paquets extrême-à-extrême au premier lien, inhabilité pour mesurer un lien trop rapide après un lien trop lent, les attentes tout au long du chemin perturbe les mesures de tous les liens dans le chemin, les accumulations des erreurs dans les calcules. La source doit être capable d'envoyer paquets rapidement dans le premier lien. La solution de l'inhabilité pour mesurer un lien trop rapide après un lien trop lent consiste en utiliser la technique à un paquet pour mesurer le lien problématique et utiliser cette technique ailleurs. Nous divisons le chemin en exécutant la phase sigma trois fois : une pour le nœud juste avant le lien avec la plus haute vitesse, une pour le nœud juste après ce lien et une pour le nœud récepteur. Il faut faire cela quand le taux des mesures de bande passante de deux liens proches est prés de 37.5. Cette solution incrémente le nombre de paquets loin du numéro normal mais reste mineur que la technique à un paquet pour un chemin.

Une autre limitation est que les attentes tout au long du chemin perturbaient les mesures des liens. En contraste, aux techniques à un paquet les attentes aux liens avant le lien en question affectaient leurs mesures. Cela peut produire un désavantage pour cette technique s'il existe un lien très congestionné dans le reste du chemin. Il sera nécessaire de prévenir des mesures précises et rapides dans les liens antérieures. La solution est similaire. Il faut exécuter la phase sigma au nœud juste avant le lien congestionné et au nœud récepteur. Alors, nous exécutons la phase "tailgating" au nœud pre-congestionné pour mesurer tous les liens avant le lien congestionné.

Une limitation finale est que les erreurs peuvent s'accumuler pendant les calcules tel que les liens éloignés sont très difficiles à mesurer. Les techniques à un paquet seulement propageaient les erreurs dans un sens alors elles sont plus robustes en ce qui concerne les erreurs.

9.4.3 Technique à ACCIG

Cette technique [**?**] [**?**] sert à estimer la bande passante du goulot d'étranglement.

Principe

Cette technique est basée sur les observations d'histogrammes des variations de délais.

Cette technique utilise une combinaison de paires de paquets test, chacune comprenne un paquet de test que suit un paquet leader, d'où la durée de vie du paquet leader est limitée.

Cette technique produise d'estimateurs du type "pathchar" aux chemins à multiples liens sans utiliser les messages ICMP en réduisant ainsi l'invasion de paquets test parmi d'autres avantages.

Modèle

Cette technique propose une approche basée sur la variation de délais, laquelle est utilisée pour estimer la bande passante du goulot d'étranglement, et en identifiant le trafic croisé qu'intervient entre sondes consécutives comme la source qui ajoute un décalage additionnel, nous pouvons aussi obtenir des mesures de la bande passante disponible.

Définition : Un flux de paquets test est défini par ses temps de départ dans l'équipement source.

Modèle générique à multiples sauts :

Les routes sont modelées comme une concaténation de H $\geqslant 1$ liens. C'est un modèle à multiples liens. Un lien est une file d'attente de taille infinie avec politique de service de type PAPS, arrivées ponctuels et services fluides déterministes (u^h), la file d'attente à la sortie des routeurs est de type Store-and-Forward suivi par un lien avec vitesse de transmission égal à la vitesse de service (u^h) et délai de propagation (D^h).

L'architecture du routeur de type Store-and-Forward nous permet de considérer la file d'attente du router (h) appartenant au lien (h-1) et de faire la modélisation comme file d'attente à la sortie. Il est assumé que les paquets traversent le commutateur dans un temps égal à zéro.

L'approche de cette technique est basée sur les échantillons du chemin à la place d'être probabiliste. Il diffère des techniques traditionnelles basées dans la théorie des files d'attente. Elle ne prend pas en compte les considérations sur les statistiques du trafic en non plus les mesures de performance comment la moyenne ou la variance du délai étudie ; mais plutôt les caractéristiques des flux de données qui sont dérivés directement des événements d'attentes liés aux paramètres d'intérêt, tel que le débit de transmission des liens.

L'analyse se focalise aux temps d'estampillage. Les pertes ne changeant pas sa validité, il n'est travaille pas avec temps de vie explicites. L'analyse est valable pour tout paquet test arrivant au récepteur mais aussi pour tous les paquets traversant le chemin réseau.

Le délai d'un paquet test dans un lien est :

$$d_i = \tau_i^* - \tau_i = \omega_i + x_i + D$$

D'où :

τ_i^* : temps de départ

τ_i : temps d'arrivée

ω_i : délai d'attente $\geqslant 0$

x_i : temps de service > 0

D : délai de propagation > 0

En comparant deux paquets test consécutifs nous avons :

Le temps d'inter arrivé : $t_i = \tau_i - \tau_{i-1}$

Le temps d'inter départ : $t_i^* = \tau_i^* - \tau_{i-1}^*$

La variation du délai : $\delta_i = d_i - d_{i-1} = t_i^* - t_i = (x_i - x_{i-1}) + (\omega_i - \omega_{i-1})$

Traditionnellement, les méthodes basées au temps d'aller-retour se sont concentré dans les séries du temps $\{d_i\}$ tandis que les méthodes basées à paires de paquets dans les séries de temps d'inter arrivée $\{t_i^*\}$. Cette technique suggère se concentrer dans la variation du délai $\{\delta_i\}$ pour les raisons suivantes :

- On connait $\{t_i\}$ et δ_i.
- · $\{\delta_i\}$ est plus enrichie car existe la liberté de choisir $\{t_i\}$ et en plus elle comprenne $\{t_i^*\}$.
- $\{\delta_i\}$ est une constante alors est égale à $\{d_i\}$. Comme $\{t_i^*\}$, elle a besoin seulement d'horloges à débit de transmission précise pour effectuer les mesures et pas de la synchronisation de bout en bout.
- L'histogrammes de $\{\delta_i\}$ comprennent les corrélations les plus importants dans $\{d_i\}$.

La variation du délai dans un chemin à H sauts est donc :

$$\delta_i = \sum_{h=1}^{H} (x_i^h - x_{i-1}^h) + \sum_{h=1}^{H} (\omega_i^h - \omega_{i-1}^h)$$

Le premier terme de l'équation représente la contribution des temps de service. Il est déterministe et indépendant du trafic croisé. Le deuxième terme est dû au trafic croisé, on peut le prendre comme un bruit aléatoire et sa nature dépends de l'interaction entre les paquets test et le trafic croisé.

Les temps d'attente des paquets successifs dans une file d'attente infinie de type PAPS est :

$\omega_i = [\omega_{i-1} + x_{i-1} - t_i]^+$

D'où :

$[x]^+ = \max(0,x)$

Les périodes occupées d'une file d'attente sont les intervalles du temps auxquelles le serveur est actif. Les périodes vides sont les intervalles auxquelles la file d'attente est vide.

Nous introduisons deux concepts clés pour cette approche. La sonde initiale (I) : dite la première sonde de paquets d'une période occupée et (B) les autres sondes qui sont occupées. La définition des sondes de type B généralise la condition d'extrême-à-extrême du concept à paires de paquets (c'est à dire les sondes qui portent la signature du lien d'où la dispersion temporelle a lieu) pendant que la sonde I permet la compréhension de la nature du bruit créé par signatures emportant des paquets test, habilitant la détection, l'interprétation, et le filtrage des séries de temps des paquets test mesurés.

Il y a deux classes de signatures type B : la signature à débit de transmission qui est liée à la majorité des méthodes existantes et la signature à distribution qui formalise l'approche pris par la technique ToPP expliquer après [**?**]. Les deux peuvent être analysées par le temps d'inter arrivé ou par la variation du délai.

I : $\delta_i = [x_i - x_{i-1}] + (\omega_i - \omega_{i-1})$

B : $\delta_i = [x_i - t_i] + (c_i)$

D'où :

c_i : temps de service agrégé du trafic croisé entrant la file d'attente entre les paquets test i-1 et i.

Nous pouvons voir ces équations comme : [terme deterministe du test] + (bruit dû au trafic croisé). La nature des bruits est différente pour chaque type de test I ou B. Ces équations sont opérateurs à un saut. En étendant ces formules à H sauts :

Opérateur de temps d'inter arrivé : $t_i^h = t_i^{h-1} + \delta_i^{h-1}$

Variation du délai de la route : $\delta_i = \sum_{h=1}^{H} \delta_i^h$

Si le paquet test i est I ou B au lien actuel et B au prochain lien, l'information codé en δ_i au premier lien est sur écrit et remplacé pour l'équation de B pour le deuxième lien seulement. Si le paquet de test est I au deuxième lien, les termes de l'équation d'I pour le deuxième lien sont ajoutés.

Ils existent deux scénarios. Soit $x_i^h = \frac{p_i^h}{u^h}$, d'où p_i^h, d'où p_i^h est la taille du paquet test et u^h le débit de transmission du lien h.

Le premier scénario est II...I correspondant à la variation du délai de routage. Ici rien est sur écrit et les termes de l'équation I s'accumulaient.

$$\delta_i = \sum_{h=1}^{H} \frac{[p_i^h - p_{i-1}^h]}{u^h} + \sum_{h=1}^{H} (\omega_i^h - \omega_{i-1}^h)$$

Le deuxième scénario est XX...XBII...I :

$$\delta_i = [x_i^{S_i} - t_i] + (c_i^{S_i}) + \sum_{h=S_i+1}^{H} \frac{[p_i^h - p_{i-1}^h]}{u^h} + \sum_{h=S_i+1}^{H} (\omega_i^h - \omega_{i-1}^h)$$

D'où :

S_i : dernier lien que le paquet de test est du type B.

Les liens avant S_i sont sur écrits, mais l'histoire n'est pas complètement enlevée, car le service agrégé de trafic croisé $c_i^{S_i}$ rentre dans la file d'attente S_i pendant l'intervalle $t_i^{S_i}$ et non t_i comme dans le cas à un lien. Cette dernière équation peut s'appliquer à n'importe quel lien et pas nécessairement au lien avec goulot d'étranglement. Si nous graphiquons cette équation nous pouvons observer un mélange de propretés des différents liens.

Ces équations capturent toutes les forces actant dans paquets de test arbitraires. Au contraire, dans la littérature traitant des modèles comparables, des effets particulières réagissent sur paquets de test prédéterminés. Ces équations aussi habilitaient l'acheminement de l'information codée en $\{\delta_i^h\}$ pendant que la route est traversée. En traitant la distorsion de l'information utile, ou signatures, par ses points de création, est crucial pour l'utilité pratique de ces méthodes de test.

Signatures

Une signature est une propriété d'un observable tel que $\{\delta_i\}$ ou $\{t_i^*\}$ peuvent être détectées, interprétées, et exploitées. Nous allons définir quatre signatures, la figure 9.10 sera notre référence.

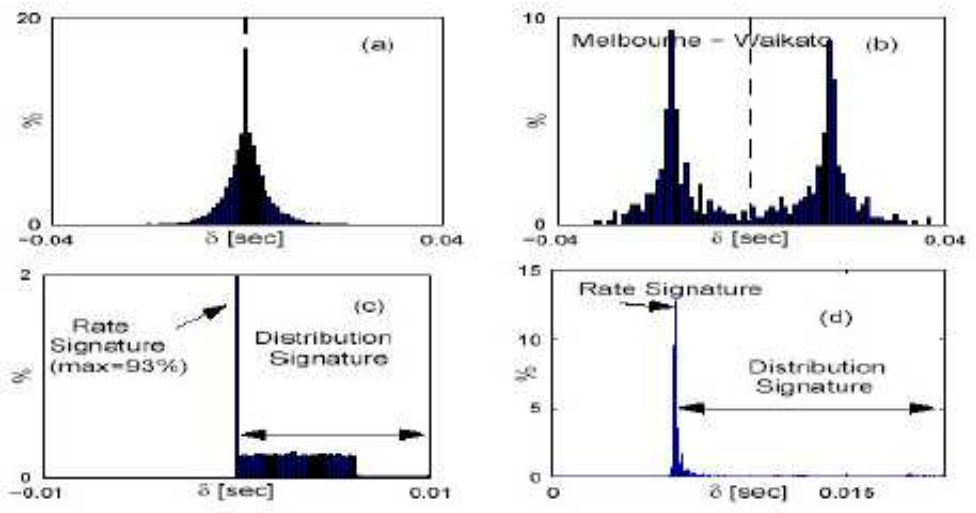


FIG. 9.10 – Effets et signatures des paquets de test

Signature d'Indépendance

Elle est due au deuxième terme de l'équation du primer scénario (voir figure 9.10.a). La symétrie correspond à la signature. Être détectée ou absentée, informe si un nombre significatif de paquets de test étaient du type B, qui a beaucoup d'applications avec d'autres signatures. En la combinant avec d'autres termes, elle peut être vue comme un bruit indépendant et sa symétrie est interprétable et filtrée dans l'histogramme du δ_i. Le point essentiel est que chaque paquet de test est de type I.

Signature d'Accumulation

Elle est due au premier terme de l'équation du primer scénario (voir figure 9.10.b). Elle est l'effet déterministe et indépendant du trafic croisé lequel s'accumule tout au long du chemin si les paquets de test ont des temps de service différents. Si un paquet de test est de type I, sa signature d'accumulation s'additionne à la signature des liens antérieurs. La figure montre l'exemple avec deux tailles de paquets test. Elle aussi montre les effets de la signature d'indépendance.

L'estimation de la bande passante à un paquet est l'effet basé sur l'extraction de $\frac{1}{u^h}$ de cette signature. Néanmoins, cela est réalisé via $\{d_i\}$ avec $\{p_i\}$ constant à la place de l'alternance des valeurs basés en $\{\delta_i\}$. Même si la signature d'accumulation peut être supprimée avec p_i=p, la signature d'indépendance ou de bruit est inévitable. Le composant discret de ce bruit (valeur crête égale à zéro dans la figure) correspondant le délai d'attente de retour, dépends aux conditions du trafic croisé et peut être absent.

Signature du débit de transmission

Ces effets sont dus au terme $x_i^{S_i}$ de l'équation du deuxième scénario. Elles correspondent aux paquets de test qui ne sont pas seulement de type B, mais aux paquets de test de source-à-source ($c_i^{S_i} = 0$), forçant le temps d'inter départ au lien S_i de prendre un nouveau valeur indépendant du passé mais dépendant du débit de transmission du lien.

En fournissant ces paquets test on trouve des attentes nulles plus loin du lien S_i, et en sélectionnant p_i=p pour annuler la signature d'accumulation, ces valeurs formaient un pic pointu dans l'histogramme de t_i^* mesurable au récepteur. Si t_i=t, comme dans la figure 9.10.c le pic est aussi présent pour δ_i.

En utilisant δ_i avec paquets de test d'extrême à extrême produise la valeur pic du goulot d'étranglement à apparaître avec valeurs négatives (une aide pour l'identification). Cette signature est la base de l'effet d'espacement dans le lien aux techniques basées à paires de paquets. Le pic doit être identifie et mesuré. Les méthodes existantes appliquaient des filtrages qui ne prennent pas en compte la nature de l'accumulation et l'indépendance du bruit dans cette équation. En faisant cela nous permet d'améliorer la détection du pic et en avoir une meilleure précision.

Signature de Distribution

Ces effets sont occasionnés par le terme $c_i^{S_i}$ de l'équation du deuxième scénario. Comme dans la signature à débit de transmission elle décrit aussi des événements dans un lien spécifique mais elle retiens l'histoire. La contribution de $c_i^{S_i}$ à t_i^* et δ_i crypte l'information sur la distribution et les temps d'inter arrivée de la taille des paquets du trafic croisé (voir figures 9.10.c et figures 9.10.d) pour tailles de paquets test fixes et variables respectivement.

Les méthodes d'estimation pour le trafic croisé comprennent différents aspects de la distribution de $c_i^{S_i}$. Le trafic croisé est très lié avec l'idée de bande passante disponible et peut être utilisé pour estimer la même.

En envoyant paquets test très proches, dans la plus part de cas nous pouvons assurer que beaucoup seront du type B. En générale l'histogramme de $\{\delta_i\}$ contiendra un mélange de signatures d'histoires en obéissant les deux équations avec différentes valeurs de S_i.

Phase d'Échantillonnage

Cette technique est basée sur le concept de signature d'accumulation, sous les hypothèses que les paquets test sont suffisamment espacés pour qu'ils puissent être effacés après le lien h_{ttl} (le lien d'où la valeur de la durée de vie soit égale à zéro) et que les paquets ICMP sont re-envoyés à la source tout au long un chemin réseau de retour.

La variation du délai aller-retour (sender-to-sender) peut s'écrire comme :

$$\delta_i = \sum_{h=1}^{h_{ttl}} (x_i^h - x_{i-1}^h) + \sum_{h=k_{ttl}}^{K} (x_{icmp,i}^h - x_{icmp,i-1}^h) + \sum_{h=1}^{h_{tl}} (\omega_i^h - \omega_{i-1}^h) + \sum_{h=k_{ttl}}^{K} (\omega_i^h - \omega_{i-1}^h)$$

D'où :

K : Nombre de liens du chemin réseau de retour.

k_{ttl} : Lien d'où les paquets ICMP rentrent le chemin réseau de retour.

La deuxième composante de cette équation est égal à zéro car la taille des paquets ICMP est égale en laissant un membre d'accumulation qui est généré en alternant des paquets test de taille supérieure jusqu'au lien h_{ttl} seulement, mais les membres du bruit correspondent aux chemins réseau aller et retour. Ces bruits sont symétriques, en résultant un histogramme des délais de variation symétrique. En assumant que le temps pour générer les paquets ICMP est indépendant de la taille des paquets, l'équation devient :

$$\delta_i = \sum_{h=1}^{h_{ttl}} (x_i^h - x_{i-1}^h) + N_i = (p_i - p_{i-1}) \sum_{h=1}^{h_{ttl}} \frac{1}{u^h} + N_i$$

D'où :

N_i : Bruit symétrique

Un groupe de paquets test est envoyé pour mesurer $\sum_{1}^{h_{ttl}} \frac{1}{u^h}$. Le nombre de paquets test sélectionné est égal au nombre de paquets utilisés dans l'outil "pathchar" (voir section IV.1.3).

Phase d'Estimation

L'estimation des liens est réalisée d'une manière récursive, similaire à l'outil "pathchar" et "clink". La procédure est répétée pour assurer un incrément stable du champ de durée de vie jusqu'au récepteur soit atteignit. La valeur de u^h est déterminée à partir des différences des estimations entre les différénts étapes de récursivité.

Plusieurs méthodes peuvent être utilisées pour détecter le pic dans l'histogramme de δ_i. Par exemple :

- La méthode basée dans la densité d'estimateur noyau (Kernel Density Estimation) [**?**].
- La méthode basée sur l'espérances conditionnelles d'échantillons (méthode de quantiles).

La méthode utilisée est celle de la moyenne car elle est considérée plus précise que la méthode basée dans la densité d'estimateur noyau. En plus, elle évite la sensitivité aux grandes déviations des moments.

Commentaires

La signature d'accumulation utilise tous les paquets test reçus. Cela réduit largement le nombre de paquets test à envoyer, spécialement pour dans le cas de longues routes et liens avec utilisations grandes, comme dans le filtrage à détection minimale qui demande l'arrivée des paquets test aux files d'attentes vides à chaque lien tout au long de la route.

Le nouveau modèle basé sur variations de délais permets : l'évaluation des effets de la taille des tests, comprendre le bruit de retour, reconnaître que la détection du pic est supérieure à la mode ou le filtrage basé au minimum, proposer des nouvelles méthodes d'estimation.

Cette approche décrit facilement les goulots d'étranglement secondaires ou liens post-étroits, éclaircit les avantages de la détection à pic sur celles basées au minimum et la mode. Elle aussi éclaircit le potentiel de la signature de distribution pour mesurer, sous certains conditions, le débit de transmission de tous les liens tandis que la signature de débit de transmission est limitée au lien avec goulot d'étranglement et les liens avec goulots d'étranglements secondaires.

La signature de distribution souffre de désavantages pratiques lesquelles limitaient son utilité. La considération que la taille des paquets test est constante à chaque lien n'est pas valable en pratique due aux effets de la couche deux du modèle ISO/OSI. La figure 9.11 montre ces effets [**?**].

Plusieurs méthodes sont implicitement basées aux différences des délais. Cette technique nous déplace vers une architecture basée explicitement dans la variation du délai à cause de ses bénéfices théoriques.

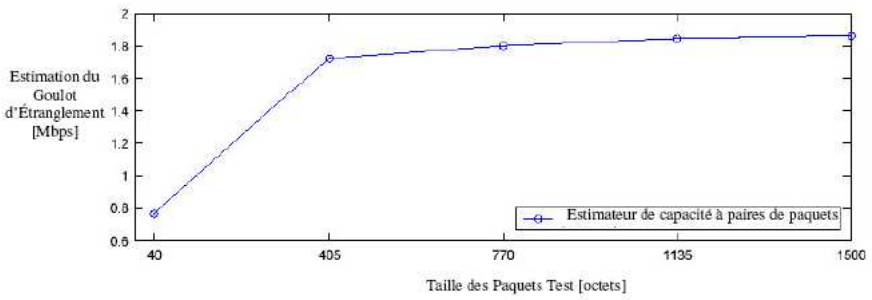


FIG. 9.11 – Estimation de la bande passante par rapport à la taille des paquets test

Le temps de convergence varie de 10 tests jusqu'à 200 tests, en dépendant de la distance depuis la source et leur débit de transmission. L'incrément introduit par un lien dans la variation du délai est inversement proportionnel à son débit de transmission.

Cette méthode considère que les paquets arrivaient aux différents périodes occupées. Il n'est pas possible d'obtenir d'estimations rapides en envoyant des paquets test avec une sépartion très proche. Nous allons mentionner plusieurs problèmes pratiques dans cette technique :

- En connaissant la présence de liens invisibles (équipements travaillant dans la couche deux du modèle ISO/OSI) et ses effets, les estimations peuvent être corrigées. Malheureusement cette correction n'est pas possible si la présence de liens invisibles est inconnue.
- Les estimations pour liens à haut débit sont assez certains pour supporter une comparaison significative.
- Les outils "pathchar" et "clink" déterminaient les temps de transmission des paquets test en se basant sur les temps d'aller-retour. Pour éviter l'invasion de paquets test cette méthode envoie les paquets test avec espacement incrémental. En incrémentant l'espacement des tests réduira la probabilité de partager des périodes vidées, mais, si la quantité de tests resta la même le temps de mesures augmentera proportionnellement.

– Cette technique est basée aux différences des temps de service alors les entêtes additionnelles d'une couche deux (liaison) peuvent affecter les estimations mais laissent sans affectation les extensions de la technique à un paquet (pathchar) car la variation de la taille des paquets test cancel cet effet.
– Quand le délai est non-linéaire par rapport à la taille des paquets test la variation des tailles des paquets test est un avantage. Cette technique peut augmenter le nombre de tailles de paquet avec la possibilité d'augmenter aussi les erreurs de détection des pics.
– Des estimations erronées sont produites aussi si les temps pour générer les messages ICMP ne sont pas constants. Ce problème peut se propager car les débits de transmission sont obtenus en utilisant une valeur accumulée des durées de vie antérieures.

ACCIG utilise des paquets ICMP comme l'outil "pathchar" et "clink", et donne des résultats similaires mais plus efficaces. Les méthodes basées sur paquets ICMP souffrent de grand bruit parce qu'elles sont basées sur le temps d'aller-retour et dû aussi aux délais de procès qui dépendaient des routeurs.

Les méthodes estimant la bande passante disponible sont aussi basées dans l'observation du temps d'inter-départ des sondes de paquets consécutifs en prenant en compte le trafic croisé insérer entre eux.

Cette technique permet d'évaluer l'affectation de la taille des paquets d'une sonde active, l'entendement du bruit de la liaison descendante, la détection creuse reconnue comme supérieure à la mode o filtrage basé dans le minimum, et des nouvelles méthodes.

Nous utilisons un modèle de routage à multiple-liens. Chaque lien consiste d'une file d'attente de type PAPS avec taux de service déterministe u^h suivi du débit de transmission de lien u^h et un délai de propagation D^h.

On fait l'hypothèse que les paquets traversent le commutateur dans un temps égal à zéro pour arriver à la file d'attente à l'instant ou ils quittent le lien antérieur.

Avantages :

– Permet la description d'une manière naturelle les goulots d'étranglements ou liens post-étroits.
– Éclaire les avantages de la détection creuse sur la détection basée dans le minimum ou dans la mode.
– Éclaire comme la distribution de la signature aie potentiel, sous certains conditions, pour mesurer le débit de transmission de tous les liens, sans importance que la signature de débit de transmission est limitée pour le goulot d'étranglement ou liens de goulot d'étranglement secondaires.

La signature de distribution souffre de désavantages pratiques qui limitaient son utilisation, et l'hypothèse que la taille des paquets est constante est limitée.

Méthodes basées sur les signatures à débit de transmission

Les statistiques basées dans les temps d'inter-arrivée ont l'avantage pratique sur la variation de délais de ne pas nécessiter la récollection des temps d'estampillage dans le récepteur et ne sont pas soumisses aux variations non-désirables des temps de inter-départ au récepteur. Néanmoins, il ne faut pas oublier que la variation du délai aide à identifier le lien avec goulot d'étranglement.

La figure 9.12 montre la variation de la signature à débit de transmission par rapport à la taille de paquets de test. La subfigure (a) est pour paquets test de taille 40 octets La subfigure (b) est pour paquets test de taille 1500 octets d'où les pics des paquets test de type I dominent.

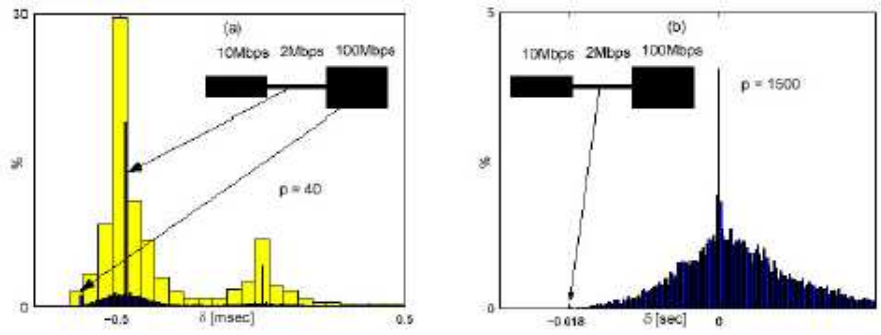


FIG. 9.12 – Variation de la signature du débit de transmission par rapport à la taille de paquets test

Méthodes basées sur les signatures de distribution

Avec ces méthodes seulement certains liens et en particulier aucun lien avec goulot d'étranglement dans le chemin d'aller peut avoir leur débit de transmission mesurés. En formalisant la méthode de Train de Paires de Paquets [**?**] on obtient deux nouvelles méthodes d'estimation de bande passante des liens basées sur les signatures de distribution. Elles se basent dans le cas d'où les paquets test sont de type B dans un lien, lien s, lequel correspond au goulot d'étranglement.

Méthode DS-1 : Proportionnel à la Taille de Paquets

En utilisant l'équation suivante avec p=p_i nous pouvons écrire :

$$t_i^* = x^S + c_i^s + n(s+1, H)$$

$$\delta_i = [x_i^{S_i} - t_i] + (c_i^{S_i}) + \sum_{h=S_i+1}^{H} \frac{[p_i^h - p_{i-1}^h]}{u^h} + \sum_{h=S_i+1}^{H} (\omega_i^h - \omega_{i-1}^h)$$

D'où :

n(s+1,H) = bruit de retour.

Alors une signature de distribution est vue comme la distribution de c_i^s avec $x^s = \frac{p}{u^s}$ comme un paramètre indépendant de la localisation. En mesurant le décalage de la distribution en fonction de p, par exemple via $\overline{t^*}(p)$, nous obtenons u^s en régressant en p et en utilisant :

$$\overline{t^*}(p) = const + \frac{p}{u^s}$$

.

Il faut faire attention pour varier p comprenant un rang consistent avec la signature de distribution. Cette méthode requiert que les mesures séparées soient faites avec différent p pour avoir trafic croisé stationnaire. La figure 9.13 montre l'histogramme $sub - t_i^*$ sans pics et pour p fixé ainsi que la moyenine des décalages d'histogrammes avec p (la pente est de $\frac{1}{u^S}$).

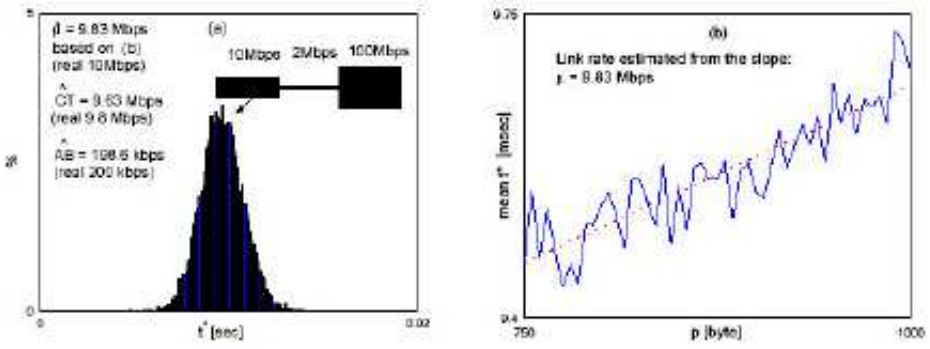


FIG. 9.13 – Estimation de la bande passante à partir de la signature de distribution

Méthode DS-2 : Proportionnel au Débit de transmission des Paquets

Cette méthode est une variante de l'antérieur d'où le composant t_i du test est exploité, car il n'est pas désirable pour varier la taille du test :

$$\frac{t_i^*}{t_i^s} = \frac{x^s}{t_i^s} + \frac{c_i^s}{t_i^s} + \frac{n(s+1,H)}{t_i^s}$$

D'où :

t_i^s : temps sur lequel c_i^s arrive au lien s.

En changeant t_i^s, change la distribution de c_i^s (et le bruit est re-dimensionné), néanmoins $\frac{c_i^s}{t_i^s}$ représente une moyenne de le débit de transmission de trafic croisé par unité de temps, lequel doit être une espérance constante assumant trafic croisé stationnaire. De la même manière nous dévions extraire u^s en régressant en $\frac{1}{t_i} = \frac{1}{t} : \frac{\overline{t^*}}{t} = const + \frac{p}{t^* u^s}$.

Nous avons substitue t (le paramètre d'entrée du test accessible) pour t_i^s dans la gauche en assumant que s est le lien avec la bande passante disponible minimale et que les test dans les liens d'aller sont de type I, alors $t = \overline{t_i^s}$.

Commentaires

Ces deux méthodes ont problèmes pratiques pour être implémentées. Le problème essentiel est que la précision d'estimateur requiert une variation large pour le paramètre de régression. Le rang est déterminé par le trafic croisé, nécessite éviter d'effacer les flux de retour, il doit être plus petit et variante dans le temps.

Pour contrôler le temps nécessaire pour effectuer des mesures ainsi comme limiter leur invasion, l'espace entre paquets doit être déterminé d'une manière adaptative. Un problème additionnel est la nécessité du trafic croisé stationnaire pendant les mesures.

Les requis des signatures de distribution sont opposées aux requis des signatures du débit de transmission. Des tailles de paquets test grandes sont appropriées pour éviter pics et paquets test associés aux source-à-source.

En envoyant des petits paquets test d'extrême à extrême est une stratégie optimale pour amplifier la signature du débit de transmission dans n'importe quel lien. Des paquets test plus grands réduiront l'effet de vol de paquets pour le bénéfice du pic au goulot d'étranglement. Des tests avec paquets plus grands sont plus utiles pour détecter le lien avec goulot d'étranglement dans la pratique, tandis que les test avec paquets de taille plus petite augmente les goulots d'étranglements secondaires.

9.4.4 Technique à Quartets de Paquets

Cette technique est appelée en anglais comme Packet Quartets[**?**], et elle est aussi basée sur les variations de délais. Elle propose résoudre les désavantages de la technique à paires de paquets courts-et-longs présentée dans la séction (IV.2), liées à l'utilisation du filtrage minimale et la nécessité d'exécuter deux phases.

Cette technique introduit une nouvelle famille de méthodes d'estimation de bande passante des liens dans une route. Elle a beaucoup des similitudes avec la technique ACCSIG présentée.

Principe

Un quartet de paquets est composé par deux paires de paquets. Chacun composée par un paquet test qui suit un paquet leader. Le paquet leader est limité par le champ de durée de vie.

Phase d'Échantillonnage

Les paires sont suffisamment séparées et envoyées pour éviter que soient mis en attente dans la même file d'attente pendant les périodes occupés. Les paquets test et leaders doivent être dans le même lien. La figure 9.14 montre le modèle d'échantillonnage de cette technique.

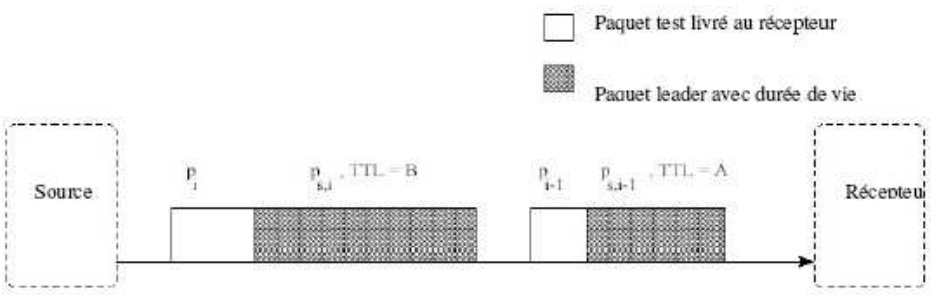


FIG. 9.14 – Modèle de test de la technique à Quartets de Paquets

Modèle et Phases d'Estimation

Les paires de paquets ont six paramètres, à dire :

1. Les tailles des paquets test : p_{i-1} et p_i.
2. Les tailles des paquets leaders : $p_{s,i-1}$ and $p_{s,i}$.
3. Les derniers liens que les paquets leaders traversant avant d'être effacés.

Soit un quartet de paquets envoyés d'extrême à extrême et un paquet leader qu'arrive dans un lien avec la file d'attente vide. Dû aux routers de type Store-and-Forward, les paquets test ne rentreront pas la file d'attente après un intervalle x_i^{h-1} contrôlé par le lien antérieur tandis que les paquets leader quitte la file d'attente après un temps de x_i^h contrôlé par le lien h.

L'habilité des paquets leader de garantir que les paquets test seront mis en attente derrière les premiers est limitée par le taux de service des débits de transmission des liens. Alors, les tailles des paquets est un paramètre de conception important et limitant en pratique l'applicabilité de ces méthodes sur liens à haut débit.

La séparation des paires n'invalide pas les quartets de paquets pour les raisons suivantes :

– Il n'est pas essentiel que les paquets test et les paquets leader soient d'extrême à extrême, le principal fondement est qu'eux partagent la même période occupée.
– La séparation pourra seulement persistée sur plusieurs liens.
– Il n'est pas nécessaire que tous les paires soient ensemble, mais seulement quelques-uns pour produire une signature suffisamment grande pour être detectée.

Il est assumé que les paquets test sont ceux qu'on observe et mesure au récepteur. Vu que les paquets leaders sont normalement effacés avant le récepteur, alors ils ne sont pas classifiés comme paquets test. Néanmoins, ils manifestaient des composants spéciaux sur les temps d'attente par rapport aux paquets test, ils sont plutôt caractérisés par sa durée connue et sa persistance à travers les liens.

La variation du délai jusq'au lien j peut être exprimée :

$$\delta_i^{(j)} = \delta_i^{(j-1)} + (x_i^j - x_{i-1}^j) + (\omega_i^j - \omega_{i-1}^j)$$

Avec : $\omega_i = [\omega_{i-1} + x_{i-1} - t_i]^+$. En assumant que le paquet de test i rentre au lien h avec son paquet leader pour joindre la même période occupée, le temps d'attente du paquet test dévient :

$$\omega_i^h = \omega_{s,i}^h + x_{s,i}^h + c_{s,i}^h - (\tau_i^h - \tau_{s,i}^h)$$

D'où :

$c_{s,i}^h$: Temps de service agrégé des paquets de trafic croisé arrivant au lien h entre le paquet leader à l'instant $\tau_{s,i}^h$ et le paquet test à l'instant τ_i^h.

Comme le temps d'arrivée au lien h est le temps du départ au lien h-1, en assumant que le paquet test et leur paquet leader partagent le même période occupée au lien h-1. En appliquant d'une manière récursive la variation du délai des paquets test dévient :

$$\delta_i = t^1_{s,i} - t^1_i + \sum_{h=1}^{A}(x^h_{s,i} - x^h_{s,i-1}) + \sum_{h=A+1}^{B} x^h_{s,i} - \sum_{h=A}^{B-1} x^h_{i-1} + \sum_{h=B}^{H}(x^h_i - x^h_{i-1}) +$$
$$+c^B_{s,i} - c^A_{s,i} + \sum_{h=1}^{A}(\omega^h_{s,i} - \omega^h_{s,i-1}) + \sum_{h=A+1}^{B}(\omega^h_{s,i} - \omega^h_{i-1}) + \sum_{h=B+1}^{H}(\omega^h_i - \omega^h_{i-1})$$

D'où :

$t_{s,i}{}^1$: Temps d'inter arrivé des paquets leader au premier lien.

Les propretés de la variation de délais sont différentes selon les trois scénarios suivants par rapport à l'occurrence de la durée de vie (voir figure 9.15) :

– $0 < A = B \leqslant H$
– $0 < A < B \leqslant H$
– $0 = A < B \leqslant H$

D'où :

H : Nombre de liens dans le chemin réseau.

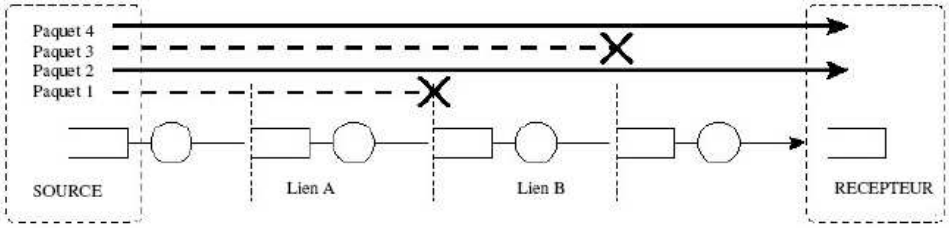


FIG. 9.15 – Quartets de paquets : Paquets leader 1 et 3 ont durée de vie limitée de lien A et B, paquets test 2 et 4 arrivent au récepteur

Le cas avec A = B = 0 correspond aux paquets test envoyés depuis le récepteur. Nous allons définir des méthodes d'estimation de bande passante à quartets de paquets dans le goulot d'étranglement pour chacune des régions des quartets.

PQ1 et PQ2 : 0 < A = B

On identifie deux temps de service, le premier est de signature d'accumulation, il est dû aux différences de taille des paquets leader est crée entre la lien 1 et le lien A, après les paquets leader sont effacés. Le deuxième est dû aux différences de taille des paquets test et crée du lien A jusqu'au récepteur. Si tous les deux sont actives il sera difficile d'identifier leur impact sur δ_i et sa proportion. La figure 9.16 montre ce cas. L'équation devient :

$$\delta_i = t^1_{s,i} - t^1_i + \sum_{h=1}^{A}(x^h_{s,i} - x^h_{s,i-1}) + \sum_{h=A}^{H}(x^h_i - x^h_{i-1}) + c^B_{s,i} - c^A_{s,i} + \sum_{h=1}^{A}(\omega^h_{s,i} - \omega^h_{s,i-1}) +$$
$$+ \sum_{h=B+1}^{H}(\omega^h_i - \omega^h_{i-1})$$

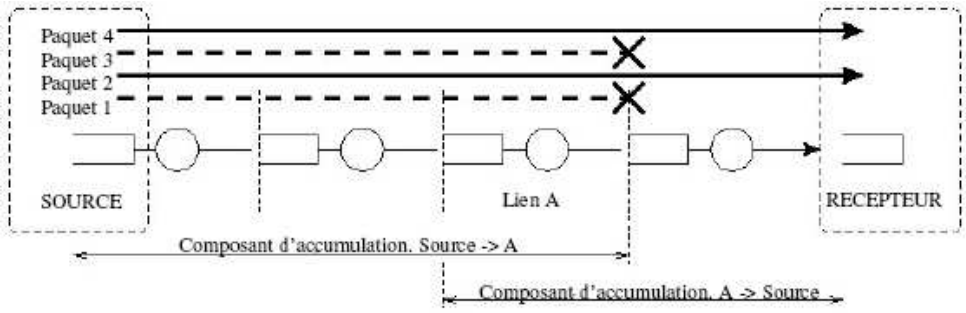


FIG. 9.16 – Quartets de paquets avec 0 < A = B

Méthode PQ1

En établissant p_i = p = constante, pour chaque paquet test effacer la deuxième composante d'accumulation, l'équation dévient :

$$\delta_i = \sum_{h=1}^{A}(x^h_{s,i} - x^h_{s,i-1}) + N^A_i$$

Cette équation diffère de l'équation de la technique ACCSIG dans le composant bruit. Alors, l'estimation de la bande passante aux liens est faite avec la même méthode que pour la technique ACCSIG.

Méthode PQ2

En établissant $p_{s,i}$ = p = constante, pour chaque paquet leader effacer la première composante d'accumulation, l'équation dévient :

$$\delta_i = \sum_{h=A}^{H}(x^h_i - x^h_{i-1}) + N^A_i$$

Cette équation est similaire à l'équation de la méthode PQ1, mais les méthodes définies par eux ont différentes propretés. Les estimations obéissent :

$$\frac{1}{u^{IP,ttl}} = \sum_{h=h_{ttl}}^{-1+h_{ttl+1}} \frac{1}{u^h}$$

Comparaison des méthodes PQ1 et PQ2

La différence essentiel de PQ2, pour déterminer la bande passante au lien h_{ttl}, est que nous dévions prendre les différences des variations du délai des étapes h_{ttl} et h_{ttl+1} à la place de h_{ttl} et h_{ttl-1} comme dans PQ1, ACCSIG et les méthodes basées sur l'outil "pathchar".

PQ1 et PQ2 produisent différentes estimations de bande passante dans chemins réseaux avec équipements travaillant dans la couche deux.

Dans PQ2 la détection du pic est basée sur la différence des tailles de paquets test. Comme la définition des quartets de paquets, les paquets test sont significativement plus petits que les paquets leader. Alors la distance entre les pics en utilisant PQ2 sera beaucoup plus petite que dans PQ1 dans le même lien et donc PQ2 est moins précis que PQ1.

Pour PQ2, ils existaient quelques problèmes de détection basée dans les pics de la moyenne dans liens trop chargés car les tailles des paquets test sont de taille différente. D'autres techniques de détection de pics doivent être utilisées pour PQ2. PQ2 a une performance mineure avec trafic croisé.

Avec PQ1, le nombre de composants de bruit dépend du nombre de liens à la place des variations de la durée de vie pendant les mesures, donc le bruit agrégé est plus bas et alors plus adaptable pour routes longues. Cette caractéristique est de PQ1 es partagée avec PQ2. Aucune de ces techniques est basée dans les réponses de messages ICMP. Le coût de ces avantages existe une sensibilité très haute aux erreurs dans la détection à pics.

Méthode PQ3 : 0 < A < B

Cette méthode est illustrée dans la figure 9.17. Dans cette méthode, même si les paquets test sont de taille similaire, p_i = p, et les paquets leader le sont aussi, $p_{s,i}$ = p_s, un membre d'accumulation reste dans l'équation et l'équation de la variation du délai dévient :

$$\delta_i = \sum_{h=A+1}^{A} x_{s,i}^h - \sum_{h=A}^{B-1} x_{i-1}^h + N_i^{A,B}$$

D'où :

$N_i^{A,B}$: Temps de non service.

L'équation de la méthode PQ3 est pour le cas de B = A + 1 :

$$\delta_i = x_{s,i}^{A+1} - x_{i-1}^{A} + N_i^{A,A+1}$$

D'où :

x_i : temps de service > 0

La bande passante du premier lien normalement est connue, permettant aux bandes passantes postérieures d'être estimées en appliquant cette équation d'une manière récursive.

Avec cette méthode un lien est isolé, une différence de plus est que l'histogrammes ont seulement un pic si nous posions t_i = t et $t_{s,i}$ = t_s. Si nous assumons que u^A es connu, et le bruit a des caractéristiques symétriques, alors l'estimateur de l'unique pic peut être obtenu par :

$$u^{h_{ttl}} = \frac{p_s}{(\delta - t_s + \bar{t} + \frac{p}{u^{h_{ttl-1}}})}$$

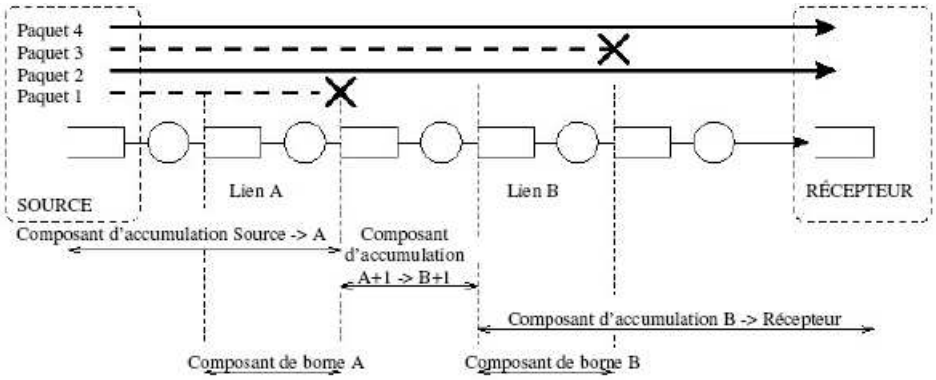


FIG. 9.17 – Méthode PQ3

L'estimation d'une étape est utilisée pour obtenir l'estimation de l'étape suivante. Celui peut produire d'erreurs d'estimation. Ce n'est pas le cas dans "pathchar", PQ1, PQ2 car elles estimaient en utilisant les différences des estimations de deux étapes.

Les estimations dans segments avec liens invisibles souffrent d'erreurs d'estimation de nature différente des ceux que nous avons rencontré auparavant. En mettant A = h_{ttl-1} et B = h_{ttl} et manipulant les équations antérieures la bande passante estimée est :

$$\frac{1}{u^{IP,h_{ttl}}} = \frac{1}{u^{h_{ttl}}} + (\frac{1-p}{p_s}) \sum_{h=1+h_{ttl-1}}^{-1+h_{ttl}} \frac{1}{u^h}$$

Les estimations de cette méthode sont similaires avec "pathchar", "clink" and ACCSIG pour liens jusqu'à 30 Mbps, néanmoins des différences plus larges sont par rapport à PQ1. La sensibilité à détecter les erreurs des pics est similaire à ACCSIG étant un avantage sur PQ1. PQ3 permet des mesures de bande passante plus grandes que PQ1 et PQ2 dû à que les tailles de paquets test et leader sont fixes.

PT1 et PT2 : 0 = A < B

Le cas spécial avec A = 0 est illustré dans la figure 9.18. Dans ce cas les quartets dévient triplets. Les tailles des paquets peuvent être choisies à canceller les effets des accumulations dans les équations. Néanmoins il est impossible de canceller les deux effets d'accumulations en même temps, sauf si les paquets test et ses paquets leader ont la même taille, lequel est incompatible avec le modèle de la technique. Alors deux méthodes correspondant à éliminer un effet à la fois chacune.

Méthode PT1

En posant p_i = p pour chaque paquet test élimine la deuxième composante d'accumulation :

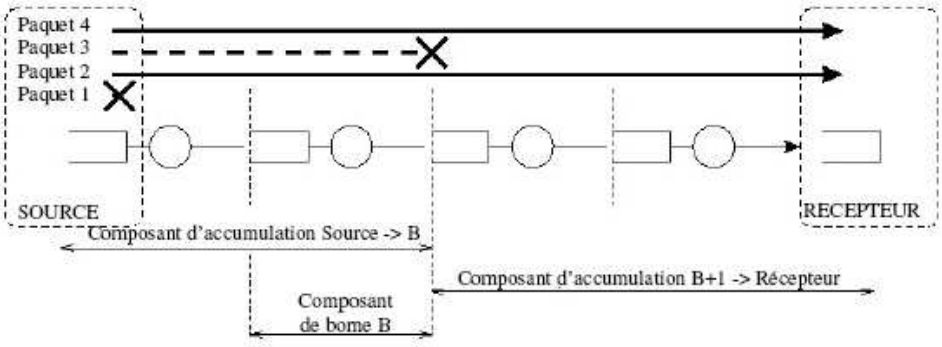


FIG. 9.18 – Quartets de paquets avec 0 = A < B

$$\delta_i = \sum_{h=1}^{B} (x_{s,i}^h - x_{i-1}^h) + x_i^B + N_i^{0,B}$$

Les propretés de cette méthode sont similaires à la méthode PQ3. L'estimation de la bande passante se fait avec l'unique variation du délai δ (en mettant $t_i = t$ at $t_{s,i} = t_s$) :

$$u^{h_{ttl}} = \frac{p_{s,i}}{[\delta - t_s + t - (p_{s,i} - p) \sum_{h=1}^{-1+h_{ttl}} \frac{1}{u^h}]}$$

Les mêmes erreurs potentielles que dans PQ3 restaient valables dans ces méthodes et les effets de liens invisibles sont exactement les mêmes. Cette méthode est très similaire à la technique de paire de paquets court-et-long (Packet Tailgating). Donc, les problèmes liés à cette dernière technique restaient valables pour la technique PT1.

Méthode PT2

En posant $p_{s,i} = p_{i-1} = p$ = constante pour chaque paquet leader cancelle la première composante d'accumulation :

$$\delta_i = \sum_{h=B+1}^{H} (x_i^h - x_{i-1}^h) + x_i^B + N_i^{0,B}$$

Cette méthode n'est pas pratique due aux incréments de sensibilité et les erreurs de détection de pics. Pour estimer la bande passante :

$$\frac{1}{u^{IP,h_{ttl}}} = \frac{1}{u^{h_{ttl}}} + (1 - \frac{p}{p_i}) \sum_{h=1+h_{ttl}}^{-1+h_{ttl-1}} \frac{1}{u^h}$$

Comparaison PT1 et PT2

Les erreurs de propagation et la sensibilité de la méthode PT2 généraient que ses estimations soient détériorées rapidement, même sans trafic croisé. Les résultats de la méthode PT1 son généralement bons et très consistants avec ceux de la méthode PQ3. Même pour les résultats avec liens invisibles.

La méthode PT1 en conjonction avec d'autres méthodes a le potentiel de détecter les liens invisibles, par contre la sensibilité à la détection de pics sont similaires à la méthode PQ2. Comme les estimations de PT1 peuvent souffrir des erreurs des entêtes dans la couche liaison, alors PQ2 est plus efficace pour détecter l'anatomie des liens invisibles.

La méthode PT1 est très proche de la technique à paire de paquets court-et-long en ce qui concerne à l'utilisation du champ de durée de vie.

PQ1 est la meilleure méthode puisqu'il offre des estimations précises avec peut de paquets test et bruit bas. ACCSIG est aussi efficace et précise. PQ2 peut aussi être utile tandis que PQ3 est considérablement moins utile. PT1 est très similaire à PQ3 mais moins précise. PT2 est très sensible et magnifie les erreurs ; elle n'est pas utile la plus part du temps.

9.4.5 Technique à Dispersion Temporelle de Paires de Paquets

Cette technique est connue en anglais comme Packet Pair Dispersion (PPD). Elle sert à estimer la capacité de bout en bout. Elle est utilisée dans l'outil "bprobe".

Cette technique a était proposé par Keshav dans [**?**] pour observer et contrôler les réseaux.

Principe

Consiste à mesurer la dispersion temporelle de deux paquets test tout au long le chemin à tester. La dispersion temporelle dans un lien spécifique c'est la distance temporelle entre le dernier bit de chaque paquet test. La figure 9.19 montre cette idée quand une paire de paquets test traversent un routeur.

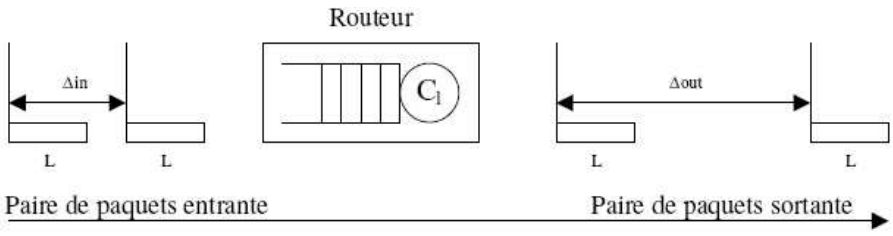


FIG. 9.19 – Idée Clé de la Technique de la Dispersion Temporelle par Paires de Paquets

Modèle

Le modèle de cette technique est connu aussi comme modèle à paire de paquets. Le modèle assume que la paire de paquets test ne trouvera pas du trafic tout au long du chemin. Si un lien de

capacité C_o connecte l'équipement source au chemin et les paquets test sont de taille L, la dispersion avant le lien de capacité C_i est Δ_{in}, la dispersion temporelle après le lien pourra être, en prenant en compte qu'il n'y a pas du trafic dans le lien :

$$\Delta_{out} = \max(\Delta_{in}, \frac{L}{C_i})$$

Ce modèle assume que les deux paquets sont mis en attente un après l'autre dans le lien avec goulot d'étranglement et non dans un autre lien. Le routeur dans le lien avec goulot d'étranglement suit une politique de type PAPS (Premier Arrivé Premier Servi, en anglais First Input First Output ou FIFO).

Ce modèle assume que le délai de transmission est linéaire par rapport à la taille (c'est pourquoi cette technique est aussi considérée comme de type déterministe) et que tous les routeurs dans le chemin réseau testé sont de type Store-and-Forward. Si le routeur utilise la politique de service d'attente à poids juste (Weighted Fair Queueing) la technique mesure la bande passante disponible du lien avec goulot d'étranglement [**?**].

Phase d'Échantillonnage.

- L'équipement source envoi multiples paires de paquets vers l'équipement récepteur.
- L'équipement récepteur mesure la dispersion entre les paires de paquets test.

Phase d'Estimation

En ayant un chemin vide (aucun lien sans trafic), la dispersion jusqu'au récepteur est de :

$$\Delta_R = \max_{i=0,\ldots,H}(\frac{L}{C_i}) = \frac{L}{[\min_{i=0,\ldots,H}(C_i)]} = \frac{L}{C}$$

D'où :

C : Capacité bout en bout du chemin.

L : Taille des paquets.

Δ_R : Dispersion jusqu'au récepteur.

H : Nombre de liens dans le chemin entre l'équipement source et récepteur.

Alors, le récepteur peut estimer la capacité du chemin à partir de :

$$C = \frac{L}{\Delta_R}$$

Commentaires

La technique à paires de paquets a était développée pour des réseaux aux attentes à poids justes mais non pour des réseaux aux attentes de type PAPS [**?**].

Ils existaient plusieurs problèmes dans cette technique, en particulier pour l'outil "bprobe", à dire :

1. D'erreurs d'attente aux routeurs.
2. Du trafic compétitif (trafic croisé) aux routeurs.
3. Des pertes de paquets test.
4. Des congestions dans le chemin de retour.
5. D'erreurs dans les processus de filtrage.
6. Étant donné que cette technique est basée sur réponses des paquets ICMP. Alors, cette technique produise des erreurs dû aux équipements de la couche de niveau deux du modèle ISO/OSI qui introduisent des délais et de la dispersion mais qui ne décrémentent pas le champ de la durée de vie (TTL).

Le trafic croisé de paquets, peut incrémenter o décrémenter la dispersion Δ_R en produisant sous-estimation ou sur-estimation de la capacité du chemin.

– Une augmentation de la valeur de la dispersion à plus de L/C produise une sous-estimation de la capacité. Ceci passe si le trafic croisé est transmit au milieu du chemin dans un lien spécifique.
– Une diminution de la valeur de la dispersion produise une sur-estimation de la capacité. Ceci passe si le trafic croisé retarde le premier paquet d'une sonde plus que le second paquet dans un lien qui est suivi du lien le plus étroit du chemin.

Keshav a montré dans [**?**] que la distribution de la dispersion par paires de paquets n'a aucune relation avec la bande passante disponible si les files d'attente des routeurs sont de type PAPS. En plus, si tous les routeurs utilisaient la discipline d'attente équitable (Fair Queueing), la technique à paires de paquets peut estimer la bande passante disponible dans le chemin.

Bolot a utilisé dans [**?**] un flux de paquets séparés à intervalles fixes pour prouver plusieurs chemins d'Internet et caractériser le délai et les pertes. Il a mesuré le temps d'aller-retour de paquets UDP ECHO et appliqué la technique à paires de paquets pour estimer les liens avec goulot d'étranglements

Dovrolis a montré dans [**?**] que la dispersion de la technique à paires de paquets suit une distribution à multi-modes qui selon lui est produite par les effets d'attente d'où la mode principale de la distribution des dispersions ne corresponde pas à la capacité du chemin c'est pourquoi nous ne pouvons pas utiliser des techniques statistiques traditionnelles.

Il faut examiner la distribution de la bande passante en termes des files d'attente, analyser les causes de chaque mode et les différences entre la mode de la bande passante et les autres modes.

Dans les techniques basées à paires de paquets la capacité d'un chemin réseau ne peut pas être déterminée par simple mesure de la dispersion minimale car cette valeur peut être le résultat d'un lien postérieur au lien avec moindre capacité.

La solution a ce problème consiste à envoyer plusieurs paires de paquets ; en utilisant des méthodes statistiques pour filtrer les mesures de bande passante erronées se réduit l'affectation du trafic croisé.

Dans la même référence [**?**], il est montré aussi qu'en utilisant la taille de paquet maximal n'est pas toujours un choix optimal. Si les paires de paquets de distincts tests sont de tailles différentes, les modes des sous-capacité de la distribution devient plus longes et faibles.

Plusieurs solutions ont été proposées pour résoudre les problèmes mentionnés ci-dessus :

1. Pour les erreurs d'attente : il est utilisé un nombre de paquets à plusieurs tailles. Les paquets de taille supérieure prendront plus de temps aux routeurs et augmenteront la possibilité d'attentes.
2. Pour le trafic croisé : En envoyant un grand nombre de paquets il est incrémenté la probabilité que quelques paires de paquets ne seront pas affectées par le trafic croisé.
3. Pour les pertes de paquets test : En envoyant des paquets avec plusieurs tailles les pertes de quelques paquets n'affectent pas les estimations.
4. Pour la congestion du chemin de retour : En utilisant des techniques de filtrage nous pouvons extraire les bruits d'attente ajoutés cela nous permets de récupérer les bonnes estimations.
5. Pour les erreurs dans le filtrage : Deux méthodes de filtrage sont proposés et basés au calcul d'une intervalle d'erreur autour de chaque estimation. L'intervalle d'erreur peut être étendu jusqu'à en avoir une estimation satisfaisant de la bande passante du lien avec goulot d'étranglement. Le filtrage à intersection trouve l'intersection entre les intervalles d'estimations et calcule leur intersection, avec l'idée de trouver l'estimation des deux ensembles. Comme les paquets longs fournissent meilleurs estimations les estimations commençaient avec les paquets longs vers les paquets petits. Le filtrage à union combine le sur emplacement d'intervalles en utilisant l'union d'ensembles et en sélectionnant l'intervalle seulement si suffisamment d'ensembles contribuent à lui. Les deux méthodes produisent un intervalle et le point au milieu de l'intersection est retourné comme l'estimation finale. La figure 9.20 montre ces deux méthodes de filtrage. L'approche à union fournisse moins de dispersion que les estimations avec l'approche d'intersection. L'outil "bprobe" ai implementé la méthode de filtrage à union.

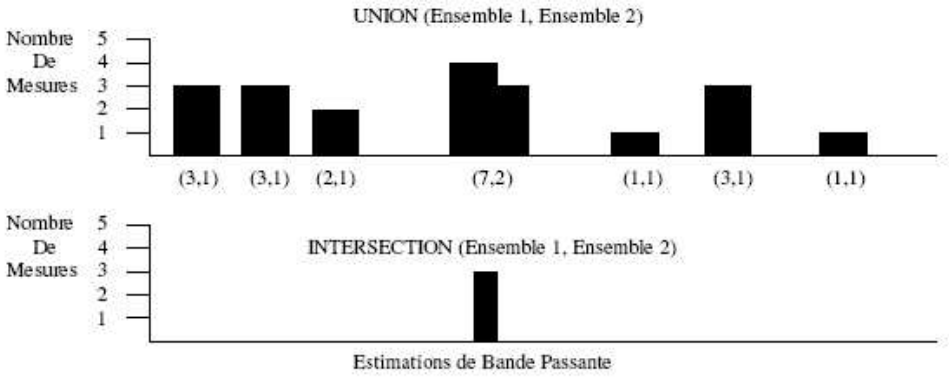


FIG. 9.20 – Techniques de Filtrage

Un des problèmes des techniques à paires de paquets est de ne pouvoir pas mesurer une bande passante supérieur à la bande passante avec laquelle la source envoie les paquets test [**?**]. Le problème augmente si la source envoie petits paquets ou envoie lentement paquets ou les deux. Alors, la bande passante potentielle est inférieure à la bande passante réelle du goulot d'étranglement dans un chemin, et toutes les mesures peuvent être mauvaises. La plus part des paquets HTTP et FTP ont

une bande passante potentielle. Malheureusement la plus part des paquets dans un flux n'ont pas une bande potentielle.

L'outil "bprobe"

L'outil "bprobe" provisionne une estimation du lien moins congestionné dans le chemin réseau [**?**]. La source envoie une séquence de paquets ICMP ECHO et l'outil mesure les temps d'inter arrivée des paquets retournés.

9.4.6 Technique à Filtrage de la Bande Passante Potentielle

Elle est connue en anglais avec le nom de Potential Bandwidth Filtering (PBF)[**?**]. Cette technique propose des solutions pour résoudre quelques problèmes qui présentaient les techniques basées dans la technique de l'outil "pathchar" et à paires de paquets :

1. Des calculs lents.
2. Pauvre précision.
3. Pauvre passage à l'échelle.
4. Peut de robustesse statistique.
5. Pauvre agilité pour s'adapter aux changements de bande passante.
6. Peut flexibilité au déploiement.
7. Imprécision avec différents types de trafic.
8. Les techniques de l'outil "pathchar" et à paires de paquets ont des estimations lentes.

Dans [**?**] les auteurs nomment les algorithmes traditionnels pour filtrer les échantillons de bande passante comme Filtrage de Bande Passante Mesurée (Measured Bandwidth Filtering ou MBF). Cette technique, étant un avancement des techniques traditionnels, ils l'ont nommée, technique de Filtrage de Bande Passante Potentielle.

Principes

Les solutions proposées aux problèmes mentionnés sont :

- Utiliser une fenêtre de paquets pour s'adapter rapidement aux changements de bande passante. Une petite fenêtre est 144% plus précise que une fenêtre infinie.
- Utiliser la technique à Paires de Paquets Basés Seulement au Récepteur (Receiver Only Packet Pair ou ROPP) pour combiner précision et facilité de déploiement.
- Utiliser la technique de filtrage potentiel de bande passante (Potential Bandwidth Filtering) pour incrémenter la précision en présence d'une variété de tailles de paquets. En particulier l'estimation basée au noyau. Cette implémentation à améliorer 37% la précision.
- Eviter l'utilisation d'heuristiques.

Modèle

Le principal problème avec la technique à paires de paquets est le filtrage du bruit ajouté. Le modèle de cette technique utilise l'algorithme d'estimateur à densité noyau (kernel density estimator algorithm). La fonction kernel définit est :

$$\int_{-\infty}^{+\infty} K(t)dt = 1$$

Alors, la densité dans n'importe quel point x est :

$$\frac{1}{n}\sum_{i=1}^{n}[K(x - x_i)h^2]$$

D'où :

h : largeur noyau

n : nombre de points dans h de x

x_i : l'i-éme point

La fonction largeur noyau utilisée est :

$$y = \{^{1+x, x \leqslant 0}_{1-x, x > 0}$$

Cette fonction a les propriétés désirées que provisionnent plus de poids aux échantillons près du point lequel on veut estimer la densité. En plus elle est simple et rapide à déterminer. L'algorithme d'estimateur noyau est valable et provisionne des résultats plus précis. Il n'a pas non plus des considérations sur leur distribution.

Phase d'Échantillonnage

La phase de test est similaire à celle de la technique IV.1 ou de la technique IV.2.

Phase d'Estimation

La principale changement de cette technique est dans cette phase d'estimation. En particulier l'idée du filtrage de la bande passante potentielle.

L'idée générale du filtrage de bande passante potentielle consiste à corréler la bande passante potentielle et la bande passante mesurée d'un échantillon en décidant comment les filtrer. Les échantillons avec la même bande passante potentielle et bande passante mesurée ne sont particulièrement informatifs car la bande passante réelle ne peut pas être supérieure.

Les échantillons avec bande passante mesurée supérieur et bande passante potentielle inférieure sont compressés temporellement et doivent être filtrés. Les échantillons avec bande passante potentielle supérieure et bande passante mesurée inférieure sont moins informatives car ils indiquaient la vraie bande passante. La figure 9.21 montre comment travail le filtrage de bande passante potentielle. Tous les échantillons au-dessus de la ligne de x=y sont filtrés.

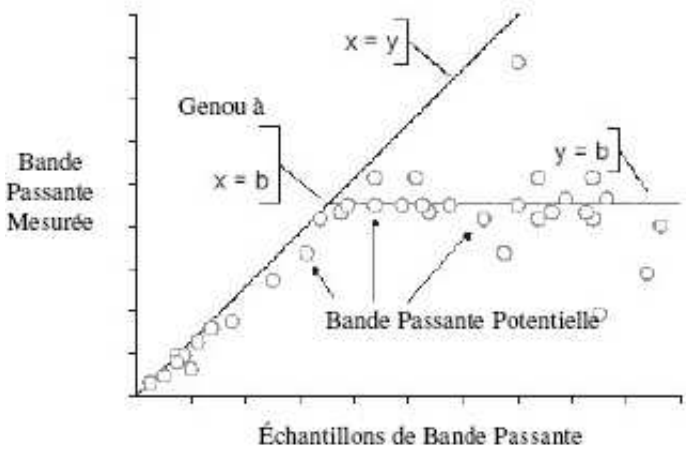


FIG. 9.21 – Filtrage Potentiel de Bande Passante

Commentaires

Cette technique prenne l'estimateur noyau (Kernel estimator) de la densité de probabilité comme outil pour le filtrage statistique à la place d'histogrammes pour détecter la mode de la distribution par paires de paquets.

Cette technique assume que la capacité du chemin est liée à la gamme de mesures de bande passante la plus ordinaire (la mode locale de la distribution la plus forte).

9.4.7 Technique ABwE

Connue en anglais avec le même nom ainsi que l'outil[**?**] cette technique utilise des paires de paquets avec même taille décalée par un délai initial entre les paquets. Elle sert à mesurer la bande passante disponible dans un rang depuis quelques Mbps à 1000 Mbps. Elle sert aussi pour détecter changements de bande passante produite par le routage ou les congestions.

Les techniques compliquées nécessitaient plus temps de calcul. Particulièrement les techniques basées dans la modélisation mathématique. Quelques techniques essaient de remplir les files d'attente au goulot d'étranglement et de trouver les paramètres de cette file d'attente. Apparemment, elles sont des bons résultats aux liens lents. Néanmoins, il est plus difficile de remplir une file d'attente sur liens avec vitesse à 622 Mbps ou supérieur.

Principe

Cette technique est basée sur le principe de mesurer le délai (Td) entre deux sondes de paires de paquets test (PP1 et PP2). Elle assume que le chemin est composé des équipements travaillant avec la politique PAPS. Elle assume aussi que les paquets test peuvent être étalés par le trafic croisé dans les nœuds intermédiaires du chemin à tester.

Modèle

Le Td augmentera d'une manière discrète dû aux paquets du trafic croisé lesquels ont une taille fixe et discrète. Le Td au récepteur est le résultat de la superposition du trafic croisé et les capacités différentes dans le chemin :

- Quand la capacité du lien entrant est supérieur à celle du lien sortant dans un équipement réseau les paquets des sondes sont serrées (contracted). Dans ce cas le temps de séparation entre les sondes augmente et peut être remplit par paquets de trafic croisé aux équipements qui suivent cette contraction. Il existe donc la possibilité que cet étalement soit fait dans le lien avec goulot d'étranglement. Il est assumé que cette séparation de temps ne soit pas remplit au total pour le trafic croisé qui sera ajouté dans tout le reste du chemin.
- Quand la capacité du lien entrant est inférieure à celle du lien sortant dans un équipement réseau les sondes de paquets sont étalées (stretched).

D'après mesures la plus part du trafic est composé en grande partie pour paquets de tailles longues et moindres avec paquets de tailles petites. Le délai est donc plus affecté par les paquets de taille longue. Un paquet long puisse produire un délai équivalent à la somme des délais produite par 20-35 petits paquets. Il est plus facile à détecter l'affectation d'un paquet long que d'un paquet petit. Alors ce-sont les paquets longs qui déterminent la valeur moyenne.

Le Td augmente de deux façons :

1. Td augmente d'une manière lineaire : $\text{Td}_i = \text{Td}_{i-1} + \delta_i$

Quand le lien actuel H_i a un facteur d'utilisation ρ supérieure au lien H_{i-1} ou $\delta_i \geqslant \text{Td}_{i-1}$.

D'où : δ_i est un incrément de Td que corresponds à la charge du trafic croisé (CT) dans le lien H_i (y compris la sonde des paires de paquets PP) : $\delta_i = \frac{L_{PP}}{C_i} + \frac{E(N)*L_{CT}}{C_i}$.

1. Td augmente d'une manière non-linéaire : $\text{Td}_{i-1} = 0$ et $Td_i = \frac{L_{PP}}{C_i} + \frac{E(N)*L_{CT}}{C_i}$.

Ce cas est déterminé par l'étalement dans la bande étroite. Il peut aussi apparaître dans plusieurs endroits tout au long du chemin. Le lien avec le plus long $(\frac{L_{PP}}{C})$ déterminera le Td.

Nous savons que les Td's dans un groupe des mesures est produit par le même lien. Nous savons aussi que les valeurs des Td's sont discrèts et qui dépendent de n*NTT. Nous essayons de superposé les différences de Td avec les classes NTT. Si la superposition est réussit nous pouvons déterminer le QDF comme le taux de Td_{var} et NTT_{classe} :

$$QDF_i = \frac{(Td_{ij} - Td_{jinit})}{NTT_{classe}}$$

$$Td_{ij} = \frac{L_{PP}}{C_i} + \frac{QDF_i * L_{CT}}{C_i}$$

$$C_i = \frac{(L_{PP} + QDF_i * L_{CT})}{Td_{ij}}$$

Phase d'Échantillonnage

1. La source envoie un train de plusieurs sondes de paires de paquets (typiquement 20) très proches vers un récepteur.
2. Le délai est converti en bande passante.
3. Cette technique utilise une valeur empirique et attendue des tailles de paquets du trafic croisé (L_{CT}).
4. Ensuite elle utilise le facteur du délai d'attente (Queueing Factor Delay ou QFD) à la place de la valeur réelle de la file d'attente.
5. Les résultats sont les capacité résiduelles d'un chemin (nommé aussi bande passante disponible ou bande passante offerte).

Phase d'Estimation

Nous commençons notre évaluation en déterminant le Td le plus fréquent dans toutes les mesures des sondes (la valeur creuse de la distribution de fréquence de Td) car il est déterminé par le lien avec la bande passante étroite ou le lien avec le facteur d'utilisation le plus élevé.

La largeur de la fenêtre dans lequel nous observons cette valeur est définit dynamiquement comme le niveau de pourcentage du Td mesuré. Nous pouvons ensuite filtrer et éliminer les valeurs fausses de Td qui sont beaucoup plus supérieures ou inférieures à la valeur creuse de Td.

Ils existaient des différences trop grandes entre les statistiques des mesures aux périodes courtes et longues. Dans les mesures à périodes courtes nous avons seulement une valeur de Td ou maximum deux valeurs. Dans les mesures à périodes longues nous avons plusieurs valeurs de Td. Ces dernières aussi comprennent des valeurs rares. En observant le comportement des statistiques pour les mesures à périodes courtes, nous pouvons voir l'étalement des valeurs creuses de Td. Elles deviennent plus fortes ou plus débiles ou disparaîtrent ou réapparaîtrent. Les mesures à périodes plus longues cachaient ce phénomène.

Si nous ne pouvons pas trouver une valeur de Td représentative. Il va falloir obtenir le moyennes de toutes les valeurs. La valeur de Td obtenue correspond au goulot d'étranglement maximal de la bande passante dans le chemin pendant l'intervalle des mesures. Si la charge du chemin augmente le Td aussi augmente.

Le Td est toujours composé para deux parties : Td = Td_{init} + Td_{var}.

La première partie est commun à toutes les mesures individuelles et est une valeur stable (Td_{init}). Elle est déterminée par le lien avec la bande passante étroite. $Td_{init} = \frac{L_{CT}}{C_{nb}}$, d'où L_{CT} est la valeur moyenne de la taille des paquets du trafic croisé et C_{nb} est la capacité du lien avec la bande passante étroite. Si le lien avec bande passante étroite n'existe pas nous assumons que la plus longue

séparation des paires de paquets est déterminée par le lien le plus chargé et la valeur de Td es déterminée par $Td_{init} = \frac{L_{PP}}{C} + \frac{E(N)*L_{CT}}{C}$. D'où C est la capacité du lien qui produise la plus longue séparation (le goulot d'étranglement).

La deuxième partie est variable et il se croit qu'elle représente les changements des files d'attente (Td$_{var}$). Pour notre valeur mesurée de Td, nous calculons : $Td_{jinit} = \min_i(Td_{ij}|1 \leqslant i \leqslant 20)$. D'où Td$_{jinit}$ est un des mesures Td$_{ij}$ dans le j-ème groupe des mesures. D'après plusieurs mesures nous pouvons identifier si la valeur creuse est due à la bande passante étroite ou bien au facteur multiplicatif.

L'outil utilise la moyenne de Td pour obtenir ses estimations. La conversion résiduelle ce fait comme un pas final de tout le processus.

Commentaires

Pour chaque sonde de 20 paires de paquets, nous essayons de trouver le Td le plus fréquent, les facteurs multiplicatifs, la moyenne, etc. Cette technique présente les problèmes suivants :

- Avec cette technique, il n'est pas possible de dire que le goulot d'étranglement es dans un lien du chemin particulier mais seulement de dire que le goulot d'étranglement a une valeur de capacité C.
- La conversion du domaine temporelle à la bande passante est un point débile dans toutes les techniques basées dans la dispersion.
- Les interprétations du L$_{CT}$ avec les PP's et les MTU ont faillit. Avec une valeur de 700 octets ont bien marché.
- Cet outil est capable de montre en temps réel (en périodes de deux minutes) des changements dans les performances du réseau.
- Pour créer des graphes plus propres l'algorithme EWMA (Exponentially Weighted Moving Averages) est utilisé.
- Cet outil peut mesurer la capacité disponible dans un lien entre les valeurs de plusieurs Mbps à 1 Gbps.
- L'outil peut être utilisé en mode continu et détecte tous les changements importants dans la bande passante produits par des congestions ou routages impropres.

9.4.8 Technique de Séparation Initiale à Incréments

Cette technique est connue en anglais comment Initial Gap Increasing ou IGI [**?**, **?**] Elle sert à estimer la bande passante disponible. Elle est une technique combinée car elle utilise d'autres outils actives pour mesurer le goulot d'étranglement.

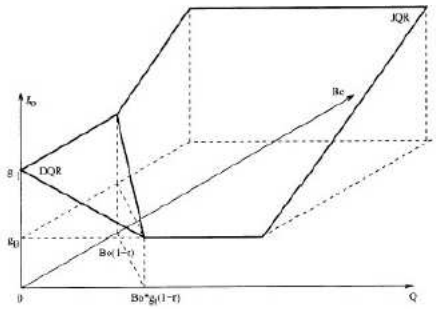
Principe

La sépartion initiale des paquets test joue un rol très important pour estimer la bande passante disponible d'une route.

Cette technique fixe une séparation initiale, entre paires de paquets, produissant une correlation haute au récepteur entre le trafic compétitif au goulot d'étranglement et la séparation des paquets.

Modèle

Cette technique propose un modèle d'un lien basé à la séparation de paquets, il capture la rélation entre le trafic compétitif et la variation de la séparation tout au long de la route. Elle assume que les files d'attente aux routeurs sont de type PAPS et les paquets test ont une taille fixe. La figure 9.22 montre la séparation de sortie (g_O) comme une fonction de la taille des files d'attente (Q) et le débit du trafic compétitif (B_c).



g_I	the **initial gap**, the time between the first bits of P1 and P2 when they enter the router; it includes P1's transmission delay on the input link
g_B	the **bottleneck gap**, the transmission delay of the probing packet on the output link; it is also the gap value of two back-to-back probing packets on the bottleneck link
g_O	the **output gap**, the time between the first bits of P1 and P2 when they leave the router, i.e., on the bottleneck link
B_O	the bottleneck link capacity
B_C	the competing traffic throughput for the time interval between the arrival of packets P1 and P2
Q	the queue size when packet P1 arrives at the router
L	the probing packet length
r	$r = g_B/g_I$

FIG. 9.22 – Modèle d'un lien basé à la séparation des paquets

Car l'intervalle de temps entre deux paquets test est de 1ms, le modèle assume aussi que le trafic compétitif entre deux paquets (P1 et P2) est constant.

La différence clé entre les deux régions (Disjoint Queueing Region ou DQR et Joint Queueing Region ou JQR) du modèle consiste en si les deux paquets tombaient dans la même période d'attente. Une période d'attente est définie comme le temps auquel la file n'est pas vide. Deux périodes d'attente consecutives sont séparées par une file d'attente vide.

Si la file d'attente dévient vide après le départ de P1 et avant l'arrivée de P2 (région DQR). Alors, la séparation de sortie est :

$$g_O = g_I - \frac{Q}{B_O}$$

Sous autres conditions (région JQR), la séparation de sortie est :

$$g_O = g_B + \frac{B_c g_I}{B_O}$$

D'où :

g_O : Séparation à la sortie

g_I : Séparation d'entrée

g_B : Séparation au goulot d'étranglement

B_O : Débit à la sortie

Le modèle d'un lien basé à la séparation de deux paquets n'applique pas directement aux trains de paquets. Soit un train de paquets test auquel M séparations sont incrémentales, K sont sans chan-

gement, et N sont réduites. Le trafic compétitif estimé par la formule IGI est (seulement pour la région JQR) :

$$\frac{B_O \sum_{i=1}^{M} (g_i^+ - g_B)}{(\sum_{i=1}^{M} g_i^+ + \sum_{i=1}^{K} g_i^= + \sum_{i=1}^{N} g_i^-)}$$

D'où :

$g^+ = \{g_i^+ | i = 1, ..., M\}$

$g^= = \{g_i^= | i = 1, ..., K\}$

$g^- = \{g_i^- | i = 1, ..., N\}$

Le numérateur est la quantité de trafic compétitif arrivant au routeur R1 pendant la période de test. Le denominateur est la durée totale de test.

Le taux de transmission moyenne de paquets est donée par la formule PTR :

$$\frac{[(M + K + N)L]}{[\sum_{i=1}^{M} g_i^+ + \sum_{i=1}^{K} g_i^= + \sum_{i=1}^{N} g_i^-]}$$

D'où :

L : Taille des paquets test.

Phase d'Échantillonnage

- La source envoie vers le récepteur une séquence de trains de paquets avec une séparation initiale et incrémentale.
- Calculer la différence entre la séparation de paquets moyenne à la source et au récepteur.
- Terminer si cette différence est égale à zéro.

Pseudocode de l'algorithme IGI :

{

/*Initialiser*/

probe.num = PROBENUM ;

packet_size = PACKETSIZE ;

g_B = GET.GB() ;

init.gap = g_B/2 ;

gap.step = g_B/8 ;

src_gap_sum = probe.num * init.gap ;

dst _gap_sum = 0 ;

```
/*chercher la valeur de la séparation au point de changement*/
while ( !GAP_EQUAL(dst_gap_sum, src_gap_sum)) {
init.gap+ = gap_step ;
src_gap_sum = probe_num * init_gap ;
SEND_PROBING_PACKETS(probe_num, packet_size, init_gap) ;
dst.gap.sum = GET_DST_GAPS() ;
}
/*calculer la bande passante disponible en utilisant la formule IGI*/
inc_gap_sum = GET_INCREASED_GAPS() ;
c_bw = b_bw * inc_gap_sum / dst_gap_sum ;
a_bw = b_bw – c_bw ;
}
```

Phase d'Estimation

Si la différence est égale à zéro cela signifie que les algorithmes IGI et PTR travaillent dans le point de changement montré sur la figure 9.23.

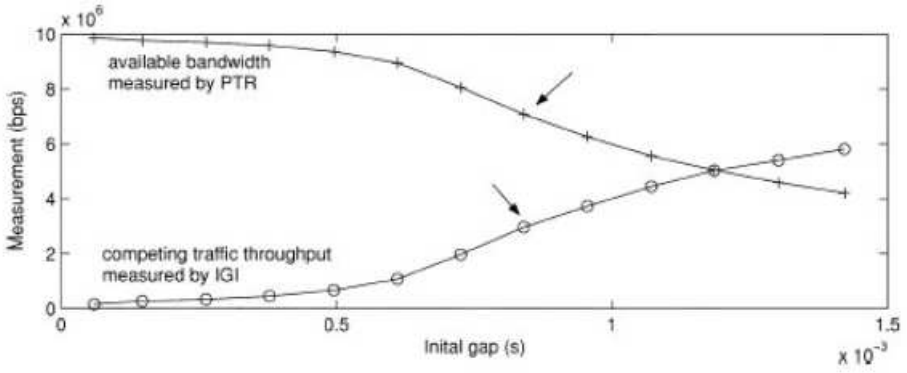


FIG. 9.23 – Point de changement des algorithmes IGI et PTR

Une fois les algorithmes IGI et PTR travaillent dans le point de changement la bande passante disponible est determinée en soustrayant le débit estimé du trafic compétitif de la capacité du lien avec goulot d'étranglement. D'erreurs dans les estimations de la capacité du lien avec goulot d'étranglement affécteront la précision d'estimateur de la bande passante disponible.

La capacité du goulot d'étranglement peut être mesurée par d'autres outils comme : bprobe, nettimer ou pathrate, lesquels nous donnent des précisions adéquates.

Commentaires

La séparation initiale entre paires de paquets est un paramètre critique quand on estime la bande passante disponible. Elle doit être sélectionnée d'une manière dynamique pour atteindre plus de précision.

Les formules assument que n'existaient pas des pertes et re-ordre de paquets. Une des clés consiste en bien choisir le pas de séparation (gap_step) et la séparation initiale (init_gap). L'algorithme découvred'une manière automatique le point de changement par le test suivant :

$$\frac{|src\_gap\_sum - dst\_gap\_sum|}{\max(src\_gap\_sum, dst\_gap\_sum)} < \delta$$

Cette technique estime la bande passante disponible plus vite que la technique Pathload [**?**, **?**] présentée dans la section IV.10 de ce rapport et avec presque la même précision.

À part la séparation initiale entre paire de paquets, trois autres parametres aussi affectaient la précision de l'algorithme :

1. La taille des paquets test (700 octets).
2. Le nombre de paquets test (60 paquets).
3. D'erreurs de mesures aux séparations. L'erreur de séparation initiale à la source est petit tandis que l'erreur au récepteur aura plus impact car elle varie les valeurs des séparations utilisées dans les formules.

Technique à Taux de Transmission de Paquets

Cette technique est connue en anglais comment Packet Transmission Rate ou PTR [**?**, **?**]. Elle sert aussi à estimer la bande passante disponible. L'unique différence entre cette technique et la technique IGI consiste en les dernières lignes de l'algorithme de la phase de test. Tout le reste est similaire.

Phase d'Échantillonnage

Algorithme PTR :

{

/*Initialiser*/

probe.num = PROBENUM ;

packet_size = PACKETSIZE ;

g_B = GET.GB() ;

init.gap = g_B/2 ;

gap.step = g_B/8 ;

src_gap_sum = probe.num * init.gap ;

```
dst _gap_sum = 0 ;
/*chercher la valeur de la séparation au point de changement*/
while ( !GAP_EQUAL(dst_gap_sum, src_gap_sum)) {
init.gap+ = gap_step ;
src_gap_sum = probe_num * init_gap ;
SEND_PROBING_PACKETS(probe_num, packet_size, init_gap) ;
dst.gap.sum = GET_DST_GAPS() ;
}
/*calculer la bande passante disponible en utilisant la formule IGI*/
ptr = [ packet_size * 8 * (probe_num – 1) ] / dst_gap_sum.
}
```

9.4.9 Technique des Modes à Groups de Paquets

Cette technique est appelée en anglais Packet Bunch Modes (PBM)[**?**]. Elle sert à mesurer la bande passante du goulot d'étranglement. Elle propose résoudre plusieurs problèmes avec la technique à paires de paquets évoqués dans [**?**], à dire :

1. Des livraisons de paquets hors ordre.
2. Des limitations dus à la résolution des horloges.
3. Des changements de bande passante aux liens avec goulot d'étranglement.
4. Des liens avec goulot d'étranglement à canaux multiples.

Principe

Le principe consiste à envoyer des trains de paquets de différentes largeurs pour détecter des liens à multiples canaux.. Il s'agit aussi de composer d'estimateurs pour un rang de taille de groups de paquets en permettant multiples valeurs de goulots d'étranglements ou valeurs apparentes de goulots d'étranglements.

En considérant plusieurs tailles de groupes, nous pouvons placer, au récepteur, des résolutions limitées d'horloge et la possibilité de canaux multiples ou partage de charge entre multiples liens en évitant le risque de sous-estimation dû au bruit et en diluant larges groupes de paquets car cette technique seulement considère des petits groups de paquets.

En permettant de trouver multiple valeurs de goulots d'étranglements, nous permettons des effets à multiples liens (multiple canaux), et la possibilité de change du goulot d'étranglement.

En permettant plusieurs valeurs à multiples liens il est possible d'utiliser l'estimateur le plus commun, la moyenne, car elle assume une mode. Et donc se focaliser en identifier le maximum local dans la densité d'estimateurs.

1. S'ils existaient deux modes fortes, pour lesquelles le premier est trouvé seulement au début de la connexion et un autre à la fin. Alors il existe évidence que le goulot d'étranglement change.
2. S'ils existaient deux modes fortes lesquelles étendaient la même portion de la connexion, et si un se trouve seulement pour une taille de groupe de paquets "m" et l'autre pour taille de groupe de paquets >"m", alors il existe évidence que se soit un lien avec goulot d'étranglement à m-canaux.
3. Ils existaient deux situations, pour un lien qu'évoque un changement et un lien à multiple-canaux.

Modèle

- La taille des groups est noté k, elle est appelée ampleur (extent).
- Pour chaque ampleur, il faut avancer une fenêtre sur les intervalles au récepteur. Cette fenêtre est de valeur nominale k mais elle est toujours étendue pour inclure (k*MSS) octets de données. Il n'est pas étendu pour k=1.
- Si toutes les arrivées occurrentes sans que l'horloge du récepteur ait avancé ($\Delta T_r < C_r$), il faut étendre aussi la fenêtre.
- Si les arrivées dans la fenêtre sont hors ordre ou si elles sont transmises dû au temps de retransmission, il faut sauter l'analyse du group de paquets, car les temps d'arrivées ne reflètent pas la bande passante du goulot d'étranglement.
- Si un événement arrive quand il y a k paquets dans le chemin (l'air) l'analyse de ce group est éliminée.

Les bornés sont calculés par :

$$\Delta T_r^- = \max(\Delta T_r - C_r, 0)$$
$$\Delta T_r^+ = \Delta T_r + C_r$$

D'où :

ΔT_r = Espace de temps entre un groupe de paquets au récepteur.

C_r = Résolution d'horloge au récepteur.

Pour les analyses, il est considéré le terme de Paires de Paquets Basés au Récepteur (Receiver Based Packet Pair ou RBPP) avec lequel il est observé la forme d'arrivées au récepteur. Il est assumé que le récepteur connaît toute l'information du temps, le récepteur aussi connais si les paquets n'étaient pas étirés par le réseau, avec la possibilité d'effacer ces paquets de l'analyse. L'analyse RBPP est considérablement plus précise que les techniques de Paires de Paquets Basés à la Source (Source Base Packet Pair ou SBPP), car elle élimine le bruit additionnel et l'asymétrie du chemin de retour, ainsi comme le bruit dû aux délais des acquittements. Il était trouvé que ce bruit est très large en pratique.

Phases d'Échantillonnage et d'Estimation

Cette technique essaie tailles de groupes de paquets entre deux et cinq. S'il est requis la résolution d'horloge limitée ou pour la faille à trouver une estimation de bande passante (près d'un quart de

toutes les mesures dû à la résolution d'horloge limitée), elle essaie des tailles de groupes progressives jusqu'à la valeur maximale de 21 paquets. Tout cela requiert de connaître le temps relatif entre les paquets, comment ils arrivaient les uns par rapport aux autres et leur taille.

Cette technique utilise une grande composante d'heuristiques et elle est très difficile à comprendre.

Commentaires

PBM est une technique d'estimation basée sur une grande quantité d'heuristiques. C'est pourquoi elle est très compliquée à comprendre.

Implémentée dans l'outil "nettimer", cette technique est basée sur un modèle déterministe du délai des paquets que lui utilise pour calculer la propriété de paires de paquets dans réseaux avec politique d'attente PAPS ainsi comme cette technique pour mesurer la bande passante des liens [**?**].

Cette technique consume moins des ressources du réseau, elle n'utilise pas des réponses ICMP et elle n'est pas liée dans le temps avec les acquittements. Malheureusement cette technique n'a pas de précision dans chemins avec plusieurs sauts [**?**].

Cette technique produise des estimations finales en termes de barres d'erreur $\pm$20% autour d'estimation du goulot d'étranglement, mais peut être plus étroite si les estimations sont autour d'une valeur particulière ou plus ample si la résolution d'horloge prévient des bornés plus fines.

Si le chemin réseau est complètement sans charge sauf la charge de la connexion (c'est à dire sans trafic croisé), alors nous avons : φi = γi, toutes les variations du délai sont dues aux délais antérieurs :

$$\beta = \frac{\sum_i (\varphi_i + \phi_i)}{\sum_j (\gamma_j + \phi_j)}$$

D'où :

β : Proportion du délai de paquets dû â la connexion propre de la charge du réseau. Proportion des ressources totales consommées par la connexion ou bande passante disponible.

Si $\beta \approx 1$, toute la variation du délai est dû à la propre charge d'attente de la connexion au réseau. Si $\beta \approx 0$, la charge de la connexion est sans signification comparée avec d'autre trafic au réseau. Plus généralement :

$\Sigma_i (\varphi_i + \phi_i)$: Ressources consommées par la connexion.

$\Sigma_j (\gamma_j + \phi_j)$: Ressources consommées par connecxions compétitives.

Des valeurs $\beta \approx 1$ signifiaient que le goulot d'étranglement est disponible tandis que les valeurs $\beta \approx 0$ signifiaient que rien de bande passante est disponible. Il est possible d'estimer β sans trop charger le chemin réseau.

9.4.10 Technique de Dispersion par Train des Paquets

Cette technique est connue en anglais comme Packet Train Dispersion (PTD) [**?**, **?**]. La dispersion par train de paquets dans un lien spécifique est la distance temporelle entre le dernier bit du premier et dernier paquet du train de paquets. Elle est utilisée dans l'outil "cprobe". Cette technique sert à estimer la bande passante disponible dans un chemin réseau.

Principe

Cette méthode étend la méthode par paires de paquets et utilise des réponses multiples.

Modèle

Si un lien appartenant au chemin n'agrége pas du trafic croisé, le taux de dispersion est égal à la capacité du chemin, le même que pour la méthode par paires de paquets. Néanmoins le trafic agrégé peut rendre le taux de dispersion significativement mineur que la capacité. Le taux de dispersion est :

$$D = \frac{[(N-1)L]}{[\Delta_R(N)]}$$

D'où :

n : Numéro de paquets dans le train.

L : Taille des paquets.

Δ_R(N) : Dispersion de bout en bout du train de paquets.

Phase d'Échantillonnage et d'Estimation

La seule différence avec la technique par paires de paquets consiste à envoyer plus de deux paquets.

Commentaires

- Cette méthode a besoin des mesures dans les deux extrêmes du chemin, c'est à dire avec le logiciel installé/démarré dans la source et le récepteur.
- Cette méthode ne peut pas mesurer d'une manière précise la charge dans chemins à haut débit.
- Il était montré que la dispersion des trains de paquets longs ne mesure pas la bande passante disponible dans le chemin, mais plutôt une autre métrique du débit appelée Taux de Dispersion Asymptotique (Asymptotic Dispersion Rate ou ADR) [**?**].
- En incrémentant la largeur des trains de paquets nous obtiendrons des variances de mesures plus réduites, mais l'estimation converge à une valeur, référée comme l'ADR, lequel est inférieure à la capacité et qui n'est pas lié à la bande passante disponible [**?**].
- Les techniques qui demandant l'accès aux deux extrêmes impactant les mesures de bande passante en faisant très difficile le découplage des caractéristiques dans les deux sens.

– Les techniques basées dans des trains de paquets sont très intrusives puisqu'elles envoient un grand nombre de paquets dans le réseau et possiblement elles enlevaient du trafic courant.
– Les techniques basées dans des trains de paquets puissent surcharger les files d'attente des routeurs si les trains sont trop longs.

L'augmentation de la taille des paquets test réduit la variance des mesures mais les estimations convergeaient à une valeur référencé comme la Taux de Dispersion Asymptotique (Asymptotique Dispersion Rate ou ADR).

Dans un lien à k-cannaux de capacité totale C, les canaux individuels acheminaient les paquets d'une façon parallèle à une vitesse de C/k et la capacité de ce lien peut être mesurée comme la dispersion des paquets dans le train avec N = k+1.

Solutions

– Il est possible d'exécuter des mesures par train de paquets sans l'accès au récepteur, en forçant le récepteur d'envoyer un message d'erreur comme réponse à chaque paquet appartenant au train (par exemple ICMP port-unreachable ou paquets TCP RST). Dans ce cas les capacités des liens et le trafic agrégé de retour peut affecter les résultats.

Pour les techniques basées à train de paquets les observations suivantes ont était présentés dans [**?**] :

1. La mode de la capacité et leurs modes postérieurs sont réduites si le nombre de paquets dans les trains est réduit (elles disparaissent si le nombre de paquets dans les trains continu à augmenter).
2. La meilleure valeur du nombre de paquets dans les trains pour obtenir une mode de capacité significative est de deux.
3. Si le nombre de paquets dans les trains augmente la distribution de la bande passante dévient uni modale.
4. Le rang de la distribution de la bande passante est réduit si N augmente.
5. Si N est suffisamment large est la distribution de la bande passante est uni modale, le centre de cette distribution est indépendant du nombre de paquets dans les trains.

L'outil "cprobe".

L'outil "cprobe" provisionne une estimation de la bande passante disponible dans le chemin réseau [**?**]. La source envoi un flux de paquets ICMP ECHO jusqu'au récepteur en enregistrant le temps entre la réception du premier et la réception du dernier paquet. Il est possible mesurer le trafic croisé dans le lien avec goulot d'étranglement.

Il est possible de mesurer la bande passante disponible, en divisant le nombre d'octets envoyés par le temps d'inter arrivé du premier et dernier paquets. Il y a deux types d'erreurs : La source peut être délayée dû au système d'exploitation en retardant les flux de paquets de retour. Les effets d'ordonnancement au récepteur puissent produire temps d'inter arrivés trop courts. Pour éliminer ces deux erreurs il ne faut pas compter la valeur la plus haute et la plus basse des mesures d'inter arrivé pendant la phase de calcul de la bande passante disponible. Cette procédure améliore la précision des estimations d'une manière très significative. Pour donner de la robustesse à l'outil contre les pertes

de paquets et le changement d'ordre des paquets des calculs avec quatre flux de huit paquets sont utilisés.

9.4.11 Technique "Pathrate"

Cette technique [**?**, **?**] est une combinaison des techniques à paires de paquets et à trains de paquets. Cette technique requiert la participation des deux extrêmes du chemin réseau à mesurer. Elle mesure la capacité d'un chemin réseau.

Bien que d'autres techniques seulement utilisent l'accès à un extrême cette technique est mois flexible mais plus précis. Les autres techniques basées dans l'accès seulement dans l'équipement source utilisaient des réponses à paquets ICMP, UDP-echo ou TCP-FIN.

Principe

Envoyer trains de paquets de taille optimale pour déterminer le taux de dispersion asymptotique à partir des mesures de la distribution de la bande passante, puis déterminer le taux d'utilisation ensuite la capacité.

Modèle

La capacité n'est pas toujours estimée d'une manière correcte par des techniques statistiques qui comprennent la bande passante la plus commun ou le rang. Il faut examiner la distribution de bande passante en termes d'attentes, analyser les causes de chaque mode, et les différences entre la mode de capacité et les modes locales.

Le modèle général du taux de dispersion asymptotique (Asymptotic Dispersion Rate ou ADR) est :

$$R = \frac{(N-1)L}{\overline{\Delta_H}}$$

D'où :

N : Nombre de paquets dans les trains.

L : Taille des paquets.

$\overline{\Delta_H}$: Dispersion moyenne à la sortie du lien H (récepteur).

Pour un chemin avec $C_0 \geqslant C_1 \geqslant \ldots \geqslant C_H$ et si tous les paquets du trafic croisé ajoutés sortaient au récepteur (persistent cross traffic), alors :

$$\overline{\Delta_i} = \frac{\overline{\Delta_{i-1}}(C_{i-1} + r_i)}{C_i}$$

$$R = \frac{C_H}{\prod_{i=1}^{H}(1+\frac{r_i}{C_{i-1}})}$$

D'où :

C_i : Capacité du lien i.

r_i : Taux du trafic croisé jusqu'à lien i (toujours avec trafic croisé persistent).

$$r_i = \sum_{k=1}^{i} r_k = u_i C_i$$

u_i : Utilisation (charge) du lien i.

Si les capacités tout au long du chemin réseau ne diminuent pas l'analyse est plus compliquée. Il faut résoudre récursivement :

$$\Delta_i = \{ \begin{matrix} [\overline{\Delta_{i-1}}(C_{i-1}+r_i)]/C_i ; Si C_{i-1} \geq C_i our_i \geq C_i - C_{i-1} > 0 \\ \overline{\Delta_{i-1}} ; Si r_i < C_i - C_{i-1} \end{matrix}$$

Avec : $\Delta_0 = \frac{L(N-1)}{C_0}$

Si les capacités tout au long du chemin réseau sont toutes de la même valeur :

$$\frac{C}{\prod_{i=1}^{H}(1+u_i)} \leqslant R \leqslant \frac{C}{(1+\max_{i=1,\ldots,H} u_i)}$$

Une taille de paquet supérieur donne une dispersion plus large laquelle est plus facile à mesurer, plus robuste aux délais d'attentes, et moins sensitive à la résolution des temps d'échantillonnage. Une taille de paquet plus grande implique une probabilité d'interférence par le trafic croisé, et le rang de dispersion inférieur à la capacité est plus grand dans la distribution de la bande passante.

Une taille minimale de paquet n'est pas optimale. Si la taille des paquets dans le train est petite, les modes post capacité sont plus significatives et la mode de la capacité est moins significative.

Dispersion minimale mesurable par un récepteur :

Un récepteur peut mesurer la dispersion d'une paire de paquets si elle est supérieure à Δ_m. Ce borné inférieur est déterminé par la latence à recevoir un paquet dans le système d'exploitation (Operating System), déplacer le paquet du noyau au espace utilisateur en utilisant l'appel du système d'exploitation recvform, au temps d'échantillonnage d'arrivée, et aux autres taches de réception dans le logiciel avant l'arrivée du deuxième paquet.

Etant donné Δ_m pour un récepteur spécifique, la capacité maximale disponible que peut être mesurée pour la taille du paquet L est de C = L/Δ_m. Si une estimation de la capacité est connue C la taille du paquet est borné par L > CΔ_m.

Phase 0 : Mesures préliminaires

La source envoie trains de paquets avec incréments graduels pour déterminer si les liens sont multi-canaux. S'ils existent alors un pas de bande passante est réduit quand N augmente de k à k+1 et le lien avec la bande passante la plus étroite consiste en k canaux. Le nombre initial de paquets dans le train est utilisé pour déterminer le nombre maximal de paquets aux trains que le chemin peut transférer sans surcharger les routeurs ou les systèmes d'exploitation de la source/récepteur.

Phase I : Échantillonnage à paires de paquets

En se basant dans les résultats présentés dans [**?**] il est bien meilleur d'observer la mode de la capacité du chemin en utilisant deux paquets que un train de paquets. Par conséquent un grand nombre de paquets sont utilisé pour découvrir toutes les modes locaux d'histogramme de la bande passante en espérant que la mode de la capacité se trouve parmi eux.

En se basant aussi dans les résultats présentés dans [**?**], la taille de paquets utilisée pour les mesures est de 800 octets. Alors cette phase consiste en 2000 expérimentes à paires de paquets. L'utilisateur doit spécifié le "bin" d'histogramme qui est aussi la résolution de la capacité à estimer. La figure 9.24 montre un exemple de modes locales.

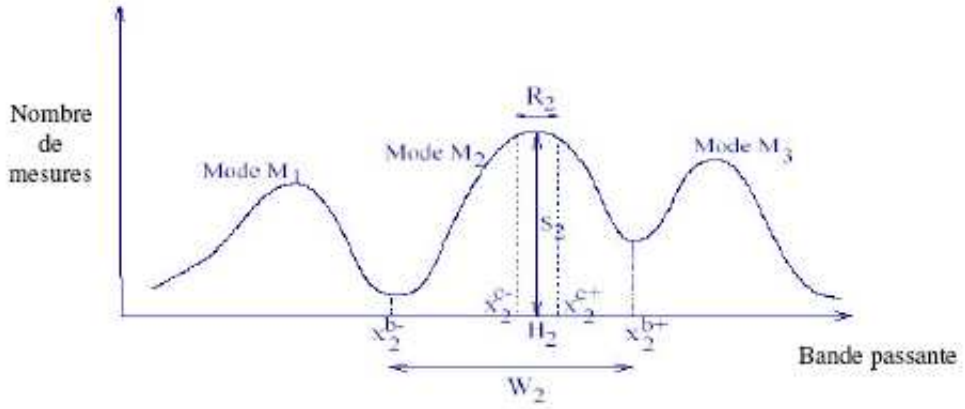


FIG. 9.24 – Caractéristiques des modes locales

Phase II : Échantillonnage à train de paquets et Estimation

Soit $\hat N$ la valeur minimale de N pour laquelle la distribution de la bande passante B(N) est uni modale. Soit $[\zeta^-, \zeta^+]$ le rang de cette unique mode. La règle pour laquelle l'estimateur de la capacité C est sélectionné est que la mode de la capacité soit la mode minimale m_i dans $\boldsymbol{M} = \{m_1, m_2, \ldots, m_M\}$ laquelle est supérieur à ζ^+. $\boldsymbol{M}$ est la séquence de modes locales :

$$\hat{C} = m_k = \min\{m_i \in M : m_i > \zeta^+\}$$

Cette heuristique est basée dans le raisonnement suivant : Quand N est suffisamment grand tel que B(N) dévient uni modale, au moins tous les paquets des trains ont trouvé de la dispersion due au trafic croisé, alors $\zeta^+ < C$. Car N est la longueur minimale du train de paquets que génère une distribution de la bande passante mono modale, le rang de l'unique mode est toujours suffisamment grand pour couvrir les modes locales dans le rang de dispersion de sous-capacité dans B(N) entre R et C.

Cette deuxième phase consiste en 400 expériences de trains de paquets avec une taille de paquet L = 1500 octets pour chaque longueur N. Si B(N) n'est pas uni modale, alors N est doublée et le processus est répété. Quand la longueur $N = \hat N$ est atteinte, le seuil supérieur ζ^+ est mesuré et la capacité m_k déterminée. La figure 9.25 montre les histogrammes des phases I et II.

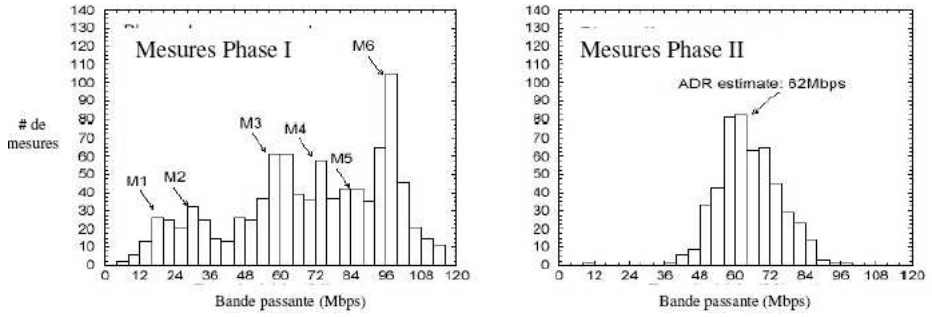


FIG. 9.25 – Histogrammes des phases I et II

Commentaires

Cette technique utilise le protocole UDP pour envoyer les paquets test. Elle aussi établit des connections TCP comme canaux de contrôle acquittent chaque paire ou train de paquets et les utilisé pour échanger de l'information de contrôle entre les deux extrêmes du chemin réseau.

Toutes les paires ou trains de paquets qui trouvent des pertes sont ignorés pour le processus de mesure. Le processus de mesure est arrêté quand les pertes dans le chemin réseau sont significatives en évitant ainsi de la congestion.

L'intervalle du temps entre paires ou trains de paquets est de 500 ms, alors le taux de moyen de transmission est de 240 kbps pour L=1500 octets et N=10. La première version de l'outil "Pathrate" utilisait estampillage au niveau utilisateur, cela produisait des estimations supérieures aux cartes réseaux dans le récepteur, car deux paquets reçus trop proches peuvent être mis en attente dans le noyau et puis délivrés à l'application avec un temps indicatif similaire à la bande passante du noyau de l'utilisateur. Ces estimations sont trop grandes pour produire des erreurs. Si la bande passante de la carte réseau au récepteur est connue les mesures supérieures à C_H peuvent être produites dans l'équipement récepteur et on peut les limiter à C_H.

9.4.12 Technique par Auto-Charge de Rafales Périodiques

Cette technique est connue en anglais comme Self-Loading Periodic Streams (SLoPS)[**?**, **?**]. Elle sert à estimer la bande passante disponible. Elle est utilisée dans l'outil appelé "Pathload".

Principe

Cette technique examine les variations des Délais à Un Sens (One-Way Delay ou OWD) d'une sonde de paquets test. La figure 9.26 montre cette idée.

Notation :

T : Temps de inter transmission [sec]

L : Taille des paquets [bit]

R : Débit de transmission [bit/sec]

D# : Dispersion du paquet # [sec]

A : Bande passante disponible

K : Nombre de paquets dans la rafale

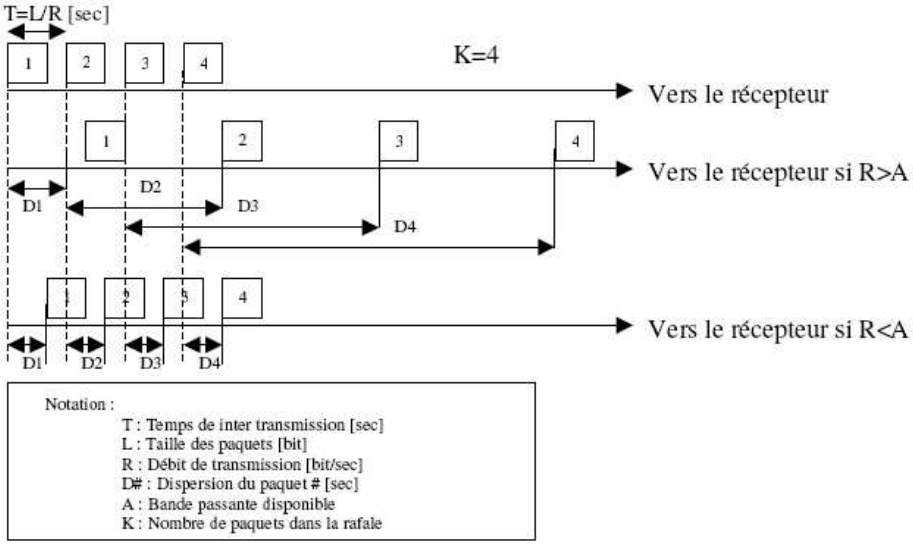


FIG. 9.26 – Exemple de monitoring des variations du OWD de paquets avec K=4

Modèle

$$A_i^\tau(t_0) = \min_{i=1,\ldots,H}\{C_i[1-u_i^\tau(t_0)]\}$$

D'où :

τ : échelle de temps moyenne de la bande passante disponible.

C_i : Capacité du lien i.

u_i^τ : Utilisation du lien i pendant la période (τ).

$A_i^\tau(t_0)$ = Portion de la bande passante sans utilisation dans la route.

Le délai dans un sens pour le paquet k est :

$$D^k = \sum_{i=1}^{H}(\frac{L}{C_i} + \frac{q_i^k}{C_i}) = \sum_{i=1}^{H}(\frac{L}{C_i} + d_i^k)$$

Si le débit de transmission (R) de la sonde de paquets test est plus grand que la bande passante disponible (A) du chemin, le flux de paquets produira une sur charge dans la file d'attente au lien à bande passante disponible étroite. Ceci fera que les OWD des paquets de la sonde s'accumulaient dans la file d'attente de ce lien.

Si le débit de transmission (R) est plus petit que la bande passante disponible (A) du chemin, les paquets voyageront sans problème et le OWD des paquets ne s'incrémentera pas. La figure 9.27 montre les variations des OWD dans les deux situations exprimés ci-dessus.

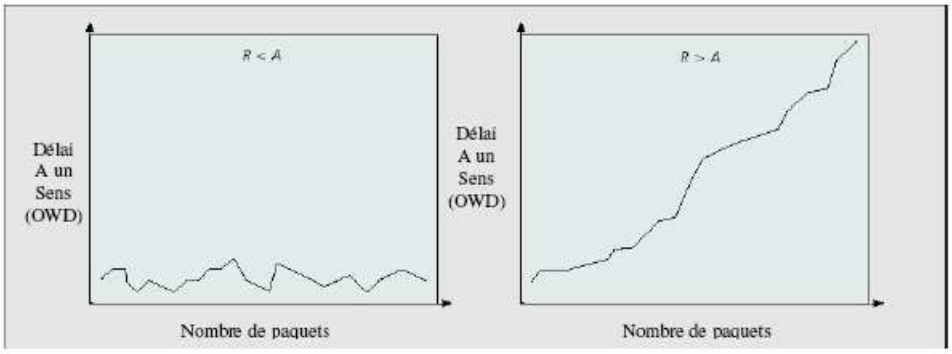


FIG. 9.27 – La variation du OWD avec R<A et R>A

Il est assumé que le chemin réseau est fixé et unique [il n'y a pas des changements ou acheminements à multiple diffusion (multicast) pendant les mesures].

Phase d'Échantillonnage et d'Estimation

- La source envoie d'une manière périodique (période=T) des trains de paquets (nombre de paquets=K≈100) UDP (User Datagram Protocol) de même taille (L) à différents débit de transmission (R=L/T) pour chaque flux.
- Le récepteur mesure le OWD ($D^K = T_{arrive}^{RCV} - T_{send}^{SND}$).
- La variation du OWD est déterminée par $\Delta D^K = D^{K+1} - D^K$ et est indépendante du décalage de l'horloge.
- Le récepteur envoie à la source la tendance du OWD pour chaque train de paquets.

- La source essaie d'approcher le débit de transmission (R) le plus proche à la bande passante disponible en suivant an algorithme itératif/adaptatif similaire à la celui par recherche binaire. Pour un modèle de trafic croisé stationnaire et fluide et files d'attente avec politique PAPS ; Si R>A=minA_i, alors ΔD^K>0 pour k=1,..,K-1, si non, ΔD^K=0 pour k=1,..,K-1.
- La source surveille qu'il exista seulement un train de paquets tout au long du chemin.
- La source créas une période de silence entre deux trains de paquets pour assurer que le taux moyen de transmission est borné à dix percent de la bande passante disponible.
- Si le OWD ne montre pas la tendance d'une manière claire (augmentation ou réduction), il se montre une région grise laquelle est liée à la grandeur de variation de (A) pendant les mesures.

Commentaires

La bande passante disponible devient significativement plus variable aux chemins plus chargés, même sur chemins avec capacité limité (ceci probablement du au bas degré de multiplexage statistique).

Des sondes si longes détériorant le chemin d'une manière significative. En plus, cette technique a besoin d'un temps si long de calcul pour donner les résultats. Cette latence interdit de l'utiliser avec applications en temps réel.

Cette technique n'est pas intrusive, en plus à la place de rapporter une valeur simple pour une bande passante disponible moyenne dans un intervalle (t_0, $t_0 + \theta$), elle estime le rang auquel le processus stochastique de la bande passante disponible $A^\tau(t)$ varie dans (t_0, $t_0 + \theta$), quand elle est mesuré avec une échelle de temps moyenne $\tau < \theta$. Les échelles de temps τ et θ sont liées à deux paramètres : la durée du flux et la durée de flotte.

D'après tests, la bande passante disponible devient plus variable si augmente l'utilisation du lien avec goulot d'étranglement.

La différence majeure de cette technique avec d'autres est l'observation et adaptation des tests par rapport aux variations du délai.

Il y a deux possibilités pour améliorer cette technique : à la place d'analyser les variations des délais d'un flux il faut regarder la présence d'un incrément dans la tendance pendant tout le flux. Nous devons accepter la possibilité que la bande passante disponible peut varier autour du taux R pendant la phase de test.

Cette technique a les applications suivantes :

1. Déterminer le produit bande passante versus délai pour le protocole TCP.
2. Nœuds de réseaux overlay et systèmes terminaux à multiple diffusion.
3. Adaptation du débit de transmission dans applications en temps réel.
4. Vérification de les Accords de Niveau de Service (Service Level Agreements ou SLAs).
5. Control d'admission de bout en bout.
6. Sélection de serveur et toute diffusion (anycasting).

9.4.13 Technique de Trains à Paires de Paquets

Cette technique est connue en anglais comme Train of Packet Pairs (ToPP)[**?**, **?**, **?**, **?**]. Elle sert à estimer ne seulement le goulot d'étranglement surplus et la bande passante du lien mais aussi ceux pour plusieurs liens sous certains conditions. En particulier cette technique était développée pour estimer la bande passante disponible dans une route. Elle donne aussi une estimation de la bande passante du lien le plus congestionné.

Principe

L'idée est similaire à celle de la technique SLoPS avec quelques différences dans les traitements statistiques des mesures (analyse par régression segmentée). Cette méthode incrémente aussi le débit de transmission d'une manière linéaire tandis que SLoPS utilise la recherche binaire pour ajuster le débit de transmission. La figure **??** montre cette idée.

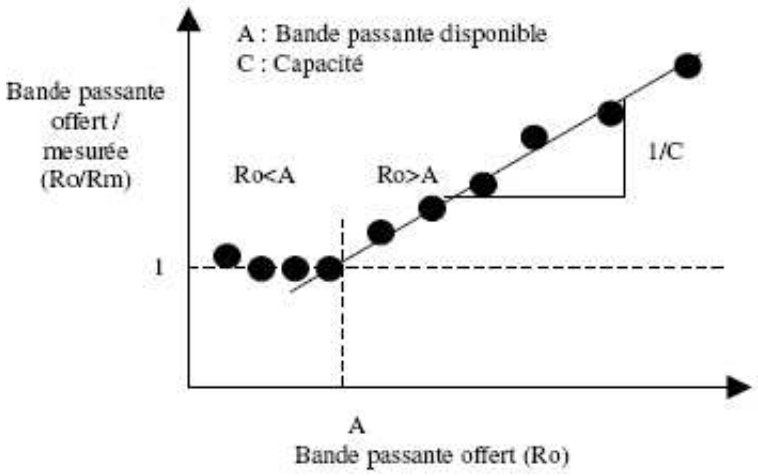


FIG. 9.28 – Bande passante offert/mesurée (Ro/Rm) par rapport à bande passante offert (Ro)

Modèle

En assumant un partage proportionnel des liens aux chemins. Le partage proportionnel de bande passante disponible, $\boldsymbol{p}_n$, au lien i quand la charge offerte de la source $\boldsymbol{p}_0 = 0$ et le trafic croisé au lien i est x_i.

$$p_j = \begin{cases} p_{j-1}, si p_{j-1} < s_j \\ p_{j-1} l_j / (x_j + p_{j-1}), si p_{j-1} \geqslant s_j \end{cases}$$

D'où :

$s_i = l_i - x_i$

Le débit de transmission après le récepteur sera f = $\boldsymbol{p}_n$. Cela signifie que si on augmente le débit de transmission offerte $\boldsymbol{p}$, nous atteindrons le débit de transmission f = $\boldsymbol{p}$ jusqu'à $\boldsymbol{p}$ atteint la bande

passante surplus du goulot d'étranglement, s_b, au lien i. Alors, le débit de transmission au récepteur correspond à la bande passante partagée proportionnelle du lien i. Toujours en augmentant $\boldsymbol{p}$ plus loin de s_b nous allons atteindre la deuxième bande passante surplus la plus inférieure dans la route. En assumant que les liens surplus apparaissent dans j et j+p, alors :

$$p_j = pl_j/(x_j + p), pour p > s_j$$
$$p_{j+p} = \frac{[pl_j/(x_j + p)]}{[x_{j+p} + pl_j/(x_j + p)]} pour p_j > s_{j+p}$$

La bande passante partagée proportionnelle, f, est de : f = $\boldsymbol{p}_{j+p+1}$. Ce modèle qui consiste à observer la bande passante, f, comme une fonction de la bande passante offerte, $\boldsymbol{p}$, sera utilisée pour estimer la bande passante du goulot d'étranglement, la bande passante surplus du goulot d'étranglement et la bande passante partagée proportionnelle de la route.

Nous appelons lien congestif (congestible) i pour une charge particulière $\boldsymbol{p}$ si $\boldsymbol{p}_i > s_i$. Les liens congestifs sont détectés par cette méthode. Pour qu'un lien soit ou non congestif dépend de la bande passante surplus et de tous les liens de retour. Pour une charge particulière $\boldsymbol{p}$ ils existent des liens congestifs. Si cet ensemble est ordonné dans l'ordre détecté en incrémentant la charge. Cet ordre est appelé Le plus petit surplus en premier (Smallest Surplus Firts ou SSF).

Phase d'Échantillonnage

- La source envoie n paires de paquets de la même taille vers le récepteur avec une débit de transmission minimale ($\boldsymbol{p}^{min}$).
- Après avoir envoyé ces paires de paquets, le débit de transmission est augmentée par $\Delta\boldsymbol{p}$ et encore n paires de paquets de la même taille sont envoyés vers le récepteur. Le temps de séparation entre deux trains de paquets test, ΔT^p, est choisit de tal manière que la probabilité d'être mis en attente soit très petite.
- Cette procédure est répété jusqu'à le débit de transmission atteigne la valeur de $\boldsymbol{p}^{max}$ (cela marque la fin de la phase de test). La figure 9.29 montre cette phase de test.
- Les paquets test sont estampillés au récepteur. Une fois tous les paquets sont reçus (ou un temps est excédé), les temps d'estampillage sont envoyés vers la source. Cela implique que toutes les mesures sont faites dans un sens (One Way Measurements).

Phase d'estimation

- Cette phase est basée sur le principe d'espacement temporel au goulot d'étranglement.
- Déterminer la valeur moyenne de chaque ensemble d'estampillages alors nous obtiendrons une valeur ΔR_i pour chaque débit de transmission offerte, $\boldsymbol{p}^i$. Les estimations de bande passante sont calculées par : f = b/ΔR. D'où b : taille des paquets, nl : Niveaux offerts de débit de transmission.
- Les paquets test envoyés à un niveau de débit de transmission offerte $\boldsymbol{p}^i$ peuvent seulement faire sauts avec une bande passante surplus s$\leqslant\boldsymbol{p}^i$ congestionné. Comme la séquence des débit de transmission offertes est incrémentale [$\boldsymbol{p}^1$,..., $\boldsymbol{p}^{nl}$] plusieurs liens congestifs peuvent être détectés en étudiant la séquence [f^1, ..., f^{nl}].

Des techniques traditionnelles de régression linéaire ne peuvent pas être utilisées. En transformant l'équation :

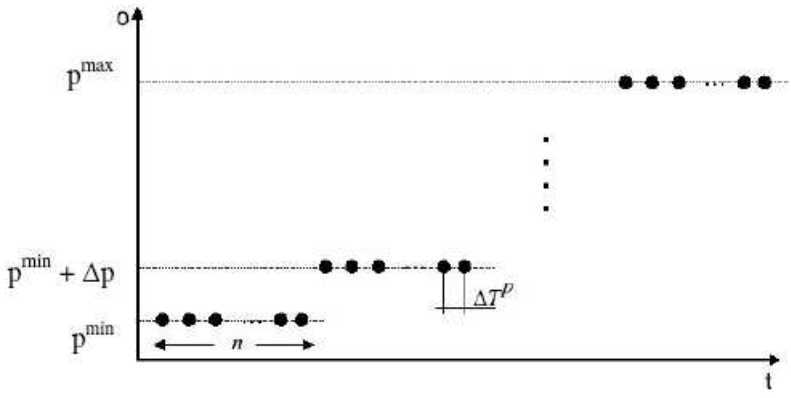


FIG. 9.29 – Phase de test pour la technique de trains à paires de paquets

$$\frac{p}{f} = [1 - \frac{(l-x)}{l}] + \frac{p}{l} = (1 - \frac{s}{l}) + \frac{p}{l} = \alpha + \beta p$$

D'où :

s : bande passante surplus = l – m.

La figure 9.30 montre p/f comme fonction de p. On peut voir que cette technique détecte plusieurs liens potentiellement congestifs d'une manière visuelle.

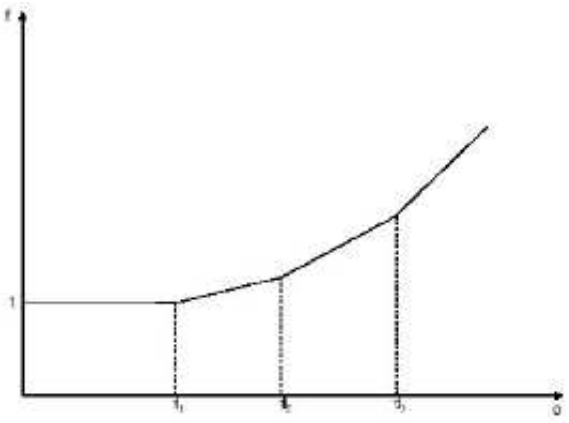


FIG. 9.30 – Graphique de p/f comme fonction de p. Les points de changements de pente correspondent aux points de bande passante surplus

Commentaires

Cette technique est considérée comme technique non-intrusive. Elle n'est pas limitée par le débit de transmission à laquelle on peut injecter des paquets test dans le réseau. Elle est une extension de la technique à paires de paquets pour estimer la bande passante des goulots d'étranglements.

Cette technique résoudre quelques problèmes que les autres techniques ne soient pas capables de résoudre comme : Identification des situations d'où différents types de goulot d'étranglements apparaissent dans liens séparés. Ce problème est appelé à goulot d'étranglement caché (hidden bottleneck problem).

Cette technique est capable de détecter liens les goulots d'étranglements invisibles. Plusieurs scénarios d'où cette méthode est capable de calculer estimations de bande passante pour plusieurs liens congestionnés en se basant dans une seule session de test de bout en bout ont était montrés.

L'outil "cprobe" peut seulement trouver le goulot d'étranglement surplus si elle est de la même valeur que le goulot d'étranglement dans la route. "cprobe" seulement mesure la bande passante surplus proportionnelle dans le débit de transmission extrême à extrême. "cprobe" prédiras une valeur surplus très haute. Ce problème est aussi le problème à goulot d'étranglement caché car le goulot d'étranglement surplus n'est pas visible pour certains techniques.

Cette méthode est aussi capable d'estimer la capacité du lien avec bande passante disponible minimale dans le chemin. Cette capacité peut être supérieure que la capacité du chemin si les liens les plus étroits en capacité et bande passante disponible sont différents.

Même que pour la technique SLoPS. Des sondes si longes détériorant le chemin d'une manière significative. En plus, cette technique a besoin d'un temps si long de calcul pour donner les résultats. Cette latence interdit de l'utiliser avec applications en temps réel.

Cette technique produise des estimations erronées quand le chemin comprend plusieurs points d'attente et ses résultats dépends d'ordre de liens dans le chemin.

Les liens à multiples canaux est un problème potentiel car il existe la possibilité que le deuxième paquet test ne soit pas mis en attente derrière le premier paquet. Dû a ce problème cette technique peut sur-estimer la bande passante disponible. Un possible solution a ce problème est de remplacer les paires par groupes de paquets [**?**].

La bande passante partagée proportionnelle n'est pas une bonne métrique pour la bande passante disponible car n'est pas un débit soutenu.

Améliorations

Parmi les améliorations de cette technique se trouve la formalisation de l'algorithme basé sur la régression linéaire à contraintes et l'automatisation de la procédure d'estimation de bande passante disponible[**?**]. Cette amélioration ne demande pas que les routeurs travaillent avec la politique d'attente juste. Elle peut aussi travailler avec routes composées par plusieurs liens.

Algorithme d'amélioration

Dans cette amélioration, il est assumé que le nombre de segments dans la figure 9.31 est connu et il est de valeur K. Les points de changements sont de deux types : (i) points de changement de pente entre deux débits de transmissions successives offertes, et (ii) points de changement de pente avec la même valeur que le débit de transmission offerte. L'algorithme commence en cherchant les solutions d'où les points de changement de pente appartient à la catégorie (i) après avec la catégorie (ii).

Soit $R^* = \sum_{k=1}^{K} R_k^*$ = Segment avec la plus petite somme-carrée résiduelle sans contrainte. Soit R(τ1, τ2) = La plus petite somme-carrée résiduelle pour la régression linéaire contrainte, en assumant K = 3 (nombre de liens congestionnés) :

set $R_{min} = \infty$

for i = 2 to n_l- 4

Assume $o^i < \tau_1 < o^{i+1}$ (*)

Obtain R_1^*

by linear regression

for j = i + 2 to n_l –2

Assume $o^j < \tau_2 < o^{j+1}$ (**)

Obtain R_2^*

and R_3^*

by linear regressions.

if R* < R_{min} then

Calculate τ_1^*

and τ_2^*

if (*) holds for τ_1^*

and (**) holds for τ_2^*

then

/*$\beta_{est} \equiv \beta$* */

set R_{min} = R*

else

Perform constrained regressions with

(τ_1^* ,τ_2^*) = (o^i, o^j), (o^{i+1}, o^j), (o^i, o^{j+1}), and (o^{i+1}, o^{j+1})

if R(τ_1^* ,τ_2^*

) < R_{min} then

set R_{min} = R*(τ_1^* , τ_2^*)

end /*if*/

end /*if*/

end /*if*/

end /*for*/

end /*for*/

La solution générale, (β_{est1}, β_{est2}, τ_{est1}, τ_{est2}), est la que provisionne R_{min}. Le calcul de τ_k^*, k = 1, 2 s'est fait en utilisant les contraints :

$$\tau_k^* = \frac{\beta_{k0}^* - \beta_{(k+1)0}^*}{-\beta_{k1}^* + \beta_{(k+1)1}^*}$$

Les régressions à contraintes sont faites en utilisant la méthode du multiplier Lagrange. Cette étape a été simplifiée. Il était seulement testé les sommets d'hyper cube, car en réalité l'optimale se trouve dans tout la surface d'hyper cube. Si n_l est trop petit (environ 70), les erreurs introduites sont acceptables. Quand les paramètres β de chaque segment on était estimés les valeurs de l et s pour la goulot d'étranglement surplus sont calculées. Si $K > 3$ il faut ajouter plusieurs boucles dans l'algorithme, un pour chaque point de changement de pente extra. Avant de démarrer l'algorithme, la valeur de K est estimée en comptant le nombre de valeurs maximales dans l'approximation de la deuxième dérivée. Cette procédure n'est pas difficile à automatiser.

Optimalisations de l'algorithme

- La deuxième dérivée peut être utilisée pour réduire les régions de recherche des points de changements de pentes plus étroite.
- En exécutant les régressions dans l'ordre décrit dans l'algorithme, les sommes et produits n'ont pas besoin d'être ré calculés à chaque fois. A sa place, elles besoin des ajustements pour l'addition et la soustraction d'observation simple.
- Il est possible d'argumenter que la plus part des points de changements de pentes appartient à la catégorie (i) même si Δp est trop petit. C'est à dire n'est pas chercher des solutions dans la frontière d'hyper cube même pas aux sommets.

9.4.14 Technique "Pathchirp"

Pathchirp est proposée pour estimer la bande passante disponible[**?**].

Principe

Cette technique est basée sur le principe de auto-congestion induite. Elle est basée aussi aux délais d'attente des paquets test et les délais relatifs entre paquets test.

L'heuristique de base consiste-en : Si le débit des paquets test est supérieur à la bande passante disponible dans la route, alors les paquets test dévient en attente dans un routeur, en résultant un

incrément dans le temps de transfert. Par contre, si le débit des paquets test est inférieur à la bande passanmte disponible, les paquets test ne présentaient pas d'attentes. La bande passante disponible peut être estimée comment le débit que produise la congestion.

Modèle

Cette technique assume d'équipements Store-and-Forward, files d'attente de type PAPS à taux de service constant. La bande passante disponible pendant l'intervalle de temps [t-τ, t] est déterminé par :

$$B[t-\tau,t] = \min_i(C_i - \frac{A_i[t-\tau+p_i, t+p_i]}{\tau})$$

D'où :

C_i : Capacité du nœud i.

A_i[a, b] : Trafic total autre que les paquets test.

p_i : Temps minimal un paquet envoyé par la source atteint le routeur i. Il comprenne les délais de propagation et les temps de services du paquet.

Soit le taux de diffusion entre paquets γ, le délai d'attente du paquet k comme $q_k^{(m)}$, le temps de transmission du paquet k comme $t_k^{(m)}$, l'espace entre les paquets k et k+1 comme $\Delta_k^{(m)}$ et le temps instantané du groupe au paquet k comme :

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

Dans un environnement à trafic croisé de type CBR (Constant Bit Rate) :

$$q_k^{(m)} = 0; si B[t_1^{(m)}, t_N^{(m)}] \geqslant R_k$$
$$q_k^{(m)} > q_{k-1}^{(m)}; Autre$$

Alors, un estimateur simple est : $B[t_1^{(m)}, t_N^{(m)}] = R_{k^*}$, d'où k* premier paquet avec délai incrémental.

Phase d'Échantillonnage

La source envoi m groupes de N paquets de taille P-octets séparés exponentiellement. La figure montre la forme de ce train.

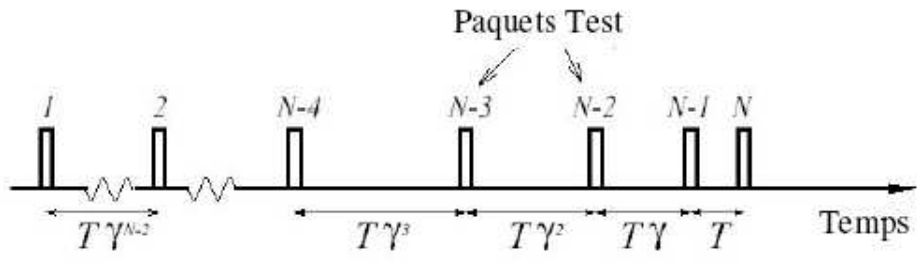


FIG. 9.31 – Train de paquets test avec incrément exponentiel

Phase d'Estimation

Cette technique utilise la signature du délai pour estimer $E_k^{(m)}$ la bande passante disponible par paquet $B[t_k^{(m)}, t_{k+1}^{(m)}]$. Alors elle prenne la moyenne des $E_k^{(m)}s$ de chaque groupe m pour obtenir la bande passante disponible par groupe $B[t_1^{(m)}, t_N^{(m)}]$:

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

Finalement, elle determine les estimateurs ρ[t-τ, t] de la bande passante disponible $B[t_1^{(m)}, t_N^{(m)}]$ en moyenant les estimateurs $D^{(m)}$ obtenus pendant l'interval [t-τ, t].

Pour bien calculer $E_k^{(m)}$, cette technique divise chacune des signatures en régions appartenant aux excursions ou non excursions. La figure 9.32 montre la signature du délai et les régions des excursions.

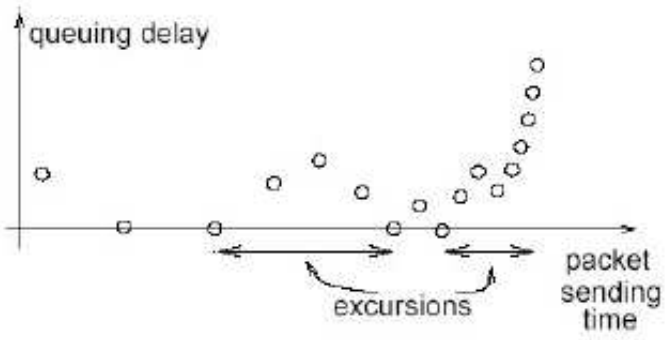


FIG. 9.32 – Signature du délai

Algorithme pathChirp :

Procedure estimate_D(q){

/*q est le vecteur des délais d'attente d'un groupe*/

for (k = 1 to N-1) E_k=0 ; /*initialise*/

i = 1 ; /*Paquet actuel*/

l = N-1 ; /*N=nombre de paquets au groupe*/

while (i $\leqslant$ N-1){

if (q_i $< q_{i+1}$){

j=excursion(q, i, F, L) /*F = Facteur de réduction, L = Seuil de la période occupée*/

choose case(j) ;

Case (a) : (j $>$ i) and (j $\leqslant$ N)

for (s = i to j-1)

if (q_s $< q_{s+1}$) E_s = R_s ;

Case (b) : j = N + 1

for (s = i to N-1) E_s = R_i ;

l = i ;

/*fin du choose*/

if (j = i) j = j + 1 ;

i = j ;

} /*fin d'if*/

else

i = i + 1 ;

} /*fin du while*/

D = 0 ;

For (i = 1 to N – 1){/*calcule de D*/

If (E_i == 0)

D^+ = $R_l \Delta_i$; /*Case (c)*/

else

D^+ = $E_i \Delta_i$;

}/*fin du boucle for*/

$$D = D \Big/ \sum_{1 \leqslant i \leqslant N-1} (\Delta_i);$$

Return D ;

}

Procedure excursion(q, i, F, L){

j = i +1 ;

max_q = 0 ;

while ((j $\leqslant$ N) and (q(j) – q(i) $>$ max_q/F))

{

max_q = maximum (max_q, q(j) – q(i)) ;

j = j + 1 ;

}

if ((j $\leqslant$ N)) return j ;

if (j – i $\geqslant$ L)

return j ;

else

return i ;

}

Commentaires

Un groupe de N paquets a N-1 espaces entre paquets lequel dévient 2N-2 paquets pour techniques à paires de paquets. En incrémentant exponentiellement l'espacement entre paquets, les groupes test sur un rang de débits [G_1, G_2] Mbps en utilisant juste [log(G_2) – log(G_1)] paquets. Les trains de paquets capturent information critique de la correlation du délai. L'algorithme présenté dans [**?**] ai une chose en commun avec cette technique, le groupe de paquets.

Cette technique fait référence au problème pratique du contexte à commutateur. Les paquets sont mis en attente tandis que le CPU (Central Process Unit) travail d'autres processus. Si la différence entre deux estampillages est inférieure à un certain seuil le groupe et éliminé, la valeur de 30 us est fixée comme seuil. Cette technique élimine tous les groupes avec paquets jetés.

Cette technique dépasse les performances des outils comme ToPP [**?**] et Pathload [**?**, **?**] en ce qui concerne la précision de l'estimation et l'efficacité.

9.4.15 Technique à Estimateur de Capacité non-Utilisée par Pair Étalé

Cette technique est connue en anglais avec le nom de SPRUCE (Spread PaiR Unused Capacity Estimate) [**?**], et sert à estimer la bande passante disponible.

Principe

Cette technique examine le taux d'arrivée au lien avec goulot d'étranglement en envoyant des paires de paquets espacés pour assurer que le deuxième paquet test arrive à une file d'attente au lien

avec goulot d'étranglement avant que le premier paquet quitte la file d'attente. Alors, elle calcule le nombre d'octets arrivant la file d'attente entre les deux paquets à partir de l'espacement entre paquets au récepteur.

La bande passante disponible est déterminée par la différence entre la capacité du chemin réseau et le taux d'arrivée des paquets au goulot d'étranglement.

Modèle

Le modèle de base est similaire à la technique IGI [**?**] et Delphi [**?**], il est basé dans la séparation des paquets test. La figure 9.33 montre le modèle à séparation des paquets test. D'après cette figure la bande passante disponible est :

$$A = C * (1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}})$$

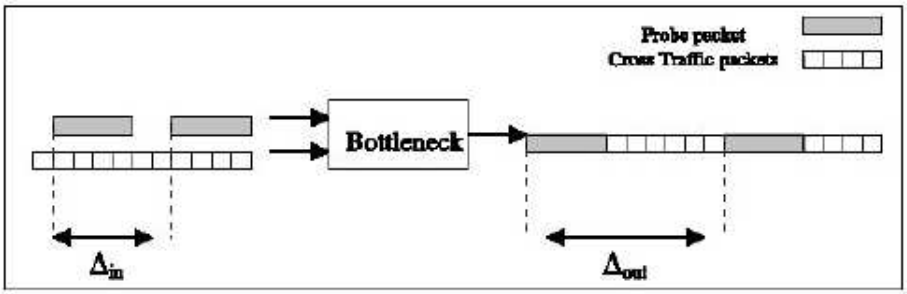


FIG. 9.33 – Modèle à séparation des paquets test

Le modèle assume : routeurs tout au long du chemin réseau avec politiques d'attente de type PAPS, le trafic croisé suit un modèle fluid, les taux moyens de trafic croisé changent doucement et est constant pendant la durée d'une mesure, le goulot d'étranglement est un et les deux, le lien avec la moindre capacité et moindre bande passante disponible.

Le modèle assume que la quantité C est connue, il est introduit Δin à la source et mesure Δout au récepteur.

Phase d'Échantillonnage et d'Estimation

- La source calcule la séparation entre les paires de paquets, Δin, à la valeur du temps de transmission de 1500 octets au goulot d'étranglement. On calcule le goulot d'étranglement avec d'autres techniques.
- La source envoie ensembles de paires de paquets UDP de taille à 1500 octets.
- La source adapte la séparation moyenne entre les paires de paquets pour assurer que le débit des paquets test est le minimum de 240 Kbps et 5% de la capacité du chemin réseau.
- Au récepteur, cette technique mesure Δout, le temps de transmission des deux : le trafic croisé et les paquets test.

– Avec ces valeurs, cette technique calcule le nombre d'octets arrivant à la file d'attente entre les deux paquets.
– En utilisant la formule ci-dessus, la technique estime la bande passante disponible.
– Pour améliorer la précision, cette technique effectue plusieurs mesures et fournisse la valeur moyenne (100 paquets).
– La technique applique le temps entre les deux paires de paquets en suivant une fonction à répartition exponentielle, avec une moyenne τ supérieure à Δin, en resultant un processus d'echantillonnage de type Poisson. Cette décision à était prise pour observer le débit moyen du trafic croisé et pour assurer que la technique reste non-intrusive.

Commentaires

Cette technique utilise un processus de Poisson des paires de paquets à la place de trains de paquets (chirps).

En sélectionnat avec attention la valeur Δin, cette technique assure que le goulot d'étranglement ne seras pas vidé entre deux tests dans une pairen (condition pour le modèle à séparation de paquets).

Cette technique doit être installée aux deux extrêmes car elle consiste en logiciels au niveau de la couche utilisateur aux source et récepteur.

Cette technique n'a pas des paramètres ajustables. Cette technique et les techniques Pathload et IGI ont besoin d'un ordonnancement très soigneux.

Cette technique est plus précise que les techniques Pathload [**?**, **?**] et IGI [**?**]. L'outil "Pathload" génère entre 2.5 et 10 MB de trafic test per mesure alors que le trafic test moyen de l'outil "IGI" est de 130 KB et celui de cette technique est de 300 KB.

9.4.16 Technique "ABdis"

Le nom de cette technique est dérivé de l'anglais (Available Bandwidth DIStribution)[**?**]. La plus part des techniques exprimées auparavant et après mesurent ou estimaient la valeur moyenne de la bande passante disponible et non sa distribution. L'estimation de la distribution de la bande passante disponible peut servir à mieux gérer et contrôler les réseaux, mieux sélectionner un serveur de type proxy et pour le control d'admission de bout en bout.

Cette technique utilise multiples sondes à différent débit avec une technique de couplage paramétrique pour estimer la distribution de la bande passante disponible de bout en bout.

Les mesures des délais dans un sens sont prises en compte pour déterminer la tendance et assigner une valeur de 0, 1 ou 0.5. Une distribution normale est approchée en utilisant la méthode de minimaux carrés pour minimiser l'erreur carrée entre l'assignation à chaque sonde et la distribution normale laquelle est utilisée comme la distribution de la bande passante disponible à estimer.

Principe

Le principe est similaire a celle de la technique SLoPS. Elle est bassée sur les délais dans un sens.

Phase d'Échantillonnage

La source envoie, avec période T, une sonde de K paquets de taille L. Le débit de transmission de la sonde est de R=L/T. Le débit de transmission de chaque sonde augmente avec une relation R(n) = n x R(1) ; d'où n est la numéro de sonde. Le premier débit de transmission est mis en valeur R(1) = B/N. La figure 9.34 montre la règle d'augmentation. Chaque sonde est transmise avec une période de x secondes. La longueur du paquet ne peut pas être inférieure à certain numéro de bits et ne doit pas être supérieure au MTU. La bande passante de la capacité du goulot d'étranglement (B) est mesuré par avance en utilisant des méthodes standard. La source aussi estampille chaque paquet i avec le temps d'envoi Ti.

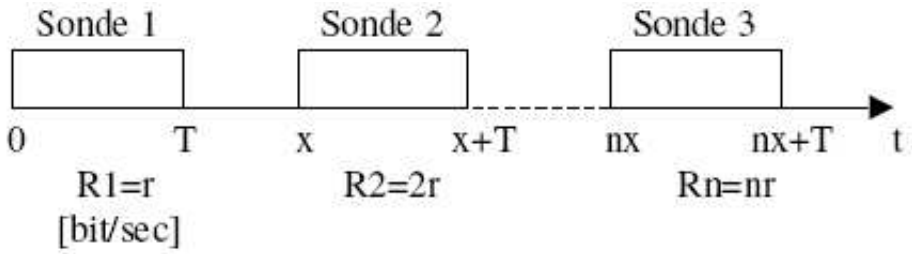


FIG. 9.34 – Règle d'augmentation de débit de transmission

Le récepteur reçoit chaque paquet et calcule le délai relatif dans un sens pour chaque paquet :

$$D_i = a_i - T_i \pm \theta$$

D'où :

a_i : Temps d'arrivée du paquet i

θ : Décalage horloge entre source et récepteur.

Les mesures du délai dans un sens sont partitionnées en m classes des délais consécutifs calculés. La valeur moyenne du délai dans un sens pour chaque classe est aussi calculée. Ils existaient deux test pour montrer s'il existe une tendance d'augmentation dans les délais dans un sens. La première est appelée test de comparaison par paire (Pairwise Comparison Test ou PCT), la deuxième est appelée test de différence par paire (Pairwise Difference Test ou PDT) :

$$S_{PCT} = \frac{[\sum_{k=2}^{m} I(D_k > D_{k-1})}{(m-1)}$$

$$S_{PDT} = \frac{(D_m > D_1)}{\sum_{k=2}^{m} |D_k > D(k-1)|}$$

PCT mesure les fractions de paires consécutives avec délai dans un sens tandis que PDT quantifies si la variation du délai dans un sens est forte dès le début à la fin. Tous les deux s'approchaient à la valeur de un s'il y a une claire tendance d'augmentation du délai dans un sens.

Le récepteur monitor la séquence des délais relatifs dans un sens pour vérifier si le débit de transmission R est supérieur à la bande passante disponible A. La métrique de jugement est appelée H(n). Cette étape est similaire à la technique SLoPS. H(n) a trois possibilités de valeurs soit (0, 1, ou 0.5). Chacune appartenant a trois cas comme suit :

1. Si un parmi PCT ou PDT montre tendance en augmentation tandis que l'autre aussi en augmentation ou ambiguë la valeur de H(n) = 1.
2. Si un parmi PCT et PDT montre tendance en non augmentation tandis que l'autre montre tendance en non augmentation ou ambiguë la valeur de H(n) = 0.
3. Si PCT et PDT montraient ambiguës ou une montre tendance en augmentation et l'autre en non augmentation la valeur de H(n) = 0.5

Phase d'Estimation

Le récepteur estime la distribution de la bande passante disponible en utilisant la méthode de minimaux carrés pour minimiser l'erreur carrée entre H(n) et la distribution normale. C'est à dire que cette technique assume que la distribution de la bande passante disponible suit une loi normale et elle essaie de trouver la distribution normale qui décrit la relation entre R(n) et H(n) aussi précis comme possible. La technique assume aussi que la relation entre R(n) et H(n) puisse exprimer la distribution de la bande passante disponible.

Commentaires

On peut prendre assez d'échantillons des délais dans un sens quand R est trop petit et parce que la taille du paquet L(n) est contrôlée. Si R(n) est trop grand et K est aussi trop grand, la sonde puisse inonder la file d'attente du goulet d'étranglement quand R>A.

La méthode a montré d'estimer la distribution de la bande passante disponible d'une manière précise. Néanmoins, si la charge du réseau est grande, la précision des mesures de la bande passante disponible est détériorée.

Cette méthode était validée par simulation et encore il reste des évaluations à faire : Performance sous trafic et topologies réelles ainsi que les effets dans les protocoles de transport.

9.4.17 Technique de Meditions des Statistiques de la Bande Passante Disponible par Trains de paquets Aléatoires

SMART [**?**] (Statistics Measurement for Available Bandwidth by Random Trains) mesure la bande passante disponible ou la bande passante disponible non-intrusive d'une route (la bande passante disponible d'un lien est égal à sa bande passante disponible non-intrusive). Cette technique peut être clasifiée comme probabiliste.

Principe

Consiste à envoyer des trains de paquets test aux moments aléatoires et calculer le délai minimal de tous les échantillons de ces trains aléatoires. La bande passante disponible est déterminée par analyse statistique et probabiliste. Pour faire cela il a fallu définir la bande passante disponible en termes de probabilité et statistique.

Modèle

Les auteurs proposent différentes définitions pour la bande passante disponible et la distinguent en deux types : bande passante disponible au certain moment du temps et bande passante disponible pendant une période.

Définition 1 :

Un lien est en état disponible quand l'équipement source est en vielle. Donc, la bande passante disponible d'un lien au certain moment du temps (t) est définit comme :

$$a(t) = \{^{C; Si source libre}_{0; Si source occupee}$$

D'où C = capacité du lien. La bande passante disponible (A) pour une période [t_1, t_2] est définit comme :

$$A(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} a(t)dt$$

Théorème 1 :

Un lien est un état disponible seulement si la file d'attente de la source est vide et l'équipement source ne transmettre pas dans ce moment.

Théorème 2 :

Un lien est en état disponible, quand le temps entre la réception et la transmission des paquets dans la file d'attente de l'équipement source est de zéro.

Déduction 2.1 :

Si le théorème 2 est réalisé le lien est un état disponible quant il reçoit un paquet.

Théorème 3 :

Une route à multiple-liens est en état disponible au moment t, seulement si les conditions suivantes sont accomplies.

$$a_1(t) = C_1$$
$$a_2(t + d_1) = C_2$$
$$\ldots$$
$$a_n(t + \sum_{i=1}^{n-1} d_i) = C_n$$

D'où :

d_i : Délai de transmission physique de chaque lien. Il ne comprend pas les délais

d'attente.

La bande passante disponible non-intrusive pour le chemin du lien 1 à n au moment t est :

$$na_n(t) = \{^{C_{\min}(n) si a_k(t+\sum_{m=0}^{k-1} d_m)=1, 1 \leqslant k \leqslant n}_{0 autre cas}$$

La bande passante disponible pour le chemin pendant la période [t_1, t_2] est de :

$$NA_n(t_1, t_2) = [\frac{1}{(t_2 - t_1)}] \int_{t_1}^{t_2} na_n(t) dt$$

Théorème 4 :

Un chemin à multiple-liens est en état disponible, si pour un paquet entrant, le délai (y compris le délai d'attente) entre la transmission et la réception à chaque nœud est de zéro.

Nous appelons bande passante disponible non-intrusive au taux que un chemin peut provisionner à un flux, sans l'influence d'autre trafic dans ce chemin.

Phase d'Échantillonnage et d'Estimation

- La source envoie de manière aléatoire un train de N (200 à 400) petits paquets test de même taille (L = 40 octets).
- L'intervalle entre deux trains de paquets est suffisamment long pour éviter leur influence.
- Chaque paquet est estampillé.
- Le récepteur calculer le délai de transmission.
- Déterminer le délai d'attente de la sonde de paquets pendant la transmission. Cet délai est égal au délai de transmission minus le délai de mini-transmission. Le délai de mini-transmission est le délai de transmission pour la sonde de paquets dans le chemin quand le chemin est en état disponible.

Commentaires

Si nous voulons obtenir la bande passante disponible dans un chemin, nous avons besoin de ne seulement mesurer le délai d'attente de la source jusqu'au récepteur mais aussi le délai d'attente depuis la source jusqu'au chaque point avec goulot d'étranglement dans le chemin réseau.

Pour déterminer la bande passante disponible d'un lien, nous devons localiser d'abord son goulet d'étranglement. Ceci peut être fait avec une algorithme par paires de paquets appliqué avant la mesure de la bande passante disponible. Cette solution correspond à une autre technique d'estimation de bande passante.

Le trafic test est plus petit que celle d'autres techniques. C'est pourquoi cette technique est considérée comme non-intrusive et peut aussi s'appliquer aux systèmes sans fils d'où la bande passante est limitée.

Cette technique résoudre les problèmes tels que la grande latence et le trafic test long. Cette technique n'est pas basée dans la dispersion par paires ou train de paquets.

Les problèmes suivants se posent à manière d'une alerte pendant l'implémentation de cet algorithme.

a) Délai de mini-transmission :

Il est assumé que le délai minimal est le délai de mini-transmission et donc le chemin est fixe. Pour éliminer la variation due aux congestions ils sont proposés deux solutions :

- Une augmentation de la taille des échantillons procure une probabilité supérieure de trouver le délai de mini-transmission.
- Un enregistrement du numéro de paquets des trains reçus par le récepteur.

b) Taille du paquet :

La taille optimale de paquet n'est pas l'Unité de Transmission Maximale (en anglais Maximum Transmission Unit ou MTU) car une taille supérieure puisse implique une possibilité d'interférence du trafic croisé. La taille est un peut mineure à celle des paquets ICMP (40 bytes) qui est de 32 bytes (taille du paquet test UDP).

c) Intervalle entre les trains de paquets :

La valeur entre 10∼20 fois T/L a était choisit. T est la valeur de précision espérée et L est la taille des paquets. Un intervalle plus longe crée une durée de calcul plus longue.

d) Temps d'enregistrement et synchronisation :

SMART envoi des paquets de preuve ICMP type message d'estampillage (13/14) pour obtenir l'estampillage. Il adopte aussi la technologie Source IP Address Spoofing (qui signifie placer l'adresse source et destination). Quand les deux routeurs reçoivent les paquets ils répondaient des paquets d'écho avec l'estampillage local du récepteur et quand ces paquets arrivaient au récepteur lui peut obtenir tous les estampillages et le délai de transmission peut être obtenu.

Il existe une autre méthode pour l'enregistrement du temps et qui fait utilisation de l'option d'estampillage à l'Internet dans les paquets IP (Type 68, Drapeau 3) pour obtenir l'estampillage des deux extrêmes du lien avec goulet d'étranglement.

9.4.18 Technique "Delphi"

La technique Delphi[**?**] sert à estimer le volume instantané du trafic croisé dans un chemin de bout en bout ou la dynamique de la bande passante disponible pendant une période T déterminée.

Principe

Cette technique utilise le délai d'attente experimenté par les paquets test pour estimer, à plusieurs échelles, la charge induite par le trafic croisé au goulot d'étranglement d'un chemin réseau. Les paquets test doivent être très proches.

Le procedure est basé aux inférences, l'algorithme est basé à la source, ne réquiert pas coopération du réseau et seulement peu de réaction de la part du récepteur.

Modèle

Le modèle de cette technique combine un modèle simplifié de bout en bout d'un chemin réseau et un modèle statistique du flux de trafic croisé. Tout le chemin réseau peut être modélisé comme une file d'attente. La figure 9.35 montre le modèle du chemin simplifié et la dynamique de la bande passante disponible avec le trafic croisé.

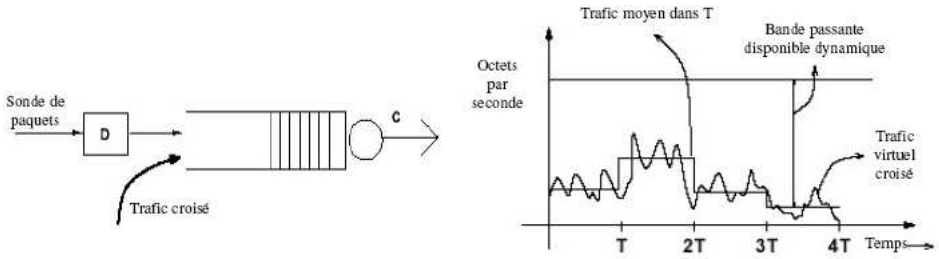


FIG. 9.35 – Modèle du chemin simplifié et dynamique de bande passante disponible

Le modèle utilisé pour le trafic croisé est le modèle multifractal à ondelettes (Multifractal Wavelet Model ou MWM). Il répresente la charge du trafic croisé à plusieurs échelles d'agrégation d'une arborecence binaire. Le modèle MWM est très proche du système à ondelettes nommées "Haar". Le modèle MWM est un modèle à paramètres pour trafic à rafales non-Gaussien. La figure 9.36 montre le format du train de paquets test et le modèle multifractale à ondelletes.

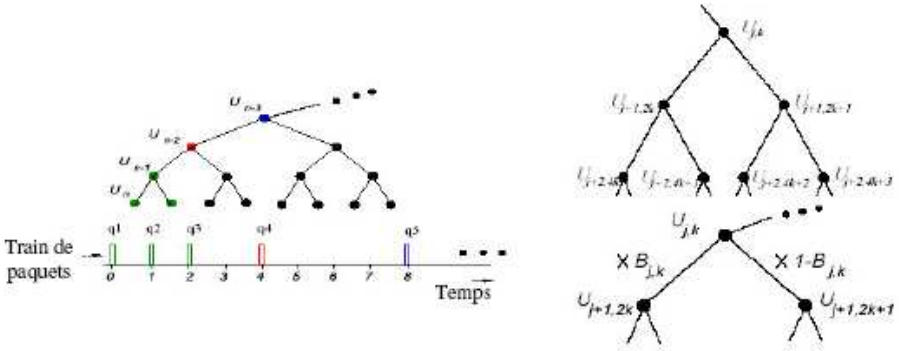


FIG. 9.36 – Format du train de paquets et le modèle multifractale à ondelletes

Cette technique assume que la plus part d'attentes des paquets test ont lieu au goulot d'étranglement.

Cet algorithme assume que les délais excedant les délais de propagation fixes tout au long du chemin ont lieu seulement au goulot d'étranglement, les files d'attente aux routeurs sont de type PAPS.

Le taux de service (C) est égal à la vitesse du lien minimal ou taux de service d'attente minimal tout au long du chemin (file d'attente avec le goulet d'étranglement).

$$D_T = \min_{i \in Q}\{\frac{[C_i T - B_T(i)]}{T}\}$$

D'où :

D_T : Quantité moyenne de trafic croisé par seconde que puisse être inséré pour la source dans un intervalle de temps T ou dynamique de la bande passante dans un intervalle de temps T.

C_i : Bande passante du lien i.

T : Intervalle de temps en secondes.

B_T(i) : Nombre d'octets de trafic croisé qui arrivent dans un intervalle de temps T au lien i.

Soient deux paquets test de taille P aux temps t_0 et t_1 = t_0 + T et leurs différences de temps entre les temps d'arrivée au récepteur a_0 et a_1 respectivement. Alors le temps d'inter départ de la file d'attente au goulot d'étranglement est proportionel au nombre d'octets dans la file d'attente pendant la période T :

$$B_T = C(a_1 - a_0) - P$$

Pour paquets de taille P, $T_{NEQ} = P/C$est utilisé pour assurer que la file d'attente ne sera vidée entre les arrivées des paquets test.

Cette technique utilise des paquets test suffisamment séparés pour améliorer la précision d'estimations du trafic croisé en permettant la connaissance statistique de la dynamique du réseau fournit par un modèle de trafic souple, le modèle multifractal à ondelettes (Multifractal Wavelet Model ou MWM).

Phases d'Échantillonnage et d'Estimation

La source envoie des groupes de trains de paquets test bien proches au début et hautement disséminés à la fin. Les primers trois trains sont séparés par un temps $T_n = T_{NEQ}$, les espaces entre les trains subséquents est incrémenté par un facteur de deux à chaque temps.

Le délai d'inter arrivé dans le récepteur est utilisé pour estimer la charge de trafic croisé dans une gamme large d'échelles. L'estimation se fait avec techniques d'inférence Bayésiennes.

Pour estimer la charge du trafic croisé à plusieurs échelles le chemin réseau est testé avec plusieurs groupes de trains de paquets.

Pseudo code :

Procedure main (q, To, C, n) {

u_o = determine_traffic (q, To, C, n)

return u_o

}

Procedure determine_traffic (q, T, C, k) {

If (T<2T_{NEQ})

u = q_k –q_{k-1} + TC

return u

else

m = determine_traffic (q, T/2, C, k-1)

u = infer (q, T, C, m, k)

return u

end

}

Procedure infer (q, T, C, m, k)

u_min = m si q_k = 0 ; m+q_k+C-max(q_{k-1}+C-(T-1)C,0) si q_k>0

u_max = m+q_k-q_{k-1}-C+TC

return û ∈ [u_min, u_max] that minimizes p(q_k, m | u)

}

D'où :

q : Vecteur des mesures de la file d'attente.

To : Intervalle de temps d'intérêt.

C : Taux de service du modèle de la file d'attente.

n : Numéro de paquets envoyés dans la sonde.

u_{max} et u_{min} : La gamme des valeurs possibles de trafic. Elles sont calculées-en

assumant une file d'attente discrète avec service FIFO en prenant en compte l'effet dans la taille de la file d'attente de la sonde antérieur

Le délai de la file d'attente est calculé comme :

Délai de la file d'attente = Temps de réception – Temps de transmission – Délai constant

Commentaires

L'efficacité du modèle MWM et l'algorithme Delphi permet de les appliquer pour estimations en temps réel.

Cette technique n'a pas besoin en avance des statistiques du trafic. Elle est basée sur un modèle déterministe de la bande passante disponible, elle utilise seulement des mesures de bout en bout en évitant la collection de données dans tout l'Internet.

Cette technique est adaptative car elle poursuit les changements des conditions du réseau. Elle estime d'une manière précise le trafic croisé pour liens à haut niveau d'utilisation pendant qu'elle sur estime le trafic agrégé pour liens à bas niveau d'utilisation.

Cette technique est du type basé dans la source et ne requiert pas la collaboration du réseau mais seulement la réponse du récepteur quand il reçoit les paquets.

Cette technique n'est peut être utiliser quand les liens avec la bande passante disponible étroite et la capacité minimale sont différentes car cela produise une interprétation des délais de file d'attente quelque part dans le chemin aussi comme dans le lien avec la bande passante la plus étroite. En plus cette technique n'est peut pas être utilisée quand il y a des délais d'attente significatifs.

La technique doit être modifiée pour prendre en compte les paquets perdus de la sonde.

9.4.19 Technique à Estimation de Bande Passante Disponible Spatio-Témporel

Cette technique est connue en anglais comme Spatio-Temporal Available Bandwidth ou STAB[**?**, **?**, **?**]. Elle sert à estimer la bande passante disponible en espace et en temps. Elle est implementée dans l'outil appelé STAB.

Principe

Cette technique combine les principes de différentes techniques exposées auparavant : le principe de charge auto-induite, le principe à paires de paquets court-et-long et le principe de groupes de paquets en une nouvelle façon. D'après cette considération on peut dire que cette technique échoue dans un autre type de classification : technique combinée.

Le principe de charge auto-induite se base dans l'heuristique suivante : Si le débit des paquets test R est supérieur à la bande passante disponible alors les paquets test sont mis en attente dans un routeur. Par contre, si R $<$ A, les paquets ne sont pas affectés par un délai extra. Alors la bande passante disponible peut être estimée dans le point de changement entre ces deux conditions.

Le principe à paires de paquets court-et-long consiste en paquets longs insérés avec paquets courts. Les paquets longs sont jétés au milieu de la route car ils sont limités par le TTL mais les paquets courts emportent information importante du temps jusqu'au récepteur.

Le principe à groupes de paquets consiste en que les temps d'inter arrivé des paquets sont réduits en suivant une loi exponentielle. Alors les groupes balayent un grand rang de débits avec peu des paquets.

Cette technique utilise groups de paires de paquets courts-et-longs pour localiser le lien avec moindre bande passante disponible. Dès groupes de paires de paquets avec TTL de paquets longs mis en valuer l fournissent estimateurs de $A(1,l)$. En variant la valeur TTL des paquets longs entre différents groups de paires de paquets il est possible d'obtenir un estimateur de $A(1,l)$ pour l = 1, 2, ...,N.

$A(1,l)$ diminue par rapport à la valeur de l jusqu'au lien avec avec moindre bande passante disponible, puis reste constant.

Modèle

Nous définissons le sub chemin de bande passante disponible jusqu'au lien m comme la bande passante disponible minimale avant m. Cette fonction ne dépend pas de m. Elle reste constante entre

deux liens à moindre capacité disponible. Le dernier lien avec moindre capacité disponible est le lien avec moindre bande passante disponible du chemin réseau.

En rémplaçant les temps d'échantillonnage dans l'algorithme "Pathchirp" nous obtenons le sub chemin de la bande passante disponible jusqu'au lien m. Il n'est pas possible d'obtenir les temps d'échantillonnage des arrivées de paquets au lien m pour m arbitraire en pratique, heureusement il est possible de les estimer avec les temps d'éstampillage des paquets courts.

Les groupes des paquets courts (après m) ont un débit trop bas pour ajouter de la congéstion et délais d'attentes après le lien m. Alors les paquets courts vont au récepteur avec leur inter espace de temp du lien m plus et moins sans changement.

Phase d'Échantillonnage et d'Estimation

- Fixer le nombre de liens tout au long du chemin en incrémentant la valeur de TTL des groupes succesifs en commençant par un (1). La valeur minimale de TTL de tous les paquets qui atteint le récepteur nous donne le nombre de liens.
- Envoyer des groupes de paires de paquets en variant la valeur de TTL pour estimer le sub chemin de bande passante disponible.
- Determiner la probabilité que le lien m soit le lien avec moindre capacité par rapport au temps pour lequel le sub chemin de bande passante disponible jusqu'au lien m-1 soit plus grand que jusqu'au lien m par un facteur multiplicatif de α. Le dernier lien avec la plus haute probabilité d'être la lien avec capacité minimale est le lien avec la moidre bande passante disponible du chemin (la valeur de α=1.2 a était choisit).

La figure 9.37 montre les concepts de groupes de paquets et celui de cette technique.

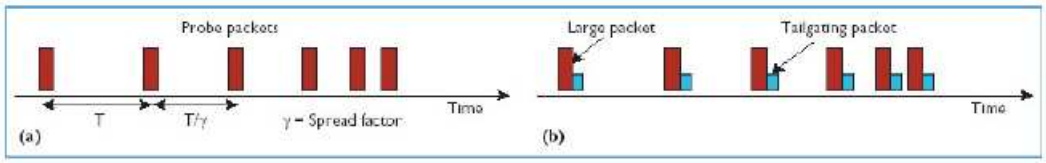


FIG. 9.37 – Groupes de paquets à temps entre paquets exponentiellement réduit (a) et paire de paquets courts-et-longs à temps entre paire de paquets exponentiellement réduit

9.4.20 Technique à Idéntification des Goulots d'Étranglements

Cette technique (Bottleneck Finder ou "BFind") [**?**] sert à identifier et estimer les goulots d'étranglements.

Principe

Cette technique produit de la congestion en envoyant continuement du trafic de paquets de type UDP ; Ensuite elle détermine le lien, appartenant à une route, avec la plus petite bande passante

disponible à partir des mesures du temps aller-retour prises par l'outil "traceroute". Comme l'outil "traceroute" n'est pas un outil d'estimation de bande passante on ne peut pas considerer cette technique dans la catégorie de techniques combinées.

Modèle

Un goulot d'étranglement est un lien d'un chemin réseau d'où la bande passante disponible d'un flux TCP est la minimale. Un goulot d'étranglement n'implique pas que le lien est hautement utilisé ou congestioné.

Phase d'Échantillonnage et d'Estimation

1. La source détermine le délai de propagation de chaque lien jusqu'au récepteur. Pour chaque lien tout au long du chemin, le délai minimal mésuré dans le lien est utilisé comme estimateur du délai de propagation du lien (si la différence entre deux délais consécutifs est negative le délai au lien est mis en zéro). La valeur minimale est prise sur cinq mesures en utilisant l'outil "traceroute".
2. La source envoie paquets UDP vers le récepteur à un débit inférieur à 2Mbps. Une procedure de trace se démarre en même temps, la procedure de trace répète des traceroutes vers le récepteur. Les délais obtenus lien par lien pour chaque valeur de traceroute sont combinés avec les premières mesures pour obtenir estimations des longueurs des files d'attente. Le processus idéntifi la file d'attente que potentiellement augmente sa taille si pendant trois mesures consécutives le délai d'attente dans la file est au moins 5ms et 20% le délai d'attente de son lien. Cette information est accesible à la procedure UDP. Le procedure UDP utilise cette information pour ajuster son taux de transmission.
3. Si aucune file d'attente ne montre pas augmentation de leur taille, la procedure UDP augmente son taux de transmission en 200Kbps (l'augmentation se fait à chaque mesures de traceroute). Alors, cette technique émule le comportement incrémental de TCP, mais plus agréssif, en testant pour trouver la bande passante disponible.
4. Au contraire, si au moins une file d'attente montre augmentation de leur taille, la technique marque la file d'attente comme goulot d'étranglement potentiel et continu avec la procedure traceroute et observer les tailles des files d'attente. En plus la procedure UDP mantient le taux de transmission constant jusqu'au un parmi les événements suivants arrivent :
   (a) Le lien continu à être marqué par "BFind" sur mesures consécutives par le processus de trace et un nombre de 15 observations sont faites par le lien.
   (b) Le lien est marqué un nombre de 50 fois au total.
   (c) "BFind" a demarré pendant un temps maximal de 180 secondes.
   (d) Le processus de trace rapport qu'il n'y a aucune file d'attente en augmentation aux liens en impliquant que les augmentations des files d'attente ont était transitoires.

Dans les deux permiers cas, "BFind" quitte et identifie le lien responsable comme le goulot d'étranglement. Dans le troisième cas, "BFind" quitte sans donner une conclusion confiable sur les goulots d'étranglements tout au long du chemin réseau. Dans le quatrième cas, "BFind" continue à incrémenter son taux de transmission en cherchant le goulot d'étranglement.

1. Si le processus de trace observe que les files d'attente sont entre les premiers trois liens depuis la source, elle sorte inmediament pour eviter surcharger le réseau local. Le taux de transmission est limité à 50 Mbps pour n'utiliser pas la capacité des réseaux locaux. Donc, "BFind" trouve goulots d'étranglements avec $<$50Mbps de capacité disponible.

Commentaires

Cette technique ne trouve seulement le goulot d'étranglement mais elle estime la capacité disponible au goulot d'étranglement juste avant de quitter le programme. Pour chemins d'où les goulots d'étranglements ne sont pas identifies "BFind" fournisse une borne inférieure de la capacité disponible. Cette technique est similaire mais non identique au protocole TCP Vegas.

Cette technique a la disavantage d'être très lourde puisqu'elle envoie beaucoup de données. Elle n'est pas souhaitable pour le monitoring continu de la bande passante disponible mais pour des mesures rapides à courte durée.

Cette technique introduit pertes au goulot d'étranglement, il est possible que d'autre trafic contrôlé par congestion réagisse et rélentisse. Elle peut rapporter quelquechose entre le taux de TCP à poids juste dans le chemin réseau et la capacité du chemin réseau.

9.4.21 Technique NCS - Pipechar

Cette technique[**?**] sert à estimer la bande passante disponible pour chaque lien d'une route pendant une tranche de temps fixe et à mantenir l'information acquise pour son futur utilisation et accès rapide. Elle demarre dans chaque réseau comme un démon et les démons de différents réseaux peuvent coopérer pour faire des mesures.

Cette technique est installée dans l'outil coopératif appellé Network Characterization Service ou NCS ainsi que dans l'outil appellé "Pipechar".

Principe

Il consite en envoyer deux paquets de taille différente et déterminer la bande passante disponible en utilisant des différences de temps d'inter arrivé. Cette technique est basée et réquiert accès seulement à la source.

Modèle

Le modèle pour cette technique est montré dans la figure 9.38

Phase d'Échantillonnage

Il y a deux méthodes possibles : Trains de paquets synchrones et trains de paquets asynchrones. Dans le cas de paquets asynchrones, des trains de différente longueur sont envoyés en temps différents. Chaque train contient seulement une voiture et leurs trains voyagent sur le réseau à temps

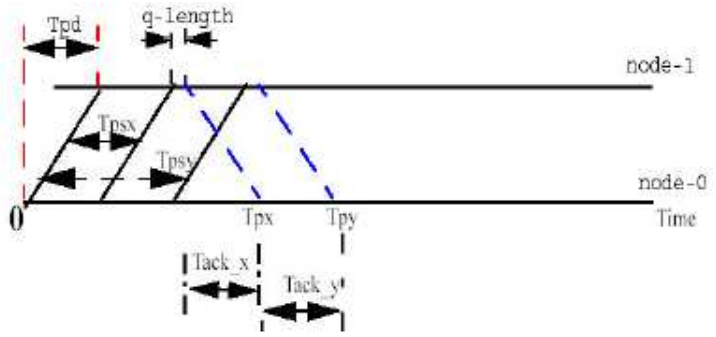


FIG. 9.38 – Ligne de temps montrant la transmission de paquets X et Y et leurs acquittements

différents. Donc, ils experiencent différentes conditions réseau alors nous ne pouvons pas calculer le difference de temps entre trains. À sa place, les temps pour demarrer les paquets de la même taille sont calculés et les résultats pour trains individuels sont procéssés à l'aide d'un algorithme convergent.

L'envoi de trains continu jusqu'à la convergence des équations différentielles est atteignit. La vitesse de convergence dépend de l'état du réseau et la longeur du train. Si l'état du réseau est bon la convergence est atteignit et le nombre de trains nécessaires est réduit. S'ils sont utilisés des longs trains la convergence arrive plus vite car les longs trains tendent à lisser les variations du réseau. Ces trains longs sont bons pour ajuster les tailles des files d'attente pour TCP car ce protocole besoin des tailles d'attente stables. Les trains courts sont utilisés pour l'information du contrôle de congestion mais ils démandaient une convergence plus longue.

La méthode synchrone énvoi un train de multiples paquets vers un nœud pour analyser le délai d'attente, et mesurer la bande passante disponible pendant un temps déterminé. Les paquets sont mis en attente dans la machine de test, et envoyer vers le réseau.

La méthode syncrhone limite le nombre de paquets dans les trains. Si l'êntete du train arrive à la source avant que le dernier paquet du train parte de la source il aura une erreur de mesure introduit par la charge extra. Donc, la taille du train dépend au RTT et à la vitesse du lien.

$$\max(cars) < \frac{RTT}{T_{car}}$$

D'où :

T_{car} : Temps de propagation d'un paquet sur le lien le plus lent du chemin réseau.

La méthode synchrone ne sera capable de tester les pasarelles d'une manière correcte si le RTT est court. La taille du train synchrone est déterminée par demarrant un train asynchrone avant d'un train synchrone. La longueur du train synchrone est alors varié de lien à lien en se basant dans le RTT du lien récepteur et la vitesse du chemin réseau.

L'approche synchrone analyse la variation du temps entre chaque paquet. Elle mesure aussi le délai d'attente produit par le trafic croisé de chaque paquet, lequel est utilisé pour estimer la bande passante disponible.

Phase d'Estimation

Cette technique propose trois algorithmes d'estimation de bande passante disponible.

Paquet Simple avec Cacul à Taille Différentielle au même Nœud (Single Packet with Size Differential calculus ou SPSD)

Le récepteur enregistre le temps entre la première et dernière unité binaire d'un paquet (T_{ps}), la bande passante du lien peut être calculée en divisant la taille du paquet par T_{ps}. Comme il est très difficile à déterminer T_{ps}, deux paquets sont envoyés avec taille différente vers un nœud distant, la différence de taille est utilisée pour calculer T_{Δ} (différence des temps de passage entre tailles différentes de paquets) :

$$T_{\Delta} = T_{psy} - T_{psx}$$

$$BW = (S_Y - S_X) \div T_{\Delta}$$

La différence temporelle entre le paquet le plus long et le paquet le plus court que peut être transmisse depuis un équipement source vers un routeur du cœur est imprécise quand Δ_{RTT} a une magnitude similaire à $T_{\Delta(zerotrafic)}$ car T_{Δ} domine. Cet algorithme est seulement bon pour tester réseaux avec capacité jusqu'à 100 Mbps.

Multiples Paquets avec Calcul à Taille Différentielle au même nœud (Multiple Packets with Size Differential calculus ou MPSD)

Cet algorithme utilise multiples trains de paquets pour mesurer un chemin réseau lien par lien. L'algorithme à taille différentielle est utilisé pour determiner T_{Δ} dedans un train de paquets et T_{Δ} entre train de paquets.

Les goulots d'étranglement dynamiques sont produits par l'intensité du trafic croisé. Ils peuvent être détectés par les délais d'attente. La distribution des délais d'attente peut être utilisée pour calculer la bande passante disponible dans un nœud particulier. La bande passante disponible peut être aussi determinée en utilisant la régression linéaire pour trouver l'endroit d'où les délais d'attente mesurés convergent et la bande passante physique maximale peut être estimée en eliminant les délais d'attente avec calcul différentiel.

Paquet Simple avec même taille en calcul à Différence de Liens (Single Packet with the same size on hop differential calculus ou SPHD)

Cet algorithme est utilisé dans les outils "pathchar" et "pchar". La bande passante d'un lien entre deux nœuds est determinée avec :

$$BW = \frac{S_p}{T2 - T1 - T_{pd1-2}}$$

D'où :

S_p : Taille de paquets.

T1 : Temps de transmission du paquet P de la source au nœud 1.

T2 : Temps de transmission du paquet P de la source au nœud 2.

T_{pd1-2} : Délai de propagation entre les nœuds 1 et 2.

Malheuresement, certains routeurs fournissent temps de réponses ICMP différentes et les mesures basées à liens différentiels ne sont pas précises. Ce pour cela que cette technique n'utilise pas cet algorithme.

L'outil "Pipechar"

"Pipechar" est une application que fournisse l'accèss aux capacités à travers une ligne de commande disponible seulement dans le mode basé à la source de NCS. "Pipechar" offre la possibilité d'analyser un chemin réseau depuis n'importe quel nœud.

Cet outil contient fonctionnalités pour tests à un temps ; il n'est pas dessiner pour collecter l'information sur différentes tranches de temps ou aider à determiner les variations de la bande passante disponible. Il reports la bande passante disponible mesurée pendant la tranche de temps specifié dans la ligne de commande. Vu que cet outil seulement mesure les valeurs inférieures à la bande passante maximale de la couche de transmission, donc il aussi utilise des heuristiques pour deviner l'infraestructure de la couche inferieure et la bande passante réelle.

Commentaires

Les algorithmes SPSD et MPSD peuvent être utilisés pour tester le nœud d'un chemin réseau déterminé. Pour mesurer réseaux avec vitesses de lien supérieures à 100 Mbps, des paquets multiples avec algorithmes à taille différentielle sont utilisés.

Le modèle de mesure de cette technique assume que les paquets sont envoyés en même temps mais cela n'est pas possible physiquement en utilisant une seule interface réseau

NCS est étendu comme un démon (NCSD) pour mesurer d'une manière continue le cœur du réseau. Le module cœur de NCSD est responsable de detecter, acquerir, analyser et cacher l'information sur les liens du chemin réseau à analyser. Un test est reprogrammé à hautes fréquences sur les liens avec hautes utilisations pour trouver les changements des goulots d'étranglements. Un autre test est reprogrammé à utilisations basses pour trouver les chagements globales. À fin de monitorer les changements globaux, les tests reprogrammés traversent le chemin complet nœud par nœud dans un procès de confirmation. Les deux tests reprogrammés intruduissent trafic avec impact minimal sur les réseaux observés.

Si le service est demarré en mode mutuel, il partage information entre les paires NCS et utilise paquets pour échanger information et completer le procès de vérification du goulot d'étranglement. Au mode mutuel, le procès de monitoring est un mode passifet utilise la technique à paires de paquets basés seulement au récepteur dans un serveur lointin. Le mode actif de tests reprogrammés est mis en opération si le serveur lointin envoi information anormale sur le trafic à paires de paquets.

Si NCS fonctionne sur une passarelle adaptative, le procès de test utilisera le mecanisme de type ferroutage que réduit les tests actifs à zéro. Cela est fait en insertant deux petits paquets UDP dans une file d'attente de contrôle de congestion pour une destination spécifique, un paquet dans l'entête d'une file d'attente et l'autre dans la queue. Les deux paquets sont envoyés vers le même récepteur alors deux paquets ICMP seront retournés avec le temps de séparation pour calculer la bande passante disponible.

9.4.22 Techniques à Réaction de Contrôle Adaptatif - Réaction de Convergence Asymptotique et Effet d'Écoulement Fluide

Les algorithmes FAC^2 (Feedback Adaptive Control and Feedback Asymptotic Convergence) et FSE (Fluid Spray Effect ou FSE)[**?**, **?**, **?**] servent à mesurer la bande passante disponible lien par lien (outil NCS) ou de bout en bout (outil Netest).

Principe

Cet algorithme est basé sur la dispersion de paquets. Il est utilisé avec la technique à train de paquets pour construire d'algorithmes pour mesurer la bande passante du réseau et non pour l'estimer. Alors cette technique n'a pas de phase d'estimation.

L'Internet a toujours du trafic, lequel produira l'effet d'écoulement fluide que consiste en si plusieurs flux de trafic arrivaient de plusieurs interfaces et sont acheminés vers une autre interface alors tous les paquets sont ensemblés, et donc la théorie de la dispersion de paquets ne marche pas toujours.

Théoreme FSE :

Assume deux trains de paquets, tous les deux ont un débit de transmission inférieure à la vitesse du lien de sortie d'un routeur, se groupant au routeur. Si le débit de transmission des deux trains est égal ou majeur à la capacité du routeur, tous les paquets sont groupés pour former un nouveau groupe. Quand ce dernier groupe quitte le routeur, le débit de transmission du nouveau train est la vitesse de transmission de l'interface de sortie.

Vu que la longueur du groupe est différente de chaque routeur dû aux bandes passantes et flux des trafics. Cela ne nous donne pas de l'information utile du trafic de la source au récepteur pour mesurer la capacité du chemin réseau. Néanmoins si la longueur du groupe est envoyée via messages ICMP, chaque paquet ICMP pourra emporter cette information à la source pourqu'elle puisse déterminer la bande passante physique (capacité). Cette méthode est employé pour mesurer la capacité après le lien avec moindre capacité dans NCS [**?**].

Modèle

Déf. Intervalle d'échantillonnage :

Les mesures de la bande passante disponible dépendaient fortement d'intervalle du temps utilisé dans les mesures. La précision dépend de cet intervalle. Pour intervalles courts du temps nécessitaient

de la précision ; Pour intervalles longs du temps, la bande passante disponible doit avoir deux valeurs, une moyenne et un rang entre la valeur minimale et maximale.

Le chemin réseau est modélisé avec deux flux individuels : Le trafic de paquets croisés (XT) et le trafic de paquets test (PT). La figure 9.39 montre ce modèle simplifié.

PT_i	XT_j	PT_i	...	XT_j	PT_i	XT_j	PT_i

FIG. 9.39 – Modèle de trafic réseau simplifié

Un concept clé pour cet algorithme est la MBS (Taille Maximale de Rafale), définie précédemment. Si la MBS est plus petite que le Produit Bande Passante – Délai (Bandwidth Delay Product ou BDP), la bande passante effective du chemin sera réduite.

La longueur maximale du train de paquets doit être inférieure à la MBS, sinon les paquets à la queue du train seront rejetés, en causant d'erreurs dans les tests. La longueur maximale du train de paquets dépend de la variation du trafic croisé.

En pratique, choisir la meilleure taille du rafale est un problème rusé car la MBS dépends du trafic croisé, lequel varie au cours du temps. La MBS mésurée pendant un période de temps court expose le trafic croisé et les situations d'attentes pour ce moment, et la MBS obtenue pendant une période plus longe expose la stabilité moyenne pendant l'intervalle du temps. Aucune de ces MBS ne pourra garantir qu'en envoyant à la vitesse maximale le routeur avec goulot d'étranglement de jetera des paquets.

Dans l'algorithme FAC^2, la longueur maximale du train de paquets est moins qu'un quart de la MBS pour arranger la variation du trafic croisé et réduire la possibilité de réjeter paquets du trafic.

Algorithme FAC^2 :

Pendant un intervalle de temps déterminé et en utilisant un train de paquets.

R_{snd} : Taux de transmission du train de paquets.

R_{rcv} : Taux de réception du train de paquets.

C_p : Capacité du chemin mesuré.

A_{bw} : Bande passante disponible.

Si R_{snd} est supérieur à la bande passante disponible et la longueur du train de paquets est inférieure à MBS pour éviter perte de paquets :

$$R_{rcv} = \frac{C_p * \sum_{i=2}^{n} PT_i}{(\sum_{i=2}^{n} PT_i) + \sum_{j=0}^{m} XT_j} = \frac{C_p * PT}{PT + XT} = \frac{C_p * R_{snd}}{R_{snd} + R_{xt}}$$

$$R_{xt} = \frac{C_p * R_{snd}}{R_{rcv}} - R_{snd}$$

$$A_{bw} = C_p - R_{xt} = C_p - R_{snd} * (\frac{C_p}{R_{rcv}} - 1) = R_{snd} - C_p * (\frac{R_{snd}}{R_{rcv}} - 1)$$

Comme C_p est inconnu, on utilise une estimation de la capacité basée en mesures du débit maximal. Alors la bande passante estimée est :

$$A'_{bw} = R_{snd} - C'_p * (\frac{R_{snd}}{R_{rcv}} - 1)$$

Et l'erreur d'estimation :

$$erreur = (C'_p - C_p) * (\frac{R_{snd}}{R_{rcv}} - 1)$$

Si $R_{snd} = R_{rcv}$, alors $A_{bw} = R_{rcv}$. Cela peut être obtenu aussi en minimisant l'erreur.

En itérant la première équation n fois, d'où n est le nombre d'itérations pour atteindre la convergence :

$$R_{rcvn} = \frac{R_{snd0} * C_p^{2n}}{R_{snd0} * (C_p + R_{xt})(...)(C_p^n + R_{xt}^n) + R_{xt}^{2n}}$$

Par exemple, si le réseau est vide ($R_{xt} = 0$) :

$$R_{rcvn} = C_p^{2n - n\frac{(n+1)}{2}}$$

Avec n = 1 ou 2, alors $R_{rcvn} = C_p$.

Le nombre d'itérationes est proportionel à l'utilisation. Si l'utilisation est presque 100% le nombre d'itération tend vers l'infini. Cela veut dire $C_p = R_{xt}$. La capacité est calculée par :

$$C_p = \frac{(R_{snd1} - R_{snd2})}{(\frac{R_{snd1}}{R_{rcv1}} - \frac{R_{snd2}}{R_{rcv2}})}$$

Commentaires

L'algorithme FAC^2 est capable de mesurer (et non estimer) la bande passante disponible d'un chemin réseau d'une manière précise et rapide. Il n'est pas intrusif, n'impacte pas le trafic et les hôtes de test et consomme peut de bande passante. En plus, il n'est pas besoin des privileges pour obtenir information des routeurs. Néanmoins, il ne peut pas mesurer les bandes passantes d'éléments réseau au dèla d'un goulot d'étranglement dans un schéma à lien par lien dû aux caracteristiques des trains de paquets.

FSE est l'algorithme pour mesurer la capacité du lien au dèla du goulot d'étranglement. Un système de mesure plus complet incluira les deux algorithmes.

Pour ne déranger pas le trafic croisé la longeur du train de paquets doit être très inférieur à la MBS.

En mesures de lien par lien, quand un train de paquets traverse un lien étroit, son taux de transmission est réduit, et des erreurs sont produites aux liens suivants. La solution est de relier avec l'algorithme FSE car le trafic croisé existe toujours.

Un algorithme de contrôle de transmission basé sur MBS ai une meilleure adaptation à la variation de trafic croisé qu'un mecanisme basé sur la fenêtre de congestion (TCP). Le mecanisme basé sur MBS peut mesurer la taille réelle et la bande passante disponible au routeur avec goulot d'étranglement, aussi bien que le trafic croisé, et calculer la taille effective de la file d'attente – MBS. Cet algorithme peut rapidement ajuster la taille de rafale envoyer et s'adapte aux changements de bande passante. Alors, un protocole de transmission basé sur MBS peut d'une manière efficace eviter les pertes de paquets pour utiliser completement la bande passante disponible.

La bande passante actuelle a supérée la bande passante des matériels aux ordinateurs et continuera à le faire dans l'avenir. Alors deux problèmes sont envisages :

1. Est-il possible qu'un équipement avec moindre capacité mesure une bande passante supérieure à la carte réseau ou les systèmes d'éxploitation ?. Les algorithmes actuels sont capables de mesurer la bande passante disponible si les équipements aux extrêmes ont un débit supérieur à la bande passante disponible et cette situation ne sera plus valide dans l'avenir.
2. Si le débit est haut, le temps pour transmettre/récevoir un paquet dévient court (problème de résolution).

9.4.23 Technique TRENO

TRENO [**?**] estime le volume de transfert d'un flux dans un chemin réseau. Elle est une technique hybride dans le sens qu'elle utilise l'outil "traceroute" et une version idéalisée des algorithmes de contrôle de flux du protocole TCP Reno. Elle est implementée dans l'outil avec le même nom.

Principe

Cette technique emule le protocole TCP à acquitements sélectifs (SACK) avec quelques différences :

– L'état est mantenu au testeur. Le récepteur seulement répondre les paquets ICMP pour chaque paquet test. Les pertes d'aller et retour sont traitées de la même manière.
– Elle ne retransmet pas les paquets perdus néanmoins elle toujours fait les adjustements propres de la taille de fenêtre.

Cette technique utilise les numéros de séquence pour emuler le protocole TCP, en performant une version idéalisée des algorithmes de contrôl de congestion du type TCP. Elle utilise le paramètre *cwnd* pour mesurer les paquets au réseau.

Modèle

Le volume de transfert d'un flux est défini comme la capacité d'un transfert assez long et soutenu pour atteindre l'équilibre avec le réseau.

Les paquets emportent numéros de séquences réflexés aux réponses, alors cette technique peut déterminer quel paquet produit quelle réponse.

Les paquets test sont summis aux même effets de congestion comme les paquets du protocole de transport TCP : délais d'attente et pertes liées à la congestion.

Propriétés :

– Un très bon volume de transfert demande un réseau clair ou netoyer.
– La plus part des problèmes réseau interferant avec la performance interactive desestabilize la performance du volume.

Phase d'Échantillonnage

Cette technique teste le réseau avec paquets ICMP ECHO ou paquets TTL UDP, lesquels démandaient paquets ICMP.

Commentaires

Cette technique et le protocole TCP utilisaient le même algorithme de contrôle de congestion. La différence essentielle entre cette technique et le protocole TCP est pendant la phase de découverte car chacune a une connaissance précise sur les différents extrêmes et chacune doit estimer l'extrême contraire.

Sous certains circunstances cette technique sur estime la performance des implementations actuelles de TCP car elles n'utilisaient pas l'aquittemments sélectifs.

9.5 TECHNIQUES PASSIVES

La grande majorité de techniques pour estimer la bande passante sont actives, c'est à dire qu'elles interfèrent avec le trafic normal du réseau, elles sont donc intrusives. On a vu que plusieurs techniques présentaient des évolutions pour devenir le moins intrusives sans jamais atteindre la non intrusivité. Pour développer des techniques non intrusives il faut penser aux techniques passives ou à inférence, lesquelles n'ont pas des phases de test mais à sa place elles ont des phases de récollection. Les techniques passives sont aussi accessibles aux utilisateurs sans besoin d'avoir les privilèges d'administrateur réseau.

9.5.1 Technique Nettimer

Cette technique [**?**] sert à estimer le goulot d'étranglement en temps réel. Elle peut mesurer la bande passante dans une direction en capturant un paquet à l'équipement hôte et dans deux directions en capturant deux paquets aux équipements d'extrêmes.

Principe

Si deux paquets sont envoyés si prés l'un de l'autre pour être mis en attente tous les deux au goulot d'étranglement, alors les paquets arrivent au récepteur avec la même séparation laquelle ils sont sortis du goulot d'étranglement.

Modèle

En pratique des considérations de base ne peuvent être appliquées :

1. Les deux paquets sont mis en attente l'un après l'autre au goulot d'étranglement et pas après lui. Des techniques de filtrage peuvent être appliquées pour mitiger cette phenomene.
2. Les deux paquets sont très proches d'une manière qu'eux sont mis en attente au goulot d'étranglement. Cela répresent un problème pour liens avec goulot d'étranglement à haut débit et pour les mesures passives.
3. Le routeur au goulot d'étranglement suit une politique de type PAPS.
4. Les délais de transmission sont proportionnels aux tailles de paquets test.

$$t_n^1 - t_n^0 = \max(\frac{s_1}{b_1}, t_0^1 - t_0^0)$$
$$b_l = \frac{s_1}{(t_n^1 - t_n^0)}$$

D'où :

t_n^m : Temps d'arrivée du paquet m au lien n.

s_m : Taille du paquet m.

b_l : Bande passante au goulot d'étranglement.

Phase d'Échantillonnage

Une fenêtre d'echantillonnage de valeurs de bande passante déclarée comme w est utilisée. La taille optimale d'échantillons de bande passante w n'est pas démontré. Elle est considerée comme paramètre définit par l'utilisateur dans cette technique.

Phase d'Estimation

Le but des techniques de filtrage est de déterminer l'echantillons de bande passante qui répresentaient la bande passante et ceux qui ne les répresentaient pas. La figure 9.40 montre une situation que satisfait les considérations de cette technique et trois que ne satisfaient pas.

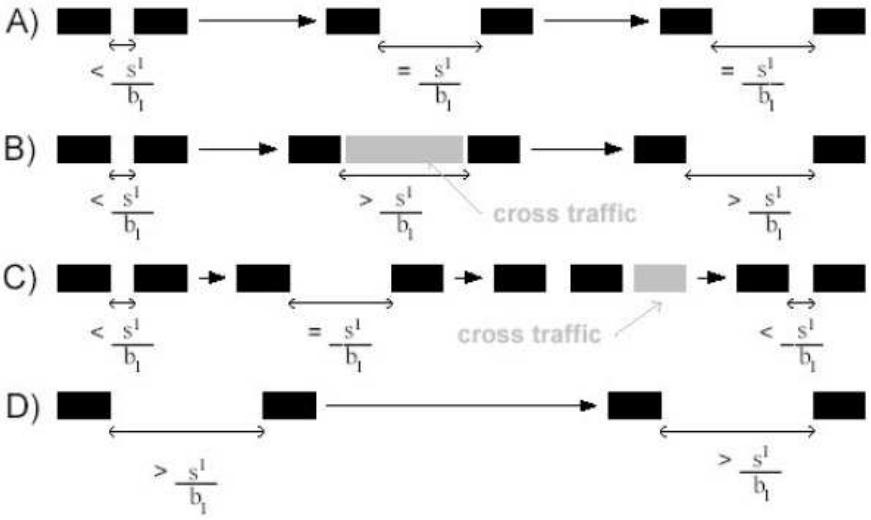


FIG. 9.40 – Quattre situations des séparations de paquets tout au long du chemin réseau

Filtrage par Estimation de la Densité de Probabilité

En utilisant le filtrage par estimation de la densité pour les cas B et C nous considérons que les echantillons influencées par le trafic croisé ne sont pas correlées tandis que les echantillons du cas A sont très correlées. Les paquets envoyés avec une bande passante inférieure arrivant avec une bande passante supérieure sont dans le cas C et peuvent être supprimées. La figure 9.41 montre ces situations hypotetiques, les echantilons de type C supérieures à la ligne constante sont enlevées, on calcule à partir d'echantillons restantes la densité de probabilité, la bande passante est la valeur maximale dans la densité de probabilité.

Les histogrammes ont les disavantages de longueur de classes (bin) fixes, alignement de classes fixes et uniformité de poids dedans chaque classe. Ils ont l'avantage de la vitesse de calcul. C'est pourquoi l'estimation choisit est celle de la densité noyau. L'idée est définir une fonction noyau K(t) avec la propieté :

$$\int_{-\infty}^{+\infty} \mathrm{K}(t)dt = 1$$

Alors, la densité pour un échantillon x est :

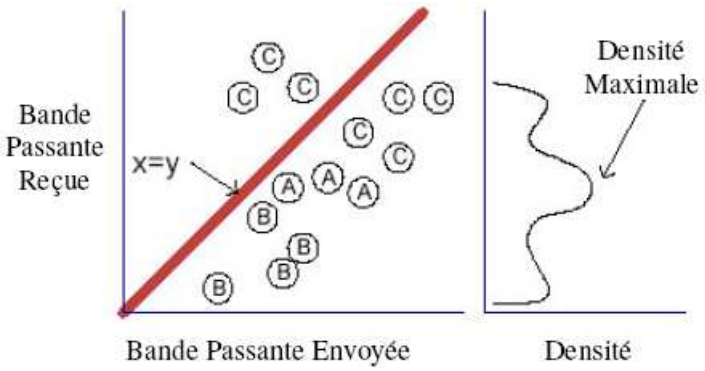


FIG. 9.41 – Situations hypotetiques et leur densité de probabilité

$$d(x) = \frac{1}{n}\sum_{i=1}^{n} K(\frac{x - x_i}{cx})$$

D'où :

c : taux de largueur noyau.

n : nombre de points dans cx appartenant à x.

x_i : iéme point.

Le taux de largueur noyau est utilisé pour contrôler le lisage de la densité de probabilité. Des valeurs plus grandes de c fournissent des résultats plus précis avec plus puissance de calcul utilisée (c est mis à la valeur de 0.10). La fonction noyau utilisée qui fournisse du poids supérieur aux échantillons prés au point où nous voudrions estimer la densité, en plus simple et rapide à calculer est :

$$K(t) = \{^{1+t;t\leqslant 0}_{1-t;t>0}\}$$

Filtrage en Utilisant la Rélation Bande Passante Reçue/Envoyée

La figure 9.42 montre une situation d'où le filtrage par estimation de la densité de probabilité peut se tromper car il fournira une bande passante inférieure à la réelle.

La rélation de bande passante reçue/envoyée pour un échantillon est :

$$p(x) = 1 - \frac{\ln(x)}{\ln(s(x))}$$

D'où :

s(x) : bande passante envoyée de x.

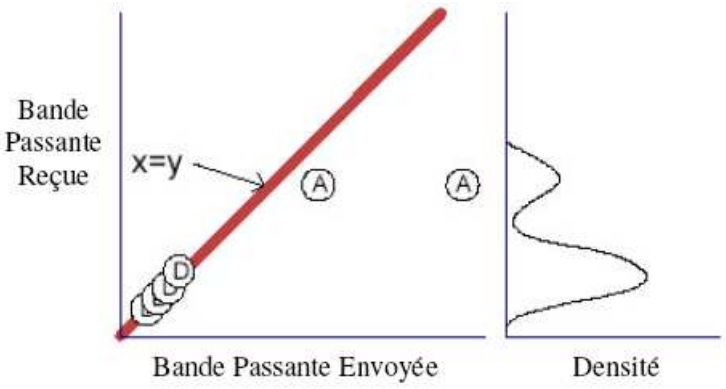


FIG. 9.42 – Rélation des bandes passantes reçues/envoyées

La rélation de la bande passante reçue est :

$$r(x) = \frac{\ln(x) - \ln(x_{\min})}{\ln(x_{\max}) - \ln(x_{\min})}$$

Filtrage Composé

En normalisant les techniques de filtrages présentées auparavant et en prennant la combinaison linéaire :

$$f(x) = 0.4 * \frac{d(x)}{d(x)_{\max}} + 0.3 * p(x) + 0.3 * r(x)$$

D'où :

$d(x)_{max}$: Valeur maximale de la densité noyau.

En choisissant la valeur maximale de $f(x)$ comme le goulot d'étranglement, il est possible de prendre en compte la densité et la rélation de bande passante reçue/envoyée sans favoriser les petites valeurs de x. Les poids de chaque membre est arbritraire.

Commentaires

Cette technique a moins de 10% d'erreur en plusieurs technologies (100 Mbps Ethernet, 10Mbps Ethernet, 11Mbps WaveLAN, ADSL, V.34 modem ou CDMA cellular data). En mesurant un goulot d'étranglement à 100Mbps, cette technique consume 6.34% du trafic mesuré et 4.52% des cycles dans un serveur à 366MHz et 57.6% des cycles dans une ordinateur à 266MHz.

9.5.2 Technique Basée sur la Minimisation de l'Entropie de la Distribution de Probabilité des Temps d'Inter Arrivée de Flux de Paquets TCP

Cette technique [**?**, **?**] sert à estimer les goulots d'étranglements partagés. Elle utilise la minimisation de l'entropie de la distribution des temps d'inter arrivée des flux de paquets TCP comme variable aléatoire.

Principe

L'idée générale consiste à utiliser les données collectées dans un lien pour estimer sa capacité à partir de la distribution des connexions TCP qui traversent ce lien.

La plus part des techniques sont basées sur le temps minimal d'inter arrivé de paquets, d'autres techniques suggèrent la mode de la distribution des temps d'inter arrivée de paquets. Dans [**?**] était montré que la distribution de probabilité des temps d'inter arrivée ne fournissent pas un bon estimateur de la capacité minimale d'une route.

Cette technique est basée sur l'observation que l'arrivé des flux de paquets agrégés partageant le goulot d'étranglement ont différentes statistiques que les flux de paquets que ne le partagent pas. En particulier, l'entropie des temps d'inter arrivée est plus petite pour le trafic agrégé partageant le même goulot d'étranglement.

Étant donné que le trafic d'entrée dans le goulot d'étranglement est plus grand que sa capacité, la plus part du temps ce goulot d'étranglement est occupé. Alors, les paquets sortent ce goulot d'étranglement avec un espacement de temps de valeur égal. Le temps entre deux paquets consécutifs est l'inverse de la bande passante du goulot d'étranglement. Les routeurs vidés après ce goulot d'étranglement maintiennent cet espace.

Avant d'atteindre le récepteur l'espacement entre paquets du même flux est varié par le trafic croisé d'autres routeurs non-vidés après le goulot d'étranglement. Pour découvrir les flux qui partagent le goulot d'étranglement, l'observateur doit récupérer l'espace du temps constant entre paquets à partir de l'espace aléatoire qu'elle/il observe.

En assumant que l'observateur connaisse une borne supérieure du nombre de goulots d'étranglements cette méthode repère le groupement de paquets correct pour minimiser l'entropie d'espacement entre paquets des groupes.

Modèle

Cette technique introduise quelques définitions :

1. Entropie : Le concept d'entropie est utilisé comme mesure de l'incertitude d'une variable aléatoire. L'entropie H(x) d'une variable aléatoire discrète x, avec probabilité p(x), est définie par :

$$H(x) = \sum_i p_i(x) \log_2 p_i(x)$$

1. Une fonction distribution de probabilité uniforme a une entropie supérieure a celle d'une fonction distribution de probabilité avec pics, soit continue ou discrète.
2. Observateur : Les termes observateur, récepteur et récepteur intermédiaire sont équivalents.
3. Flux : Ensemble de paquets provenant de la même source. Il est assumé que tous les paquets du même flux parcourent la même route et partagent le même goulot d'étranglement.
4. Groupe : Ensemble de flux. Un groupe correct est un ensemble de flux qui partagent le même goulot d'étranglement tandis qu'un groupe incorrect est le contraire.
5. Espacement entre paquets : En ordonnant tous les paquets dans un groupe et en respectant l'ordre d'arrivée, alors l'espacement entre paquets est la différence entre les arrivées de deux paquets divisés par la taille du premier paquet dans la paire.
6. Goulot d'étranglement : Un lien est un goulot d'étranglement sur un intervalle du temps T, si son trafic d'entrée est continûment plus grand que son capacité sauf pendant instants petits comparés avec l'intervalle T.
7. Temps de Transmission Normal (Normal Transmission Time ou NTT) : Temps qu'un lien prenne pour transmettre un paquet de 1500 octets.

La figure 9.43 montre quatre possibles formes de la distribution de probabilité des temps d'inter arrivée de flux de paquets :

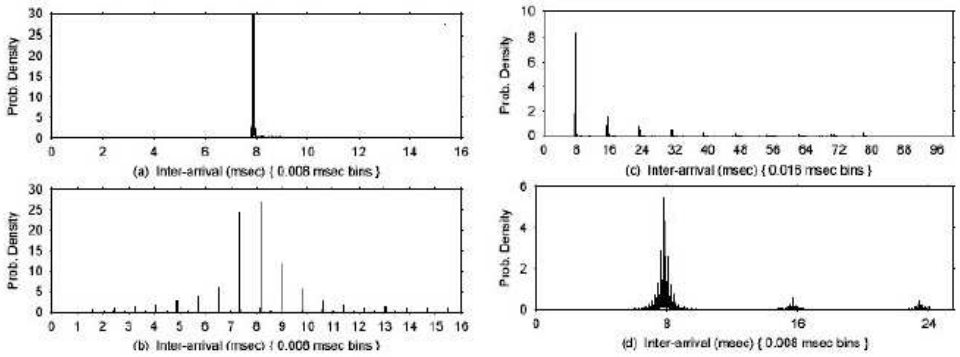


FIG. 9.43 – Quatre possibles formes de la distribution de probabilité des flux de paquets TCP à l'observateur

Chacune de formes a un nom particulier comme assigné dans [**?**], la subfigure (a) montre un seule pic (Single Spike), ce le cas d'où le flux de paquets traversa le goulot d'étranglement sans trafic croisé significatif, le pic corresponds au NTT du goulot d'étranglement. La sufigure (b) montre une gibbosité de pics (Spike Bump), ce le cas d'où le flux de paquets traversa un goulot d'étranglement de bande passante inférieur suivi par un goulot d'étranglement de bande passante supérieur. La gibbosité est centrée dans le NTT du goulot d'étranglement de bande passante inférieur, la séparation entre les pics a une valeur du NTT du goulot d'étranglement de bande passante supérieur. Une gibbosité de pics emporte information sur deux goulots d'étranglement. La subfigure (c) montre un train de pics (Spike Train), ce le cas d'où le goulot d'étranglement traversé est partagé avec trafic croisé substantiel. La séparation entre les pics est le NTT du goulot d'étranglement. Finalement la subfigure (d) montre un train de gibbosités (Train of Bumps), ce le cas d'où le flux traversa d'abord un goulot d'étranglement à basse bande passante partagé avec trafic croisé substantiel. Ensuite le

flux traversa un goulot d'étranglement avec trafic croisé moins important. La séparation entre les pics d'une gibbosité est le NTT du goulot d'étranglement de bande passante supérieure alors que la séparation entre les gibbosités est le NTT du goulot d'étranglement avec bande passante inférieure.

La détection des goulots d'étranglement partagés est un problème de groupement, d'où les objets groupés sont des flux. La distribution de probabilité d'inter arrivé d'un groupement correct de flux de paquets montre une aire presque uniforme avant la première mode. La distribution de probabilité pour groupements incorrects de flux de paquets montre plus d'aléas. Une mesure quantitative de ces aléas discrimine entre combinaisons de flux partageant le même goulot d'étranglement et des combinaisons ne partageant pas des goulots d'étranglements (Voir la figure 9.44

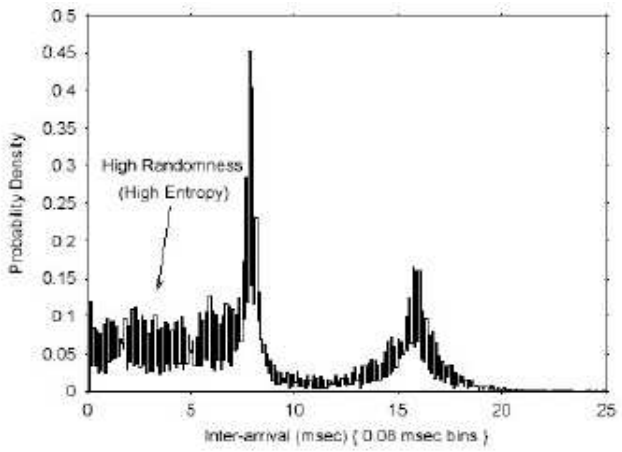


FIG. 9.44 – Distribution de probabilité des temps d'inter arrivée entre paquets

Pour développer une technique de groupement basée dans la minimisation de l'entropie, il faut résoudre deux problèmes :

1. La complexité de calcul pour chercher parmi tous les possibles groupements de flux le groupement qui minimise l'entropie (quelques techniques itératives sont considérées).

Technique Itérative de Groupement Basée à l'Entropie de Pairs :

1. Prendre un groupement aléatoire pour initialiser
2. À chaque itération :
3. Prendre une source S_i (aléatoire ou round-robin)
4. Pour chaque source S_j différente de S_i :
5. Calculer l'entropie du pair H_{ij} de $\{S_i, S_j\}$ en résultant de combiner les paquets des deux sources.
6. Pour chaque groupement C_k, calculer la moyenne H_{ij} sur toutes les sources $S_j \in C_k$.
7. Déplacer S_i vers le groupement Ck avec l'entropie de pairs moyenne minimale.
8. Répéter jusqu'à aucune source change leur groupement.

Techniques Itératives de Groupement Basé à l'Entropie de KMoyens (KMeans) :

Ces techniques comparaient un flux avec un groupement entier. Elles différè par le poids elles assignent à l'entropie d'un groupement particulier.

1. Technique de KMoyens à Poids de Groupement : La fonction de coût est un poids moyen des entropies des groupements, d'où le facteur de poids est le nombre de sources dans le groupement :

$$coût = \frac{\sum_{c=1}^{N} N_c H_c}{N}$$

D'où :

N_c : Nombre de sources dans le groupement c.

H_c : Entropie du groupement c.

N : Nombre de groupements.

1. Technique de KMoyens à Poids d'Echantillons : La fonction de coût est un poids moyen des entropies des groupements, d'où le facteur de poids est le nombre de paquets dans un groupement :

$$coût = \frac{\sum_{c=1}^{N} P_c H_c}{N}$$

D'où :

P_c : Nombre de paquets dans le groupement c.

Algorithme :

1. Prends un groupement aléatoire pour initialiser.
2. À chaque itération :
3. Prends une source S_i (aléatoire ou round-robin)
4. Effacer S_i du groupement.
5. Pour chaque groupement C_j
6. Additionne S_i à C_j et calcule le coût du groupement.
7. Déplace Si au groupement avec le coût minimal.
8. Répéter jusqu'à aucune source change leur groupement.
9. Choisir une fonction à minimiser. L'équation précédente n'indique pas comment combiner les entropies de plusieurs groupements pour obtenir une quantité à minimiser. Une fonction de coût est la quantité à minimiser, différentes fonctions de coût donnent différents niveaux de précision.

Phase d'Échantillonnage et d'Estimation

Cette technique applique la technique itérative de KMoyens à poids d'échantillons. Elle n'applique pas l'entropie de Shannon mais l'entropie de Rènyi. Celui est dû à que l'entropie de Shannon discrimine les agrégations de flux partagées et non-partagées. Néanmoins la nature en épi des distributions de probabilité des temps d'inter arrivée des paquets produise des problèmes que l'entropie de Rènyi résoudre.

L'entropie de Rènyi est une généralisation de l'entropie de Shannon, dans le limite (q→1) l'entropie de Rènyi converge à l'entropie de Shannon ($\lim_{q\to 1} K_q(x) = H(x)$) :

$$K_q(x) = \frac{\log_2 \sum_i p_i^q}{1-q}$$

D'où :

q : Ordre de l'entropie de Rènyi.

L'entropie de Rènyi et de Shannon partagent plusieurs propretés. Les deux entropies atteignent leur maximum avec une distribution uniforme. Aucune ne dépend pas de la valeur d'où la probabilité coïncide. L'entropie de deux sub-ensembles indépendants d'un ensemble de données est la somme des entropies individuelles. L'entropie de Rènyi donne plus de poids aux valeurs de haute probabilité plus que base probabilité (dû aux effets d'échantillons petits). Il ne faut pas choisir des hautes valeurs de q. D'après expériences réalisées les valeurs de q=4 et q=5 donnent des bons résultats. L'équation du coût change à :

$$coût = \frac{\sum_{c=1}^{N} P_c K_c}{N}$$

D'où :

K_c : Entropie Rènyi du groupement c.

L'algorithme itératif modifié est le suivant :

1. Commence avec chaque flux dans un groupement lui-même.
2. Prends une source Si d'une manière round-robin.
3. Essaie de déplacer Si de son groupement à chaque autre groupement.
4. Accepte le déplacement que minimise le coût total.
5. Répète à partir du pas deux autant que possible.

Commentaires

Les observations passives des temps d'inter arrivé des paquets emportent information sur la capacité et l'ordre relatif de liens congestionnés tout au long de la route traversée par le flux.

En notant les temps d'inter arrivé de paquets, un observateur passif peut regrouper les flux alors que tous les flux dans un groupe partagent le même goulot d'étranglement. Cette approche est basée sur l'hypothèse que le regroupement correct minimise l'entropie du temps d'inter arrivée vu par l'observateur.

La plus part de propositions pour estimer les caractéristiques ne marchent pas bien au trafic du protocole UDP car elles sont basées sur l'hypothèse que les pertes de paquets sont détectables.

Cette technique n'envoie pas paquets test, ne requiert aucune coopération des sources, elle n'assume rien sur le contenu des paquets ou le protocole des équipements aux extrêmes. Elle n'assume que les paquets emportent numéros de séquences alors elle peut pas être basée aux pertes. Elle n'assume pas que les liaisons montantes utilisaient des politiques d'attente particulières alors elle travaille avec toutes les disciplines d'attente ainsi que les protocoles de transport (TCP, UDP et Flux à Multiple diffusion).

La détection passive des goulots d'étranglements partagés est très précise en utilisant la minimisation de l'entropie si l'observateur est bien placé pour que le trafic des goulots d'étranglements traversa le lien observé tandis qu'elle devient imprécise et non-pratique si une petite partie du trafic des goulots d'étranglements est observée.

Pendant les périodes de sous-utilisation, le goulot d'étranglement n'estampille pas les paquets et l'espace entre paquets et leur sortie ne sont pas constants. Ces périodes d'utilisations sont peut probables quand le numéro de flux est large. En plus, leur durée au goulot d'étranglement est courte comparée avec leur durée des périodes d'estampillage. En outre, les routeurs après le goulot d'étranglement produise d'attentes sans être goulots d'étranglements en modifiant les espacements des paquets.

Si les files d'attente entre le goulot d'étranglement et l'observateur sont vides, les techniques à minimisation de l'entropie peuvent potentiellement trouver le groupement correct. La Plus part du temps, une petite fraction du trafic de sortie du goulot d'étranglement finisse au lien de l'observateur.

Une difficulté d'agroupement qui partagent le goulot d'étranglement augmente vu que le problème de groupement ne passe pas à l'échelle par rapport à la taille de l'entrée. Par exemple, trouver le groupement que minimise l'entropie peut être fait en examinant tous les possibles groupements et en choisissant la solution qui minimise l'entropie totale. Cette approche incrémente le problème d'une manière exponentielle en devenant un problème NP-Difficile.

Les techniques de groupement proposées ignorent toute dépendance du temps qui peut exister entre deux points consécutifs. Pour améliorer ces techniques on peut prends en compte que si un paquet est mis en attente au goulot d'étranglement le paquet prochain traversant le même goulot d'étranglement sera mis en attente. Cette modification est simple à réaliser, à la place de calculer la fonction de probabilité d'une variable aléatoire que représente l'espacement entre paquets, il faut calculer l'entropie d'un vecteur avec premier composant l'espace entre paquets actuel et comme deuxième composant l'antérieur espacement entre paquets. Comme ce vecteur est à deux dimensions la technique est appelée "2D-KMeans".

La précision est haute (90% à 99%) si une fraction longue du trafic au goulot d'étranglement traversa le lien observé. Cette précision est dégradée si la fraction du trafic au goulot d'étranglement traversant le lien observé est moins de 15%. Cela est dû que le trafic croisé est considéré comme bruit. Alors l'observation passive en utilisant ces techniques ne dévient pas pratique si l'observateur

est un récepteur dans l'Internet. Néanmoins, ces techniques sont plus précises si l'observateur est placé dans un lien traversé par trafic du goulot d'étranglement.

Les techniques à groupement ont besoin un nombre petit de paquets. S'il n'y a pas du trafic croisé et la topologie réseau n'est pas très complexe, près de 20 paquets par flux sont suffisants pour un groupement correct. La technique "KMeans" a toujours besoin près de 100 paquets. Ce nombre petit de paquets est dû que ces techniques ne comparent seulement les paquets de deux flux mais les paquets d'un flux avec les paquets d'un groupement. Les nombre exact de paquets par flux dépends du nombre de goulots d'étranglements, flux à classifier et le type d'erreurs à traiter.

La méthode peut détecter n'importe quel goulot d'étranglement partage centaines de flux. En plus, elle travaille bien avec une charge lourde de trafic croisé et les erreurs sont réduits exponentiellement si le trafic croisé au goulot d'étranglement est réduit. La méthode peut s'appliquer en temps réel.

9.5.3 Technique EMG

EMG [**?**] (Equally-spaced Mode Gaps) infère les capacités des liens en se basant sur les temps d'inter arrivé des paquets. Cette technique est intégrée dans l'outil de mesures passives appelé "M&M".

Principe

Cette technique utilise la séparation entre les modes consécutives des distributions de probabilité d'inter arrivée de paquets tandis que les techniques traditionnelles utilisaient la localisation des modes dans la même distribution de probabilité. Elle est basée dans l'outil "multiQ".

Les modes aux égales séparations dans les distributions de probabilité d'inter arrivée de paquets d'un flux correspondent aux temps de transmissions de paquets de 1500 octets dans le goulot d'étranglement tout au long de la route.

L'enveloppe de la distribution de probabilité des temps d'inter arrivée décrit la capacité minimale du goulot d'étranglement tout au long de la route, laquelle la sortie est modulée par les liens congestionnés.

L'outil "multiQ"

Cet outil [**?**] estime automatiquement la densité de probabilité des temps d'inter arrivée de paquets en se basant sur une progression d'échelles lissées correspondant à un ensemble des capacités de liens communs. À chaque échelle, elle construit un estimateur noyau de la distribution de probabilité d'inter arrivée de paquets à différentes échelles, et la balaie pour chercher des modes statistiquement valables. Les séparations entre les modes sont calculées. La distribution de ces séparations a aussi ses propres modes. La localisation de la première mode dans la distribution des séparations est le temps de transmission de 1500 octets d'un routeur simplifié par une échelle lissée. La figure 9.45 montre la distribution des temps d'inter arrivée pour la même connexion â différentes échelles de résolution (10 usec, 45 usec et 240 usec).

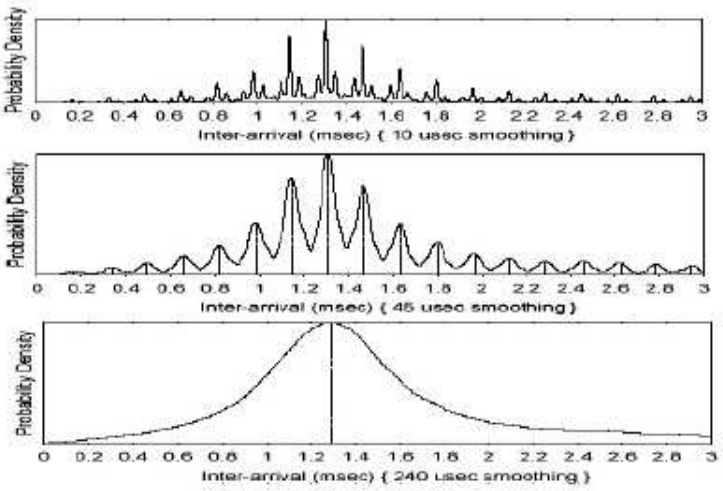


FIG. 9.45 – Distribution de probabilité des temps d'inter arrivée de paquets à différentes échelles de résolution

Avec la plus basse résolution, nous voyons une mode dans la distribution, laquelle correspond à la capacité du goulot d'étranglement avec moindre capacité. Si on augmente la résolution, cette mode devient fracturée en petits pics qui correspondaient au goulot d'étranglement avec capacité supérieur. L'enveloppe des pics suit la mode originale.

Cet outil "multiQ" peut découvrir jusqu'à trois goulots d'étranglement à partir d'un flux, et peut travailler avec les acquittements aussi comme l'inter arrivé des données. Selon [**?**] cet outil fournisse la même précision que l'outil "Pathrate" présentée [**?**, **?**].

Des connexions TCP trop courtes sont considérées comme des test trop pauvres. Ici seulement sont considérés les flux significatifs définis comme les connexions TCP durant au moins deux secondes, atteignant un débit moyen d'au moins 20 paquets par seconde ($\approx$2 paquets/RTT) et contenant au moins un paquet de 1500 octets. La rafale du trafic croisé est alors définie comme le trafic intervenant entre deux paquets consécutifs d'un flux significatif.

À chaque lien la rafale individuelle entre paires de paquets peut varier. On cherche à comprendre les probabilités de plusieurs quantités de trafic intervenant entre paires de paquets dans un flux significatif au lien congestionné.

Les rafales de trafic croisé sont intéressantes pour les techniques de mesures passives car elles traduisent les attentes en temps d'inter arrivé observées aux récepteurs.

Algorithme :

1. Calculer l'inter arrivé des flux à partir de la trace
2. Initialiser l'échelle := 10 us
3. Pendant que l'échelle est $<$ 10,000 us
   (a) Calculer l'estimateur PDF noyau avec une longueur = échelle
   (b) Trouver les modes

(c) S'il existe une mode, à M :
   i. Fournir la capacité de (1500*8/M) Mb/s
   ii. Sortir
(d) Déterminer la mode des séparations
(e) Déterminer la PDF des séparations
(f) Initialiser G := la mode la plus petite dans la PDF des séparations
(g) Si la probabilité dans G>0.5
(h) Fournir la capacité de (1500*8/G) Mb/s
(i) Incrémenter l'échelle

Les modes sont identifiés comme la valeur maximale locale dans la densité estimée qu'ait "dips" statistiquement significatifs. Un "dip" significatif est défini comme celui auquel les "dips" dans les deux cotes de la mode baissent plus que la déviation standard de la densité noyau au maximum local. La déviation standard est donnée par :

$$StdDev(g(x)) = sqrt[\frac{g(x) * R(K)}{nh}]$$

,

D'où :

g(x) : estimateur au point x.

R(K) : Rugosité de la fonction noyau.

n : Nombre de points.

h : Longueur du noyau.

Cette analyse est bonne si nous avons contrôle du récepteur ou d'un point d'observation près de lui. Si la trace est prise à la source, les données d'inter arrivées ne sont pas intéressantes car les paquets sont espacés pour le lien appartenant à la source ou liens près de la source.

Les flux des acquittements maintiennent cet accomplissement n'importe quelle information peut être récupérée. Si le point d'observation est au milieu du réseau, les données et les acquittements doivent être étudiées pour découvrir les goulots d'étranglements montants et descendants à partir du point d'observation. En générale la PDF d'inter arrivé des acquittements contient plus d'information que l'inter arrivé de données, mais elle a plus de bruit.

La version actuelle de l'outil "multiQ" ne mesure pas goulots d'étranglements avec une capacité supérieure à 155 Mbps.

Cette technique ajoute les définitions suivantes :

1. Flux Significatif : Un flux TCP qui atteint un débit moyen $>$ 10 paquet par seconde ($\approx$1pkt/RTT), qui contient au moins 50 paquets, et ai une MTU de 1500 octets (la plus part des flux de données moyens à longs ont cette taille).
2. Rafale de trafic croisé : Trafic intervenant entre deux paquets consécutifs appartenant à la trace d'un flux.

3. Demi-RTT : Latence entre le temps d'envoi d'un paquet et sa réception de l'acquittement correspondant au paquet tel qu'il apparaît dans la trace du flux.

EMG est plus robuste aux erreurs produites par le trafic croisé et peut découvrir la capacité de multiples liens congestionnés et leur ordre relatifs dans la route, à partir d'un flux de paquets TCP.

EMG est aussi plus robuste avec paquets de donnés qu'avec paquets d'acquittements. EMG est basée sur les rafales de trafic croisé, qui depends de la distribution de la taille de paquets. Si les paquets n'ont pas la taille dominante de 1500 octets, alors cette technique ne fonctionne pas.

Cet outil contient un analyseur TCP de pertes et RTT appelé "mystery" qui travaille avec l'outil "multiQ". Elle peut mesurer les caractéristiques de la route et corréler différents types de mesures de la même route, en produisant des nouveaux résultats. Les deux outils analysent les flux TCP des moyens aux longs.

Sur des outils passifs, cette technique peut découvrir les capacités et l'ordre relatif jusqu'à trois goulots d'étranglements tout au long de la route à partir de la trace d'un flux TCP.

L'analyseur "mystery" rapport les événements de pertes TCP, les paquets perdus et des mesures demi-RTT granulaires en utilisant la durée d'un flux. Avec l'outil "multiQ" permets de corréler les caractéristiques de la route tel que la capacité, le nombre de goulots d'étranglement, l'existence de goulots d'étranglement de retour, avec le performance TCP.

9.5.4 Technique ABEst

ABEst (Available Bandwidth Estimator) [**?**] propose un estimateur de la bande passante disponible d'un lien dans un réseau à services différentiés (Diffserv).

Principe

Cette technique est proposée pour un système centralisé de gestion réseau (au niveau domaine) que déterminera la bande passante disponible de tous les liens dans le domaine. L'approche le plus adéquat seras de récolter l'information de toutes les sources possibles à la plus haute fréquence en permettant que l'information de la basse de données pour la gestion soit renouvelle dans les contraintes. Néanmoins, cette approche peut être chère pour la signalisation et la conservation de données.

Cette technique est basée sur le protocole SNMP (Simple Network Management Protocol). Un réseau basé sur SNMP consiste en trois composants : les dispositifs gérés (parfois appelés éléments du réseau comme les serveurs, les commutateurs et les ponts, les hubs, les ordinateurs, ou les imprimantes), les agents, et les systèmes de gestion réseau (Network Management Systems ou NMSs). Un dispositif géré est un nœud du réseau qui contient un agent SNMP et qui réside dans un réseau géré. Les dispositifs à géstionner collectaient et sauvegardaient information de la gestion sur les basses de donnés dédiées à tel fin (Management Information Bases ou MIBs) et provisionnaient cette information aux NMSs en utilisant SNMP. Un agent est un logiciel basé dans la gestion du réseau et qui réside dans le dispositif à géré. Un agent ait connaissance locale de l'information de gestion et traduit cette information dans un format compatible avec SNMP. Le NMS démarre les applications de monitoring et contrôle des dispositifs à gérer. Les NMSs provisionnaient le volume du travail et les

ressources de mémoire requises pour la gestion du réseau. SNMP peut être utilisé comme technique passive pour observer un dispositif spécifique.

L'outil MRTG (Multi Router Traffic Grapher) utilise le protocole SNMP et provisionne les calculs de l'utilisation du réseau chaque cinq minutes. Cette technique proposa de modifier l'outil MRTG à MRTG++ en améliorant le temps de calcul à dix secondes.

Modèle

L'algorithme indique la durée de validation de l'estimateur avec un degré de confiance très haute. Cet algorithme est basé sur la régression linéaire. Il sert à prédire l'utilisation du lien. L'algorithme est adaptatif car, dépendant du trafic et son profil, il varie le nombre d'échantillons pour calculer la régression linéaire.

Pour un lien entre nœuds i et j :

C : Capacité d'un lien [bps].

A(t) : Capacité disponible au temps t [bps].

L(t) : Charge de trafic au temps t [bps].

τ : Durée de calcul pour MRTG.

$L_\tau[k]$, k∈N : Charge moyenne dans [(k-1)τ, kτ].

p : Cantité de mesures passées pour la prédiction.

h : Cantité de mesures futures pour un prédiction assurée.

$A_h[k]$: La valeur estimée au kτ valide dans [(k+1)τ, (k+h)τ].

La capacité disponible peut être obtenue comme A(t) = C – L(t). Le problème peut être modélisé comme une prédiction linéaire :

$$L_\tau[k+a] = \sum_{n=0}^{p-1} L_\tau[k-n]\omega_a[n]$$

pour a∈[1,h]

D'où le membre droit de l'équation sont les valeurs du passé et les coefficients de prédiction $\omega_a[n]$ et le membre gauche sont les valeurs prédites. Ce problème peut être résolu en utilisant la méthode de la covariance.

Les auteurs proposa de changer les valeurs de p et h en se basant dans la dynamique du trafic (cet changement est la distinction de cette algorithme par rapport aux autres basées dans la régression linéaire).

Algorithme :

1. À l'instant k, la mesure de la bande passante disponible est souhaitée.
2. Trouver les vecteurs ω_a, a∈[1,h] en utilisant la méthode de la covariance étant donné p et les mesures antérieures.

3. Trouver [Å$_\tau$[k+1], ... , Å$_\tau$[k+h]]^T et [Å$_\tau$[k-p+1], ... , Å$_\tau$[k]]^T.
4. Prédire A$_h$[k] pour [(k+1)τ, (k+h)τ]
5. Au temps (k+h)τ, obtenir [L$_\tau$[k+1], ... , L$_\tau$[k+h]]^T.
6. Trouver le vecteur d'erreur [e$_\tau$[k+1], ... , e$_\tau$[k+h]]^T.
7. Mis à la valeur k = k+h
8. Obtenir les nouvelles valeurs pour p et h.
9. Allez dans le pas 1.

p$_0$ et h$_0$ sont les valeurs initiales de p et h. Dans l'etape deux il faut résoudre les équations de la covariance. Elles sont données en forme de matrice comme R$_L\omega_a$ = r$_a$, pour a=1, ... , h :

$$R_L = \begin{bmatrix} r_L(0,0) & ... & r_L(0,p-1) \\ ... & ... & ... \\ r_L(p-1,0) & ... & r_L(p-1,p-1) \end{bmatrix}$$

$$\omega_a = [\omega_a(0)\omega_a(1)...\omega_a(p-1)]$$

$$r_a = [r_L(0,-a)r_L(1,-a)...r_L(p-1,-a)]$$

Pour déterminer la covariance à partir des mesures :

$$r_L(n,m) = \sum_{i=k-N+p}^{k} L_\tau[i-n]L_\tau[i-m]$$

D'où N affecte la précision de l'estimation.

Phase d'Échantillonnage

L'algorithme change d'une manière dynamique le nombre d'échantillons utilisés dans le procedure de prédiction, ainsi comme la durée de temps de la prédiction.

Le nombre d'échantillons nécessaires pour un n et N donnés est de (n+N). La covariance est actualisée à chaque fois que la valeur de p change dans l'etape huit de l'algorithme. La solution des équations de la covariance nous donne ω_a qui seras utilisé pour prédire Å$_\tau$(k+a), a=1, ... , h. Avec les coefficients de prédiction ω_a nous prédictions [Å$_\tau$[k+1], ... , Å$_\tau$[k+h]]^T.

Phase d'Estimation

Le prochain pas consiste à estimer la bande passante disponible dans l'intervalle [(k+1)t, (k+h)t]. Cela ce fait en obtenant une valeur représentative dedans l'intervalle. Nous pouvons le faire avec deux méthodes dependant des demandes des opérateurs réseaux. La valeur représentative de la bande passante disponible A$_h$[k] peut être donné par :

– A_h[k] = C – max {Å$_\tau$[k+1], … , Å$_\tau$[k+h]} qui donne un estimateur très conservateur de la bande passante disponible dans le lien pour la durée totale.
– A_h[k] = C - α ; d'où α est la bande passante effective, qui donne un estimateur plus réaliste et qui peut être syntonisé avec les demandes des opérateurs réseaux.

L'algorithme pour estimer la bande passante effective en ligne est :

1. Initialiser M=0 et i=k,
2. Obtenir la prédiction Å$_\tau$[i],
3. Actualiser $M = (1 - \frac{1}{i})M + \frac{\exp^{s\tau_\tau[i]}}{i}$.
4. Si i<k+p, allez à l'étape 2,
5. α(s) = log(M)/(sτ) ; Arreter.

Après avoir obtenu la charge actuelle [L$_\tau$[k+1], … , L$_\tau$[k+h]]^T au temps (k+h)t, nous trouvons le vecteur de prédiction d'erreur [e$_\tau$[k+1], … , e$_\tau$[k+h]]^T d'où chaque élément est donné par : e$_\tau$[k+a] = (L$_\tau$[k+a] - Å$_\tau$[k+a])^2 pour a = 1, … , h.

L'algorithme pour déterminer les nouvelles valeurs de p et h est :

1. Si σ/μ>Th$_1$, décrémenter h jusqu'à h$_{min}$ et incrémenter p jusqu'à p$_{max}$ multiplicativement.
2. Si Th$_1$>σ/μ>Th$_2$, décrémenter h jusqu'à h$_{min}$ et incrémenter p jusqu'à p$_{max}$ sommativement.
3. Si σ/μ < Th$_2$, alors :
   (a) Si $u = Th_3 * M_E^2$, décrémenter h jusqu'à h$_{min}$ et incrémenter p jusqu'à p$_{max}$ sommativement.
   (b) Si $Th_3 * M_E^2 > u > Th_4 * M_E^2$, garde h et p constantes.
   (c) Si $u < Th_4 * M_E^2$, incrémenter h et décrémenter p jusqu'à p$_{min}$ sommativement.

D'où :

M$_E$: Erreur maximal

Th$_i$: Seuils pour les décisions d'actualisations des paramètres p et h.

Commentaires

Il est désirable d'obtenir la bande passante disponible en mesurant la bande passante des LSP actuels car ils donnent une mesure plus réelle. La bande passante disponible peut aussi être obtenue en soustraient la réservation nominal pour les tuyaux a partir de la capacité du lien qui nous donne un borne inférieur.

Cet algorithme peut être fortment amélioré en utilisant une méthode pour obtenir les valeurs des seuils qui sont basés dans les caractéristiques du trafic.

9.6 TECHNIQUES HYBRIDES

Comme nous avons décrit au debut de ce rapport ; les techniques hybrides sont des techniques combinées par techniques actives et passives mais pas d'un seul type de technique.

9.6.1 Technique TEMB

TEMB [**?**] (Tool for End-to-end Measurement of available Bandwidth) sert à mesurer la bande passante disponible d'une route. Elle est précise, flexible et passe à l'échelle. Elle détecte le lien de la route avec moindre bande passante disponible et la valeur de la bande passante disponible.

Principe

TEMB est une technique que combine les techniques actives et passives pour obtenir des mesures de la bande passante disponible de bout en bout d'une manière plus précises et plus fiables. Elle est efficace et facile à implémenter. Les mesures des paquets sont traitées au même temps de calcul que la réexpédition au niveau de la couche IP (trois). TEMB utilise les MIBs des routeurs.

Modèle

Soient les temps d'estampillage des paquets test de retour à la source T={t_1, t_2, ... , t_N} avec $t_1 < t_2 < \ldots < t_N$. Soient les interfaces trouvées tout au long du chemin P={I_1, I_2, ... , I_N} avec N liens dans le chemin. Soient les compteurs pour chaque interface C_I={c_{I1}, c_{I2}, ... , c_{IN}} avec c_{Ik} le compteur de l'interface I jusqu'au temps t_k. Soit S_I le vitesse de l'interface prise par les paquets de mesures. L'utilisation de l'interface est determinée avec le k-iéme échantillon :

$$U_{IK} = \frac{[C_{IK} - C_{I(k-1)}]}{[t_k - t_{(k-1)}]}$$

pour k = 2, 3, ... , N

La bande passante disponible est :

$$A_{IK} = S_I - U_{IK}$$

Phase d'Échantillonnage

Les formats des paquets test sont du registre de données ; ils sont montrés dans la figure 9.46

(a) (b)

– La source envoie paquets test lesquels collectaient information de la route, ces informations sont renvoyés du récepteur à la source (10 paquets de mesures pendant un seconde). Le numéro dix approvisionne des estimations raisonnables sans être trop intrusive.

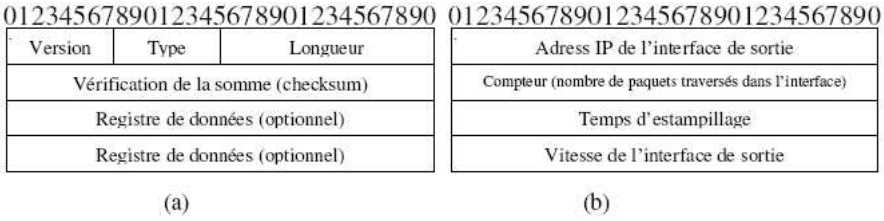


FIG. 9.46 – Formats des paquets test (a) et du registre de données (b)

- Quand les paquets arrivaient au récepteur, celui obtient l'information des liens compatibles avec TEMB. Le récepteur permute les champs source et destination de l'entête IP du paquet de mesure, change à 2 le champ type des paquets pour indiquer le retour à la source et l'envoi aux files de transmission.
- Chaque lien de la route ajoute leur information aux paquets de mesure, s'ils sont compatibles avec TEMB. Comme le routeur a modifié le paquet de mesure, il doit déterminer la longueur du paquet et le CRC (cycle de vérification rédundante ou Cyclic Redundancy Check) pour le nouveau paquet. Ces valeurs doivent être modifiées avant de mettre le paquet de mesure dans les files de transmission.
- La majorité de la charge du travail est donnée au routeur source du chemin à tester. Plusieurs paquets sont envoyés pour obtenir une estimation plus adéquate de l'identification du lien avec la moindre bande passante disponible et sa valeur.
- Si le profil du trafic tout au long du chemin est trop variable, TEMB est dessiné pour envoyé une autre série de dix paquets pendant un autre seconde afin d'obtenir une meilleur identification du lien avec moindre bande passante disponible dans la route.

Phase d'Estimation

L'opération est divisée en deux parties :

- La première consiste a obtenir des estimations rudes de la bande passante disponible dans tous les liens du chemin et déterminer leur quantité minimale comme une méthode pour déterminer le lien du chemin avec moindre bande passante disponible.
- La deuxième partie consiste à obtenir une mesure plus précise de la bande passante disponible dans ce lien particulier. Cette phase est faite en utilisant la technique passive "ABEst" présentée dans [**?**] (détaillée en bas). La technique "ABEst" est basée sur l'outil MRTG et appelée aussi MRTG++.

Pour la première partie. Une fois tous les (N-1) estimateurs ont été obtenus, TEMB essai d'identifier le lien avec moindre bande passante disponible.

Si tous les estimations indiquaient le même interface comme le lien avec moindre bande passante disponible et leurs valeurs sont similaires, alors TEMB identifie le lien avec la moindre bande passante disponible. Par contre, si les estimations différènt en interface ou avec leurs valeurs, alors TEMB envoi les dix prochains paquets de mesures.

- L'accord sur l'identification du lien avec moindre bande passante disponible est atteignit si au moins un certain pourcentage des estimations sont en accord (agree$_{link}$).

– L'accord sur la valeur de la bande passante disponible est atteignit si les (N-1) estimations de l'interface I sont au moins dans un certain pourcentage de l'estimation minimal (agree$_{avail}$).

Les valeurs d'accord doivent être très proches avec agree$_{link} > 100$ et agree$_{avail} < 100$. La bande passante disponible moyenne dans l'interface I est :

$$A_I = \frac{1}{[n * (N-1)]} \sum_{k=1}^{p*(N-1)} A_{Ik}$$

D'où :

n : nombre d'essais de la première partie de l'estimation.

Si l'estimation de la bande passante disponible pour n'import quelle interface tombe sur 150% de la valeur minimale, l'interface est marquée en état critique est doit être révisée avec plus de détail.

Commentaires

Cette technique est dessinée pour travaillé en deux situations :

1. Quand le chemin entre la source et le récepteur est le lien minimal. Dans ce cas, les paquets des mesures ont-leur registres de données vides, le champ type est mis en valeur de 0, et les mesures des paquets son encapsulés IP. En voyageant chaque lien ajoute leur registre de données et fait suivre avec les tables " lookup " IP préexistantes.
2. Quand la mesure de la bande passante disponible est nécessaire pour un chemin avec lien non minimal entre la source et le récepteur. Les registres de données sont inclus dans les paquets de mesures à la source. Les registres de donnés contiennent l'adresse IP des liens tout au long du chemin. Dans le champ de type la valeur 1 est mise. Les liens modifient les registres de données en incluant l'information de l'interface et cheminent les paquets vers les files de transmission.

Les mesures de la bande passante disponible peuvent être obtenues pour chemins prédéterminés entre deux extrêmes. Les paquets de mesures ne puissent pas être supérieurs à la taille de 556 octets. La configuration des paquets de mesures est la suivante :

– Version : Valeur de 0
– Type : mettre la valeur de 0 si le paquet est envoyé de la source au récepteur et (si) est routée lien-par-lien dans le réseau, mettre la valeur de 1 si le chemin de la source à la destination est déjà fixé et encodé dans le paquet, et mettre la valeur de 2 si le paquet est retourné du récepteur, comme expliqué avant.
– Longueur : longueur totale des paquets TEMB exprimé en octets.
– Vérification de la somme : Appliquer la technique de fiabilité de type CRC pour tout le paquet.
– Registre de données : Il est modifié dans chaque lien, comme expliqué auparavant.

Les estampillages des routeurs sauvegardées aux paquets de mesures ont de signification locale et pourtant ils ne sont pas corrélés avec d'autre routeurs. Les routeurs n'ont pas besoin de l'information sur la référence du temps réel ou ses estampillages. L'unique différence avec les estampillages consécutifs est utilisée pour calculer l'utilisation pendant l'intervalle de mesure. L'ordre des paquets de mesures n'importe s'ils atteignent leur destination.

9.7 Conclusions

Cet état de l'art a était réalisé dans l'axe de recherche comprenant le développement de mécanismes d'ingénierie de trafic basés sur mesures de la bande passante pour le routage et le contrôle d'admission. La bonne compréhension du processus et des techniques d'estimation des métriques de la bande passante est une composante clé, en particulière pour la définition d'un modèle conceptuel de contrôle à boucle fermé en incluant le processus de mesure et d'estimation.

Nous avons évalué autour d'une trentaine de techniques estimant les métriques liées à la bande passante de bout en bout tel que la bande passante lui-même (capacité), la bande passante disponible et le débit. Nous constatons le fait qu'il existent beaucoup plus de techniques d'estimation de bande passante actives que passives et par l'instant une seule technique d'estimation hybride (consistant en utiliser au moins une technique active et une technique passive pour estimer la métrique d'intérêt).

Ce travail propose une architecture des techniques d'estimation et une classification étendue aidant à mieux comprendre et conceptualiser les techniques d'estimation de bande passante pour les réseaux. Nous remarquons la nécessité de développer un cadre de normalisation pour le processus d'estimation aux équipements réseau collaborant avec les groupes de travail PSAMP et IPFIX à l'IETF.

Nous constatons l'apparition de techniques d'estimation de type combinées (union de plusieurs techniques de la même catégorie, soit passives ou actives) et techniques mesurant multiples-métriques. Notre travail dans le cadre du projet est orienté vers les changements nécessaires pour évoluer les plate-formes existantes (e.g. Saturne2) vers l'estimation à multiple-métriques.

Nous avons trouvé que la plus part des techniques étaient validées par simulation ou aux environnements contrôlés en comparant les résultats avec les techniques traditionnelles. Uniquement le rapport final du projet d'estimation de bande passante de CAIDA et leur article à la conférence PAM2005 présentaient les validations/comparaisons sur un environnement réel à haute charge.

Selon cet étude, les outils de mesure active travaillant avec le principe de congestion auto-induite par exemple "pathload" et "pathchirp" donnent les estimations les plus précises. Nous pensons que ce principe doit être plus exploité car selon nous il se base dans le fondement d'identification de systèmes (application d'un stimulus appelé rafale comprenant plusieurs taux de transmissions).

Les évaluations des techniques d'estimation trouvées ne prennent pas en compte la durée d'estimation mais la précision et les intrusions. Nous pensons que la stabilité de l'estimation est aussi importante pour évaluer la robustesse de ces techniques.

Nous avons aussi constaté que la métrique la plus répondue est celle de la bande passante disponible. Nous pensons que cela est dû à l'importance pour le développement de mécanismes adaptatifs d'ingénierie de trafic se basant sur cette métrique comme par exemple notre sujet de recherche. En plus nous avons trouvé peux des techniques détectant plus d'un goulot d'étranglement.

Nous avons trouvé une seule technique estimant la distribution de probabilité d'une métrique de bande passante à la place de la valeur moyenne. Nous considérons qu'une telle approche est si important pour mieux comprendre la dynamique des estimations des métriques.

Nous avons trouvé une technique mesurant les caractéristiques de bande passante au niveau de la couche deux, appelée Spectroscopie de l'Internet, utilisant la transformée de Radon des distributions

des délais inter-paquets et la minimisation de l'entropie. Cette technique n'est pas présente dans le rapport pour raisons de taille.

Nous avons trouvé que la plus part des techniques sont développées sous un modèle basé aux files d'attente avec politique de service de type PAPS. Un effort de recherche doit être fait pour développer techniques d'estimation en environnements aux routeurs de politiques de service variées. En plus, une seule technique proposa un modèle déterministe. En autre, le modèle aux variations des délais de la technique ACCIG est aussi utilisé pour la synchronisation précise des horloges dans un réseau sans l'utilisation d'un GPS.

Avec l'arrivée des équipements avec capacités de services différentiés des nouveaux enjeux sont posés sur les estimations des métriques de la bande passante par clase de service. Nous n'avons pas trouvé une technique estimant les métriques de la bande passante sur chemins avec qualité de service (Diffserv).

Pour conclure, les techniques d'estimation modernes sont plus effectives, la plus part des techniques d'estimation trouvées sont développées en outils informatiques à licence libre, les phénomènes affectant la précision des estimations restaient toujours un sujet de recherche (l'existence des équipements de la couche deux, la différence des tailles des files d'attente aux routeurs tout au long d'un chemin réseau et la charge du trafic croisé). Une évaluation de ces affectations et leur modélisation est un sujet de recherche important pour améliorer les techniques d'estimation.

Nous pensons que la technique idéelle doit fournir la radiographie complète du chemin réseau (capacité, bande passante disponible, délai dans un sens, pertes, variation des délais, etc.) pour avoir une ingénierie de trafic intelligente basée sur mesures qui prenne en compte un nombre varié de métriques dépendant de la fiabilité des estimations.

Chapitre 10

Evaluation of active measurement tools for bandwidth estimation in real environment

[1]

10.1 INTRODUCTION

Having an accurate estimation of available bandwidth on network links or on end-to-end paths is of high interest for many functions in networking as admission control, load balancing, (QoS-)routing, congestion control, etc. Passive monitoring tools are certainly the most appropriate tools for this purpose. But they are most of the time not accessible to users that need such information. Even for carriers or ISP that manage their own domain or autonomous system, and that then can have access to any information they need about their own network state, they miss the same type of information for the networks of other carriers or ISP they are connected to. As a consequence, tools for estimating available bandwidth on an end-to-end path are based on active measurement techniques, which are said to be user oriented, at the opposite of passive measurements which are carrier or ISP oriented. With the active approach, these tools then provide a solution for having an easy access to such network feature estimations, and this can be used for any network structures and technologies. Many tools for estimating available bandwidth have appeared in the recent years as Abing [**?**], Spruce [**?**], Pathload [**?**], [**?**], IGI-PTR [**?**], Pathchirp [**?**], etc. But tools based on active measurements for available bandwidth only make possible to get estimations on this parameter, while passive monitoring tools can measure it in a very accurate way. The question then deals with the accuracy of available bandwidth estimation tools based on active techniques. In addition, there are very few comparisons between all these tools in real environments as the actual Internet. Existing literature essentially focuses on evaluating these tools on local and fully controlled platforms. It is then very difficult for potential users to select the best tool depending on their requirements. And we

[1] par Yann Labit et Philippe Owexrzarski

are facing this kind of problem : we need to estimate available bandwidth, but we are unable based on the current literature to find out the best suited tool for our need. We then started a study on the accuracy and efficiency of the main available bandwidth estimation tools. However, it is important to recall that active measurements consist in generating probe traffic in the network, and then observing the impact of network components and protocols on traffic : loss rate, delays, RTT, etc. Therefore, as active measurement tools generate traffic in the network, one of their major drawbacks is related to the disturbance introduced by the probe traffic which can make the network QoS change, and thus provide erroneous measures. Sometimes, active probing traffic can be seen as denial of service attacks, scanning, etc ; but in any case as hacker acts. Probe traffic is then discarded, and its source can be blacklisted. Intrusiveness of probe traffic is then one of the key features which active measurement tools have to care about. Besides, much work addresses this issue of probe traffic intrusiveness, trying to minimize the number of sent packets as well as their impacts on the network QoS. In addition, if an active measurement tool generates only few packets, it would certainly provide estimation results in a very short time, what is an important performance parameter in the Internet whose traffic is very versatile. This work evaluates the accuracy of active measurement tools aiming at measuring the available bandwidth on a path from a source to a destination workstation, as well as its intrusiveness level and response time. This evaluation relies on the use of very accurate passive monitoring tools, based on the DAG card [**?**] which is an absolute reference.

This chapter then first presents the main metrics for active measurement tools, and a list of tools which have been evaluated : these tools are classified according to their estimation / measurement technique, but also according to the kind of parameters they measure / evaluate (section 2). For instance, some, already quoted, measure available bandwidth, while other, as Clink [**?**], Pchar [**?**], Pathchar [**?**], etc, measure links or paths capacities. These two families of tools are important for this evaluation work as some of the available bandwidth estimation tools need to know the link or path physical capacity. The study and analysis proposed in this paper is being performed in the framework of the French Metropolis project which is presented in section 3. In particular, it is also explained in this section how the evaluation and analysis is going to be performed. It describes how active and passive measurement equipments, composing the Metropolis monitoring and measurement platform, can be jointly used for this purpose. Finally, section 4 presents results for the two families of tools we are considering, i.e. the one of tools measuring link or path capacity, whose results will be used for evaluating the results of the second tool family dealing with available bandwidth estimation.

10.2 METRICS, TECHNIQUES and TOOLS

Before sending data on a path of the network, users may want to know some information concerning QoS. It can be the same for network operators who want to optimize their routing strategy. Evaluating QoS and performances on a path most of the time deals with measuring or estimating capacity, available bandwidth, utilization level, loss ratio, etc. These parameters will give an idea of the QoS and performances users can expect. The following definitions present the main metrics to be used in this paper.

– Concerning data transmission, the term bandwidth or “capacity” is related to the width of the communication pipe and how quickly bits can be sent. The capacity can be defined as the maximum quantity of data per time unit when there is no cross traffic. We will speak about capacity of a link or a path. By considering a path of N links l_1, l_2, l_3, …, l_N, we define the capacity of each link by C_1, C_2, C_3, …, C_N. The capacity C of a path is determined

by the minimum capacity of a link. This link is called the narrow link. Let's note : $C = min(C_1, C_2, C_3, \ldots, C_N)$. In the following example (Figure 10.1), the capacity of the path corresponds to the one of the narrowest link which is C_1.

– The utilization of a link is the consumed part of the link capacity. Let's note U_i the utilization of a link.

– The available bandwidth is defined as the unused capacity in the link independently of the transport protocol. The available bandwidth is a function resulting from the utilization and the capacity. Let's consider the first path, made of N links : the available bandwidth for the i-th link is defined by :

$$AvB_i = C_i(1 - U_i) \tag{10.1}$$

The available bandwidth of a path is designed by the link which has the lowest available bandwidth :

$$AvB = min(AvB_1, AvB_2, AvB_3, \ldots, AvB_N) \tag{10.2}$$

The link having the minimum available bandwidth is called the tight link. In the example below, the tight link defining the available bandwidth is l_3 and AvB equals to AvB_3.

– Intrusiveness can be defined as the percentage of capacity that is consumed for the measurements. It means that intrusiveness I_X is equal to :

$$I_X(\%) = 100\frac{C_X}{C} \tag{10.3}$$

where C_X and C are respectively the amount of traffic sent in one second and the link capacity. C_X is the amount of bits generated in one run by the tool X during the probing time (time between the first and the last probe bit generated by the tool X). The probing time is different from the response time : The first probing bit is not necessarily sent to the destination at the moment the tool starts and the tool does not necessarily return its estimation right after sending the last probe bit.

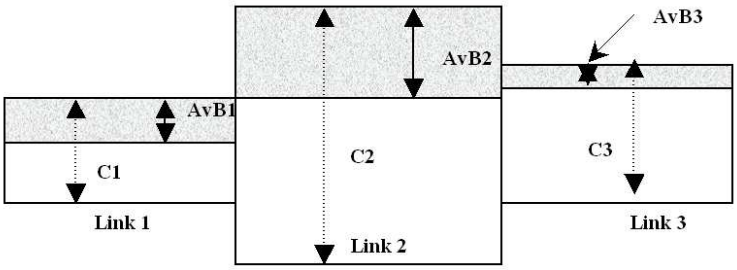


FIG. 10.1 – Narrow link (link 1) and tight link (link 3) on a path.

This chapter section then focuses on active probing tools estimating AvB, as well as the ones estimating the capacity. This last value is important for estimating AvB, as explained in equation (10.1). All these tools use theoretical network properties that are described in the already quoted literature. We do not describe here existing techniques of active probing tools as many papers already present this state of the art [**?**], [**?**] and provide taxonomies of these tools according to their measurement techniques. According to these taxonomies, the tools belong to the four following families :

– Variable Packet Size (VPS) probing which estimates the capacity of individual hops ; Examples : Clink, Pchar, Pathchar, Bing [**?**].
– Packet Pair/Train Dispersion (PPTD) which estimates end-to-end capacity ; Examples : Abing, Spruce, Pipechar [**?**], bprobe [**?**], cprobe [**?**], Pathrate [**?**], sprobe [**?**].
– Self-Loading Periodic Streams (SLoPS) which estimates end-to-end available bandwidth ; Examples : Pathchirp, IGI, Pathload.
– Trains of Packet Pairs (TOPP) which estimates end-to-end available bandwidth [**?**].

For more details about active probing techniques, the reader can look at [**?**]. A short taxonomy of tools evaluated in this paper is proposed in table in the figure 10.2. It gives for each tool, the name, author, version (Release), technique (Methodology), protocols, some interesting characteristic, the target (path or link), the need of root privileges, operating systems (L : Linux, B : BSD, Su : Sun, So : Solaris, I : Irix, F : FreeBSD, N : NetBSD, O : OpenBSD, A : AIX) and the number of hosts required (sender only (S) or, sender and receiver (S & R)).

Name	Authors	Release	Methodology	Protocol	Characteristic	Path / Link	Root	OS	Host
Clink	Downey	1-0	VPS	UDP	Bandwidth	Link	Yes	L,B,Su	S
Pchar	Mah	1-4+	VPS	UDP, ICMP	Bandwidth	Link	Yes	L,So,I,F,N,O	S
Pathchar	V. Jacobson	2-0-30	VPS	UDP, ICMP	Bandwidth	Link	Yes	F,N,O,L,So	S
Abing	Navratil	2-4-4	Packet Pair	UDP	Av Bandwidth	Path	No	L	S & R
Spruce	Strauss	0-2	Packet Pair	UDP	Av Bandwidth	Path	No	L,B	S & R
Pipechar	Guojon	2K1	Packet train	UDP	Av Bandwidth	Link	Yes	L,So,I,F,A	S
Pathchirp	Ribeiro	2-3-7	SLoPS	UDP	Av Bandwidth	Path	No	L,F,Su,A,I	S & R
IGI	Hu	1-0	SLoPS	UDP	Av Bandwidth	Path	No	F,L,Su	S & R

FIG. 10.2 – Taxonomy of evaluated tools.

10.3 EVALUATION METHODOLOGY

The evaluation of these tools is being performed on the Metropolis monitoring and measurement platform (Figure 10.3). The monitoring and measurement platform has been designed and developed, and it is now deployed on the Renater network (the French network for education and research). To keep in mind this platform, it consists of :

– Some passive microscopic monitoring probes based on the famous DAG card (designed and provided by the University of Waikato and Endace society in New Zeeland) [**?**] ;
– Some active measurement equipments. These active probes rely on the RIPE boxes that have been extended to support the NIMI software, as well as a self designed and developed measurement software environment called MetroMI (derived from the NIMI.s one). The Metropolis monitoring and measurement platform is depicted on the Figure 10.3.

The evaluation procedure under consideration in this paper relies on the use of passive measurements for evaluating the accuracy of active probing tools. We run the tools under concern for estimating the available bandwidth, and compare their results with the measurements made with the DAG based monitoring systems. Indeed, we are able to collect traffic traces at the edge of the Renater network,

and then to capture the traffic going to and from the core network, including the probe traffic. In addition, the core network is largely over provisioned : links have several Gbps capacity and are never charged more than 50%. At the opposite, access links are Fast Ethernet (100 Mbps) or GigE (1 Gbps) links. FastEthernet as well as GigE links are then the narrowest links on the paths between any of the RIPE equipments. These narrow links are then monitored using the DAG systems. It is important to note, and this is the proof for our evaluation methodology, that DAG systems are perfectly provisioned for capturing the traffic and very accurate : packet headers are extracted on the fly and timestamped with a GPS clock on the dedicated DAG hardware, and the resulting sample is stored on the hard drive of the machine after crossing an overprovisioned bus. This system is then insensitive to network load compared to some software dump solutions (as tcpdump for example). It is also important to mention the accuracy of the timestamp, the clock of the DAG system being synchronized on a GPS signal. Our RIPE boxes are also synchronized on a GPS signal, meaning that all the monitoring and measurement boxes are perfectly synchronized on the universal temporal reference. It also avoids any temporal drift as boxes resynchronized on the GPS pulse every second. The vendors of the GPS cards ensure less than 2 microseconds accuracy, what is largely sufficient in our case. The DAG based monitoring system is then an absolute reference and will allow us to evaluate the accuracy of the active measurement tools estimating the available bandwidth on the path from a given source to a given destination.

The evaluation results presented in section 4 of this chapter have been obtained between LAAS in Toulouse and LIP6 in Paris. LAAS is connected to RENATER with a FastEthernet link, while LIP6 has a GigE link (Figure 10.3). Note that these Figures are largely simplified compared to the real complexity of the network between LAAS and LIP6. The detail of links between LAAS and LIP6 as seen by traceroute is shown in table in the figure 10.4. This traceroute, as the experiment described in the following, has been run between polka.laas.fr (140.93.192.71) and adonis.lip6.fr (132.227.74.18).

However, for fully evaluating the accuracy and performances of active probing tools in different network and traffic conditions, we use a traffic generator. Its first ability is to generate a constant traffic at a given rate. By changing this rate, we will emulate networks with different load / capacities, and it will then be possible to evaluate the accuracy of the active tools in different networks proposing a full range of capacities. As we generate only constant traffic, this additional traffic has a limited impact on the dynamics of the global Internet traffic throughput which keeps the same variations as without the traffic generator.

In addition, by generating other traffic models than the constant one, we will also be able to evaluate the accuracy of available bandwidth estimation tools when confronted to cross traffic having very different properties and in particular the ones of current Internet traffic. This would help us to analyze in what conditions these tools are providing accurate and efficient results (or not), and why ?

The results presented in section 4 have been obtained following this methodology. The first part of section 4 presents the evaluation results of Clink, Pchar, and Pathchar. It gives the per hop bandwidth estimation. After evaluating the capacity of the links, especially the one of the narrow link, the second part of the section 4 presents evaluation results for the tools estimating available bandwidth : Abing, Spruce, Pipechar, IGI and Pathchirp. As for capacities, this second part presents the end-to-end available bandwidth estimation on a path which has its beginning at LAAS and its end at LIP6. IPERF has been used to generate constant UDP traffic on the LAAS' access link. Destination of IPERF traffic was ENSICA, an engineering school in Toulouse area, thus "reducing"

the available bandwidth on the LAAS' access link without impacting the rest of the path from LAAS to LIP6.

10.4 RESULTS

This section describes the experiments with available bandwidth estimation tools previously discussed (see table in the figure 10.2). We present results in two steps : We first present the results of bandwidth (capacity) estimation using clink, Pchar and Pathchar, and discussed their estimation error ratio, response time and intrusiveness. We conclude this first experiment with the reliability (and utility) of these tools in networking. We secondly present the results of the available bandwidth estimation at the output of LAAS using Abing, Spruce, IGI, Pathchirp and Pipechar. We discuss their estimation error ratio, response time, intrusiveness and reliability. We compare the available bandwidth estimation with DAG measurements. DAG measurements will also give information (probing time, amount of traffic generated by each tool) to calculate intrusiveness. All tools have been run more than 600 times to get consistent results.

10.4.1 EVALUATION OF CLINK, PCHAR, PATHCHAR

Clink, Pchar and Pathchar have been used for estimating the bandwidth for every link of the path between LAAS and LIP6 (15 hops in the path). We are especially interested by the estimation bandwidth for the LAAS output, normally a fast Ethernet link (100 Mbps). Figure 10.5 describes the bandwidth estimation without any Iperf cross traffic and with a 50 Mbps Iperf cross traffic respectively. One can observe that all these tools produce a bandwidth estimate far from the actual value. Clink proposes three values of the bandwidth : a low one, a high one and a best supposed one. Clink and Pathchar are approximately constant but these tools overestimate the bandwidth (case without Iperf traffic). At the same time, Pchar is very unstable. And, it presents unrealistic disruptions (when there is no Iperf cross traffic). With a 50 Mbps Iperf cross traffic, these three tools propose different estimations but the conclusion is similar : none of the tools produce good values. The capacity measured by Clink is negative for its three values. Pchar and Pathchar most of the time crash. The preceding experiments show very bad estimation results at least when confronted to two cases of cross traffic. Figure 10.6 (left) then extends these results by showing the bandwidth estimation for many cross traffic values ranging from 0 to 100 Mbps (every 5 Mbps). For each value of the cross traffic, 30 experiments have been run. Figure 10.6 (left) presents the estimation average for each of these cases, when possible. Indeed, the average for Pchar is not exploitable because results from one experiment to the other differ so much that it is not possible to get an useful estimate. For some values of cross traffic, Pchar also crashes most of the time, thus making the computing of an average impossible. Identically, Clink bandwidth estimation values (Low, High, Best) are very far from the real values : The best-supposed value appears as the worst estimate. Figure 10.6 (right) shows the estimation error rate which confirms the inaccuracy of Clink and Pathchar.

The preceding experiments clearly demonstrate that the tools are not accurate, reliable and robust. We nevertheless require a method for having an acceptable estimate of the narrow link bandwidth on the path from LAAS to LIP6. We then designed a very basic tool based on the use of 4 ping - StupPing - to compute a rough estimate of the bandwidth of the narrow link. Given the limited effort spent for designing this almost "stupid" tool, we do not consider it as a possible contribution in this

research area. In addition, it should work only in our specific case for which the narrow link on the path is the closest one from the probing source. The principle of StupPing is illustrated on our specific case, i.e. for estimating bandwidth between LAAS (Braveheart) and REMIP (Remip-v2). The first requirement for using StupPing is to get the list of routers and links which will be crossed between the source and destination : this can be obtained using traceroute, for example. Then, the StupPing process uses ping four times. First, ping the near end of a link with two different packet sizes. Next, use the same two packet sizes to ping the far end of the link. Let us call P_l and P_s the largest and smallest packet sizes (in bytes), $T1_l$ and $T1_s$ the ping times for the largest and smallest packets to the nearer interface (in seconds), and $T2_l$ and $T2_s$ the ping times for the largest and smallest packets to the distant interface (also in seconds). Finally, the difference $(T2_l - T2_s) - (T1_l - T1_s)$ represents the amount of time to send the additional data over the last link in the path. Therefore, the formula for bandwidth estimation is :

$$AvB = 16\frac{(P_l - P_s)}{(T2_l - T2_s) - (T1_l - T1_s)} \tag{10.4}$$

Figures 10.7 and 10.8 show the results got with StupPing. This 5-lines binary program computes a bandwidth near the actuel capacity when there is no IPerf cross traffic. The average of the bandwidth estimation is around 92 Mbps, what is quite close from the actual value. As for other tools, when there is Iperf cross traffic (higher than 35 Mbps), this binary program is out of range. Finally, StupPing performs better than other tools when cross traffic is less than 35 Mbps. Another advantage of this tool is that it does not require the root privileges.

The table in the figure 10.9 summarizes the results got with the probing tools for intrusiveness, and response time, analyzed thanks to the DAG card. We define the “Laas response time” as the time for estimating the bandwidth on the link between Braveheart and Remip-v2 (as this link is the first on the path between LAAS and LIP6, bandwidth estimation tools first provide the bandwidth estimation for this link, and then successively the bandwidth estimation for the following links). In all the tools analyzed, there are some differences in response and probing time, as well as in the probing traffic amount. Clink and StupPing have short (laas)response time (figure 10.9-right). Pchar is also quite fast but Pathchar takes 5 minutes (300 s) to provide the result for the Braveheart and Remip-v2 link. The probing time also varies a lot for Clink, Pchar and Pathchar. It is quite constant for StupPing which only evaluates the bandwidth for one link (LAAS- Remip-v2). But all other tools generate a lot of packets : Pathchar which takes more than 37 minutes (2232 s) for the probing time generates 110,3 Mbits per estimation, Clink around 54,3 Mbits and Pchar 6,9 Mbits. StupPing is the less intrusive tool with 0,13 Mbits sent. We finished this evaluation description with the intrusiveness. For all these four tools, results are good as they appear as lowly intrusive (less than 0,06% of additional probing traffic compared to actual operational traffic). StupPing shows an intrusiveness less than 0,01% (thanks to few ping and ICMP echos). The table in the figure 10.9 nevertheless points out that Clink and Pathchar generates lot of traffic even if their intrusiveness is low. But this means that the probing and response times are consequently too long.

Few concluding remarks for these first experiments :

- Most of these tools either rely on ICMP or UDP probe packets. Such ICMP packets are useful for determining information about a network like its capacity, RTT, loss, etc. But one of their biggest drawbacks is that normal users are not permitted to generate ICMP packets : root privileges are most commonly required.
- ICMP packet can overflow some hosts / routers if not used carefully (ICMP flooding).

– These tools are somewhat basic at this point, slow, not robust and not accurate. Dealing with accuracy of the results, StupPing is the only one to provide a correct estimation of the bandwidth (without overestimation), but of course, just for the closest link on the path. And when cross traffic increases (because of IPerf in our experiments), these active tools crash, thus exhibiting the limitations of the VPS probing technique.

10.4.2 EVALUATION OF ABING, SPRUCE, PATCHIRP, IGI, PIPECHAR

The first tools evaluated were supposed to give the bandwidth of the link between LAAS and Remip : we concluded that none of the tested tools produce accurate enough results. We just have a good estimation with StupPing. This part now focuses on tools able to estimate the available bandwidth on the path between LAAS and LIP6, i.e. on the link between LAAS and Remip. Figure 10.10, left and right sides, describe respectively the available bandwidth estimation without cross traffic and with a 50 Mbps IPerf cross traffic, for Abing, Spruce and Pipechar (which use the same probing technique). We describe at the end of this section some experiments with IGI and Pathchirp, two tools using the SLoPS probing methodology. Results on Figure 10.10 (left) show an inaccurate available bandwidth estimation, with Abing and Spruce (when no Iperf traffic is generated). In this case, these tools underestimate the available bandwidth. Moreover, Abing provides unstable estimations. On the other side, when IPerf generates 50 Mbps of cross traffic, estimation results are better (Figure 10.10, right). The third tool, Pipechar, gives good estimations in both cases (Figure 10.10, left and right sides). The major drawback of this tool is that it crashes very often (Figure 10.10 and 10.11). Given these first good results when evaluating these tools with 50Mbps IPerf cross traffic, Figure 10.11 (left) presents the estimation results obtained with the three tools when IPerf cross traffic is ranging from 0 to 100 Mbps (for each value, each tool has been run 30 times). Figure 10.11 (left) shows the average for the 30 estimations for each tool and each IPerf traffic level. Figure 10.11 (right) presents the related estimation error. Not that we got the same results with Spruce and Abing : this was expected as both tools use the same technique (packet pair). However, the hypothesis of Spruce which assumes that there is only a narrow link on the path, and that the narrow link is also the tight link is very strong. This can explain why we got bad estimation results. And it is difficult to conclude for Pipechar accuracy as it crashes very often (and also needs root privileges). Figure 10.12 summarizes the results we got with the 3 considered tools for response time and intrusiveness : It then appears that these tools are not very intrusive (less than 0,6%) and have short response time. But the available bandwidth estimations are not good, except for Pipechar with a low level of cross traffic. We conclude these experiments with some evaluation of IGI and Pathchirp (with 50Mbps of IPerf cross traffic), shown on Figure 10.13. It appears that Pathchirp overestimates the available bandwidth and IGI is unstable (but its average is not really far from the actual value).

10.5 CONCLUSION

In this chapter, we presented an evaluation of active probing tools for estimating capacity and available bandwidth on a link/path in a real Internet environment. The tools which have been evaluated are Clink, Pchar and Pathchar for estimating link capacity, and Abing, Spruce, Pipechar, Pathchirp and IGI for available bandwidth on a path. These experiments show bad results for all these tools in real environment. In addition, most of the tested tools hugely overestimate bandwidth. Such overestimation is really dangerous. For example, for a rate based congestion control, using Pa-

thChirp estimations will cause huge congestion phenomena. It would be safer to underestimate the available bandwidth. But, functions using such results would not be optimal. In addition, these tools are not robust enough especially when there is cross traffic. Given the bad results got while cross traffic was constant, we even did not evaluate them with a highly variable traffic. But we can guess that the results would not be very good given the large response time of these tools : there is a level of magnitude between the variation rate of current Internet traffic and response time of these active probing tools. As a conclusion, we are not convinced by any of these tools.

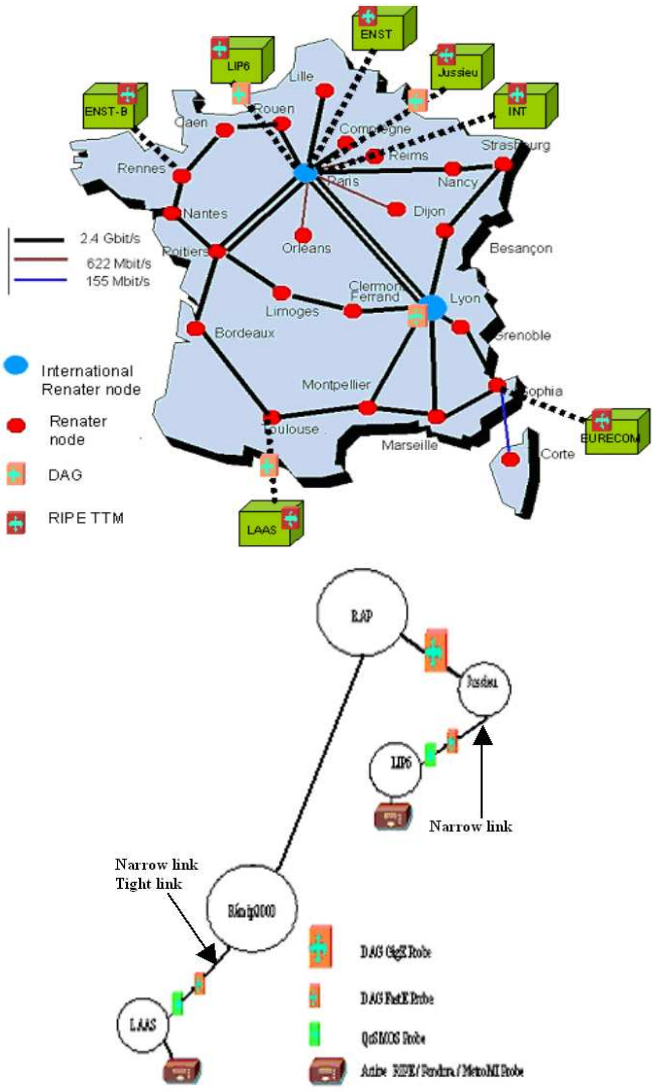
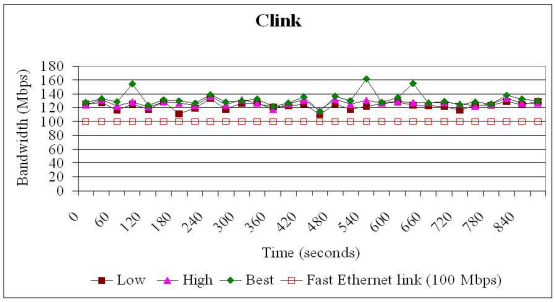
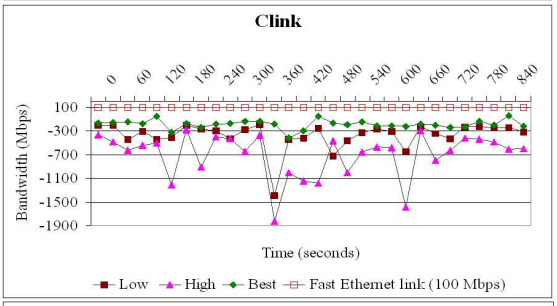


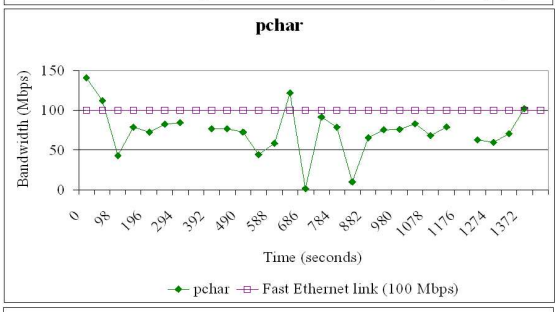
FIG. 10.3 – Metropolis monitoring and measurement platform.

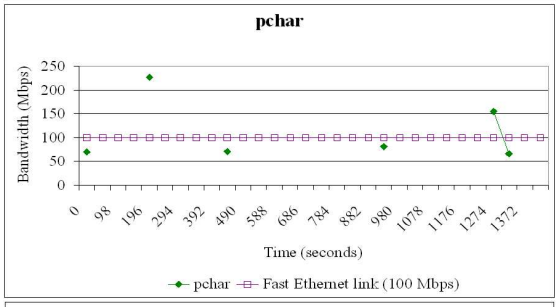
traceroute to 132.227.74.18 (132.227.74.18), 30 hops max, 40 byte packets				
1 braveheart	(140.93.0.75)	0.744 ms	0.721 ms	0.813 ms
2 remip-v2	(195.83.132.129)	0.213 ms	0.211 ms	0.207 ms
3 195.220.57.25	(195.220.57.25)	0.734 ms	0.700 ms	0.669 ms
4 193.52.8.1	(193.52.8.1)	0.951 ms	0.984 ms	0.884 ms
5 193.55.105.238	(193.55.105.238)	1.353 ms	0.880 ms	0.993 ms
6 toulouse-g3-1.cssi.renater.fr	(193.51.181.178)	1.137 ms	1.086 ms	1.008 ms
7 bordeaux-pos2-0.cssi.renater.fr	(193.51.180.13)	12.025 ms	11.524 ms	11.608 ms
8 poitiers-pos1-0.cssi.renater.fr	(193.51.179.253)	11.611 ms	12.234 ms	12.103 ms
9 nri-b-pos4-0.cssi.renater.fr	(193.51.179.133)	11.969 ms	12.201 ms	11.798 ms
10 jussieu-pos4-0.cssi.renater.fr	(193.51.180.157)	11.469 ms	11.753 ms	15.453 ms
11 193.50.20.73	(193.50.20.73)	15.667 ms	11.702 ms	11.301 ms
12 jussieu-rap.rap.prd.fr	(195.221.127.182)	11.962 ms	11.680 ms	11.463 ms
13 r-scott.reseau.jussieu.fr	(134.157.254.10)	13.327 ms	13.334 ms	15.345 ms
14 olympe-gw.lip6.fr	(132.227.109.1)	12.668 ms	13.334 ms	12.409 ms
15 adonis.lip6.fr	(132.227.74.18)	12.526 ms	13.147 ms	12.328 ms

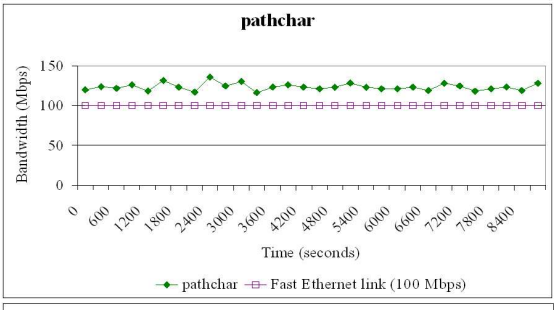
FIG. 10.4 – Traceroute results.

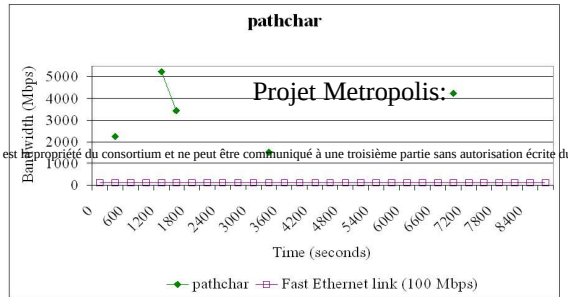












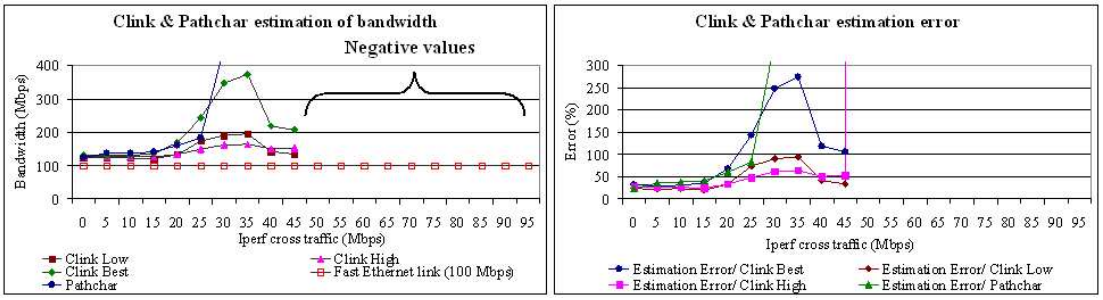


FIG. 10.6 – Bandwidth estimation average and error rate.

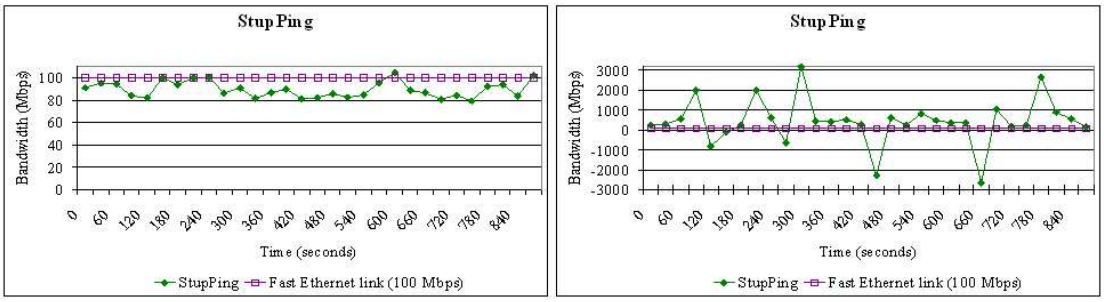


FIG. 10.7 – Bandwidth estimation (Left : no Iperf cross traffic - Right : with 50 Mbps Iperf cross traffic.

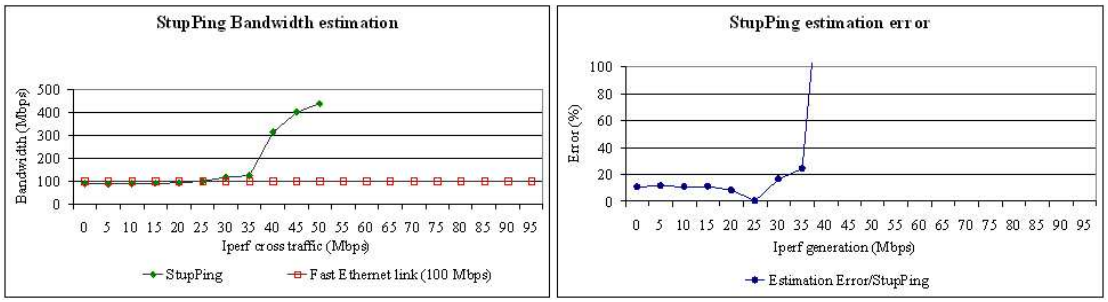


FIG. 10.8 – Bandwidth estimation average and error rate.

	Clink	Pchar	Pathchar	StupPing
Response time (s)	1112	354	2400	16
Probing Time (s)	980	318	2232	14,3
Laas response time (s)	21	49	300	16
Probing traffic Amount (Mbits)	54,3	6,92	110,31	0,132
Intrusiveness (%)	0,0583	0,0217	0,05202	0,009737

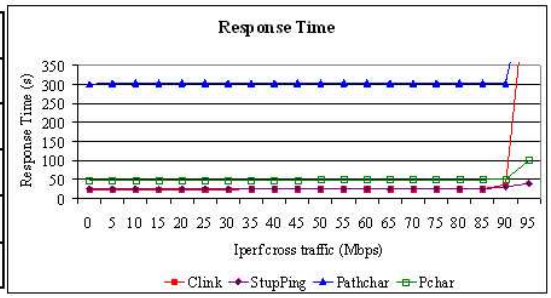


FIG. 10.9 – Evaluation results (left) and response times (right).

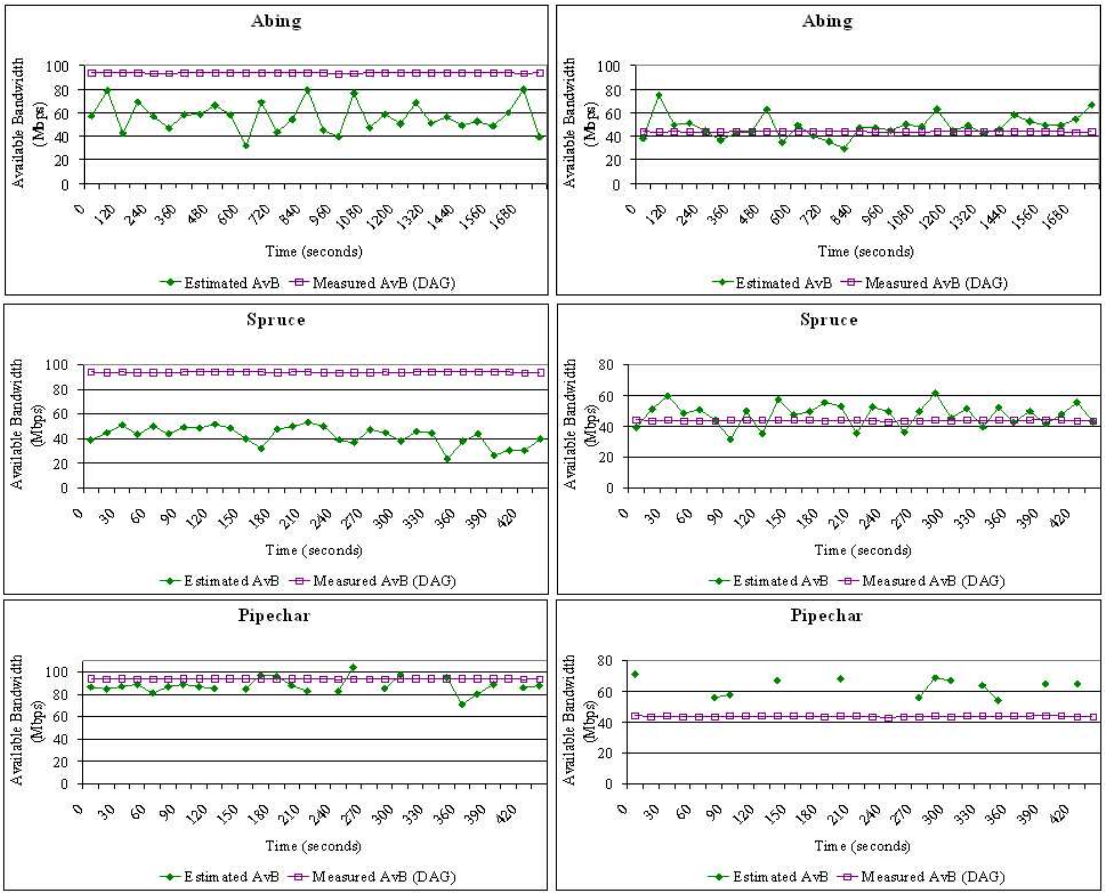


FIG. 10.10 – Bandwidth estimation (Left : no Iperf cross traffic - Right : with 50 Mbps Iperf cross traffic.

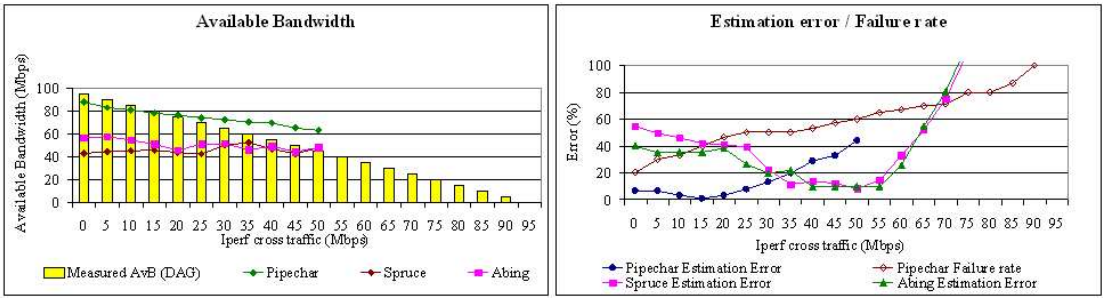


FIG. 10.11 – Bandwidth estimation average (Left) and error/failure rate (right).

	Abing	Spruce	Pipechar
Response time (s)	1,1	11	161
Probing Time (s)	0,96	9,8	141,2
Laas response time (s)	N/A	N/A	161
Probe traffic Amount (Mbits)	0,464	2,34	38,5
Intrusiveness (%)	0,509	0,251	0,286

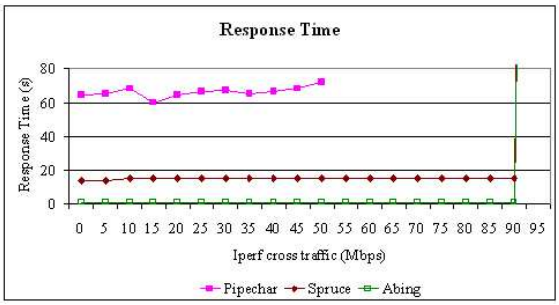


FIG. 10.12 – Tools results (left) and response times (right).

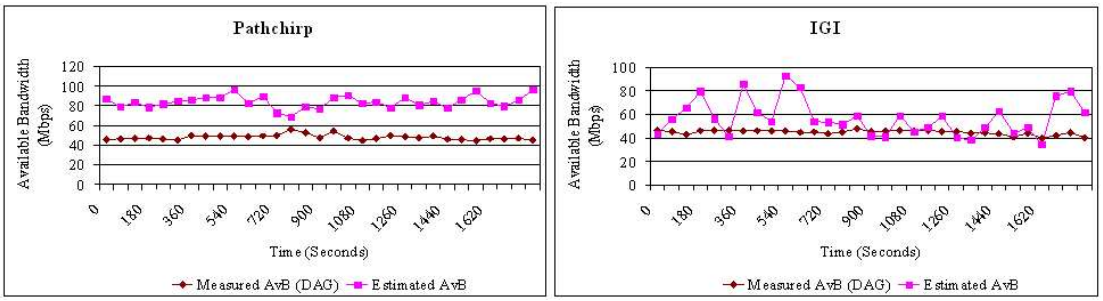


FIG. 10.13 – Available bandwidth estimation with 50 Mbps Iperf cross traffic.

Troisième partie

Sous Projet 5

Modélisation

Chapitre 11

Introduction

11.1 Intégration des flots TCP “souris” et des flots TCP “éléphants”

Les principaux efforts de l'INRIA et de France Telecom dans la partie modélisation du projet Metropolis ont été consacrés à l'étude de l'intégration des souris et des éléphants. L'influence du trafic des souris sur les performances du trafic des éléphants est étudiée. Les deux types de clients coexistent : Les souris (i.e. des flots TCP de moins de 20 paquets IP) sont servies sont servies immédiatement, les éléphants (les autres flots TCP) utilisent la capacité restante du serveur via la discipline *processor sharing* avec un taux de service variable.

On utilise la représentation obtenue dans les travaux de la période précédente pour représenter le trafic des souris : un processus d'Ornstein Uhlenbeck $(X(t))$ de moyenne m. Dans ce cadre les éléphants arrivent suivant un processus de Poisson de paramètre λ et sont servis à l'instant t au taux $\mu g(X(t))$, où g est une fonction décrivant la variabilité du trafic des souris. De cette façon, le processus gaussien $(X(t))$ “dirige” le taux de service. Nous nous sommes intéressés à deux situations pour le cas perturbatif :

1. $g(x) = \mu(1 + \varepsilon h(x - m))$ où h est une fonction arbitraire. Voir les parties 12 et 14.
2. $g(x) = \mu(1 + \varepsilon(x - m))$; voir la partie 13. pour une présentation de ce travail.

L'étude de ce problème est notoirement difficile. Pour cette raison, la variable ε est supposée très petite, de façon à permettre une expansion certaines des caractéristiques en ε. Quand ε vaut 0, c'est bien entendu le cas indépendant, sans interaction avec le processus des souris $(X(t))$.

Résultats.
Partie 12. Le cas (1) a été abordé sous l'angle probabiliste. La fonction de perturbation g est a priori plus générale que dans le cas (1). En revanche les expansions de la durée moyenne d'une période d'occupation et du nombre moyen d'éléphants à l'équilibre n'ont été obtenues que pour les termes en ε et ε^2. L'approche consiste à quantifier précisément l'impact d'une ou deux arrivées et d'un ou deux départs supplémentaires dans une période d'occupation d'une file d'attente stable. Dans ce cas l'expansion en ε et ε^2 permet d'appréhender de façon concrète l'impact de la variabilité de la capacité de service.

Partie 13. Dans le cas (2) une expansion *complète* en ε de la distribution à l'équilibre (par le biais de sa fonction génératrice) du nombre d'éléphants a été obtenue. Le résultat principal montre qu'une forme de résultat de capacité équivalente est vraie pour le terme d'ordre 1 de l'expansion en ε. L'approche est essentiellement analytique, elle repose sur l'itération d'un opérateur fonctionnel.

Partie 14. On se place sous les hypothèses (1). Le développement au deuxième ordre du débit moyen obtenu par un flot TCP éléphant à l'équilibre est obtenu. Il est montré qu'au premier ordre l'approximation de charge réduite est valide : on peut remplacer la file d'attente à service variable par une file d'attente à service constant dont le taux est donné par le taux moyen de service. Au deuxième ordre la situation est très différente, il est montré que l'approxiamtion de charge réduite n'est pas valable et que la variabilité du taux de service a un impact négatif par rapport à une système à taux de service constant. Diverses applications de ces résultats sont ensuite donnés.

Chapitre 12

Perturbation Analysis of a Variable $\mathrm{M/M/1}$ Queue : A Probabilistic Approach

12.1 Introduction

We consider an $M/M/1$ queue with a time varying server rate. We specifically assume that the server rate depends upon a random environment represented by means of a process $(X(t))$, taking values in some (discrete or continuous) state space and assumed to be stationary. The study of this queueing system is motivated by the following engineering problem : Consider a transmission link of a telecommunication network carrying elastic traffic, able to adapt to the congestion level of the network, and a small proportion of traffic, which is unresponsive to congestion. The problem addressed in this paper is to derive quantitative results for estimating the influence of unresponsive traffic on elastic traffic.

In real implementations, elastic traffic is controlled by the so-called transmission control protocol (TCP), which has been designed in order to achieve a fair bandwidth allocation among sufficiently long flows at bottleneck links. If we assume that the link under consideration is the bottleneck, say, the access link to the network, then it is reasonable to assume that bandwidth is distributed among the different competing elastic flows according to the processor sharing discipline (see for instance Massoulié and Roberts [**?**]). Unresponsive traffic is then composed of small data transfers, which are too short to adapt to the congestion level of the network. Throughout the paper, it will be assumed that long flows arrive according to a Poisson process.

With the above modeling assumptions, unresponsive traffic appears for elastic flows as a small perturbation of the available bandwidth. In addition, when there is no unresponsive traffic, owing to the insensitivity property satisfied by the $M/G/1$ processor sharing queue, the number of long flows is identical to the number of customers in an $M/M/1$ queue. Hence, in order to obtain a global system able to describe the behavior of long flows in the presence of unresponsive traffic, we study an $M/M/1$ queue with a time varying server rate, which depends upon unresponsive traffic (for

instance the number of small flows and their bit rate). The problem is then to estimate the impact of unresponsive traffic on the performance of the system. A classical issue is in particular to investigate the validity of the so-called reduced service rate (RSR) approximation, which states that everything happens as if the server rate for long flows were reduced by the mean load of unresponsive traffic.

It is worth noting that queueing systems with time varying server rate have been studied in the literature in many different situations. In Núñez-Queija and Boxma [**?**], the authors consider a queueing system where priority is given to some flows driven by Markov Modulated Poisson Processes (MMPP) with finite state spaces and the low priority flows share the remaining server capacity according to the processor sharing discipline. By assuming that arrivals are Poisson and service times are exponentially distributed, the authors solve the system by means of matrix analysis methods. Similar models have been investigated in Núñez-Queija [**?**, **?**] by still using the quasi-birth and death process associated with the system and a matrix analysis. In this setting, the characteristics of the queue at equilibrium are expressed in terms of the spectral quantities of some matrices leading to potential numerical applications.

The integration of elastic and streaming flows has been studied by Delcoigne *et al* [**?**], where stochastic bounds for the mean number of active flows have been established. More recently, priority queueing systems with fast dynamics, which can be described by means of quasi-birth and death processes, have been studied via a perturbation analysis of a Markov chain by Altman *et al* [**?**]. Boxma and Kurkova [**?**] studies the tail distributions of an $M/M/1$ with two service rates.

Getting qualitative results for queueing systems with variable service rates is really difficult, to study, for example, the impact of the variability of the service rate on the performances of the system. At the intuitive level, it is quite well known that the variability deteriorates them but, rigorously speaking, few results are available. The main objective of this paper is to get some insight on these phenomena by considering a slightly perturbated system. As it will be seen, deriving such an expansion is already quite technical. The benefits of the approach are presented in Section 14.3 where, in a quite general setting for the perturbating environment, several qualitative and quantitative results are established.

In this paper, it is assumed that the server rate of the $M/M/1$ queue is equal at time t to $\mu + \varepsilon p(X(t))$ for some function p, where $(X(t))$ is the process describing the environment affecting the service rate. In Fricker *etal.* [**?**], it has been assumed that the process $(X(t))$ is a diffusion process and that $p(x) = -x$. In this paper, the perturbation function p is quite general and the environment process $(X(t))$ is only assumed to be stationary and Markovian. While in Fricker *etal.* [**?**], only the number of customers in the queue has been considered, the impact of the perturbation on the busy period duration is investigated in details in the present paper. The mean busy period duration is expanded with respect to the parameter ε, which quantifies the magnitude of the perturbation. As far as the first order is concerned, the RSR approximation is valid : the time-varying server queue is identical to an equivalent $M/M/1$ queue with a fixed service rate equal to the average service rate $\mu + \varepsilon\mathbb{E}(p(X(0))$. The analysis of the second order is much more intricate ; the correlations of the process $(X(t))$ play a key role and, consequently, the RSR approximation is no more valid.

12.2 Model description

Let $\widetilde{L}(t)$ be the number of customers at time t in a $M/M/1$ queue with arrival rate λ and service rate μ. We will denote this queue by $\widetilde{L}$.

Let $L(t)$ be the number of customers at time t in a $M/M/1$ queue with arrival rate λ and service rate which depends on a stationary diffusion process $(X(t))$. We will denote this queue by L. If $L(t) = l$ and $X(t) = x$ at time t, then the transitions of $(L(t))$ are given by

$$l \to \begin{cases} l+1 & \text{with rate } \lambda \\ l-1 & \text{with rate } \mu + \varepsilon p(X(t)) \end{cases}$$

for some function $p(x)$ on $\mathbb{R}$ and some small $\varepsilon \geq 0$.

For fixed $X(t)$, we can define the departures rates of queue L

$$\mu\left(1 - \frac{\varepsilon p(X(t))^-}{\mu}\right) + \varepsilon p(X(t))^+$$

where $p(x)^- = \max(0, -x)$ and $p(x)^+ = \max(0, x)$. In this expression, we assume that $\mu(1 - \varepsilon p(X(t))^-/\mu)$ and $\varepsilon p(X(t))^+$ are the rates of two independent Poisson processes $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. The process $\mathcal{N}_1$ is obtained by colouring a Poisson process $\mathcal{N}$ with rate μ. We call the points of $\mathcal{N}$ which are colouring with probability $\varepsilon p(X(t))^-/\mu$ the *marked departures* (departures that occur in $\widetilde{L}$ and not in L) and represent its sequence by $(\bar{t}_n)$. We call the points (t_n) of $\mathcal{N}_2$ the *additional departures* times (departures that only occur in L) of the queue.

12.3 Busy period (ε term)

Suppose that a busy period with one customer starts at time 0 in queues L and $\widetilde{L}$. Let T and $\widetilde{T}$ denote the busy periods of the queues L and $\widetilde{L}$, respectively. In this section we will compute the coefficient of ε for the difference between the expected values of both busy periods.

Only one additional jump

If there is only one additional jump and no marked jumps in $(0, T)$ then at time T, the queue L is empty and queue $\widetilde{L}$ is with one customer. Conditioning on this event, by the strong Markov property the difference between T and $\widetilde{T}$ is equal to the length of a busy period of $\widetilde{L}$. Then in the expansion of $\mathbb{E}[T - \widetilde{T}]$ we have the term

$$-\mathbb{E}[\mathbf{1}_{\{t_1 < T < t_2\}} \mathbf{1}_{\{T < \bar{t}_1\}}]\mathbb{E}[B_1] \tag{12.1}$$

where B_i is the length of a busy period in $\widetilde{L}$ which starts with i customers, or equivalently

$$-\mathbb{E}[\mathbf{1}_{\{t_1 < \widetilde{T}\}} \mathbf{1}_{\{t_2 > t_1 + B_{\widetilde{L}(t_1)-1}\}} \mathbf{1}_{\{\bar{t}_1 > t_1 + B_{\widetilde{L}(t_1)-1}\}}]\mathbb{E}[B_1]. \tag{12.2}$$

The next result gives the ε term of (12.2) with $a_+ = \mathbb{E}[p(X(0))^+]$.

Proposition 4. *The ε term of $\mathbb{E}[T-\widetilde{T}]$ from one additional jump is*

$$-\varepsilon\frac{a_+}{(\mu-\lambda)^2}. \tag{12.3}$$

Proof : Note that equation (12.2) is equal to

$$-\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}(1-\mathbf{1}_{\{t_2<t_1+B_{\widetilde{L}(t_1)-1}\}})(1-\mathbf{1}_{\{\bar{t}_1<t_1+B_{\widetilde{L}(t_1)-1}\}})]\mathbb{E}[B_1] \tag{12.4}$$

where the only term in ε is given by the expansion of $-\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}]\mathbb{E}[B_1]$. Thus, conditioning on the probability density function of t_1 and assuming that the process $(X(t))$ is fixed,

$$\begin{aligned}\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}] &= \int_0^\infty \varepsilon p(X(s))^+\exp\left(-\varepsilon\int_0^s p(X(v))^+\,dv\right)\mathbb{E}[\mathbf{1}_{\{s<\widetilde{T}\}}]ds \\ &= \varepsilon\int_0^\infty p(X(s))^+\mathbb{E}[\mathbf{1}_{\{s<\widetilde{T}\}}]\,ds + O(\varepsilon^2).\end{aligned} \tag{12.5}$$

Taking expectation on $(X(t))$ and by stationarity of this process,

$$\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}] = \varepsilon\mathbb{E}[p(X(0))^+]\mathbb{E}[B_1] + O(\varepsilon^2).$$

where $\mathbb{E}[B_1] = 1/(\mu-\lambda)$ (see Appendix). □

Only one marked jump

Suppose that there is only one marked jump and no additional jumps during the busy period of $\widetilde{L}$. In this case, at time $\widetilde{T}$, the queue L is with one customer and queue $\widetilde{L}$ is empty. If there are no marked and additional jumps during $(\widetilde{T},\widetilde{T}+B_1)$ in L then the difference between both busy periods is equal to B_1. In the expansion of $\mathbb{E}[T-\widetilde{T}]$ we have the term

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1\le\widetilde{T}\}}\mathbf{1}_{\{\bar{t}_2>\widetilde{T}+B_1\}}\mathbf{1}_{\{t_1>\widetilde{T}+B_1\}}B_1]. \tag{12.6}$$

where B_1 is independent of the random variables $\bar{t}_1$, $\bar{t}_2$, t_1 and $\widetilde{T}$. The next result gives the ε term of (12.6) with $a_- = \mathbb{E}[p(X(0))^-]$.

Proposition 5. *The ε term of $\mathbb{E}[T-\widetilde{T}]$ from one marked jump is*

$$\varepsilon\frac{a_-}{(\mu-\lambda)^2}. \tag{12.7}$$

Proof : The expression (12.6) can be written as

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1\le\widetilde{T}\}}(1-\mathbf{1}_{\{\bar{t}_2<\widetilde{T}+B_1\}})(1-\mathbf{1}_{\{t_1<\widetilde{T}+B_1\}})B_1] \tag{12.8}$$

where the only term in ε is given by the expansion of $\mathbb{E}[\mathbf{1}_{\{\bar{t}_1\le\widetilde{T}\}}]\mathbb{E}[B_1]$. To compute $\mathbb{E}[\mathbf{1}_{\{\bar{t}_1\le\widetilde{T}\}}]$, let $(D_1,\ldots,D_N)$ denote the sequence of departures times in the busy period $\widetilde{T}$ and assume that $((X(t))$ is fixed. Thus,

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1\le\widetilde{T}\}}] = \mathbb{E}\left[\sum_{i=1}^N\frac{\varepsilon p(X(D_i))^-}{\mu}\prod_{j=1}^{i-1}\left(1-\frac{\varepsilon p(X(D_j))^-}{\mu}\right)\right]$$

which has the following term in ε after taking expectation on $(X(t))$

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \leq \widetilde{T}\}}] = \frac{\varepsilon}{\mu}\mathbb{E}[p(X(0))^-]\mathbb{E}[N].$$

Using that $\mathbb{E}[N] = \mu/(\mu - \lambda)$ (see Appendix) it ends the prove. □

Conclusion

In the expansion of the busy period of L, the term in ε is given by the two events consisting in only one marked or only one additional jump during the busy period of $\widetilde{L}$. The next proposition follows from Propositions 4 and 5 with $\mathbb{E}[\widetilde{T}] = 1/(\mu - \lambda)$ (see Appendix).

Proposition 6.

$$\mathbb{E}[T] = \frac{1}{\mu - \lambda} + \varepsilon\frac{a_- - a_+}{(\mu - \lambda)^2} + O(\varepsilon^2). \tag{12.9}$$

12.4 Queue length (ε term)

Suppose that a busy period starts at time 0 with one customer in queues L and $\widetilde{L}$ and let $\tau = \inf\{t > 0, L(t) = \widetilde{L}(t) = 0\}$ be the first time that both queues are empty after time 0. In this section we compute the coefficient of ε for the difference between the mean numbers of customers in each queue.

Only one additional jump

It is easy to see that, if there is one additional jumps and no marked jumps during $(0, \tau)$ then, in the expansion of $\mathbb{E}[\int_0^\tau (L(t) - \widetilde{L}(s))\, ds]$, we have the term

$$-\mathbb{E}[\mathbf{1}_{\{t_1 < \widetilde{T}\}}\mathbf{1}_{\{t_2 > \widetilde{T}\}}\mathbf{1}_{\{\bar{t}_1 > \widetilde{T}\}}(\widetilde{T} - t_1)] \tag{12.10}$$

and $\tau = \widetilde{T}$.

Proposition 7. *The ε term of $\mathbb{E}[\int_0^\tau (L(t) - \widetilde{L}(s))\, ds]$ from one additional jump is*

$$-\varepsilon\frac{a_+\mu}{(\mu - \lambda)^3}. \tag{12.11}$$

Proof : From (12.10), the term in ε is given by the expansion of

$$-\mathbb{E}[\mathbf{1}_{\{t_1 < \widetilde{T}\}}(\widetilde{T} - t_1)].$$

Conditioning on the density of t_1 and assuming that $(X(t))$ is fixed,

$$-\mathbb{E}[\mathbf{1}_{\{t_1 < \widetilde{T}\}}(\widetilde{T} - t_1)] = -\mathbb{E}\left[\int_0^{\widetilde{T}} \varepsilon p(X(s))^+ \exp\left(-\varepsilon\int_0^s p(X(v))^+\, dv\right)(\widetilde{T} - s)\, ds\right]$$

$$= -\varepsilon\mathbb{E}[X(0))^+]\frac{\mathbb{E}[\widetilde{T}^2]}{2} + O(\varepsilon^2)$$

after integrating with respect to $(X(t))$. Using that $\mathbb{E}[\widetilde{T}^2] = 2\mu/(\mu-\lambda)^3)$ (see Appendix) the result follows. □

Let $\mathbb{E}_\pi[.]$ (resp. $\mathbb{E}_{\widetilde{\pi}}[.]$) denote the expected value with respect to the stationary distribution of $(L(t))$ (resp. $(\widetilde{L}(t))$). The difference between the means numbers of customers in each queue is

$$\mathbb{E}_\pi[L(0)] - \mathbb{E}_{\widetilde{\pi}}[\widetilde{L}(0)] = \frac{1}{\lambda^{-1} + \mathbb{E}[\tau]} \mathbb{E}\left[\int_0^\tau (L(s) - \widetilde{L}(s))\, ds\right]. \tag{12.12}$$

The ε term of this expression for only one additional jump in $(0,\tau)$ is given by the result of Proposition 7 divided by $\lambda^{-1} + \mathbb{E}[\widetilde{T}] = \mu/(\lambda(\mu-\lambda))$.

Proposition 8. *The ε term of $\mathbb{E}_\pi[L(0)] - \mathbb{E}_{\widetilde{\pi}}[\widetilde{L}(0)]$ from one additional jump is*

$$-\varepsilon \frac{a_- \lambda}{(\mu-\lambda)^2}. \tag{12.13}$$

Only one marked jump

In the case where there is only one marked jump and no additional jumps in $(0,\tau)$ then in the expansion of $\mathbb{E}[\int_0^\tau L(t) - \widetilde{L}(s)\, ds]$ we have the term

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \le \widetilde{T}\}} \mathbf{1}_{\{\bar{t}_2 > \widetilde{T} + B_1\}} \mathbf{1}_{\{t_1 > \widetilde{T} + B_1\}} (\widetilde{T} - \bar{t}_1 + B_1)] \tag{12.14}$$

and $\tau = T$.

Proposition 9. *The ε term of $\mathbb{E}[\int_0^\tau (L(t) - \widetilde{L}(s))\, ds]$ from one marked jump is*

$$\varepsilon \frac{a_- \mu}{(\mu-\lambda)^3}. \tag{12.15}$$

Proof : The term in ε from (12.14) is given by the expansion of

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \le \widetilde{T}\}} (\widetilde{T} - t_1 + B_1)] = \mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \le \widetilde{T}\}} (\widetilde{T} - t_1)] + \mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \le \widetilde{T}\}}] \mathbb{E}[B_1]$$

where the last term in the second member has been computed in the prove of Proposition 5 and is given by $\varepsilon a_-/(\lambda-\mu)^2$. As in that prove, let $(D_1, \ldots, D_N)$ denote the sequence of departures times in the busy period $\widetilde{T}$. Thus,

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \le \widetilde{T}\}} (\widetilde{T} - t_1)] = \mathbb{E}\left[\sum_{i=1}^N \frac{\varepsilon p(X(D_i))^-}{\mu} (\widetilde{T} - D_i) \prod_{j=1}^{i-1} \left(1 - \frac{\varepsilon p(X(D_j))^-}{\mu}\right)\right]$$

and the ε term after taking expectation on $(X(t))$ is given by

$$\frac{\varepsilon}{\mu} \mathbb{E}[p(X(0))^-] (\mathbb{E}[N\widetilde{T}] - \mathbb{E}[D])$$

where $D = \sum_{i=1}^{N} D_i$ is the sum of the departures times in the busy period $\widetilde{T}$. Using that (see Appendix)

$$\mathbb{E}[D] = \frac{\mu^2}{(\mu-\lambda)^3} \qquad \text{and} \qquad \mathbb{E}[N\widetilde{T}] = \frac{\mu(\lambda+\mu)}{(\mu-\lambda)^3}$$

simple algebra proves the result. □

The ε term of (12.12) for only one marked jump in $(0,\tau)$ is obtained from Proposition 9 and using that $\mathbb{E}[\tau] = \mathbb{E}[T]$ which is given by $\mathbb{E}[\widetilde{T}] + O(\varepsilon)$.

Proposition 10. *The ε term of $\mathbb{E}_\pi[L(0)] - \mathbb{E}_{\widetilde{\pi}}[\widetilde{L}(0)]$ from one marked jump is*

$$\varepsilon \frac{a_- \lambda}{(\mu-\lambda)^2}. \tag{12.16}$$

Conclusion

Using the results above we can state the following proposition for the expansion of the mean numbers of customers in queue L where we have used $\mathbb{E}_{\widetilde{\pi}}[\widetilde{L}(0)] = \lambda/(\mu-\lambda)$ (see Appendix).

Proposition 11.

$$\mathbb{E}_\pi[L(0)] = \frac{\lambda}{\mu-\lambda} + \varepsilon \frac{(a_- - a_+)\lambda}{(\mu-\lambda)^2} + O(\varepsilon^2).$$

12.5 Busy Period (ε^2 term)

Only additional jumps

If there is only two additional jump and no marked jumps in $(0,T)$, then the difference between $T - \widetilde{T}$ is equal to the busy period of $\widetilde{L}$ which starts with two customers. Similarly, as for only one additional jump, in the expansion of $\mathbb{E}[T-\widetilde{T}]$ we have the term

$$-\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{t_2<t_1+B_{\widetilde{L}(t_1)-1}<t_3\}}\mathbf{1}_{\{t_1>t_1+B_{\widetilde{L}(t_1)-1}\}}]\mathbb{E}[B_2]. \tag{12.17}$$

Through the rest of the paper, to simplify the exposition, we will present just the jumps that occur which are responsible by the ε^2 term. Thus, (12.17) simplify to

$$-\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{t_2<t_1+B_{\widetilde{L}(t_1)-1}\}}]\mathbb{E}[B_2]. \tag{12.18}$$

Moreover, from (12.2) (see also (12.4)), the ε^2 term of $\mathbb{E}[T-\widetilde{T}]$ involving just additional jumps is given from the expansion of

$$-\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}]\mathbb{E}[B_1] + \mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{t_2<t_1+B_{\widetilde{L}(t_1)-1}\}}]\mathbb{E}[B_1]. \tag{12.19}$$

Summing with (12.18), the ε^2 term of only additional jumps is given by

$$-\mathbb{E}[B_1]\left(\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}] + \mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{t_2<t_1+B_{\widetilde{L}(t_1)-1}\}}]\right).$$

Proposition 12. *The ε^2 term of $\mathbb{E}[T-\widetilde{T}]$ involving only additional jumps is*

$$\varepsilon^2\frac{1}{\mu-\lambda}\left(+\mathbb{E}\left[\int_0^{\widetilde{T}}\int_0^s \mathbb{E}[p(X(0))^+p(X(v))^+]\,dvds\right]\right.$$
$$\left.-\frac{1}{\lambda}\sum_{i=1}^{\infty}\rho^i\mathbb{E}\left[\int_0^{B_{i-1}}\mathbb{E}[p(X(0))^+p(X(v))^+]\,dv\right]\right). \tag{12.20}$$

Proof : From (12.5), the ε^2 term of $\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}]$ is

$$\int_0^{\infty}\varepsilon p(X(s))^+\left(-\varepsilon\int_0^s p(X(v))^+\,dv\right)\mathbb{E}[\mathbf{1}_{\{s<\widetilde{T}\}}]ds.$$

Taking expectation on $(X(t))$ and by the stationarity of the process we obtain

$$-\varepsilon^2\int_0^{\infty}\left(\int_0^s\mathbb{E}[p(X(0))^+p(X(v))^+]\,dv\right)\mathbb{E}[\mathbf{1}_{\{s<\widetilde{T}\}}]ds$$

and by the nonnegative of the functions is equal to first term of (12.20).

To compute $\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{t_2<t_1+B_{\widetilde{L}(t_1)-1}\}}]$ we conditionate on $t_1=s$ and assuming that $(X(t))$ is fixed

$$\begin{aligned}\mathbb{E}[\mathbf{1}_{\{s<\widetilde{T}\}}\mathbf{1}_{\{t_2<s+B_{\widetilde{L}(s)-1}\}}|t_1=s] &= \sum_{i=1}^{\infty}\mathbb{P}(t_2<s+B_{i-1}|t_1=s)\mathbb{P}(\widetilde{L}(s)=i,\widetilde{T}>s)\\ &= \sum_{i=1}^{\infty}\left(1-\mathbb{E}\left[\exp\left(-\varepsilon\int_s^{s+B_{i-1}}p(X(v))^+\,dv\right)\right]\right)\mathbb{P}(\widetilde{L}(s)=i,\widetilde{T}>s)\\ &= \varepsilon\sum_{i=1}^{\infty}\mathbb{E}\left[\int_s^{s+B_{i-1}}p(X(v))^+\,dv\right]\mathbb{P}(\widetilde{L}(s)=i,\widetilde{T}>s)+O(\varepsilon^2)\end{aligned}$$

Integrating with respect to the density of t_1 and using only its ε term

$$\int_0^{\infty}\varepsilon p(X(s))^+\varepsilon\sum_{i=1}^{\infty}\mathbb{E}\left[\int_s^{s+B_{i-1}}p(X(v))^+\,dv\right]\mathbb{P}(\widetilde{L}(s)=i,\widetilde{T}>s)\,ds+O(\varepsilon^2)$$

Taking expectation on $(X(t))$,

$$\varepsilon^2\sum_{i=1}^{\infty}\mathbb{E}\left[\int_0^{B_{i-1}}\mathbb{E}[p(X(0))^+p(X(v))^+]\,dv\right]\mathbb{E}\left[\int_0^{\widetilde{T}}\mathbf{1}_{\{\widetilde{L}(s)=i\}}ds\right]$$

where the last term is the expected time that queue $\widetilde{L}$ spend in state i during a busy period which is equal to ρ^i/λ. □

Corollaire 2. *Replacing $X(v)$ by $X(\delta v)$ in (12.20) and letting δ tends to infinity, the ε^2 term of only additional jumps is*

$$\varepsilon^2\frac{a_+^2}{(\mu-\lambda)^3}. \tag{12.21}$$

Proof : Letting δ tends to infinity and using that

$$\mathbb{E}[p(X(0))^+ p(X(\delta v))^+] \to a_+^2$$

it is easy to see that (12.20) is equal to

$$a_+^2 \mathbb{E}[B_1] \left(+ \frac{\mathbb{E}[\widetilde{T}^2]}{2} - \frac{1}{\lambda} \sum_{i=1}^{\infty} \rho^i \mathbb{E}[B_{i-1}] \right).$$

Using the results in Appendix, simple algebra proves the result. □

Only marked jumps

If there is only one marked jump in $(0, \widetilde{T})$ and only one marked jump in $(\widetilde{T}, \widetilde{T} + B_1)$ and no marked jumps during $(0, \widetilde{T} + B_1)$, then the difference between both busy periods is $B_1 + B_1^{'}$ where $B_1^{'}$ is independent and identically distributed as B_1. Then in the expansion $\mathbb{E}[T - \widetilde{T}]$ we have the ε^2 of

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 < \widetilde{T} < \bar{t}_2\}} \mathbf{1}_{\{\bar{t}_2 < \widetilde{T} + B_1\}} (B_1 + B_1^{'})]. \tag{12.22}$$

Another possibility is to have two marked jumps during $(0, \widetilde{T})$. In this case the ε^2 term is given by the expansion of

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_2 < \widetilde{T}_1\}}] \mathbb{E}[B_2]. \tag{12.23}$$

From (12.6) (see also (12.8)), the ε^2 involving just additional jumps are

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \leq \widetilde{T}\}}] \mathbb{E}[B_1] - \mathbb{E}[\mathbf{1}_{\{\bar{t}_1 \leq \widetilde{T}\}} \mathbf{1}_{\{\bar{t}_2 < \widetilde{T} + B_1\}}] \mathbb{E}[B_1] \tag{12.24}$$

and summing with (12.22) and (12.23) (see appendix for the proof) the ε^2 of only additional jumps is given by the expansion of

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 < \widetilde{T} < \bar{t}_2\}} \mathbf{1}_{\{\bar{t}_2 < \widetilde{T} + B_1\}}] \mathbb{E}[B_1'].$$

Let $(D_1, \ldots, D_N)$ and $(D_1', \ldots, D_{N'}')$ be the sequences of the departures times of two independent busy periods $\widetilde{T}$ and B_1 beginning at time 0.

Proposition 13. *The ε^2 of only $\mathbb{E}[T - \widetilde{T}]$ involving only marked jumps is*

$$\frac{\varepsilon^2}{\mu^2(\mu - \lambda)} \mathbb{E} \left[\sum_{i=1}^{N} \sum_{j=1}^{N'} p(X(0))^- p(X(\widetilde{T} - D_i + D_j'))^- \right]. \tag{12.25}$$

Proof : Assuming that $(X(t))$ is fixed, then $\mathbb{E}[\mathbf{1}_{\{\bar{t}_1 < \widetilde{T} < \bar{t}_2\}} \mathbf{1}_{\{\bar{t}_2 < \widetilde{T} + B_1\}}]$ is equal to

$$\mathbb{E}\Bigg[\sum_{i=1}^{N} \frac{\varepsilon p(X(D_i))^-}{\mu} \prod_{j=1, j \neq i}^{N} \left(1 - \frac{\varepsilon p(X(D_j))^-}{\mu} \right) \times \sum_{k=1}^{N'} \frac{\varepsilon p(X(\widetilde{T} + D_k'))^-}{\mu} \prod_{l=1}^{k-1} \left(1 - \frac{\varepsilon p(X(\widetilde{T} + D_l'))^-}{\mu} \right) \Bigg].$$

Taking the expectation on $(X(t))$ and by the stationarity of the process the coefficient of ε^2 is

$$\frac{1}{\mu^2}\mathbb{E}\left[\sum_{i=1}^{N}\sum_{k=1}^{N'}p(X(0))^{-}p(X(\widetilde{T}-D_i+D'_k))^{-}\right].$$

□

If in the evaluation of (12.25), we replace $\mathbb{E}[p(X(0))^{-}p(X(t))^{-}]$ by $\mathbb{E}[p(X(0))^{-}p(X(\delta t))^{-}]$ and let δ tends to infinity, we have the following result.

Corollaire 3. *The ε^2 of only marked jumps is*

$$\varepsilon^2\frac{a_{-}^2}{(\mu-\lambda)^3}. \tag{12.26}$$

Proof : Letting δ tends to infinity and using that

$$\mathbb{E}[p(X(0))^{+}p(X(v))^{+}]\rightarrow a_{-}^2$$

it is easy to see that (12.25) is equal to

$$\frac{\varepsilon^2 a_{-}^2}{\mu^2(\mu-\lambda)}\mathbb{E}[N]^2.$$

Replacing by the results in the Appendix, the prove is complete. □

One additional and one marked

Suppose that there is only one marked jump and no additional jumps in $(0,\widetilde{T}]$. Then if there is only one additional jump and no marked jumps during $(\widetilde{T},\widetilde{T}+B_1]$, the difference between both busy periods is

$$t_1+B_{\tilde{L}(t_1)-1}-\widetilde{T}. \tag{12.27}$$

Since this expression is difficult to evaluate we give an alternative. From subsection 2.1, we know that the difference between B_1 and $T-\widetilde{T}$ when there is only one additional jump during $(\widetilde{T},T)$ is equal to a busy period which we denote by B'_1. In this case, the ε^2 is given by the expansion of

$$\mathbb{E}[\mathbf{1}_{\{\bar{t}_1<\widetilde{T}\}}\mathbf{1}_{\{\widetilde{T}<t_1<\widetilde{T}+B_1\}}(B_1-B_1^{'})]. \tag{12.28}$$

From (12.2) and (12.6), the ε^2 involving jumps of different type are given by the expansion of

$$-\mathbb{E}[\mathbf{1}_{\{\bar{t}_1<\widetilde{T}_1\}}\mathbf{1}_{\{t_1<\widetilde{T}_1+B_1\}}B_1]+\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{\bar{t}_1<t_1+B_{\tilde{L}(t_1)-1}\}}]\mathbb{E}[B_1] \tag{12.29}$$

and summing with (12.28) (see the Appendix for the proof), the ε^2 of one additional and one marked jump is given by the expansion of

$$\mathbb{E}[B'_1]\left(-\mathbb{E}[\mathbf{1}_{\{\bar{t}_1<\widetilde{T}\}}\mathbf{1}_{\{t_1<\widetilde{T}+B_1\}}]+\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{\bar{t}_1<t_1+B_{\tilde{L}(t_1)-1}\}}]\right).$$

Before we give the ε^2 term of this expansion, we need to the concept of cycles in the busy period $\widetilde{T}$ which was introduced in the proof of *** (see the Appendix). Shortly, the number of cycles are geometric distributed with parameter $\mu/(\lambda+\mu)$, and each cycle k is the sum of a exponential time A_k with parameter $\mu+\lambda$ and a busy period B_1^k. After the last cycle i it remains an exponential time A_{i+1} with parameter $\mu+\lambda$. Let C_k be the time that cycle k finished and $(D_1^k,\ldots,D_{N^k}^k)$ the departures times in B_1^k.

Proposition 14. *The ε^2 term of $\mathbb{E}[T-\widetilde{T}]$ from one additional and one marked jumps is*

$$\frac{\varepsilon^2}{\mu(\mu-\lambda)}\left(\mathbb{E}\left[\sum_{i=1}^{N}\int_0^{\widetilde{T}+B_1} p(X(D_i))^- p(X(s))^+\,ds\right]\right.$$
$$\left.+\sum_{i=1}^{\infty}\left(\frac{\mu}{\lambda+\mu}\right)\left(\frac{\lambda}{\lambda+\mu}\right)^i\mathbb{E}\left[\sum_{k=1}^{i}\sum_{l=1}^{N^k}\int_{C_{k-1}+A_k}^{\widetilde{T}} p(X(D_l^k)^- p(X(s))^+\,ds\right]\right) \tag{12.30}$$

Proof : Assuming that $(X(t))$ is fixed, $\mathbb{E}[\mathbf{1}_{\{\bar{t}_1<\widetilde{T}\}}\mathbf{1}_{\{t_1<\widetilde{T}+B_1\}}]$ is given by

$$\mathbb{E}\left[\sum_{i=1}^{N}\frac{\varepsilon p(X(D_i))^-}{\mu}\prod_{j=1}^{i-1}\left(1-\frac{\varepsilon p(X(D_j))^-}{\mu}\right)\right.$$
$$\left.\times\int_0^{\widetilde{T}+B_1}\varepsilon p(X(s))^+\exp\left(-\varepsilon\int_0^s p(X(v))^+\,dv\right)\right]$$

From its expansion it is to see that the coefficient of ε^2 is equal to the first term of (12.30) in parenthesis after taking expectation on $(X(t))$.

To compute $\mathbb{E}[\mathbf{1}_{\{t_1<\widetilde{T}\}}\mathbf{1}_{\{\bar{t}_1<t_1+B_{\widetilde{L}(t_1)-1}\}}]$, we will conditionate on the number of cycles in the busy period $\widetilde{T}$. For $\bar{t}_1$ occurred during $(0,\widetilde{T})$, it has to be in one B_1^k. Then given $\bar{t}_1$, for $t_1+B_{\widetilde{L}(t_1)-1}>\bar{t}_1$, t_1 has to occur in the same B_1^k or in another cycle after k. As a consequence the probability will be

$$\sum_{i=1}^{\infty}\left(\frac{\mu}{\lambda+\mu}\right)\left(\frac{\lambda}{\lambda+\mu}\right)^i\mathbb{E}\left[\sum_{k=1}^{i}\left(\sum_{l=1}^{N^k}\frac{\varepsilon p(X(D_l^k))^-}{\mu}\prod_{n=1}^{l-1}\left(1-\frac{\varepsilon p(X(D_n^k))^-}{\mu}\right)\right.\right.$$
$$\left.\left.\times\prod_{j=1}^{k-1}\prod_{m=1}^{N^j}\left(1-\frac{\varepsilon p(X(D_m^j))^-}{\mu}\right)\right)\int_{C_{k-1}+A_k}^{\widetilde{T}}\varepsilon p(X(s))^+\exp\left(-\varepsilon\int_0^s p(X(v))^+\,dv\right)\right]ds$$

where the coefficient of ε^2 after taking expectation for $(X(t))$ is

$$\sum_{i=1}^{\infty}\left(\frac{\mu}{\lambda+\mu}\right)\left(\frac{\lambda}{\lambda+\mu}\right)^i\mathbb{E}\left[\sum_{k=1}^{i}\sum_{l=1}^{N^k}\frac{p(X(D_l^k))^-}{\mu}\int_{C_{k-1}+A_k}^{\widetilde{T}}p(X(s))^+\,dv\right]$$

□

If in the evaluation of (12.30), we replace $\mathbb{E}[p(X(0))^- p(X(t))^+]$ (resp. $\mathbb{E}[p(X(0))^+ p(X(t))^-]$) by $\mathbb{E}[p(X(0))^- p(X(\delta t))^+]$ (resp. $\mathbb{E}[p(X(0))^+ p(X(\delta t))^-]$) and let δ tends to infinity, we have the following result.

Corollaire 4. *The ε^2 term of one additional and one marked jump is*

$$-\frac{2a_+a_-}{(\mu-\lambda)^3}. \tag{12.31}$$

Proof : For $t>0$, letting δ tends to infinity and using that

$$\mathbb{E}[p(X(0))^- p(X(\delta t))^+] \to a_- a_+, \qquad \mathbb{E}[p(X(0))^+ p(X(\delta t))^-] \to a_+ a_-,$$

(12.30) is equal to

$$\begin{aligned}
&\frac{\varepsilon^2 a_+a_-}{\mu(\mu-\lambda)}\Bigg(-\mathbb{E}[N(\widetilde{T}+B_1)] \\
&+\sum_{i=1}^{\infty}\left(\frac{\mu}{\lambda+\mu}\right)\left(\frac{\lambda}{\lambda+\mu}\right)^i \mathbb{E}\left[\sum_{k=1}^{i} N^k(B_1^k+C_i-C_k+A_{i+1})\right]\Bigg) \\
&=\frac{\varepsilon^2 a_+a_-}{\mu(\mu-\lambda)}\Bigg(-\mathbb{E}[N\widetilde{T}]-\mathbb{E}[N]\mathbb{E}[B_1] \\
&+\sum_{i=1}^{\infty}\left(\frac{\mu}{\lambda+\mu}\right)\left(\frac{\lambda}{\lambda+\mu}\right)^i \sum_{k=1}^{i}\mathbb{E}[N^1B_1^1]+\mathbb{E}[N^1](\mathbb{E}[A_i+B_1^i](i-k)+\mathbb{E}[A_{i+1}])\Bigg)
\end{aligned}$$

where the last term is due to the cycles being independent and identically distributed. Replacing by the results in the Appendix, simple algebra completes the prove. □

Conclusion

Using the results above we can state the following result for the coefficient of ε^2 in the expansion of $\mathbb{E}[T-\widetilde{T}]$ using limiting results for $(X(t))$.

Corollaire 5. *The ε^2 term of $\mathbb{E}[T-\widetilde{T}]$ is*

$$\frac{(a_+-a_-)^2}{(\mu-\lambda)^3}.$$

12.6 The queue length in the perturbated model (ε^2 term)

Proposition 15. *The mean of difference between the number of customers in the integrated model and the reference model on a cycle has the following expansion in ε*

$$\mathbb{E}_1\left(\int_0^\tau \left(L(s)-\tilde{L}(s)\right)ds\right)=\frac{a_--a_+}{\mu^2(1-\rho)^3}\varepsilon+\beta\varepsilon^2+O(\varepsilon^3)$$

where

$$\beta = \frac{1}{\mu-\lambda}\mathbb{E}\left(\int_0^{\tilde{T}} \mathbb{E}[p(X(0))^+ p(X(u))^+](t-u)du\right) \tag{12.32}$$

$$+ \frac{1}{\mu^2(\mu-\lambda)}\mathbb{E}\left(\sum_{i,j=1,i<j}^{N} p(X(D_i))^- p(X(D_j))^-\right) \tag{12.33}$$

$$+ \frac{1}{\mu^2}\mathbb{E}\left(\sum_{i=1}^{N}\sum_{j=1}^{N_1} p(X(D_i))^- p(X(\tilde{T}+D_{1,j}))^-(B_1 - D_{1,j} + B_1')\right) \tag{12.34}$$

$$- \frac{1}{\mu(\mu-\lambda)}\mathbb{E}\left(\int_0^{\tilde{T}} dv \sum_{i=1}^{N} p(X(D_i))^- p(X(v))^+\right) \tag{12.35}$$

$$+ \frac{1}{\mu}\mathbb{E}\left(\int_0^{B_1}\sum_{i=1}^{N} p(X(D_i))^- p(X(\tilde{T}+v))^+(v-B_1)dv\right) \tag{12.36}$$

where $(D_1,\ldots,D_N)$ is the sequence of departure times of the busy-period of length $\tilde{T}$ beginning at time 0, $(D_1,\ldots,D_N)$ and $(D_{1,1},\ldots,D_{1,N_1})$ are the departure times of two independent busy-periods of lengths $\tilde{T}$ and B_1 beginning at time 0.

Proof

Step 1. Consider first a M/M/1 queue with marked service jumps described as follows. The mean of the difference between the queue length for this model and the model of reference can be expressed, if there are at most two jumps in the busy-period of the model as

$$\begin{aligned}
&-\mathbb{E}\left(1_{\{\bar{t}_1\le\tilde{T},\bar{t}_2>\tilde{T}+B_1\}}(\tilde{T}-\bar{t}_1+B_1)\right)\\
&-\mathbb{E}\left(1_{\{\bar{t}_2\le\tilde{T}\}}(\tilde{T}-\bar{t}_1+\tilde{T}-\bar{t}_2+3B_1')\right)\\
&-\mathbb{E}\left(1_{\{\bar{t}_1\le\tilde{T}<\bar{t}_2\le\tilde{T}+B_1<\bar{t}_3\}}(\tilde{T}-\bar{t}_1+\tilde{T}+2B_1-\bar{t}_2+B_1')\right).
\end{aligned}$$

This is sufficient to derive the term in ε^2 of the expansion. Indeed, the previous expression is equal to

$$\mathbb{E}_1\left(\sum_i \frac{\varepsilon}{\mu}p(X(D_i))^- \prod_{j\neq i}(1-\frac{\varepsilon}{\mu}p(X(D_j))^-)\prod_k(1-\frac{\varepsilon}{\mu}p(X(\tilde{T}+D_{1,k}))^-)(\tilde{T}-D_i+B_1)\right)$$

$$\mathbb{E}_1\left(\sum_{i<j} \frac{\varepsilon}{\mu}p(X(D_i))^- \frac{\varepsilon}{\mu}p(X(D_j))^- \prod_{k\neq i,j}(1-\frac{\varepsilon}{\mu}p(X(D_k))^-)(\tilde{T}-D_i+\tilde{T}-D_j+3B_1')\right)$$

$$\mathbb{E}_1\Bigg(\sum_{i,k} \frac{\varepsilon}{\mu}p(X(D_i))^- \prod_{j\neq i}(1-\frac{\varepsilon}{\mu}p(X(D_j))^-)\frac{\varepsilon}{\mu}p(X(\tilde{T}+D_{1,k}))^-$$
$$\prod_{l\neq k}(1-\frac{\varepsilon}{\mu}p(X(\tilde{T}+D_{1,l}))^-)(\tilde{T}-D_i+2B_1-D_{1,k}+B_1')\Bigg).$$

and its asymptotic of order 2 is

$$\begin{aligned}
&\frac{\varepsilon}{\mu}\mathbb{E}_1\Big(\sum_i p(X(D_i))^-(\tilde{T}-D_i+B_1)\Big)\\
&-\frac{\varepsilon^2}{\mu^2}\mathbb{E}_1\Big(\sum_{i<j} p(X(D_i))^- p(X(D_j))^-(\tilde{T}-D_i+\tilde{T}-D_j+2B_1')\Big)\\
&-\frac{\varepsilon^2}{\mu^2}\mathbb{E}_1\Big(\sum_{i,k} p(X(D_i))^- p(X(\tilde{T}+D_{1,k}))^-(\tilde{T}-D_i+B_1)\Big)\\
&+\frac{\varepsilon^2}{\mu^2}\mathbb{E}_1\Big(\sum_{i<j} p(X(D_i))^- p(X(D_j))^-(\tilde{T}-D_i+\tilde{T}-D_j+3B_1')\Big)\\
&+\frac{\varepsilon^2}{\mu^2}\mathbb{E}_1\Big(\sum_{i,k} p(X(D_i))^- p(X(\tilde{T}+D_{1,k}))^-(\tilde{T}-D_i+2B_1-D_{1,k}+B_1')\Big)
\end{aligned}$$

which integrating with respect to $(X(t))$ and by stationarity of $(X(t))$ gives

$$\begin{aligned}
&\frac{\varepsilon a^-}{\mu}\Big(\mathbb{E}_1(N\tilde{T}-D)+\mathbb{E}(B)\mathbb{E}(N)\Big)\\
&+\frac{\varepsilon^2}{\mu^2}\Big(\mathbb{E}\Big(\sum_{i,j=1,i<j}^{N} p(X(D_i))^- p(X(D_j))^-\Big)\mathbb{E}(B)\\
&+\mathbb{E}\Big(\sum_{i=1}^{N}\sum_{k=1}^{N_1} p(X(D_i))^- p(X(\tilde{T}+D_{1,k}))^-(B_1-D_{1,k}+B_1')\Big)\Big)+O(\varepsilon^3)
\end{aligned} \tag{12.37}$$

Step 2. Consider then a M/M/1 queue with additional service jumps. The additional service process is a Poisson process (t_n) independent of the M/M/1 queue with intensity $\varepsilon p(X(u))^+du$. The hitting time τ is equal to $\tilde{T}$ and

$$\mathbb{E}\Big(\int_0^{\tilde{T}}\Big(L(s)-\tilde{L}(s)\Big)ds\Big)=-\mathbb{E}\Big(1_{\{t_1<\tilde{T}\}}B_{\tilde{L}(t_1)}\Big)-\mathbb{E}\Big(1_{\{t_2<\tilde{T}\}}B_{\tilde{L}(t_2)-1}\Big) \tag{12.38}$$

$\mathcal{N}_2=(t_n)$ is a Poisson process with intensity $\varepsilon p(X(u))^+du$ thus

$$\begin{aligned}
\mathbb{P}(t_1>t)&=\mathbb{P}(\mathcal{N}_2(t)=0)\\
&=e^{-\varepsilon\int_0^t p(X(s))^+ds}
\end{aligned}$$

thus

$$\mathbb{P}(t_1<t)=1-e^{-\varepsilon\int_0^t p(X(s))^+ds}$$

and the density of t_1 is

$$\begin{aligned}
g_1(t)&=\varepsilon p(X(t))^+e^{-\varepsilon\int_0^t p(X(s))^+ds}\\
&=\varepsilon p(X(t))^+-\varepsilon^2 p(X(t))^+\int_0^t p(X(s))^+ds+O(\varepsilon^3).
\end{aligned}$$

In the same way,

$$\begin{aligned}\mathbb{P}(t_2 > t) &= \mathbb{P}(\mathcal{N}_2(t) = 0 \text{ or } 1)\\ &= \left(1 + \varepsilon \int_0^t p(X(s))^+ ds\right) e^{-\varepsilon \int_0^t p(X(s))^+ ds}\\ &= 1 - \frac{\varepsilon^2}{2}\left(\int_0^t p(X(s))^+ ds\right)^2 + O(\varepsilon^3)\end{aligned}$$

thus

$$\mathbb{P}(t_2 < t) = \frac{\varepsilon^2}{2}\left(\int_0^t p(X(s))^+ ds\right)^2 + O(\varepsilon^3)$$

and the density of t_2 is

$$g_2(t) = \varepsilon^2 p(X(t))^+ \int_0^t p(X(s))^+ ds + O(\varepsilon^3).$$

With standard calculations on the $M/M/1$ queue, using the expansion of the density functions of t_1 and t_2, equation (12.38) gives (see Appendix)

$$\begin{aligned}\mathbb{E}\left(\int_0^{\tilde{T}} (L(s) - \tilde{L}(s)) ds\right) &= -\frac{\varepsilon \mathbb{E}[p(X(0))^+]}{\mu^2(1-\rho)^3}\\ &\quad + \frac{\varepsilon^2}{\mu - \lambda}\mathbb{E}\left(\int_0^{\tilde{T}} \mathbb{E}[p(X(0))^+ p(X(u))^+](\tilde{T} - u) du\right).\end{aligned} \tag{12.39}$$

Step 3. Let us consider the different cases which lead to terms in ε and ε^2 in the expansion of $\mathbb{E}\left(\int_0^\tau \left(L(s) - \tilde{L}(s)\right) ds\right)$ for the integrated model. First if there is only one marked jump then $\tau = T$ and the contribution is

$$\mathbb{E}(1_{\{\bar{t}_1 \le \tilde{T}, \bar{t}_2 > \tilde{T} + B_1, t_1 > \tilde{T} + B_1\}}(\tilde{T} - \bar{t}_1 + B_1)$$

which gives terms from the so-called model with marked jumps (Step 1) and a new one, the term in ε^2 of the expansion of

$$-\mathbb{E}\left(1_{\{\bar{t}_1 \le \tilde{T}, t_1 < \tilde{T} + B_1\}}(\tilde{T} - \bar{t}_1 + B_1)\right). \tag{12.40}$$

The case with two marked jumps between 0 and τ gives a term in ε^2 which has already been computed in the model with marked jumps (see Step 1). The ε^2 term already computed of (12.37) is exactly the sum of terms (12.33) and (12.34) in the expression of β.

For the case with only one or two additional jumps, $\tau = \tilde{T}$ and the contribution is

$$-\mathbb{E}\left(1_{\{t_1 < \tilde{T}, \bar{t}_1 > \tilde{T}\}}(\tilde{T} - t_1)\right) - \mathbb{E}\left(1_{\{t_2 < \tilde{T}\}} B_{\tilde{L}(t_2)-1}\right).$$

Some terms have been already computed in the model with additional jumps (see (12.39))

$$-\frac{\varepsilon a_+}{\mu^2(1-\rho)^3} + \frac{\varepsilon^2}{\mu - \lambda}\mathbb{E}\left(\int_0^{\tilde{T}} \mathbb{E}[p(X(0))^+ p(X(u))^+](\tilde{T} - u) du\right)$$

and there is a new one which can be seen as the term in ε^2 in the expansion of

$$\mathbb{E}\left(1_{\{t_1<\tilde{T},\bar{t}_1\leq\tilde{T}\}}(\tilde{T}-t_1)\right). \tag{12.41}$$

The last case is the case of two jumps of different types (marked or additional) between 0 and τ. The contribution to the expression is new and equal to

$$\mathbb{E}\left(1_{\{t_1<\tilde{T}+B_1,\bar{t}_1\leq\tilde{T}<\bar{t}_2\}}(t_1-\bar{t}_1)\right). \tag{12.42}$$

In (12.42), the term in ε is 0. The conclusion is that the new term in ε^2 is the sum of the terms in ε^2 of (12.40), (12.41) and (12.42) which gives

$$\begin{aligned}
&-\mathbb{E}(1_{\{\bar{t}_1\leq\tilde{T},t_1\leq\tilde{T}+B_1\}}(\tilde{T}-\bar{t}_1+B_1)\\
&\quad+\mathbb{E}\left(1_{\{t_1<\tilde{T},\bar{t}_1\leq\tilde{T}\}}(\tilde{T}-t_1)\right)\\
&\quad+\mathbb{E}\left(1_{\{t_1<\tilde{T}+B_1,\bar{t}_1\leq\tilde{T}\}}(t_1-\bar{t}_1)\right)\\
&=-\mathbb{E}\left(1_{\{t_1<\tilde{T},\bar{t}_1\leq\tilde{T}\}}B_1\right)+\mathbb{E}(1_{\{\bar{t}_1\leq\tilde{T},\tilde{T}\leq t_1\leq\tilde{T}+B_1\}}(t_1-\tilde{T}-B_1)\Big)\\
&=-\mathbb{P}(t_1<\tilde{T},\bar{t}_1\leq\tilde{T})\mathbb{E}(B)+\mathbb{E}\Big(1_{\{\bar{t}_1\leq\tilde{T},\tilde{T}\leq t_1\leq\tilde{T}+B_1\}}(t_1-\tilde{T}-B_1)\Big)
\end{aligned} \tag{12.43}$$

With the notations $(D_1,\ldots,D_N)$ and $(D_1,\ldots,D_N)$ introduced in Proposition 15 the expression can be written

$$\begin{aligned}
&-\mathbb{E}(B)\mathbb{E}\Big(\sum_{i=1}^{N}\frac{\varepsilon}{\mu}p(X(D_i))^-\varepsilon\int_0^{\tilde{T}}p(X(v))^+dv\Big)+\mathbb{E}\Big(\sum_{i=1}^{N}\frac{\varepsilon}{\mu}p(X(D_i))^-\varepsilon\int_{\tilde{T}}^{\tilde{T}+B_1}dvp(X(v))^+(v-\tilde{T}-B_1)\Big)\\
&=-\frac{\varepsilon^2}{\mu(\mu-\lambda)}\mathbb{E}\Big(\sum_{i=1}^{N}\int_0^{\tilde{T}}dvp(X(D_i))^-p(X(v))^+\Big)+\frac{\varepsilon^2}{\mu}\mathbb{E}\Big(\sum_{i=1}^{N}\int_0^{B_1}dvp(X(D_i))^-p(X(\tilde{T}+v))^+(v-B_1)\Big)
\end{aligned}$$

It gives exactly the terms (12.35) and (12.36) involving the expressions of the form

$$\mathbb{E}[p(X(.))^-p(X(.))^+]$$

in the proposition. □

Then the limit result corresponding is the following.

Corollaire 6. *Replacing $(X(t))$ by $(X(\delta t))$ and letting δ tends to infinity, the term in ε^2 of $\mathbb{E}_1\left(\int_0^\tau\left(L(s)-\tilde{L}(s)\right)ds\right)$ reduces to*

$$\beta=\frac{a_+^2+a_-^2-a_+a_-(2+\rho)}{\mu^3(1-\rho)^4}$$

and letting δ tends to 0, the term in ε^2 of $\mathbb{E}_1\left(\int_0^\tau\left(L(s)-\tilde{L}(s)\right)ds\right)$

reduces to

$$\beta = \frac{A_+ + A_-(1+\rho)}{\mu^3(1-\rho)^4}.$$

where $A_+ = \mathbb{E}[(p(X(0))^+)^2]$ *and* $A_- = \mathbb{E}[(p(X(0))^-)^2]$.

Proof Letting δ tends to $+\infty$, using that

$$\mathbb{E}[p(X(0))^+ p(X(\delta u))^+] \to a_+^2,$$

it is easy to get that, in the expression of β, term (12.32) converges to

$$a_+^2 \frac{\mathbb{E}(B_1^2)}{2(\mu-\lambda)} = a_+^2 \frac{\mu}{(\mu-\lambda)^4}.$$

Term (12.33) converges to

$$a_-^2 \mathbb{E}\left(\frac{N(N-1)}{2}\right) \frac{1}{\mu^2(\mu-\lambda)} = \frac{a_-^2 \lambda}{(\mu-\lambda)^4}$$

and term (12.34) converges to

$$\begin{aligned}
&\frac{a_-^2}{\mu^2}\Big(\mathbb{E}(N)\mathbb{E}(NT) - \mathbb{E}(N)\mathbb{E}(D) + \mathbb{E}(N)^2\mathbb{E}(B)\Big) \\
&= \frac{a_-^2 \mathbb{E}(N)}{\mu^2}\Big(\mathbb{E}(NT) - \mathbb{E}(D) + \mathbb{E}(N)\mathbb{E}(B)\Big) \\
&= \frac{a_-^2 \mu}{\mu^2(\mu-\lambda)} \mu^2(\mu-\lambda)^3 \\
&= \frac{a_-^2 \mu}{(\mu-\lambda)^4}.
\end{aligned}$$

The sum of terms (12.35) and (12.36) converges to

$$\begin{aligned}
&-\frac{a_- a_+}{\mu(\mu-\lambda)}\mathbb{E}(NT) - \frac{a_- a_+}{\mu}\mathbb{E}(N)\mathbb{E}(\frac{T^2}{2}) \\
&= -a_- a_+ \left(\frac{\mu+\lambda}{(\mu-\lambda)^4} + \frac{\mu}{(\mu-\lambda)^4}\right) \\
&= -a_- a_+ \frac{2\mu+\lambda}{(\mu-\lambda)^4}
\end{aligned}$$

Summing these terms gives

$$\beta = \frac{\mu a_+^2 + (\lambda+\mu)a_-^2 - a_+ a_-(\lambda+2\mu)}{(\mu-\lambda)^4} \tag{12.44}$$

which is the expression in the corollary.

Letting δ tends to 0, using that

$$\mathbb{E}[p(X(0))^+ p(X(0))^-] = 0,$$

it is sufficient to replace a_+a_- by 0, a_+^2 by A_+ and a_-^2 by A_- in (12.44) to obtain

$$\beta = \frac{\mu A_+ + (\lambda+\mu)A_-}{(\mu-\lambda)^4}.$$

Proposition 16. *The stationary mean queue length in the perturbated model has the following expansion in* ε

$$\mathbb{E}_\pi(L) = \frac{1}{1-\rho} + \frac{\varepsilon}{\mu(1-\rho)^2} + \left(\frac{\lambda(\mu-\lambda)\beta}{\mu} - \frac{(a_- - a_+)a_-}{\mu(\mu-\lambda)^3}\right)\varepsilon^2 + O(\varepsilon^3)$$

where β *is given in Proposition 15. For the limit regime, when* δ *tends to* $+\infty$*,*

$$\mathbb{E}_\pi(L) = \frac{1}{1-\rho} + \frac{\lambda(a_- - a_+)}{(\mu-\lambda)^2}\varepsilon + \frac{\lambda(a_- - a_+)^2}{(\mu-\lambda)^3}\varepsilon^2 + O(\varepsilon^3)$$

and when δ *tends to* 0*,*

$$\mathbb{E}_\pi(L) = \frac{1}{1-\rho} + \frac{\lambda(a_- - a_+)}{(\mu-\lambda)^2}\varepsilon + \frac{\lambda\left((1+\rho)A_- + A_+ - \rho a_-(a_- - a_+)\right)}{(\mu-\lambda)^3}\varepsilon^2 + O(\varepsilon^3).$$

Proof. τ is a stopping time for both queues and $\tau = \tilde{T}$ or T. The asymptotic of order 1 in ε of τ is

$$\mathbb{E}_1(\tau) = \mathbb{E}_1(\tilde{T}) + \frac{\varepsilon a_-}{(\mu-\lambda)^2} + O(\varepsilon^2). \tag{12.45}$$

For the stopping time τ of both processes $(L(t))$ and $(\tilde{L}(t))$, it is well-known that

$$\mathbb{E}_\pi(L) = \mathbb{E}_{\tilde{\pi}}(\tilde{L}) + \frac{1}{\frac{1}{\lambda} + \mathbb{E}_1(\tau)}\mathbb{E}_1\left(\int_0^\tau \left(L(s) - \tilde{L}(s)\right) ds\right) \tag{12.46}$$

Replacing $\mathbb{E}_1(\tau)$ from equation (12.45) and $\mathbb{E}_1\left(\int_0^\tau \left(L(s) - \tilde{L}(s)\right) ds\right)$ from Proposition 15 in equation (12.46), the result follows. □

12.7 Appendix

Proof of (12.39)

By equation (12.38),

$$\mathbb{E}\left(\int_0^{\tilde{T}} \left(L(s) - \tilde{L}(s)\right) ds\right) = -\mathbb{E}\left(1_{\{t_1<\tilde{T}\}} B_{\tilde{L}(t_1)}\right) - \mathbb{E}\left(1_{\{t_2<\tilde{T}\}} B_{\tilde{L}(t_2)-1}\right). \tag{12.47}$$

The first term of the right-hand side of (12.47) is

$$\mathbb{E}\left(1_{\{t_1<\tilde{T}\}}B_{\tilde{L}(t_1)}\right) = \int_0^{+\infty} g_1(t)\mathbb{E}\left(1_{\{t<\tilde{T}\}}B_{\tilde{L}(t)}\right)dt \tag{12.48}$$

$$= \int_0^{+\infty}\left(\varepsilon p(X(t))^+ - \varepsilon^2 p(X(t))^+\int_0^t p(X(s))^+ds + O(\varepsilon^3)\right)\mathbb{E}\left(1_{\{t<\tilde{T}\}}B_{\tilde{L}(t)}\right)dt \tag{12.49}$$

But we use then that

$$\begin{aligned}\mathbb{E}\left(1_{\{t<\tilde{T}\}}B_{\tilde{L}(t)}\right) &= \sum_{i=1}^{+\infty}\mathbb{E}\left(1_{\{t<\tilde{T},\tilde{L}(t)=i\}}B_i\right)\\ &= \sum_{i=1}^{+\infty}\mathbb{E}(B_i)\mathbb{P}(t<\tilde{T},\tilde{L}(t)=i)\\ &= \mathbb{E}(B_1)\mathbb{E}\left(\sum_{i=1}^{+\infty} i1_{\{t<\tilde{T},\tilde{L}(t)=i\}}\right)\end{aligned}$$

and we integrate with respect to $(X(t))$, so

$$\begin{aligned}\mathbb{E}\left(1_{\{t_1<\tilde{T}\}}B_{\tilde{L}(t_1)}\right) &= \varepsilon\mathbb{E}[p(X(0))^+]\mathbb{E}(B_1)\sum_{i=1}^{+\infty} i\mathbb{E}\left(\int_0^{\tilde{T}} 1_{\{\tilde{L}(t)=i\}}dt\right)\\ &\quad - \varepsilon^2\sum_{i=1}^{+\infty}\mathbb{E}(B_i)\mathbb{E}\left(\int_0^{\tilde{T}} dt1_{\{\tilde{L}(t)=i\}}\int_0^t \mathbb{E}[p(X(0))^+p(X(u))^+]du\right) + O(\varepsilon^3)\end{aligned} \tag{12.50}$$

Furthermore the term in ε of the expansion in (12.50) can be computed using that

$$\begin{aligned}\mathbb{E}\left(\int_0^{\tilde{T}} 1_{\{\tilde{L}(t)=i\}}dt\right) &= \pi(i)\left(\mathbb{E}(\tilde{T}) + \frac{1}{\lambda}\right)\\ &= \frac{\rho^i}{\lambda}\end{aligned}$$

and then that

$$\sum_{i=1}^{+\infty} i\rho^i = \frac{\rho}{(1-\rho)^2}.$$

Equation (12.50) becomes

$$\begin{aligned}\mathbb{E}\left(1_{\{t_1<\tilde{T}\}}B_{\tilde{L}(t_1)}\right) &= \varepsilon\frac{\mathbb{E}[p(X(0))^+])}{\mu^2(1-\rho)^3}\\ &\quad - \varepsilon^2\sum_{i=1}^{+\infty}\mathbb{E}(B_i)\mathbb{E}\left(\int_0^{\tilde{T}} dt1_{\{\tilde{L}(t)=i\}}\int_0^t \mathbb{E}[p(X(0))^+p(X(u))^+]du\right) + O(\varepsilon^3).\end{aligned} \tag{12.51}$$

The secund term of the right-hand side of (12.47) is

$$\begin{aligned}\mathbb{E}\left(1_{\{t_2<\tilde{T}\}}B_{\tilde{L}(t_2)-1}\right) &= \int_0^{+\infty} g_2(t)\mathbb{E}\left(1_{t<\tilde{T}}B_{\tilde{L}(t)-1}\right)dt \\ &= \mathbb{E}\left(\int_0^{\tilde{T}} dt\varepsilon^2 p(X(t))^+ \int_0^t p(X(s))^+ ds B_{\tilde{L}(t)-1}\right).\end{aligned}$$

Integrating with respect to $(X(t))$,

$$\mathbb{E}\left(1_{\{t_2<\tilde{T}\}}B_{\tilde{L}(t_2)-1}\right) = \varepsilon^2\sum_{i=1}^{+\infty}\mathbb{E}(B_{i-1})\mathbb{E}\left(\int_0^{\tilde{T}} dt 1_{\{\tilde{L}(t)=i\}}\int_0^t \mathbb{E}[p(X(0)^+ p(X(u)^+]du\right) \tag{12.52}$$

Thus (12.51) and (12.52) together give, with some further calculations

$$\begin{aligned}&\mathbb{E}\left(\int_0^{\tilde{T}}\left(L(s)-\tilde{L}(s)\right)ds\right) \\ &= -\quad \varepsilon\frac{\mathbb{E}[p(X(0))^+])}{\mu^2(1-\rho)^3} + \varepsilon^2\mathbb{E}(B_1)\mathbb{E}\left(\int_0^{\tilde{T}} dt\int_0^t \mathbb{E}[p(X(0))^+ p(X(u))^+]du\right) + O(\varepsilon^3) \\ &= -\quad \varepsilon\frac{\mathbb{E}[p(X(0))^+])}{\mu^2(1-\rho)^3} + \frac{\varepsilon^2}{\mu-\lambda}\int_0^{+\infty} b(t)dt\int_0^t \mathbb{E}[p(X(0))^+ p(X(u))^+](t-u)du + O(\varepsilon^3).\end{aligned}$$

Some useful mean quantities for the M/M/1 queue

From [**?**], the first two moments of the stationary busy period are given by

$$\mathbb{E}(B_1) = \frac{1}{\mu-\lambda} \qquad\qquad \mathbb{E}(B_1^2) = \frac{2}{\mu^2(1-\rho)^3}. \tag{12.53}$$

We use the expression 2.40 p.190 of [**?**] for $\varphi(z,\xi) = \sum_{n=1}^{+\infty} z^n \int_0^{+\infty} e^{-\xi t} b_n(t)dt$, where

$$B_n(t) = \mathbb{P}(\tilde{T} < t, N = n),$$

and

$$b_n(t) = B_n'(t)$$

which is

$$\varphi(z,\xi) = \frac{1}{2\rho}\left(1+\rho+\mu^{-1}\xi - \sqrt{(1+\rho+\mu^{-1}\xi)^2 - 4\rho z}\right)$$

for $|z| \leq 1$, $Re\xi \geq 0$. It is easy to derive

$$\mathbb{E}(N) = \int_0^{+\infty} dt \sum_{n=1}^{+\infty} n b_n(t) = \frac{1}{1-\rho}$$

$$\mathbb{E}(N\tilde{T}) = \int_0^{+\infty} t dt \sum_{n=1}^{+\infty} n b_n(t) = -\frac{d^2\varphi}{dzd\xi}(1,0) = \frac{1+\rho}{\mu(1-\rho)^3}$$

$$\mathbb{E}[N(N-1)] = \int_0^{+\infty} dt \sum_{n=1}^{+\infty} n(n-1) b_n(t) = \frac{d^2\varphi}{dz^2}(1,0) = \frac{2\mu^2\lambda}{(\mu-\lambda)^3}.$$

It just remains to compute $\mathbb{E}(D)$ where $D = \sum_{i=1}^{N} D_i$. Using a branching argument used classically to derive the mean of the busy-period of the M/M/1 queue (see [**?**] for example)

$$D = \sigma + \sum_{i=1}^{N_\sigma} \left((\sigma + \sum_{j=1}^{i-1} \tilde{T}_j) N_i + D_i \right)$$

where σ is the service time of the first customer of the busy-period, N_{sigma} the number of arrivals during σ, $\tilde{T}_i$ the busy-period generated by the i-th customer arrived during σ, N_i the number of customers in $\tilde{T}_i$, D_i the sum of the departure times of $\tilde{T}_i$ from the beginning of this busy-period. Taking the expectation, it is easy to derive that

$$\mathbb{E}(D) = \mathbb{E}(\sigma) + \mathbb{E}(\sigma N_\sigma) + \mathbb{E}(B)\mathbb{E}(N_\sigma(N_\sigma - 1)/2)\mathbb{E}(N) + \mathbb{E}(\sigma N_\sigma)\mathbb{E}(D)$$

where N_σ has a geometric distribution with parameter $\frac{\lambda}{\lambda+\mu}$, thus $\mathbb{E}(N_\sigma(N_\sigma - 1) = 2\rho^2$. Simple algebra gives

$$\mathbb{E}(D) = \frac{\mu^2}{(\mu-\lambda)^3}.$$

Chapitre 13

Perturbation analysis of an $\mathrm{M}/\mathrm{M}/1$ queue in a diffusion random environment

13.1 Introduction

An $M/M/1$ queue whose server rate is time varying is investigated. The server rate is assumed to depend upon a random process $(X(t))$ so that the server rate at time t is $\phi(X(t))$ for some function ϕ. The study of this model is motivated by the following problem related to bandwidth sharing in telecommunication networks. Consider a link carrying elastic traffic corresponding to long file transfers together with a small proportion of traffic, which does not adapt to the level of congestion of the network, referred to as unresponsive traffic. On the one hand, long flows are usually controlled by TCP, which adapts the transmission rate of long flows according to the level of congestion of the network. If we consider the bottleneck link, then a reasonable assumption consists of using the processor sharing discipline for modeling how bandwidth is shared among long file transfers. Moreover, long flows are assumed to arrive according to a Poisson process. Under these classical modeling assumptions, we thus have an $M/G/1$ processor sharing queue (see for instance Massoulié and Roberts [**?**]).

On the other hand, unresponsive traffic is usually due to short file transfers, which are too small to achieve some bandwidth sharing within the network. It is however worth noting that with the emergence of multimedia applications in the Internet, unresponsive traffic may also be generated by streaming applications. This type of traffic is carried by the uncontrolled UDP protocol, which is not able to adapt to network conditions. As far as responsive flows are concerned, everything happens as if the transmission capacity seen by responsive long flows were reduced up to the aggregated bit rate of short flows.

Queueing systems with time varying server rate have been studied in the literature in many different situations. In Núñez-Queija and Boxma [**?**], the authors consider a queueing system where priority is given to some flows driven by Markov Modulated Poisson Processes (MMPP) with finite

state spaces and the low priority flows share the remaining server capacity according to the processor sharing discipline. By assuming that arrivals are Poisson and service times are exponentially distributed, the authors solve the system via a matrix analysis. Similar models have been investigated in Núñez-Queija [**?**, **?**] by still using the quasi-birth and death process associated with the system and a matrix analysis. The integration of elastic and streaming flows has been studied by Delcoigne *et al* [**?**], where stochastic bounds for the mean number of active flows have been established. More recently, priority queueing systems with fast dynamics, which can be described by means of quasi birth and death processes, have been studied via a perturbation analysis of a Markov chain by Altman *et al* [**?**].

In this paper, we assume that the process modulating the server rate is a diffusion process and more precisely an Ornstein-Uhlenbeck (OU) process. This assumption is motivated by the following facts.

1. An OU process reasonably represents the aggregated bit rate of the superposition of a large number of short flows. In particular, when those flows have exponentially distributed duration, then exact heavy traffic results show that the aggregated bit properly rescaled converges in distribution to an OU process, see Iglehart [**?**] for example.
2. An OU process has only two parameters (namely the mean and the variance), which can be empirically identified in practical situations. Furthermore, the impact of these parameters on the performance of the system will be much easier to understand, when compared with the case of MMPP environment where these variables are somewhat hidden in the numerous parameters of the MMPP environment.

In a first step, we establish the Fokker-Planck equations of the system and we show that these equations can be seen as an eigenvalue problem for a self-adjoint operator defined in some adequate Hilbert space. This last property is closely related to the time-reversibility properties of the $M/M/1$ occupation process and of the Ornstein-Uhlenbeck process. We then show that when the interaction between the OU process and the $M/M/1$ queue is weak and depends upon a small parameter ε, the problem of computing the generating function of the number of customers in the $M/M/1$ queue can be solved by means of a regular perturbation analysis. It is possible to completely compute the coefficients of the expansion in power series of ε of the solution. Also, the radius of convergence is determined. By taking into account the first order only, the above analysis shows that a reduced service rate pertains.

This paper is organized as follows : In Section 13.3, the Fokker-Planck equation is established. This equation can be interpreted as an eigenvalue problem for a self-adjoint operator is given in Section 13.4. When the interaction between the $M/M/1$ queue and the OU process is weak, the formulation as a perturbation problem is presented in Section 13.5, where the case of a linear perturbation function is completely solved.

13.2 Problem formulation

Model description

We consider a single link with transmission capacity equal to unity. We suppose that two classes of customers are multiplexed on this link and that the first class has priority over the second class. More precisely, if there are $N(t)$ customers of the first class in the system at time t, we assume

that the service rate for the customers of the second class is equal to $\phi(N(t))$ for some function $\phi(x)$ which is decreasing and such that $\phi(0) = 1$. Moreover, we assume that the number of class 1 customers is sufficiently large so that the process $N(t)$ properly rescaled converges in distribution to an OU process (X_t) satisfying the stochastic differential equation

$$dX(t) = -\alpha(X(t) - m)dt + \sigma dB(t), \tag{13.1}$$

where $(B(t))$ is a standard Brownian motion and α and σ are positive constants.

This situation typically occurs when class 1 customers arrive according to a Poisson process with rate u, require exponential service times with unit mean and have a peak bit rate which is negligible with respect to the link transmission capacity. Since class 1 customers have priority over class 2 customers and contention for those customers can be neglected, the process describing the number of class 1 customers then corresponds to the occupation process of an $M/M/\infty$ queue. When u tends to infinity, classical heavy traffic results (see Borovkov [**?**] or Iglehart [**?**]) then yield

$$\left(\frac{N(t) - u}{\sqrt{u}}, t \geq 0\right) \xrightarrow{d} (X(t), t \geq 0),$$

where the OU process $(X(t))$ satisfies Equation (13.1) with $\alpha = -1$ and $\sigma = \sqrt{2}$.

With the above assumptions, the server rate for class 2 customers is a function of $X(t)$, which is denoted by $\phi(X(t))$ (with $\phi(0) = 1$). We now assume that class 2 customers arrive according to a Poisson process with intensity λ and require exponential service times with mean $1/\mu$. If $L(t) = l$ denotes the number of class 2 customers in the system and $X(t) = x$ at time t, then the transitions of $(L(t))$ are given by

$$l \to \begin{cases} l+1 & \text{with rate } \lambda, \\ l-1 & \text{with rate } \mu\phi(x). \end{cases}$$

The process describing the number of class 2 customers is thus equal to the occupation process of an $M/M/1$ queue, which server rate depends upon a diffusion process. In the following, the function $\phi(x)$ will be referred to as perturbation function.

Throughout this paper, we assume that the diffusion process $(X(t))$ is in stationary regime. Its stationary distribution is a normal distribution with mean m and variance $\sigma^2/(2\alpha)$; its density function on $\mathbb{R}$ is therefore given by

$$p(x) \stackrel{\text{def.}}{=} \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}} \exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right). \tag{13.2}$$

Let us note that the stability condition for the system reads

$$\rho \stackrel{\text{def.}}{=} \frac{\lambda}{\mu} < \mathbb{E}[\phi(X(0))],$$

and will be assumed to hold throughout the paper. Under this assumption, it is straightforward to show the existence of a stationary probability distribution for the Markov process $(X(t), L(t))$. See Meyn and Tweedie [**?**] for example.

The independent case and its perturbation

When $\phi(x) \equiv 1$, the processes $(X(t))$ and $(L(t))$ are clearly independent one of each other. In this case, at equilibrium, for $t \geq 0$, the variable $L(t)$ has a geometric distribution with parameter ρ

and we consequently have the relation

$$\mathbb{E}\left(u^{L(t)}\mathbf{1}_{[x,x+dx]}(X(t))\right) = \frac{(1-\rho)}{(1-\rho u)}p(x)\,dx. \tag{13.3}$$

As it will be seen in Section 13.3, when ϕ is not constant, it is extremely difficult to get some explicit results on the equilibrium distribution of $(L(t))$. For this reason, this paper addresses the case when the queue is almost independent with respect to the OU process. More precisely, it is assumed that the function $\phi(x)$ is given by $1-\varepsilon x$ for some small $\varepsilon \geq 0$. The goal of this paper is to derive an expansion of the distribution of the stationary distribution of $(L(t))$ with respect to ε. In particular, the following theorem will be proved.

Theorem 1. *For ε sufficiently small, the first order expansion of the generating function of the stationary distribution of $(L(t))$ is given by*

$$\mathbb{E}\left(u^{L}\right) = \frac{1-\rho}{1-\rho u} - \frac{\rho(1-u)}{(1-\rho u)^2}m\varepsilon + o(\varepsilon).$$

Therefore, $\mathbb{E}[u^{L(t)}] \sim \mathbb{E}[u^{L_\varepsilon}]$, where L_ε has the stationary distribution of the number of customers in an $M/M/1$ queue when the server rate is $1-\varepsilon m$. This shows a principle of reduced service rate approximation.

13.3 Fokker-Planck equations

The goal of this section is to establish the Fokker-Planck equation for the process $(X(t), L(t))$ in the stationary regime, i.e., the evolution equation for the probability density function $p(x,\ell)$ for $x \in \mathbb{R}$ and $\ell \in \mathbb{N}$. By construction, it is easily checked that the process $(X(t), L(t))$ is a Markov process taking values in $\mathbb{R}\times\mathbb{N}$. The following result gives its infinitesimal generator.

Lemma 1. *The process $(X(t), L(t))$ is a Markov process in $\mathbb{R}\times\mathbb{N}$ with infinitesimal generator $\mathcal{G}$ defined by*

$$\begin{aligned}\mathcal{G}f(x,\ell) = \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial x^2}(x,\ell) - \alpha(x-m)\frac{\partial f}{\partial x}(x,\ell)\\ + \lambda[f(x,\ell+1)-f(x,\ell)] + \mu\phi(x)f(x)\mathbf{1}_{\{\ell>0\}}[f(x,\ell-1)-f(x,\ell)],\end{aligned} \tag{13.4}$$

for every function $f(x,\ell)$ from $\mathbb{R}\times\mathbb{N}$ in $\mathbb{R}$, twice differentiable with respect to the first variable.

Démonstration. According to Equation (13.1), the infinitesimal generator of an Ornstein-Uhlenbeck process applied to some twice differentiable function g on $\mathbb{R}$ is given by

$$\frac{\sigma^2}{2}\frac{\partial^2 g}{\partial x^2}(x) - \alpha(x-m)\frac{\partial g}{\partial x}(x).$$

The second part of Equation (13.4) corresponds to the infinitesimal generator of the number of customers of a classical $M/M/1$ queue with arrival rate λ and service rate $\mu\phi(x)$, when the OU process is in state x. □

Let P denote the stationary probability distribution of the couple $(X(t), L(t))$,

$$P(x, \ell) = \mathbb{P}(X \leq x, L = \ell),$$

the probability density function $p(x, \ell)$ is

$$p(x, \ell) = \frac{\partial P}{\partial x}(x, \ell)$$

and the generating function is given by

$$g_u(x) = \sum_{\ell=0}^{\infty} p(x, \ell) u^{\ell} \tag{13.5}$$

for $u \in (0, 1)$ and $x \in \mathbb{R}$.

The equation of invariant measure for the Markov process $(L(t), X(t))$ is given by

$$\sum_{\ell \geq 0} \int_{\mathbb{R}} \mathcal{G} f(x, \ell) P(dx, \ell) = 0,$$

for a twice differentiable function f with respect to the first variable. By choosing convenient test functions, one readily gets the Fokker-Planck equations.

Lemma 2 (Fokker-Planck equations). *The function $p(x, \ell)$ satisfies the relation*

$$\begin{aligned} \frac{\sigma^2}{2} \frac{\partial^2 p}{\partial x^2} + \alpha(x - m) \frac{\partial p}{\partial x} + \alpha p(x, \ell) + \lambda \mathbf{1}_{\{\ell > 0\}} p(x, \ell - 1) \\ - \left(\lambda + \mu \phi(x) \mathbf{1}_{\{\ell > 0\}}\right) p(x, \ell) + \mu \phi(x) \mathbf{1}_{\{\ell > 0\}} p(x, \ell + 1) = 0. \end{aligned} \tag{13.6}$$

An easy consequence of the above Fokker-Planck equation is the following equation for the function g_u.

Proposition 17. *The generating function $g_u(x)$ is such that*

$$\begin{aligned} \frac{\sigma^2}{2} \frac{\partial^2 g_u}{\partial x^2} + \alpha(x - m) \frac{\partial g_u}{\partial x} + \left(\lambda(u - 1) + \alpha + \mu \left(\frac{1}{u} - 1\right) \phi(x)\right) g_u(x) \\ = \mu \left(\frac{1}{u} - 1\right) \phi(x) g_0(x). \end{aligned} \tag{13.7}$$

13.4 Operator theoretic analysis of the Fokker Planck equation

Notation

By definition, the function $g_u(x)$ is twice weakly differentiable with respect to the variable x and is analytic in variable u in the open unit disk and continuous in the closed unit disk. Hence, the function $g_u(x)$ can be seen as an element of the tensor product $H^2(\mathbb{R}) \otimes S(U)$, where $H^2(\mathbb{R})$ is

the Sobolev space of functions which admit a second order weak derivative and $S(U)$ is the set of functions which are analytic in the unit disk and continuous in the closed unit disk.

From the previous section, the function $g : (u, x) \to g_u(x)$ defined by Equation (13.5) satisfies the relation

$$\Omega g(u, x) = 0$$

where Ω is the operator defined as follows : for a function $f \in H^2(\mathbb{R}) \otimes S(U)$

$$\Omega f(u,x) = \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial x^2} + \alpha(x-m)\frac{\partial f}{\partial x} + \left(\lambda(u-1) + \alpha + \mu\left(\frac{1}{u} - 1\right)\phi(x)\right) f(u,x) - \mu\left(\frac{1}{u} - 1\right)\phi(x) f(0,x). \quad (13.8)$$

In the following, we refine the domain of definition of Ω so as to obtain a self-adjoint operator defined in an appropriate Hilbert space.

By construction, the function $g_u(x)$ is given by

$$g_u(x) = \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}} \exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right) \mathbb{E}[u^L \mid X = x]$$

The function $g_u(x)$ of the variable u is analytic in the open unit disk and is continuous in the closed unit disk. Moreover, the function $g_u(x)$ is such that for all $|u| \leq 1$,

$$\frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}} \int_{-\infty}^{\infty} g_u(x)^2 \exp\left(\frac{\alpha(x-m)^2}{\sigma^2}\right) dx \leq 1.$$

This clearly implies that for fixed u with $|u| \leq 1$, the function $g_u(x)$ is in the Hilbert space H defined by

$$H = \left\{ f : \mathbb{R} \to \mathbb{C} : f \exp\left(\frac{\alpha(x-m)^2}{2\sigma^2}\right) \in H^2(\mathbb{R}) \right\}, \quad (13.9)$$

This Hilbert space is equipped with the scalar product defined by : for all $f, g \in H$,

$$< f, g > = \int_{-\infty}^{\infty} f(x)\overline{g(x)} \exp\left(\frac{\alpha(x-m)^2}{\sigma^2}\right) dx,$$

where $\overline{g(x)}$ is the complex conjugate of $g(x)$; the norm of an element $f \in H$ is

$$\|f\| = \sqrt{\int_{-\infty}^{\infty} |f(x)|^2 \exp\left(\frac{\alpha(x-m)^2}{\sigma^2}\right) dx}.$$

Now, for fixed x, we identify $g_u(x)$ with the sequence $(p(x,\ell), l \geq 0)$. By definition,

$$p(x,\ell) = \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}} \exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right) \mathbb{P}(L = \ell \mid X = x),$$

and clearly

$$\sum_{\ell=0}^{\infty} |p(x,\ell)|^2 < \frac{\alpha}{\pi\sigma^2} \exp\left(-\frac{2\alpha(x-m)^2}{\sigma^2}\right),$$

it follows that the sequence $(p(x,\ell), \ell \geq 0)$ is in the Hilbert space $L^2(\mathbb{N})$ composed of square summable sequences in $\mathbb{C}$, that is,

$$L^2(\mathbb{N}) = \left\{ f = (f_n) \in \mathbb{C}^{\mathbb{N}} : \sum_{n=0}^{\infty} |f_n|^2 < \infty \right\},$$

equipped with the scalar product defined by

$$< f, g > = \sum_{n=0}^{\infty} f_n \overline{g_n}, \tag{13.10}$$

for two elements $f = (f_n)$ and $g = (g_n)$. The norm of an element f of $L^2(\mathbb{N})$ is given by

$$\|f\| = \sqrt{\sum_{n=0}^{\infty} |f_n|^2}. \tag{13.11}$$

The operator Ω can be seen as an operator defined in the tensor product $H \otimes L^2(\mathbb{N})$, that we still denote by Ω and given by

$$\Omega = A \otimes \mathbb{I} + \mathbb{I} \otimes B + V,$$

where

- The symbol $\mathbb{I}$ denotes the identity operator in the appropriate Hilbert space.
- The operator A is defined by

  $$Af = \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} + \alpha(x - m) \frac{\partial f}{\partial x} + \alpha f;$$

  the domain of definition of A is denoted by $D(A)$ and is given by

  $$D(A) = \left\{ f \in H : x^2 f \exp\left(\frac{\alpha(x-m)^2}{2\sigma^2} \right) \in H^2(\mathbb{R}) \right\}.$$

- The operator B is defined by the infinite matrix

  $$\begin{pmatrix} -\lambda & \mu & 0 & . & . & . \\ \lambda & -(\lambda+\mu) & \mu & 0 & . & . \\ 0 & \lambda & -(\lambda+\mu) & \mu & 0 & . \\ 0 & 0 & \lambda & -(\lambda+\mu) & \mu & . \\ . & . & . & . & . & . \end{pmatrix} \tag{13.12}$$

  the domain of B will be determined in the following.
- The operator V is equal to $\Phi \otimes C$, where Φ is the operator defined on $L^2(\mathbb{R})$ by the multiplication by $\phi - 1$, i.e.

  $$\Phi(f)(x) = (\phi(x) - 1) f(x),$$

  and C is the pure death operator, defined by the infinite matrix

  $$\begin{pmatrix} 0 & \mu & 0 & . & . & . & . \\ 0 & -\mu & \mu & 0 & . & . & . \\ 0 & 0 & -\mu & \mu & 0 & . & . \\ 0 & 0 & 0 & -\mu & \mu & 0 & . \\ . & . & . & . & . & . & . \end{pmatrix} \tag{13.13}$$

Self-adjointness properties

In this section, we examine the properties of the operators A and B. We specifically determine under which conditions these operators are self-adjoint. The self-adjointness property will be crucial in subsequent sections to carry out a perturbation analysis. To prove self-adjointness, we use the classical tools of spectral analysis (see Dautray and Lions [**?**], Dunford and Schwartz [**?**], Reed and Simon [**?**] or Rudin [**?**] for basic elements of spectral theory).

The operator B defined in $L^2(\mathbb{N})$ is not symmetric. However, by reducing the underlying Hilbert space, we can obtain a symmetric operator as follows. Let us consider the Hilbert space $L^2_\rho(\mathbb{N})$ defined by

$$L^2_\rho(\mathbb{N}) = \left\{ f = (f_n) \in \mathbb{C}^{\mathbb{N}} : \sum_{n=0}^{\infty} |f_n|^2 \rho^{-n} < \infty \right\},$$

where $\rho = \lambda/\mu < 1$. The scalar product in $L^2_\rho(\mathbb{N})$ is defined by

$$<f, g>_\rho = \sum_{n=0}^{\infty} f_n \overline{g_n} \rho^{-n}$$

and the norm by

$$\|f\|_\rho = \sqrt{\sum_{n=0}^{\infty} |f_n|^2 \rho^{-n}}.$$

Since $\rho < 1$, the space $L^2_\rho(\mathbb{N})$ is clearly a subspace of $L^2(\mathbb{N})$. The operator B induces in $L^2_\rho(\mathbb{N})$ an operator, that we still denote by B and that is defined by the infinite matrix given by Equation (13.12). In the following, the symbol B refers to that operator defined in $L^2_\rho(\mathbb{N})$ by the infinite matrix (13.12).

Let $D(B)$ denote the domain of the operator B, i.e., the subset of $L^2_\rho(\mathbb{N})$ composed of those elements $f \in L^2_\rho(\mathbb{N})$ such that $Bf \in L^2_\rho(\mathbb{N})$. The adjoint of the operator B is denoted by B^* and is defined by : for $f \in D(B)$ and $g \in H$ by $<Bf, g>_\rho = <f, B^* g>_\rho$ and $D(B^*)$ is the domain of B^*.

In the following, the operator B is shown to be self-adjoint, i.e. $B = B^*$, which requires in particular that $D(B) = D(B^*)$. To get this property it is sufficient to prove that the operator B is

— symmetric : for all $f, g \in D(B)$, $<Bf, g>_\rho = <f, Bg>_\rho$;

— bounded : the quantity

$$\|B\|_\rho = \inf \left\{ |<Bf, f>_\rho| : f \in L^2_\rho(\mathbb{N}), \|f\|^2_\rho = 1 \right\}$$

is finite.

Lemma 3. *The operator B is symmetric and bounded with*

$$\|B\|_\rho \leq (\sqrt{\lambda} + \sqrt{\mu})^2, \tag{13.14}$$

the operator B is consequently self-adjoint.

Démonstration. The symmetry of the operator B is straightforward. For $f \in L^2_\rho(\mathbb{N})$,

$$<Bf, f>_\rho = \sum_{n=0}^{\infty} \left((\lambda + \mu \mathbf{1}_{\{n \geq 0\}}) f_n - \lambda f_{n-1} - \mu f_{n+1} \right) \overline{f_n} \rho^{-n},$$

consequently

$$|<Bf, f>_\rho| \leq (\lambda+\mu)||f||_\rho^2 + \left|\sum_{n=0}^{\infty} \mu f_{n+1}\overline{f_n}\rho^{-n}\right| + \left|\sum_{n=0}^{\infty} \lambda f_{n-1}\overline{g_n}\rho^{-n}\right|,$$

by using Schwarz inequality, we get

$$\left|\sum_{n=0}^{\infty} \mu f_{n+1}\overline{f_n}\rho^{-n}\right| \leq \sqrt{\lambda\mu}||f||_\rho^2,$$
$$\left|\sum_{n=0}^{\infty} \lambda f_{n-1}\overline{f_n}\rho^{-n}\right| \leq \sqrt{\lambda\mu}||f||_\rho^2,$$

and Equation (13.14) follows. □

The spectrum $\sigma(B)$ of the operator B is defined by

$$\sigma(B) = \{z \in \mathbb{C} : (B - z\mathbb{I}) \text{ is not invertible}\},$$

since B is self-adjoint, $\sigma(B) \subset \mathbb{R}$. Moreover, Equation (13.14) implies the relation $\sigma(B) \subset [-(\sqrt{\lambda}+\sqrt{\mu})^2, \infty]$. Standard spectral theory shows that the spectrum $\sigma(B)$ can be decomposed as follows :

$$\sigma(B) = \overline{\sigma_p(B)} \cup \sigma_c(B),$$

where $\overline{\sigma_p(B)}$ is the closure of the set composed of the eigenvalues of B, referred to as point spectrum, and $\sigma_c(B)$ is the continuous spectrum. The point spectrum is purely discrete and $z \in \sigma_p(B)$ if and only if there exists some $f \in L_\rho^2(\mathbb{N})$ such that $Bf = zf$. From spectral theory, the following proposition holds.

Proposition 18. *There exists a measure $d\psi(z)$, referred to as spectral measure, whose support is $\sigma(B)$, and a family of spaces $\{\mathcal{H}_z\}$, $z \in \sigma(B)$, such that*

– *the Hilbert space $L_\rho^2(\mathbb{N})$ is equal to the direct sum of the spaces $\mathcal{H}_z$, that is,*

$$H = \int^{\oplus} \mathcal{H}_z d\psi(z), \tag{13.15}$$

i.e. every $f \in L_\rho^2(\mathbb{N})$ can be decomposed into a family $(f_z, z \in \sigma(B))$, where $f_z \in \mathcal{H}_z$ and $\int \|f_z\|_\rho^2 \psi(z) < \infty$. Moreover,

$$<f, g>_\rho = \int <f_z, g_z>_\rho d\psi(z).$$

– *The operator B is such $(Bf)_z = zf_z$ for $z \in \sigma(B)$, where $(Bf)_z$ is the projection of (Bf) on the space $\mathcal{H}_z$.*

Note that z is an eigenvalue of the operator B if and only if $\psi(\{z\}) > 0$ and that the space $\mathcal{H}_z$ is a subset of $L_\rho^2(\mathbb{N})$ if and only if z is an eigenvalue. The next result gives the explicit representation of the spectral measure and the spaces $\mathcal{H}_z$ appearing in Decomposition (13.15).

Proposition 19. *The spectral measure $d\psi(x)$ is given by*

$$\int f(x)d\psi(x) = (1-\rho)f(0) - \frac{\sqrt{\rho}}{\pi}\int_{-(\sqrt{\lambda}+\sqrt{\mu})^2}^{-(\sqrt{\lambda}-\sqrt{\mu})^2} \frac{f(x)}{x}\sqrt{1-\left(\frac{x+\lambda+\mu}{2\sqrt{\lambda\mu}}\right)^2}\,dx, \quad (13.16)$$

for any non-negative Borelian function f.

The operator B has a unique eigenvalue at 0, and the associated eigenvector is up to a multiplicative constant the sequence $e(\rho)$ whose nth element is equal to ρ^n. The space $\mathcal{H}_0$ is the space spanned by the vector $e(\rho)$.

For $z \in (-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2)$, the space $\mathcal{H}_z$ is the vector space spanned by the sequence $(Q_n(z))$ defined by the following recursion :

$$Q_0(z) = 1, Q_1(z) = (z+\lambda)/\mu$$
$$\mu Q_{n+1}(z) - (z+\lambda+\mu)Q_n(z) + \mu Q_{n-1}(z) = 0, \quad n \geq 2.$$

The sequence $(Q_n(z))$ for $z \in (-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\mu}-\sqrt{\lambda})^2)$ forms an orthogonal family.

Démonstration. To determine the spectrum of the operator B, we consider the equation $Bf(z) = zf(z)$, where $f(z) = (f_n(z)) \in \mathbb{C}^{\mathbb{N}}$. By assuming that $f_0(z) = 1$, the sequence $f(z)$ satisfies the recurrence relation :

$$f_0(z) = 1, f_1(z) = (z+\lambda)/\mu,$$
$$\mu f_{n+1}(z) - (z+\lambda+\mu)f_n(z) + \lambda f_{n-1}(z) = 0, \quad n \geq 2. \quad (13.17)$$

The above three-term recurrence relation implies that $f_n(z)$ is a polynomial in variable z with degree n and that the polynomials $(f_n(z))$ form an orthogonal polynomial system since Favard's condition is obviously satisfied (see Askey and Ismail [**?**] for details). The orthogonality measure of these polynomials is precisely the spectral measure of the operator B since B is self-adjoint.

To determine $d\psi(x)$, we compute the limiting value as n tends to infinity of the ratio $f_n^*(z)/f_n(z)$, where $f_n^*(z)$, $n = 0, 1, 2, \ldots$ are the associate polynomials, which satisfy recurrence relation (13.17) with the initial conditions

$$f_0(z) = 0 \text{ and } f_1(z) = 1/\mu.$$

Straightforward computations yield that, for $z \notin [-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2]$,

$$f_n(z) = \frac{1}{Z_+ - Z_-}\left[\left(\frac{\lambda+z-\mu+\sqrt{\delta(z)}}{2\mu}\right)Z_+^n + \left(\frac{-\lambda-z+\mu+\sqrt{\delta(z)}}{2\mu}\right)Z_-^n\right],$$

where

$$Z_\pm = \frac{z+\lambda+\mu \pm \sqrt{\delta(z)}}{2\mu}$$

with $\delta(z) = (z+\lambda+\mu)^2 - 4\lambda\mu$. Moreover, the associated polynomials $f_n^*(z)$ are given by : for $z \notin [-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2]$

$$f_n^*(z) = \frac{1}{\mu(Z_+ - Z_-)}\left[Z_+^n - Z_-^n\right].$$

Stieltjes theory [**?**] states that the measure $d\psi(x)$ has a support included in $(-\infty, 0]$ and that

$$\int_{-\infty}^{0} \frac{d\psi(x)}{z-x} = \chi(z),$$

where $\chi(z)$ is the continued fraction whose nth approximant is $f_n^*(z)/f_n(z)$. The function $\chi(z)$ for $z \notin (-\infty, 0]$ is given by

$$\chi(z) = \lim_{n\to\infty} \frac{f_n^*(z)}{f_n(z)}.$$

It is easily checked that for $z > 0$, $Z_+ > Z_- > 0$ and then for $z > 0$,

$$\chi(z) = \frac{2}{\lambda + z - \mu + \sqrt{\delta(z)}}. \tag{13.18}$$

The function on the right hand side of Equation (13.18) can be analytically continued to the complex plane deprived of the segment $[-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2]$ and the origin. More precisely, the function $\chi(z)$ has a unique pole at $z = 0$ and its residue is equal to $(1-\rho)$. The eigenvector associated with the eigenvalue 0 is the vector which nth component is ρ^n. (This vector clearly belongs to $L^2_\rho(\mathbb{N})$.)

From Perron-Stieltjes inversion formula, see Askey and Ismail [**?**] and Henrici [**?**], the continuous spectrum of the measure $d\psi(x)$ is given by

$$\frac{d\psi(x)}{dx} = \lim_{\varepsilon\to 0} \frac{1}{2i\pi}(\chi(x-i\varepsilon) - \chi(x+i\varepsilon)).$$

It is easily checked that the above limit is non null only for x in the interval $(-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2)$ and, in that case,

$$d\psi(x) = -\frac{\sqrt{\rho}}{\pi x}\sqrt{1 - \frac{(x+\lambda+\mu)^2}{4\lambda\mu}}\,dx.$$

It is worth noting that $d\psi(x)$ is very close to that the corresponding spectral measure associated with Chebyshev polynomials. In fact, the polynomials under consideration here differ from Chebyshev polynomials only through the initial conditions (see Chihara [**?**] for an exhaustive treatment of classical orthogonal polynomials).

It is easily checked that

$$\begin{aligned}\int_{-(\sqrt{\lambda}+\sqrt{\mu})^2}^{-(\sqrt{\lambda}-\sqrt{\mu})^2} d\psi(x) &= \frac{2\sqrt{\lambda\mu\rho}}{\pi}\int_{-1}^{1} \frac{\sqrt{1-x^2}}{\lambda - 2\sqrt{\lambda\mu}x + \mu}\,dx \\ &= \frac{2\rho}{\pi}\sum_{n=0}^{\infty}\rho^{n/2}\int_{-1}^{1} U_n(x)\sqrt{1-x^2}\,dx = \rho,\end{aligned}$$

where $U_n(x)$, $n = 0, 1, 2, \ldots$ are the Chebyshev polynomials of the second kind, which are orthonormal with respect to the weight measure $w(x)dx$ with

$$w(x) = \sqrt{1-x^2}\;\mathbf{1}_{(-1,1)}(x).$$

It follows that the total mass of $d\psi(x)$ is

$$\int_{-\infty}^{0} d\psi(x) = 1.$$

The orthogonality of the sequences $(Q_n(z))$ for $z \in [-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2]$ and Equation (13.16) are therefore established. □

It is worth noting that the point spectrum of the operator B contains only one point and that the continuous spectrum is the interval $(-(\sqrt{\lambda}+\sqrt{\mu})^2, -(\sqrt{\lambda}-\sqrt{\mu})^2)$.

Let us now examine the properties of the operator A. This operator is closely related to the harmonic oscillator in quantum mechanics (see Reed and Simon [**?**] for details). Indeed, for $f \in D(A)$ and $h(x) = f(x)\exp[\alpha(x-m)^2/(2\sigma^2)]$ we have

$$Af = \alpha\left(\frac{\sigma^2}{2\alpha}\frac{\partial^2 h}{\partial x^2} + \left(\frac{1}{2} - \frac{\alpha(x-m)^2}{2\sigma^2}\right)h\right)\exp\left(-\frac{\alpha(x-m)^2}{2\sigma^2}\right).$$

We then have the following result, where we use the Hermite functions $H_\nu(x)$, the Hermite polynomials $H_n(x)$, as well as the parabolic cylinder functions $D_n(x)$, also referred to as Whittaker functions, see Abramowitz and Stegun [**?**] or Lebedev [**?**].

Proposition 20. *The operator A is self-adjoint in H. Its spectrum is purely discrete, composed of the numbers of the form $-2\alpha n$ for $n \geq 0$. The eigenvector associated with the eigenvalue $-\alpha n$ is the function*

$$\varphi_n(x) = \gamma_n \exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right) H_n\left(\frac{\sqrt{\alpha}(x-m)}{\sigma}\right), \tag{13.19}$$

where $H_n(x)$ is the nth Hermite polynomial and

$$\gamma_n^2 = \frac{\sqrt{\alpha}}{2^n n!\sigma\sqrt{\pi}}.$$

The sequence (φ_n) forms an orthonormal basis of H.

Démonstration. The Hilbert space H defined by Equation (13.9) and $H^2(\mathbb{R})$ are obviously isomorphic.

Let κ denote canonical isomorphism from H into $H^2(\mathbb{R})$. By construction, this isomorphism preserves the scalar product. The image of the operator A by the isomorphism κ is the operator $\alpha(-\mathcal{A} + \frac{1}{2}\mathbb{I})$ where the operator $\mathcal{A}$ is the harmonic oscillator operator defined by

$$\mathcal{A}h = -\frac{\sigma^2}{2\alpha}\frac{\partial^2 h}{\partial x^2} + \frac{\alpha(x-m)^2}{2\sigma^2}h.$$

The domain of definition of the operator $\mathcal{A}$ is

$$D(\mathcal{A}) = \left\{h \in H^2(\mathbb{R}) : x^2 h \in L^2(\mathbb{R})\right\}.$$

It is well known that the operator $\mathcal{A}$ is self-adjoint. The functions $h_n(x) = D_n(\sqrt{2\alpha}(x-m)/\sigma)$, $n \geq 0$, where the functions D_n are Whittaker parabolic cylinder functions [**?**], satisfy

$$\mathcal{A}h_n = \left(n + \frac{1}{2}\right)h_n$$

Since the functions D_n for $n \geq 0$ form an orthogonal basis of $H^2(\mathbb{R})$, the spectrum of $-\mathcal{A}$ is purely discrete and composed of the numbers $n + 1/2$ for $n \geq 0$. It follows that the operator A is self-adjoint in H. Its eigenvalues are the numbers $-\alpha n$, $n \geq 0$ and the eigenvectors associated with the eigenvalue $-\alpha n$ is

$$\tilde{\varphi}_n(x) = \exp\left(-\frac{\alpha(x-m)^2}{2\sigma^2}\right) D_n(\sqrt{2\alpha}(x-m)/\sigma).$$

By using the relation between Whittaker and Hermite functions [**?**, p. 284] and by normalizing, Equation (13.19) follows. □

The main Hilbert space $\mathcal{H}$ used in this paper is defined as the tensor product of the spaces H and $L^2_\rho(\mathbb{N})$, that is, $\mathcal{H} = H \otimes L^2_\rho(\mathbb{N})$. In view of the above results, an element of this Hilbert space is defined by a sequence $(c_{n,k})$ and can be written as

$$\sum_{n=0}^{\infty}\sum_{k=0}^{\infty} c_{n,k}\varphi_n \otimes e_k,$$

where e_k is the sequence with all elements equal to 0 except the kth one equal to 1 and φ_n is defined by Equation (13.19).

The Hilbert space $\mathcal{H}$ is equipped with the scalar product $<\cdot,\cdot>$ defined by : for $f = (f_{n,k})$ and $g = (g_{n,k})$ in $\mathcal{H}$

$$<f, g> = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty} f_{n,k}\overline{g}_{n,k}\rho^{-n};$$

the norm is defined as

$$\|f\|^2 = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty} |f_{n,k}|^2\rho^{-n}.$$

Note also that, as a consequence of Proposition 19, an element $f \in \mathcal{H}$ can also be represented as

$$f = \sum_{n\geq 0}\int_{-(\sqrt{\lambda}+\sqrt{\mu})^2}^{-(\sqrt{\lambda}-\sqrt{\mu})^2} c_n(y)\varphi_n \otimes \mathcal{Q}(y)\, d\psi(y) + \sum_{n\geq 0} c_n(0)\varphi_n \otimes e(\rho),$$

where the measure $d\psi(y)$ and the sequence $\mathcal{Q}(y)$ are defined by Proposition 19 and the function $c_n : \mathbb{R} \to \mathbb{C}$ is such that

$$\sum_{n\geq 0}\int_{-(\sqrt{\lambda}+\sqrt{\mu})^2}^{-(\sqrt{\lambda}-\sqrt{\mu})^2} |c_n(y)|^2\, d\psi(y) < \infty.$$

If we consider the equation $(A\otimes\mathbb{I} + \mathbb{I}\otimes B)f = 0$ for a non-zero function f of $\mathcal{H}$ represented by

$$f = \sum_{n\geq 0} c_n\varphi_n \otimes g,$$

for some $g = (g_n) \in L^2_\rho(\mathbb{N})$ and (c_n) a sequence of $\mathbb{C}$, then we have for all $n \geq 0$,

$$c_n(-\alpha n g + Bg) = 0.$$

Since the single eigenvalue of B is 0, we have $c_n = 0$ for $n > 1$. Now, if we want that $c_0 \neq 0$, then $g = \kappa e(\rho)$ for some $\kappa > 0$. If we impose that the sum of the g_n is 1, then $\kappa = 1 - \rho$. This shows that, for $\phi \equiv 1$, the unique non-null solution to the equation $(A \otimes \mathbb{I} + \mathbb{I} \otimes B)f = 0$ is proportional to $\varphi_0 \otimes e(\rho)$.

To conclude this section, let us examine the properties of the operator V.

Proposition 21. *If the function $|\phi - 1|$ is upper bounded by a constant K, then the operator V is bounded and its norm $||V|| \leq K\mu$.*

Démonstration. For $g \in L^2_\rho(\mathbb{N})$, $<Dg, g>_\rho \leq \mu||g||^2$. Hence, for any element $f = (f_n(x)) \in \mathcal{H}$,

$$(Vf, f) \leq \int_{-\infty}^{\infty} |\phi(x) - 1|\mu \sum_{n \geq 0} ||f_n(x)||^2 \exp\left(\frac{\alpha(x-m)^2}{\sigma^2}\right) dx \leq K\mu||f||^2,$$

and the result follows. □

The above result indicates that when the function $\phi(x) - 1$ is bounded, then the operator V appears as a nice self-adjoint perturbation of the operator $A \otimes \mathbb{I} + \mathbb{I} \otimes B$. In the following, we have to deal with a more complex perturbation function of the form $\phi(x) = 1 - \varepsilon x$. The multiplication by x is clearly not bounded in H and the above result can not be applied.

In the remainder of this paper, an element $f = (f_{n,k})$ is identified with the function in $\mathcal{H}$ defined by

$$f_u(x) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} c_{n,k} u^k \varphi_n(x).$$

13.5 Perturbation analysis

In this section, we assume that the perturbation function ϕ is of the form $1 - \varepsilon x$ for some $\varepsilon \ll 1$. The operator V thus appears as a small perturbation of the self-adjoint operator $A \otimes \mathbb{I} + \mathbb{I} \otimes B$. We then perform a classical perturbation analysis by studying the modification of the function $g_u(x)$ given by Equation (13.3) due to the perturbation.

In the following, we search for a solution to the Fokker-Planck Equation (13.7), which belongs to the reference Hilbert space $\mathcal{H}$. We specifically assume that the solution can be expanded as

$$g_{u,\varepsilon}(x) = g_u^{(0)}(x) + \varepsilon g_u^{(1)}(x) + \varepsilon^2 g_u^{(2)}(x) + \cdots, \tag{13.20}$$

where the functions $g_u^{(n)}(x)$ for $n \geq 0$ belongs to the Hilbert space $\mathcal{H}$. The function $g_u^{(0)}(x)$ corresponds to the case $\varepsilon = 0$ and is given by Equation (13.3).

The ultimate goal of this section is to prove that the elements $g^{(n)}$ has to satisfy a recurrence relation of the form $g^{(n)} = \Theta(xg^{(n-1)})$ for $n \geq 1$ and for some linear operator Θ whose norm is finite. This shows that the expansion (13.20) actually defines for sufficiently small ε an element which is in $\mathcal{H}$ and that by construction, this is the unique solution to the perturbed Fokker-Planck equation.

In the following, we assume that the expansion (13.20) is valid and we investigate the conditions which have to be satisfied by the elements $g^{(n)}$. In a first step, we prove the following property satisfied by the functions $(g_u^{(n)}(x))$.

Lemma 4. *For $n \geq 0$, the function $g_u^{(n)}(x)$ in Expansion (13.20) can be expressed as a linear combination of $\varphi_0, \ldots, \varphi_n$. In particular, for $N > n$,*

$$\lim_{x \to \pm\infty} \frac{1}{x^N} g_u^{(n)}(x) \exp\left(\frac{\alpha(x-m)^2}{\sigma^2}\right) = 0. \tag{13.21}$$

Démonstration. The proof is by induction. The result is true for $n = 0$ since $g_u^{(0)}(x)$ is given by Equation (13.3).

If the result is true for n, then for fixed u, $g_u(x)$ belongs to the vector space spanned by the functions φ_i for $i = 0, \ldots, n$, denoted by $\mathrm{span}(\varphi_0, \ldots, \varphi_n)$. From Fokker-Planck Equation (13.7), we have

$$(A \otimes \mathbb{I} + \mathbb{I} \otimes B) g^{(n+1)} = \Psi \otimes D g^{(n)},$$

where the operator Ψ is the multiplication by x in the Hilbert space H. By using the recurrence relation satisfied by Hermite polynomials, it is easily checked that the image by the operator Ψ of the vector space $\mathrm{span}(\varphi_0, \ldots, \varphi_n)$ is the vector space $\mathrm{span}(\varphi_0, \ldots, \varphi_{n+1})$. Therefore, since by assumption $g^{(n)}$ belongs to $\mathrm{span}(\varphi_0, \ldots, \varphi_n) \otimes L^2_\rho(\mathbb{N})$, we immediately deduce from the uniqueness of the decomposition on the basis (φ_n) that $g^{(n+1)}$ is in $\mathrm{span}(\varphi_0, \ldots, \varphi_{n+1}) \otimes L^2_\rho(\mathbb{N})$ and the result follows. □

First order term

In a first step, we pay special attention to the derivation of the first order term because it gives the basic arguments to derive higher order terms. Moreover, the explicit form of the first order term will be used to examine the validity of the reduced service rate approximation (see Theorem 1).

On the basis of the domination property given by Lemma 4, we explicitly compute the function $g_u^{(1)}(x)$. From Equation (13.7), it is easily checked that the function $g_u^{(1)}(x)$ satisfies the equation

$$\begin{aligned} \frac{\sigma^2}{2}\frac{\partial^2 g_u^{(1)}}{\partial x^2} + \alpha(x-m)\frac{\partial g_u^{(1)}}{\partial x} + \alpha(\nu(u)+1) g_u^{(1)}(x) & \\ = \mu\left(\frac{1}{u} - 1\right)\left(g_0^{(1)}(x) - x(g_0^{(0)}(x) - g_u^{(0)}(x))\right) & \\ = \mu\left(\frac{1}{u} - 1\right)\left(g_0^{(1)}(x) + \frac{x}{\sigma}\sqrt{\frac{\alpha}{\pi}}\frac{\rho u(1-\rho)}{(1-\rho u)} \exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right)\right) & \end{aligned} \tag{13.22}$$

where the constant $\nu(u)$ is given by

$$\nu(u) = \frac{\mu(1-u)(1-\rho u)}{\alpha u}.$$

In a first step, we search for a particular solution to the ordinary differential equation

$$\frac{\sigma^2}{2}\frac{\partial^2 \xi_u}{\partial x^2} + \alpha(x-m)\frac{\partial \xi_u}{\partial x} + \alpha(\nu(u)+1)\xi_u(x) = \frac{x}{\sigma}\sqrt{\frac{\alpha}{\pi}}\frac{\rho\mu(1-u)(1-\rho)}{(1-\rho u)}\exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right) \quad (13.23)$$

of the form

$$\xi_u(x) = (a(u)+b(u)x)\exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right).$$

Straightforward manipulations show that

$$b(u) = \frac{1}{\sigma}\sqrt{\frac{1}{\alpha\pi}}\frac{\rho\mu(1-u)(1-\rho)}{(\nu(u)-1)(1-\rho u)}\exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right) \quad \text{and} \quad a(u) = -\frac{m}{\nu(u)}b(u).$$

Noting that $\xi_0(x) \equiv 0$, it follows that if we write $g_u^{(1)}(x) = \xi_u(x) + \psi_u(x)$, then the function $\psi_u(x)$ is solution to the equation

$$\frac{\sigma^2}{2}\frac{\partial^2 \psi_u}{\partial x^2} + \alpha(x-m)\frac{\partial \psi_u}{\partial x} + \alpha(\nu(u)+1)\psi_u(x) = \mu\left(\frac{1}{u}-1\right)\psi_0(x). \quad (13.24)$$

By using the domination property of Lemma 4, we can determine the form of the function $\psi_0(x)$.

Lemma 5. *The function $\psi_0(x)$ is given by*

$$\psi_u(x) = \left(c_0 + c_1\frac{\sqrt{\alpha}(x-m)}{\sigma}\right)\exp\left(-\frac{\alpha(x-m)^2}{\sigma^2}\right)$$

for some constants c_0 and c_1.

Démonstration. By introducing the function $k_u(x)$ defined by

$$k_u(x) = \exp\left(\frac{\alpha(x-m)^2}{2\sigma^2}\right)\psi_u(x) \quad (13.25)$$

and then the change of variable

$$z = \frac{\sqrt{\alpha}(x-m)}{\sigma}, \quad (13.26)$$

Equation (13.24) becomes

$$\frac{\partial^2 k_u}{\partial z^2} + (2\nu(u)+1-z^2)k_u(z) = \frac{2\mu}{\alpha}\left(\frac{1}{u}-1\right)k_0(z). \quad (13.27)$$

The homogeneous equation reads

$$\frac{\partial^2 k_u}{\partial z^2} + (2\nu(u)+1-z^2)k_u = 0,$$

which solutions are parabolic cylinder functions (see Lebedev [**?**] for details). Two independent solutions $v_1(u;z)$ and $v_2(u;z)$ of this homogeneous equation are given in terms of Hermite functions as

$$v_1(u;z) = e^{-z^2/2} H_{\nu(u)}(z) \quad \text{and} \quad v_2(u;z) = e^{z^2/2} H_{-\nu(u)-1}(iz). \tag{13.28}$$

The Wronskian $\mathcal{W}$ of these two functions is given by

$$\mathcal{W}(z) = e^{-(\nu+1)\pi i/2}.$$

By using the method of variation of parameters, the solution to Equation (13.27) is given by

$$\begin{aligned} k_u(z) &= \gamma_1(u) v_1(u;z) + \gamma_2(u) v_2(u;z) \\ &\quad - \frac{2\mu}{\alpha}\left(\frac{1}{u} - 1\right) e^{(\nu+1)\pi i/2} \int_0^z \left[v_1(u;y) v_2(u;z) - v_1(u;z) v_2(u;y)\right] k_0(y)\, dy, \end{aligned}$$

where $\gamma_1(u)$ and $\gamma_2(u)$ are constants, which depend upon u.

The function $\psi_u(x)$ enjoys the same domination property as function $g_u^{(1)}(x)$, given by Lemma 4. Hence, for $N > 1$

$$\lim_{z \to \pm\infty} \frac{1}{z^N} e^{z^2/2} k_u(z) = 0. \tag{13.29}$$

From Lebedev [**?**], we have the following asymptotic estimates

$$H_\nu(z) \sim (2z)^\nu \tag{13.30}$$

when $|z| \to \infty$ and $|\arg z| \leq 3\pi/4 - \delta$ for some $\delta > 0$. Moreover, when $z \to -\infty$

$$H_\nu(z) \sim \begin{cases} \dfrac{\sqrt{\pi}}{\Gamma(-\nu)} |z|^{-\nu-1} e^{z^2}, & \nu \notin \mathbb{N} \\ \\ (2z)^\nu, & \nu \in \mathbb{N}. \end{cases}$$

From the above asymptotic estimates and Lemma (13.29), we deduce that for $u \in (0,1)$ such that $\nu(u) \in \mathbb{N}$ with $\nu > 1$, we have

$$\begin{aligned} \gamma_1(u) &= -\frac{2\mu}{\alpha}\left(\frac{1}{u} - 1\right) e^{(\nu+1)\pi i/2} \int_0^\infty v_2(u;y) k_0(y)\, dy \\ &= -\frac{2\mu}{\alpha}\left(\frac{1}{u} - 1\right) e^{(\nu+1)\pi i/2} \int_0^{-\infty} v_2(u;y) k_0(y)\, dy \end{aligned}$$

and

$$\begin{aligned} \gamma_2(u) &= \frac{2\mu}{\alpha}\left(\frac{1}{u} - 1\right) e^{(\nu+1)\pi i/2} \int_0^\infty v_1(u;y) k_0(y)\, dy \\ &= \frac{2\mu}{\alpha}\left(\frac{1}{u} - 1\right) e^{(\nu+1)\pi i/2} \int_0^{-\infty} v_1(u;y) k_0(y)\, dy. \end{aligned}$$

The latter equation implies that for all $n > 1$

$$\int_{-\infty}^{\infty} e^{-y^2/2} k_0(y) H_n(y)\, dy = 0, \tag{13.31}$$

where $H_n(x)$ is the nth Hermite polynomial. Property (13.29) implies that the function $y \to \exp(y^2/2)k_0(y)$ is in $L^2(\mathbb{R}, \exp(-y^2)\,dy)$, which is the Hilbert space of the functions square integrable with respect to the measure $\exp(-y^2)\,dy$, i.e.,

$$L^2(\mathbb{R}, \exp(-y^2)\,dy) = \left\{ f : \int_{-\infty}^{\infty} |f(y)|^2 e^{-y^2}\,dy < \infty \right\},$$

equipped with the scalar product

$$< f, g >_2 = \int_{-\infty}^{\infty} f(y)\overline{g(y)}e^{-y^2}\,dy.$$

Since Hermite polynomials form an orthogonal basis in this Hilbert space, equation (13.31) entails that the function $y \to \exp(y^2/2)k_0(y)$ is orthogonal to all Hermite polynomials H_n with $n > 1$ and then that this function belongs to the vector space spanned by H_0 and H_1. Hence, function $k_0(z)$ should be of the form

$$k_0(z) = (c_0 + c_1 z)e^{-z^2/2}$$

for some constants c_0 and c_1 and the result follows. □

By using the above lemma, we are now able to establish the expression of $g_u^{(1)}(x)$.

Proposition 22. *The function $g_u^{(1)}(x)$ is given by*

$$g_u^{(1)}(x) = \left(\frac{(u_1 - 1)(\tilde{u}_1 - \rho u)(u - 1)}{u_1 \tilde{u}_1 (u - \tilde{u}_1)(1 - \rho u_1)(1 - \rho u)^2} + \frac{(1 - \rho)(1 - u)}{(u - \tilde{u}_1)(1 - \rho u_1)(1 - \rho u)} x \right) \frac{1}{\sigma} \sqrt{\frac{\alpha}{\pi}} e^{\alpha(x-m)^2/\sigma^2}, \quad (13.32)$$

where u_1 and $\tilde{u}_1$ are the two real solutions to the quadratic equation

$$\rho u^2 - \left(1 + \rho + \frac{\alpha}{\mu}\right) u + 1 = 0$$

with $0 < u_1 < 1 < \tilde{u}_1$

Démonstration. By taking into account Lemma 5, the function $K_u(z)$ defined by

$$K_u(x) = g_u^{(1)}(x) \exp\left(\frac{\alpha(x - m)^2}{2\sigma^2}\right)$$

and the change of variable (13.26), satisfies the equation

$$\frac{\partial^2 K_u}{\partial z^2} + (2\nu(u) + 1 - z^2)K_u(z) = \frac{2\mu}{\alpha}\left(\frac{1}{u} - 1\right)\left(c_0 + c_1 z + \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}}\frac{\rho u(1 - \rho)}{1 - \rho u}\left(\frac{\sigma z}{\sqrt{\alpha}} + m\right)\right) e^{-z^2/2}. \quad (13.33)$$

We search for a particular solution of the form

$$K_u(z) = (a(u) + b(u)z)\, e^{-z^2/2}.$$

Straightforward computations yield

$$a(u) = \frac{1}{(1-\rho u)}\left(c_0 + \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}}\frac{\rho u(1-\rho)}{1-\rho u}m\right),$$
$$b(u) = \frac{(1-u)}{\rho(u-u_1)(u-\tilde{u}_1)}\left(c_1 + \frac{\rho u(1-\rho)}{\sqrt{\pi}(1-\rho u)}\right).$$

It follows that the general solution to the above equation can be written as

$$K_u(z) = (a(u)+b(u)z)\,e^{-z^2/2} + \gamma_1(u)v_1(u;z) + \gamma_2(u)v_2(u;z), \tag{13.34}$$

where the functions v_1 and v_2 are defined by Equation (13.28) and the constants $\gamma_1(u)$ and $\gamma_2(u)$ depend upon u.

By differentiating once Equation (13.34) with respect to z and using the fact that the Wronskian of the functions $v_1(u;z)$ and $v_2(u;z)$ is $\exp[(\nu(u)+1)\pi i/2]$, we can easily express $\gamma_1(u)$ and $\gamma_2(u)$ by means of $K_u(z)$, $a(u)$, and $b(u)$. This shows that $\gamma_1(u)$ and $\gamma_2(u)$ are analytic in the open unit disk deprived of the points 0 and u_1. From the asymptotic properties satisfied by the functions v_1 and v_2, we know that $\gamma_1(u) = 0$ and $\gamma_2(u) = 0$ for u such that $\nu(u) > 1$. It follows that $\gamma_1(u) \equiv \gamma_2(u) \equiv 0$ for $|u| < 1$.

By using the fact that $g_u^{(1)}(x)$ has to be analytic in variable u is the unit disk, we necessarily have

$$c_1 = -\frac{\rho u_1(1-\rho)}{\sqrt{\pi}(1-\rho u_1)}$$

and then,

$$b(u) = \frac{(1-\rho)(1-u)}{\sqrt{\pi}(u-\tilde{u}_1)(1-\rho u_1)(1-\rho u)}.$$

Moreover, since $g_1^{(1)}(x) \equiv 0$, we have

$$c_0 = -\frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}}\rho m$$

and then,

$$a(u) = \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}}\frac{\rho(u-1)}{(1-\rho u)^2}m.$$

By using the expressions of $a(u)$ and $b(u)$, the result follows. □

Higher order terms

We assume that $g_u^{(n)}(x)$ can be expressed as

$$g_u^{(n)}(x) = \sum_{k=0}^{n} c_{n,k}(u)\varphi_k(x), \tag{13.35}$$

where the function φ_n is defined by Equation (13.19) and the coefficients $c_{n,k}$ are analytic functions in variable u. From previous sections, this representation is valid for $n = 0, 1$. If it is valid for $n-1$, then the function $g_u^{(n)}(x)$, $n \geq 1$, satisfies the equation

$$\frac{\sigma^2}{2}\frac{\partial^2 g_u^{(n)}}{\partial x^2} + \alpha(x-m)\frac{\partial g_u^{(n)}}{\partial x} + \alpha(\nu(u)+1)g_u^{(n)}(x) = \mu\left(\frac{1}{u}-1\right)\left(g_0^{(n)}(x) - x(g_0^{(n-1)}(x) - g_u^{(n-1)}(x))\right). \quad (13.36)$$

First note that by using the recurrence relation

$$H_{n+1}(x) - 2xH_n(x) + 2nH_{n-1}(x) = 0$$

satisfied by Hermite polynomials, it is easily checked that

$$x(g_u^{(n-1)}(x) - g_0^{(n-1)}(x)) = \sum_{k=0}^{n} d_{n,k}(u)\varphi_k(x),$$

where

$$d_{n,n}(u) = \frac{\sigma}{2\sqrt{\alpha}}(c_{n-1,n-1}(u) - c_{n-1,n-1}(0)),$$

and for $0 \leq k \leq n-1$,

$$d_{n,k}(u) = \frac{\sigma}{2\sqrt{\alpha}}(c_{n-1,k-1}(u) - c_{n-1,k-1}(0)) + m(c_{n-1,k}(u) - c_{n-1,k}(0)) + \frac{(k+1)\sigma}{\sqrt{\alpha}}(c_{n-1,k+1}(u) - c_{n-1,k+1}(0)).$$

By using the above notation, we have the following result.

Proposition 23. *The coefficients $c_{n,k}$ appearing in the representation (13.35) of $g_u^{(n)}(x)$ are recursively defined as follows : we have*

$$c_{0,0}(u) = \frac{1}{\sigma}\sqrt{\frac{\alpha}{\pi}}\frac{1-\rho}{1-\rho u},$$

and for $n \geq 1$,

$$\begin{aligned} c_{n,0}(u) &= \frac{d_{n,0}(u) - d_{n,0}(1)}{1-\rho u}, \\ c_{n,k}(u) &= \frac{\mu}{\alpha}\left(\frac{1}{u}-1\right)\frac{d_{n,k}(u) - d_{n,k}(u_k)}{\nu(u)-k} \quad 1 \leq k \leq n, \end{aligned}$$

where for $k \geq 1$, u_k and $\tilde{u}_k$ are the two real solutions to the quadratic equation $\nu(u) = k$, i.e.

$$\rho u^2 - \left(1 + \rho + \frac{k\alpha}{\mu}\right)u + 1 = 0$$

with $0 < u_k < 1 < \tilde{u}_k$.

Démonstration. As in the previous section, we first search for a solution to the equation

$$\frac{\sigma^2}{2}\frac{\partial^2 \xi_u^{(n)}}{\partial x^2} + \alpha(x-m)\frac{\partial \xi_u^{(n)}}{\partial x} + \alpha(\nu(u)+1)\xi_u^{(n)}(x) = \mu\left(\frac{1}{u}-1\right)x(g_u^{(n-1)}(x) - g_0^{(n-1)}(x)).$$

Assuming that the function $\xi_u^{(n)}(x)$ is of the form

$$\xi_u^{(n)}(x) = \sum_{k=0}^{n} \delta_{n,k}(u)\varphi_n(x),$$

and we have, by using the fact that the functions $\varphi_n(x)$ are eigenfunctions of the operator A associated with the eigenvalues $-\alpha n$ and that these functions are linearly independent, for $k = 0, \ldots, n$,

$$\delta_{n,k} = \frac{\mu}{\alpha}\left(\frac{1}{u}-1\right)\frac{d_{n,k}(u)}{\nu(u)-k}.$$

It is easily checked that $\xi_0^{(n)}(x) \equiv 0$. We can then decompose $g_u^{(n)}(x)$ as

$$g_u^{(n)}(x) = \psi_u^{(n)}(x) + \xi_u^{(n)}(x),$$

where the function $\psi_u^{(n)}(x)$ is solution to the equation

$$\frac{\sigma^2}{2}\frac{\partial^2 \psi_u^{(n)}}{\partial x^2} + \alpha(x-m)\frac{\partial \psi_u^{(n)}}{\partial x} + \alpha(\nu(u)+1)\psi_u^{(n)}(x) = \mu\left(\frac{1}{u}-1\right)\psi_0(x).$$

By using the same arguments as in the proof of Lemma 5, we can easily show that $\psi_0(x)$ has the form

$$\psi_0(x) = \sum_{k=0}^{n} c_k \varphi_n(x),$$

where the coefficients $c_k \in \mathbb{C}$ for $k = 0, ..., n$. It follows that the function $g_u^{(n)}(x)$ is solution to the ordinary differential equation

$$\frac{\sigma^2}{2}\frac{\partial^2 g_u^{(n)}}{\partial x^2} + \alpha(x-m)\frac{\partial g_u^{(n)}}{\partial x} + \alpha(\nu(u)+1)g_u^{(n)}(x) = \mu\left(\frac{1}{u}-1\right)\sum_{k=0}^{n}(c_k + d_{n,k}(u))\varphi_n(x).$$

By using the same arguments as in the proof of Proposition 22, we come up with the conclusion that $g_u^{(n)}(x)$ is of the form (13.35) with the coefficients $c_{n,k}(u)$ given by

$$c_{n,k}(u) = \frac{\mu}{\alpha}\left(\frac{1}{u}-1\right)\frac{c_k + d_{n,k}(u)}{\nu(u)-k}.$$

Since the function $g_u^{(n)}(x)$ has to be analytic in the open unit disk, we have for $k \geq 1$

$$c_k = -d_{n,k}(u_k)$$

In addition, since $g_1^{(n)}(x) \equiv 0$, we have $c_0 = -d_{n,k}(1)$. □

Radius of convergence

In this section, we examine under which conditions the expansion (13.20) defines an element of $\mathcal{H}$. In a first step, note that as a consequence of Proposition 23, the function $g_u^{(n)}(x)$ can be written as

$$g_u^{(n)}(x) = x\Theta\left(g_u^{(n-1)}(x)\right) = \Theta\left(xg_u^{(n-1)}(x)\right)$$

where the operator Θ is defined in $\mathcal{H}$ as follows : for $f \in \mathcal{H}$, which gives rise to the function

$$f_u(x) = \sum_{n=0}^{\infty} c_n(u)\varphi_n(x),$$

the element $h = \Theta f$ is defined by the function

$$h_u(x) = \sum_{n=0}^{\infty} \mu\left(\frac{1}{u} - 1\right) \frac{c_n(u) - c_n(u_n)}{\nu(u) - n}\varphi_n(x).$$

It is easily checked that for $n \geq 0$, $0 < u_n < 1 < 1/\sqrt{\rho} < \tilde{u}_n$. Moreover, the function $c_n(u)$ appearing in the expression of f_u is analytic in the disk $D_\rho = \{z : |z| < 1/\sqrt{\rho}\}$ and continuous in the closed disk $\overline{D}_\rho = \{z : |z| \leq 1/\sqrt{\rho}\}$ for $n \geq 0$. Similarly, for all $n \geq 0$, the function

$$u \to \frac{\mu}{\alpha}\left(\frac{1}{u} - 1\right) \frac{c_n(u) - c_n(u_n)}{\nu(u) - n}\varphi_n(x)$$

is analytic in D_ρ and continuous in $\overline{D}_\rho$. With the above notation, we can state the main result of this section.

Proposition 24. *The operator Θ is bounded and if $\varepsilon < 1/(m\|\Theta\|)$, where $\|\Theta\|$ denotes the norm of Θ, then the sequence defined by Equation (13.20) is in $\mathcal{H}$.*

Démonstration. Let $f \in \mathcal{H}$ be defined by the function

$$f_u(x) = \sum_{n=0}^{\infty} c_n(u)\varphi_n(x).$$

For $(c_n) \in L^2_\rho(\mathbb{N})$ associated with the generating function

$$c(u) = \sum_{n=0}^{\infty} c_n u^n \text{ then } \|c\|^2_\rho = \frac{1}{2\pi}\int_0^{2\pi} \left|c\left(\frac{1}{\sqrt{\rho}}e^{i\theta}\right)\right|^2 d\theta,$$

and define the sequence $(\tilde{c}_n)$ associated with the generating function

$$\tilde{c}(u) = \frac{\mu}{\alpha}\left(\frac{1}{u} - 1\right)\frac{c(u) - c(u_n)}{\nu(u) - n}.$$

Assume first that $n \geq 1$, then

$$\tilde{c}(u) = \frac{1}{\rho}(1 - u)\frac{1}{u - \tilde{u}_n}\frac{c(u) - c(u_n)}{u - u_n}.$$

and then

$$\|\tilde{c}\|_\rho^2 \leq \frac{1}{\rho^2}\left(1+\frac{1}{\sqrt{\rho}}\right)^2 \frac{1}{(\tilde{u}_n - 1/\sqrt{\rho})^2}\frac{1}{2\pi}\int_0^{2\pi}\left|\frac{c(e^{i\theta}/\sqrt{\rho}) - c(u_n)}{e^{i\theta}/\sqrt{\rho} - u_n}\right|^2 d\theta.$$

Simple manipulations show that

$$\frac{1}{2\pi}\int_0^{2\pi}\left|\frac{c(e^{i\theta}/\sqrt{\rho}) - c(u_n)}{e^{i\theta}/\sqrt{\rho} - u_n}\right|^2 d\theta \leq \|c\|_\rho^2 \frac{1}{(1/\sqrt{\rho} - u_n)^2}\left(1+\sqrt{\frac{1}{1-\rho u_n^2}}\right)^2.$$

It follows that $\|\tilde{c}\|_\rho \leq \kappa_n \|c\|_\rho$, where

$$\begin{aligned}\kappa_n &= \frac{1}{\rho}\left(1+\frac{1}{\sqrt{\rho}}\right)\frac{1}{(\tilde{u}_n - 1/\sqrt{\rho})(1/\sqrt{\rho} - u_n)}\left(1+\sqrt{\frac{1}{1-\rho u_n^2}}\right)\\ &= \frac{1+\sqrt{\rho}}{(1-\sqrt{\rho})^2 + \frac{n\alpha}{\mu}}\left(1+\sqrt{\frac{1}{1-\rho u_n^2}}\right).\end{aligned}$$

It is easily checked that the sequence (κ_n) for $n \geq 1$ is decreasing.

When $n = 0$, we define

$$\tilde{c}(u) = \frac{\mu}{\alpha}\left(\frac{1}{u} - 1\right)\frac{c(u) - c(1)}{\nu(u)} = \frac{c(u) - c(1)}{1 - \rho u}.$$

It is then easily checked that $\|\tilde{c}\|_\rho \leq \kappa_0 \|c\|_\rho$, where

$$\kappa_0 = \frac{1}{1-\sqrt{\rho}}\left(1+\sqrt{\frac{1}{1-\rho}}\right).$$

Define $\kappa = \max\{\kappa_0, \kappa_1\}$. The above computations show that for all $f \in L^2_\rho(\mathbb{N})$, $\|\Theta f\| \leq \kappa \|f\|$. It follows that the operator Θ is bounded; its norm is denoted by $\|\Theta\| \stackrel{\text{def.}}{=} \inf\{k > 0 : \forall f \in \mathcal{H},\ \|\Theta f\| \leq k\|f\|\}$. The above computations shows that

$$\|\Theta\| \leq \frac{1+\sqrt{\rho}}{(1-\sqrt{\rho})^2}\left(1+\sqrt{\frac{1}{1-\rho}}\right).$$

We immediately deduce that the sequence $c^{(n)} = (c^{(n)}_{k,\ell})$ associated with the function $g^{(n)}_u(x)$, in the sense that

$$g^{(n)}_u(x) = \sum_{k=0}^{n}\sum_{\ell=0}^{\infty} c_{k,\ell} u^\ell \varphi_k(x),$$

is such that

$$\|c^{(n)}\| \leq \|\Theta\|^n \|c^{(0)*n}\|$$

where the sequence $c^{(0)*n}$ is associated with the function

$$\frac{1-\rho}{1-\rho u} x^n p(x),$$

where the function $p(x)$ is defined by Equation (13.2).

Straightforward computations show that

$$\|c^{(0)*n}\|^2 = \left(\frac{\sigma}{2\sqrt{\alpha}}\right)^{2n} H_{2n}\left(\frac{\sqrt{\alpha}m}{\sigma}\right),$$

where $H_n(x)$ is the nth Hermite polynomial. Using the asymptotic estimate (13.30), we have

$$\|c^{(0)*n}\| \sim m^n$$

when $n \to \infty$. It follows that $\|c^{(n)}\| \leq a_n$ with $a_n \sim (\|\Theta\|m)^n$ as n tends to infinity. It follows that the sequence defined by the expansion (13.20) is convergent in $\mathcal{H}$ if $\varepsilon\|\Theta\|m < 1$. □

By using all the above results, we are now ready to prove the validity of the reduced service rate approximation given by Theorem 1.

Proof of Theorem 1. From the above result, we deduce that, under the assumption $\varepsilon < 1/(m\|\Theta\|)$, the first order expansion of the generating function of the stationary distribution of $(L(t))$

$$\mathbb{E}\left(u^L\right) = \frac{1-\rho}{1-\rho u} - \frac{\rho(1-u)}{(1-\rho u)^2} m\varepsilon + o(\varepsilon).$$

holds. Theorem 1 is proved. □

13.6 Concluding remarks

The perturbation results presented in this paper have been obtained for a particular form of the perturbation function $\phi(x)$. Of course, the same approach could be extended to more complicated perturbation functions of the form $\phi(x) = 1 - \varepsilon p(x)$ for some function $p(x)$. The key point consists of determining how the operator corresponding to the multiplication by $p(x)$ acts on the basic functions φ_n. For computing explicit expressions, however, the main difficulty is in solving the differential equations satisfied by the coefficients of the expansion. When $p(x)$ is a polynomial, a particular solution to the equations similar to Equations (13.22) and (13.36) is obtained in the form of a polynomial times the function $\exp(-\alpha(x-m)^2/\sigma^2)$ and in that case, explicit computations can be carried out.

The perturbation function $\phi(x) = 1 - \varepsilon x$ correspond to the case when unresponsive flows have a peak bit rate ε much smaller than the transmission capacity of the link. The results of this paper show that the reduced service rate approximation yields accurate results for the performance of responsive flows.

Chapitre 14

Integration of streaming services and TCP data transmission in the Internet

14.1 Introduction

The emergence of the Internet as the universal multi-service network raises major traffic engineering problems, in particular with regard to the coexistence on the same transmission links of real time and data services. As a matter of fact, these two types of services have different requirements in terms of transfer delay and loss, data transmission being very sensitive to packet loss but relatively tolerant to delay whereas real time services have strict transfer delay constraints. While classical data transfers are usually controlled by TCP (Transmission Control Protocol), which aims at achieving a fair bandwidth allocation at a bottleneck link (see Massoulié and Roberts [**?**] for a discussion on modeling TCP at the flow level and processor sharing), real time services most of the time are supported by the unreliable UDP protocol, even if some transmission control can be performed by upper layers (e.g., RTCP). Real time services thus reduce the transmission capacity for data transfers.

This problem has been addressed by Delcoigne *etal.* [**?**], where stochastic bounds have been obtained for the bit rate seen by a TCP data transfer, when elastic traffic and unresponsive streaming flows are multiplexed on a same link (see also Bonald and Proutière [**?**]). From a theoretical point of view, this problem can be seen as the analysis of a priority system, where streaming flows have priority over data traffic. In this context, a usual approximation (referred to as Reduced Service Rate, RSR) consists of assuming that everything happens as if the service rate for data were reduced up to the bit rate offered by streaming flows. This approximation has been investigated for the number of active data flows by Antunes *etal.* [**?**], when the load offered by streaming flows is very small. We note that the same kind of problem has been addressed in the technical literature by Núñez-Queija and Boxma [**?**] in the context of ABR service in ATM networks and more recently by Núñez-Queija [**?**, **?**] via matrix analysis for systems described by means of quasi birth and death processes. In a similar context, Núñez-Queija *etal.* [**?**] use a perturbation technique for studying a priority system,

where priority traffic offers a small load. Finally, note that systems with different speeds are also of interest for analyzing the coexistence of different traffic types [**?**].

In this paper, we investigate the mean bit rate obtained by a data transfer when elastic traffic and unresponsive streaming flows are multiplexed on a same transmission link. Along the same line of investigations as Antunes *etal.* [**?**], because of the *real difficulty* of the problems, the mean load offered by streaming flows is supposed to be very small (controlled by a parameter $\varepsilon \ll 1$) and a perturbation analysis for the analysis of the mean bit rate is done. It is assumed that flows arrive according to a Poisson process and share the available bandwidth according to the processor sharing discipline. In addition, to simplify the computations, we assume that the service time required by data transfers is exponentially distributed with parameter μ. Thus, we have to deal with an $M/M/1$ queue with a time-varying server rate, which depends upon the instantaneous number of active streaming flows.

To compute the mean bit rate of a data transfer, we consider the quantity $\mathcal{A} = \int_0^B L(s)ds$, where B is the length the busy period and $L(t)$ is the number of customers at time t in the $M/M/1$ queue under consideration. The quantity $\mathcal{A}$ is equal to the cumulative waiting time in the $M/M/1$ queue and also represents the amount of data served during a busy period. In the case of the $M/M/1$ PS sharing queue, if $\mathbb{E}[d]$ represents the mean bit rate obtained by a data transfer, we have $\mathbb{E}[\mathcal{A}] = \mathbb{E}[N]\mathbb{E}[S]/\mathbb{E}[d] = \mathbb{E}[B]/\mathbb{E}[d]$, where $\mathbb{E}[S]$ is the mean service time (equal to $1/\mu$) and $\mathbb{E}[N]$ is the mean number of customers served in a busy period. Thus, the computation of $\mathbb{E}[\mathcal{A}]$ allows us to estimate $\mathbb{E}[d]$, since the quantity $\mathbb{E}[B]$ has been computed by Antunes *etal.* [**?**] as a power series expansion of ε. Note that in the case of a classical $M/M/1$ queue, we have $E[B] = 1/(\mu(1-\rho))$ and $\mathbb{E}[\mathcal{A}] = 1/(\mu(1-\rho)^2)$, which yields $\mathbb{E}[d] = (1-\rho)$, where ρ is the offered load. This is the classical result for an $M/G/1$ PS queue, which states that the mean bit rate obtained by a data transfer is $(1-\rho)$ times the server rate (taken as unity in this paper) ; see Massoulié and Roberts [**?**].

In this paper, we derive a power series expansion in ε of the quantity $\mathbb{E}[\mathcal{A}]$ in the case of an $M/M/1$ queue, whose server rate is modulated by an auxiliary process (X_t). We specifically assume that the server rate at time t is $(1+\varepsilon p(X_t)/\mu)$ for some function p satisfying regularity assumptions, the process (X_t) being stationary, ergodic, and Markovian. The objective of this paper is, first to check the validity of the RSR approximation, which claims that everything happens as if the server rate were frozen at the value $(1+\varepsilon\mathbb{E}[p(X_0)]/\mu)$ and, second, to get some qualitative insight on the impact of the variability of streaming flows over elastic traffic.

The organization of this paper is as follows : The model is described in Section 14.2, where the main result concerning the power series expansion in ε of $\mathbb{E}[\mathcal{A}]$, which is the key quantity for computing the mean bit rate of a data transfer. In Section 14.3, the main result is applied to obtain the expansion of the mean bit rate of a data transfer and to analyze different special cases. Some concluding re ;arks are presented in Section 14.4. The quite technical proof of the main result is sketched in the Appendix.

14.2 Model description

Throughout this paper we consider a stable $M/M/1$ queue with arrival rate λ and service rate μ ; the load $\rho = \lambda/\mu < 1$. Let $L(t)$ denote the number of customers at time t. The invariant distribution π of $(L(t))$ is geometrically distributed with parameter ρ.

Let B denote the duration of a busy period starting with one customer, that is, $B = \inf\{s \geq 0 : L(s) = 0\}$, given $L(0) = 1$. For $x \geq 1$, let B_x denote the duration of a busy period starting with x customers. Note that $B_1 \stackrel{\text{dist.}}{=} B$. In the following, when the variables B, B_1 and B_1' are used in the same expression, they are assumed to be independent with the same distribution as B.

The quantity $\mathcal{A}$ defined in the Introduction represents the area swept under the occupation process in a busy period. When several busy cycles are considered, the notation $\mathcal{A}_B$ will be used to indicate that the area is calculated for the corresponding busy period of length B. By definition, the relation $\mathcal{A} \geq B$ holds, the excess will be denoted by $\bar{\mathcal{A}} \stackrel{\text{def.}}{=} \bar{\mathcal{A}}_B \stackrel{\text{def.}}{=} \mathcal{A}_B - B$. This queue will be referred to as the standard queue denoted, for short, by S-Queue.

Streaming flows impact data transfers by reducing the amount of available bandwidth. This situation is described by introducing an $M/M/1$ queue with arrival rate λ and varying service rate driven by an ergodic Markov process $(X(t))$ taking value in a state space $\mathcal{S}$. Typically, the state space of the environment is a finite, countable set when $(X(t))$ is a Markov Modulated Poisson Process or $\mathcal{S} = \mathbb{R}$ in the case of a diffusion, for instance an Ornstein-Uhlenbeck process (see Fricker *etal.* [**?**]). The invariant measure of the process $(X(t))$ is denoted by ν. The Markovian notation $\mathbb{E}_x(\cdot)$ will refer only to the initial state x of the Markov process $(X(t))$.

Let $\widetilde{L}^\varepsilon(t)$ be the number of customers of the queue at time t. The process $(\widetilde{L}^\varepsilon(t), X(t))$ is a Markov process. If $X(t) = x$ and $L(t) = n > 0$, then *the service rate is given by* $\mu + \varepsilon p(x)$ for some function $p(x)$ on the state space of the environment $\mathcal{S}$ and some small parameter $\varepsilon \geq 0$. For $t \geq 0$, let us define the quantities $p^+(t) = \max(p(t), 0)$ and $p^-(t) = \max(-p(t), 0)$ so that $p(t) = p^+(t) - p^-(t)$. At time t, the additional capacity is therefore $\varepsilon p^+(X(t))$ and $\varepsilon p^-(X(t))$ is the capacity lost.

In the rest of this paper, we make the two following assumptions :
the function $p(x)$ is bounded (H_1) and $\lambda + \varepsilon \sup(|p(x)| : x \in \mathcal{S}) < \mu$ (H_2). Under assumption (H_2), the queue is stable and the duration $\widetilde{B}^\varepsilon$ of a busy period starting with one customer, $\widetilde{B}^\varepsilon = \inf\{s \geq 0 : \widetilde{L}^\varepsilon(s) = 0 \mid \widetilde{L}^\varepsilon(0) = 1\}$, is a.s. finite. The queue with time-varying service rate as defined above will be referred to as the perturbated queue, denoted, for short, by P-Queue. The case $\varepsilon = 0$ obviously corresponds to the S-Queue. The area for the perturbated queue over a busy cycle is defined as

$$\widetilde{\mathcal{A}}^\varepsilon = \int_0^{\widetilde{B}^\varepsilon} \widetilde{L}^\varepsilon(s)\, ds. \tag{14.1}$$

The basic idea of the perturbation analysis carried out in this paper for the quantity $\widetilde{\mathcal{A}}^\varepsilon$ defined by equation (14.1) is to construct a coupling between the busy periods of the processes $(\widetilde{L}^\varepsilon(t))$ and $(L(t))$. Provided that for both queues the arrival process is the same Poisson process with parameter λ, we add and remove departures as follows.

Additional departures. When $p^+(X(t)) > 0$, there is additional capacity when compared with the S-Queue and more departures can take place. These additional departures are counted by means of a point Process $\mathcal{N}^+ = (t_i^+)$, with $0 < t_1^+ \leq t_2^+ \leq \cdots$, which is a non-homogeneous Poisson process on $\mathbb{R}_+$ with intensity given by $t \to \varepsilon p^+(X(t))$. Conditionally on $(X(t))$, the number of points of $\mathcal{N}^+$ in the interval $[a, b]$ is Poisson with parameter $\varepsilon \int_a^b p^+(X(s))\, ds$. In particular the distribution of the location $t_1^+ \geq 0$ of the first point of $\mathcal{N}^+$ after 0 is given, for

$x \geq 0$, by

$$\mathbb{P}(t_1^+ \geq x) = \mathbb{P}(\mathcal{N}^+([0,x]) = 0) = \mathbb{E}\left[\exp\left(-\varepsilon \int_0^x p^+(X(s))\, ds\right)\right]. \tag{14.2}$$

Removing Departures On the other hand, when $p^-(X(t)) > 0$, the server rate is smaller than in the S-queue. Let $\mathcal{N}_\mu = (t_i)$, a Poisson process with intensity μ on $\mathbb{R}_+$ which represents the non-decreasing sequence of instants when the customer of the S-queue (if not empty) may leave the queue. We denote by $\mathcal{N}^-$ the point process obtained as follows : For $s > 0$, a point at s of the Poisson process $\mathcal{N}_\mu$ is a point of $\mathcal{N}^-$ with probability $\varepsilon p^-(X(s))/\mu$. (Note that this number is ≤ 1 by assumption (H_2).) The point process $\mathcal{N}^-$ is Poisson with intensity $s \to \varepsilon p^-(X(s))$. A point of $\mathcal{N}^-$ is called a marked departure. The points of $\mathcal{N}^-$ are denoted by $0 < t_1^- \leq t_2^- \leq \cdots \leq t_n^- \leq \cdots$. By definition,

$$\mathbb{P}(t_1^- \geq x) = \mathbb{E}\left(\prod_{t_i, t_i \leq x}\left(1 - \frac{\varepsilon p^-(X(t_i))}{\mu}\right)\right), \quad x \geq 0. \tag{14.3}$$

The processes defined above are non homogeneous Poisson processes (see Grandell [**?**] for an account on this topic). The main result of this paper is the expansion in power series of ε up to the second order of $\mathbb{E}(\widetilde{\mathcal{A}}^\varepsilon)$.

Theorem 2. *The second order expansion of the area swept under the occupation process of the perturbated queue during a busy period is given by*

$$\mathbb{E}\left(\widetilde{\mathcal{A}}^\varepsilon\right) = \mathbb{E}\left(\mathcal{A}\right) - \varepsilon \frac{\mathbb{E}[p(X(0))](\lambda+\mu)}{(\mu-\lambda)^3} - \varepsilon^2(a_+ + a_- + a_\pm) + o(\varepsilon^2), \tag{14.4}$$

where the coefficients a_+, a_-, $a_\pm$ are defined below by Equations (21) , (22), and (23), respectively.

14.3 Applications

In this section, as an application of Theorem 2, we evaluate the mean bit rate $\mathbb{E}(d_\varepsilon)$ obtained by an elastic data transfer when the server rate is perturbated by the presence of streaming flows (coming through the term $\varepsilon p(X(t))$ in the service rate of the perturbated $M/M/1$ queue, equal to $\mu + \varepsilon p(X(t))$). As mentioned in the Introduction, we have $\mathbb{E}(\widetilde{\mathcal{A}}^\varepsilon) = \mathbb{E}(\widetilde{B}^\varepsilon)/\mathbb{E}(d_\varepsilon)$. The average of the duration $\widetilde{B}^\varepsilon$ of the corresponding busy period has been studied by Antunes *etal.* [**?**] and can be expanded in power series of ε as follows.

Theorem 3. *The expansion of $\mathbb{E}(\widetilde{B}^\varepsilon)$, the mean duration of a busy period, is given by $\mathbb{E}(\widetilde{B}^\varepsilon) = 1/(\mu-\lambda) - \varepsilon\mathbb{E}_\nu(p(X(0)))/(\mu-\lambda)^2 + (b_- - b_+)\varepsilon^2 + o(\varepsilon^2)$. where b_+ and b_- are given by, with*

the notation of Proposition 30,

$$b_+ = -\frac{1}{\mu}\mathbb{E}\left(\int_0^B (B-v)\,\mathbb{E}_\nu\left(p^+(X(0))p^+(X(v))\right)\,dv\right) - \frac{1}{\mu^2(1-\rho)}\mathbb{E}\left(\sum_{i=1}^{H}\sum_{j=1}^{N_i}\int_0^{A_i} p^+(X(u))p^-(X(D_i^j))\,du\right), \tag{14.5}$$

$$b_- = \frac{1}{\mu^2(1-\rho)}\left(-\mathbb{E}\left(\sum_{i=1}^{N}\int_0^{B+B_1} p^-(X(D_i))p^+(X(s))\,ds\right) + \frac{1}{\mu}\mathbb{E}\left(\sum_{i=1}^{N}\sum_{k=1}^{N'} p^-(X(D_i))p^-(X(B+D'_k))\right)\right). \tag{14.6}$$

By using Equations (14.5) and (14.6) and Theorem 2, straightforward computations show that the quantity $\mathbb{E}(d_\varepsilon)$ can then expanded in power series of ε as follows.

Proposition 25. *The mean bit rate of an elastic data transfer can be expanded in power series of ε as*

$$\mathbb{E}(d_\varepsilon) = 1-\rho+\frac{\rho\mathbb{E}_\nu(p(X(0)))}{\mu}\varepsilon + c\varepsilon^2 + o(\varepsilon^2), \tag{14.7}$$

where the coefficient c is given by

$$c = \mathbb{E}_\nu(p(X(0)))^2\frac{\rho(1+\rho)}{\mu^2(1-\rho)} + \mu(1-\rho)^2\left((1-\rho)(a_+ + a_- + a_\pm) + b_- - b_+\right), \tag{14.8}$$

the quantities a_+, a_-, $a_\pm$, b_+ and b_- being defined by equations (21), (22), (23), (14.5) and (14.6), respectively.

From Equation (14.7), we immediately deduce that as far as the first order term is concerned, the RSR approximation is valid : for an $M/M/1$ queue with service rate $\mu+\varepsilon\mathbb{E}_\nu(p(X(0)))$, the mean bit rate denoted by $\hat{d}$ obtained by a customer is given by $\mathbb{E}(\hat{d}) = 1-\lambda/(\mu+\varepsilon\mathbb{E}_\nu[p(X(0))]) = 1-\rho+\varepsilon\rho\mathbb{E}_\nu[p(X(0))]/\mu+o(\varepsilon)$ Unfortunately, the coefficient c defined by Equation (14.8) intricately depends upon the correlation structure of the modulating process $(X(t))$ and the dynamics of the $M/M/1$ queue. Because of this complexity, three cases of practical interest in the following are considered : non-positive perturbation functions, non-negative perturbation functions, and special environments (namely, fast and slow environments).

Non-positive Perturbation Functions

We assume in this section that the perturbation function is non-positive so that the environment uses a part of the capacity of the $M/M/1$ queue with constant service rate μ. This application is motivated by the following practical situation. Coming back to the coexistence of elastic and streaming traffic in the Internet, assume that priority is given to streaming traffic in a buffer of a router. The bandwidth available for non-priority traffic is the transmission link reduced by the rate of streaming traffic. Denoting by $\varepsilon r(X_t)$ the rate of streaming traffic at time t (for instance ε may

represent the peak rate of a streaming flow and $r(X_t)$ the number of such flows active at time t), the service rate available for non-priority traffic is $\mu - \varepsilon r(X_t)$. Setting $p(x) = -r(x)$, the function $p(x)$ is non-positive.

Proposition 26. *When $p^+ \equiv 0$, if $\hat{d}$ is the mean bit rate of the $M/M/1$ queue with service rate $\mu + \varepsilon\mathbb{E}[p(X(0))]$, then with the notation of Proposition 30,*

$$\lim_{\varepsilon\to 0} \frac{1}{\varepsilon^2}\mathbb{E}\left(d_\varepsilon - \hat{d}\right) = -\frac{(1-\rho)^2}{\mu^2}\mathbb{E}\left(\sum_{1\le i<j\le N} C_p\left(X(D_j - D_i)\right)\right)$$

$$-\frac{(1-\rho)^3}{\mu}\mathbb{E}\left(\sum_{i=1}^{N}\sum_{j=1}^{N'} C_p(B - D_i + D'_j)\left(B_1 - D'_j\right)\right), \quad (14.9)$$

where the function $C_p(u)$ is defined for $u \ge 0$ by

$$C_p(u) = \mathbb{E}_\nu\left[p(X(0))p(X(u))\right] - \mathbb{E}_\nu\left[p(X(0))\right]^2 \quad (14.10)$$

and is, up to the factor ε^2, the auto-covariance function of the extra-capacity of the perturbated queue.

The above result shows that if the process $(X(t))$ is positively correlated, i.e. $C_p(\cdot) \ge 0$, then $\lim_{\varepsilon\to 0}\mathbb{E}\left(d_\varepsilon - \hat{d}\right)/\varepsilon^2 < 0$. The environment has therefore a negative impact on the performances of the system in this case.

Démonstration. The terms a_+, $a_\pm$ and b_+ in the coefficient c defined by Equation (14.8) are equal to 0. In addition, one easily checks that

$$(1-\rho)a_- + b_- = -\frac{1}{\mu^3}\mathbb{E}\left(\sum_{1\le i<j\le N} p^-(X(D_i))p^-(X(D_j))\right)$$

$$-\frac{(1-\rho)}{\mu^2}\mathbb{E}\left(\sum_{i=1}^{N}\sum_{k=1}^{N'} p^-(X(D_i))p^-(X(B + D'_k))(B_1 - D'_j)\right).$$

One concludes by using Expansion (14.7). □

Non-negative Perturbation Functions

It is assumed in this section that $p^- \equiv 0$. We have the following result, which is the analogue of Proposition 26 for this case. Contrary to the above proposition, the expansion has a more explicit expression. Its (straightforward) proof is omitted.

Proposition 27. *When $p^- \equiv 0$, with the same notation as in Proposition 26.*

$$\lim_{\varepsilon\to 0}\frac{1}{\varepsilon^2}\mathbb{E}\left(d_\varepsilon-\hat{d}\right)=(1-\rho)^3\left(\mathbb{E}\left(\int_0^B (B-v)\,C_p(v)\,dv\right)\right.$$
$$\left.-\frac{(1-\rho)\mu}{2}\mathbb{E}\left(\int_0^B (B-v)^2\,C_p(v)\,dv\right)\right) \quad (14.11)$$

An integration by parts and some calculations give the following corollary.

Corollaire 7. *When the correlation function of the environment is exponentially decreasing, i.e., when $C_p(x)=\mathrm{Var}[p(X(0))]\,e^{-\alpha x}$ for all $x\geq 0$ and some $\alpha>0$, then*

$$\lim_{\varepsilon\to 0}\frac{1}{\varepsilon^2}\,\mathbb{E}\left(d_\varepsilon-\hat{d}\right)=\frac{1}{\mu^2}\left(\mathbb{E}\left(e^{-\alpha B^{**}}\right)-\frac{1+\rho}{1-\rho}\mathbb{E}\left(e^{-\alpha B^{***}}\right)\right), \quad (14.12)$$

with the convention that, if Z is some integrable non-negative random variable, the density on $\mathbb{R}_+$ of the variable Z^* is defined as

$$\mathbb{P}\left(Z^*\geq x\right)=\mathbb{P}(Z\geq x)/\mathbb{E}(Z),$$

for $x\geq 0$, and Z^{**} stands for $(Z^*)^*$, and Z^{***} for $(Z^{**})^*$.

when α is small, the right hand side of Equation (14.12) is equivalent to the quantity $-2\rho/\mu^2<0$. This shows that negative perturbation functions have a negative impact at the second order on the mean bit rate of elastic data transfers.

Fast and slow Environments

The performance of the system in two limit regimes, called fast and slow environments, are now evaluated. These regimes are very useful, since performance in the limit regimes is insensitive and only depends on appropriately defined parameters. Such a situation has also been analyzed by Delcoigne *et al* [**?**] through stochastic bounds.

The environment is scaled by a factor $\alpha>0$, such that at time t the environment is supposed to be $X(\alpha t)$. The behavior when α goes to infinity and zero is investigated.

When the parameter α is very large, the environment process approximately averages the capacity of the variable queue. For a large α and for t and $h>0$, the total service capacity available during t and $t+h$ is given by

$$\mu h+\varepsilon\int_t^{t+h} p(X(\alpha u))\,du\overset{\text{dist.}}{=}\mu h+\varepsilon\frac{1}{\alpha}\int_0^{\alpha h} p(X(u))\,du\sim(\mu+\varepsilon\mathbb{E}[p(X(0))])h$$

using the stationarity of $(X(t))$ and the ergodic theorem. Thus, when α tends to infinity, the variations completely vanish and the service rate reduces to a constant.

On the other hand, for small values of α, the environment process remains almost constant over the busy period of the P-Queue. As α goes to 0, the variation disappears and the environment is frozen on the initial state of the process : the service rate is constant and equal to $\mu+\varepsilon p(X(0))$.

This intuitive picture is rigorously established in the next proposition. In the following a general perturbation function p is considered together with some stationary Markov process $(X(t))$ with invariant probability distribution ν. It is assumed that it verifies a mixing condition such as

$$\lim_{t\to+\infty} |\mathbb{E}[f(X(0)g(X(t))] - \mathbb{E}_\nu(f)\mathbb{E}_\nu(g)| = 0 \tag{14.13}$$

for any Borelian bounded functions f and g on the state space $\mathcal{S}$. Note that this condition is not restrictive in general since it is true for any ergodic Markov process with a countable (or finite) state space or for any diffusion on $\mathbb{R}^d$.

Under the above assumptions, we have the following result ; the proof relies on the use of the mixing condition (14.13) and can be found in the paper by Antunes *etal.* [**?**].

Proposition 28. *When the environment is given by $(X(\alpha t))$ and Relation (14.13) holds, then when ε tends to 0, $\mathbb{E}(d_\varepsilon) - \mathbb{E}(\hat{d}) = \Psi(\alpha)\varepsilon^2 + o(\varepsilon^2)$ where*

$$\lim_{\alpha\to+\infty} \Psi(\alpha) = -\rho\frac{\mathbb{E}_\nu(p(X(0)))^2}{\mu^2} \text{ and } \lim_{\alpha\to 0} \Psi(\alpha) = -\rho\frac{\mathbb{E}_\nu\left(p(X(0))^2\right)}{\mu^2}.$$

The fast and slow environment provide an explicit estimate of the second order term, where the slow environment yields a worst performance of the perturbation queue. It is not clear that these limit regimes give a lower and upper bound of the performance of the queue. If the intuition leads to such conclusion, it is in general very difficult to establish it rigorously.

14.4 Conclusion

We have investigated in this paper the impact on the performance of elastic data transfers of the presence of streaming flows, when both kinds of traffic are multiplexed on a same link of an IP network. By assuming that the perturbation due to streaming flows is of small magnitude, a perturbation analysis can be performed in order to obtain *explicit* results for the mean bit rate achieved by a data transfer, under the assumption that elastic streams share the available bandwidth according to the processor sharing discipline. It turns out that at the first order, the so-called RSR approximation is valid. This is not the case for the second order, for which the variability of streaming flows seem to have a negative impact, at least for the three cases examined here.

Further investigations are needed in order to estimate the degradation suffered by data transfers at the second order. The perturbation analysis carried out in this paper is possible, because we have assumed that streaming flows offer a very small contribution to the total load. When this is not the case, new tools have to be developed to estimate the quality of data transfers.

.1 Appendix : Proof of Theorem 2

Since the derivation of the expansion of the average area is somewhat technical, we begin with the simplest case, which is the first order expansion. In the following, we set $\Delta^\varepsilon = \mathbb{E}(\widetilde{\mathcal{A}}^\varepsilon) - \mathbb{E}(\mathcal{A})$.

First order term

We first consider an additional departure. Define $\mathcal{E}_+ = \{t_1^+ \leq B, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}\}$. On this event, an additional departure is added and the busy period of the P-Queue finishes before a departure is canceled. For the first order term of the expansion of the mean value of Δ^ε on the event $\mathcal{E}_+$, we only need to consider the case where there is only one additional departure during the busy period of the P-Queue. The probability that two additional jumps occur in the same busy period is of the order of magnitude of ε^2 since the intensity of the associated Poisson process is proportional to ε.

Lemma 6. *In the case of a single additional departure*

$$\mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\mathcal{E}_+}\right) = \varepsilon \frac{\mathbb{E}_\nu[p^+(X(0))](\lambda+\mu)}{(\mu-\lambda)^3} + o(\varepsilon). \tag{14}$$

Démonstration. The difference Δ^ε on the event $\{t_1^+ \leq B, t_2^+ > t_1^+ + B_{L(t_1^+)-1}, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}\}$ is represented as the sum of two disjoint areas (see Figure .1). The first one is given by the distance between t_1^+ and the end of the busy period of the S-Queue. By the strong Markov property at the stopping time $\widetilde{B}^\varepsilon$, conditionally on the event $\{t_1^+ \leq B, t_2^+ \geq t_1^+ + B_{L(t_1^+)-1}, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}\}$, the S-Queue starts at time $\widetilde{B}^\varepsilon$ an independent busy period with one customer (with duration B_1). The second area of Δ^ε is then given by the area of the *sub-busy periods*, i.e. the periods when L is > 1 in the second busy period.

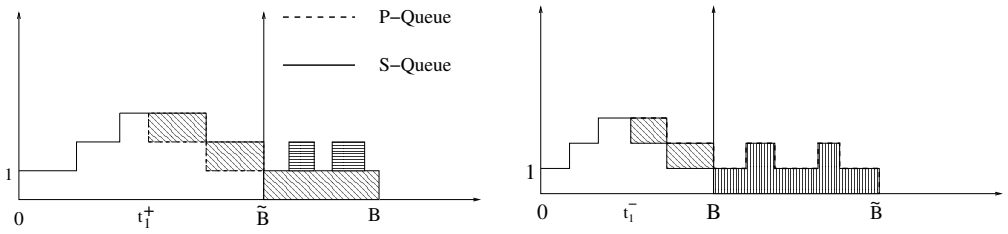


FIG. 1 – One additional departure — One marked departure

It follows that

$$\begin{aligned}
\mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\{t_1^+ \leq B, t_2^+ \geq t_1^+ + B_{L(t_1^+)-1}, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}\}}\right) = \\
\mathbb{E}\left((B - t_1^+)\mathbf{1}_{\{t_1^+ \leq B, t_2^+ \geq t_1^+ + B_{L(t_1^+)-1}, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}\}}\right) \\
+ \mathbb{E}(\mathcal{A}_{B_1} - B_1)\mathbb{P}(t_1^+ \leq B, t_2^+ \geq t_1^+ + B_{L(t_1^+)-1}, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}) \\
= \mathbb{P}(t_1^+ \leq B)\mathbb{E}(\mathcal{A}_{B_1} - B_1) + \mathbb{E}\left((B - t_1^+)\mathbf{1}_{\{t_1^+ \leq B\}}\right) + o(\varepsilon).
\end{aligned} \tag{15}$$

Equation (14.2) and the boundedness of p give that

$$\begin{aligned}\mathbb{P}(t_1^+ \le B) &= 1 - \mathbb{E}\left[\exp\left(-\varepsilon \int_0^B p^+(X(s))\,ds\right)\right] \\ &= \varepsilon\mathbb{E}\left[\int_0^B p^+(X(s))\,ds\right] + o(\varepsilon) = \varepsilon\mathbb{E}(B)\,\mathbb{E}\left[p^+(X(0))\right] + o(\varepsilon) \\ &= \frac{\varepsilon}{\mu-\lambda}\mathbb{E}\left[p^+(X(0))\right] + o(\varepsilon)\end{aligned}$$

by independence between B and $(X(t))$ and by the stationarity of $(X(t))$.

Similarly,

$$\begin{aligned}\mathbb{E}\left((B - t_1^+)\mathbf{1}_{\{t_1^+ \le B\}}\right) &= \mathbb{E}\left(\int_0^B \varepsilon p^+(X(u)) e^{-\varepsilon \int_0^u p^+(X(s))\,ds}(B-u)\,du\right) \\ &= \varepsilon\mathbb{E}[p^+(X(0))]\frac{\mathbb{E}(B^2)}{2} + o(\varepsilon)\end{aligned}$$

Since $\mathbb{E}[B^2] = 2/\mu^2(1-\rho)^3$ and $\mathbb{E}(\mathcal{A}_{B_1}) = \mu/(\mu-\lambda)^2$ (see for instance standard books such as Cohen [**?**]), Equation (14) follows. □

We now turn to the case, when there is one removed departure. On the event $\mathcal{E}_- = \{t_1^- \le B,\ B + B_1 \le t_1^+\}$, a marked departure occurs and no departures are added before the completion of the busy period B_1. We derive the first order expansion of the mean value of Δ^ε on $\mathcal{E}_-$.

Assume that there is only one marked departure and no additional jumps during the busy period of the P-Queue. In this case, at the end of the busy period of the S-Queue, the P-Queue has one customer and the difference is the distance between t_1^- and the end of the busy period B. At time B, the P-Queue starts a busy period with one customer and provided that there are no marked and additional departures during $(B, B + \widetilde{B}^\varepsilon)$, the difference has the same distribution as the area of a busy period B_1 of the standard queue (see Figure .1).

Lemma 7. *In the case of a single marked departure*

$$\mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\mathcal{E}_-}\right) = -\varepsilon\frac{\mathbb{E}[p^-(X(0))](\lambda+\mu)}{(\mu-\lambda)^3} + o(\varepsilon). \tag{16}$$

Démonstration. By using the same arguments as before, one obtains the relation

$$\begin{aligned}&\mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\{t_1^- \le B, t_2^- \ge B+B_1, t_1^+ \ge B+B_1\}}\right) \\ &\qquad = -\mathbb{E}\left((B - t_1^- + \mathcal{A}_{B_1})\mathbf{1}_{\{t_1^- \le B, t_2^- \ge B+B_1, t_1^+ \ge B+B_1\}}\right)\end{aligned} \tag{17}$$

Hence, $\mathbb{E}(\Delta^\varepsilon \mathbf{1}_{\mathcal{E}_-}) = -\mathbb{E}\left((B - t_1^- + \mathcal{A}_{B_1})\mathbf{1}_{\{t_1^- \le B, t_2^- \ge B+B_1, t_1^+ \ge B+B_1\}}\right) + o(\varepsilon)$, so that $\mathbb{E}(\Delta^\varepsilon \mathbf{1}_{\mathcal{E}_-}) = -\mathbb{E}((B - t_1^-)\mathbf{1}_{\{t_1^- \le B\}}) - \mathbb{E}(\mathcal{A}_{B_1})\mathbb{P}(t_1^- \le B) + o(\varepsilon)$.

To estimate $\mathbb{P}(t_1^- \leq B)$, let (D_i) denote the sequence of departures times and N the number of customers served during the busy period of length B, then Equation (14.3) gives the identity

$$\begin{aligned}\mathbb{P}(t_1^- \leq B) &= \mathbb{E}\left(\sum_{i=1}^{N} \frac{\varepsilon p^-(X(D_i))}{\mu} \prod_{j=1}^{i-1}\left(1 - \frac{\varepsilon p^-(X(D_j))}{\mu}\right)\right) \\ &= \frac{\varepsilon}{\mu}\mathbb{E}\left(\sum_{i=1}^{N} p^-(X(D_i))\right) + o(\varepsilon) = \frac{\varepsilon}{\mu}\mathbb{E}(N)\mathbb{E}[p^-(X(D_1))] + o(\varepsilon)\end{aligned}$$

by stationarity of $(X(t))$ and Wald's Formula with $\mathbb{E}(N) = 1/(1-\rho)$. Similarly,

$$\begin{aligned}\mathbb{E}\left((B - t_1^-)\mathbf{1}_{\{t_1^- \leq B\}}\right) &= \frac{\varepsilon}{\mu}\mathbb{E}\left(\sum_{i=1}^{N} p^-(X(D_i))(B - D_i)\right) + o(\varepsilon) \\ &= \varepsilon\frac{\mathbb{E}[p^-(X(D_1))]}{\mu}(\mathbb{E}(NB) - E(D))\end{aligned}$$

where $D = \sum_{i=1}^{N} D_i$ is the sum of the departures in the busy period of the S-Queue. Using the fact that $\mathbb{E}(D) = \mu^2/(\mu-\lambda)^3$, $\mathbb{E}(NB) = (1+\rho)/(\mu(1-\rho)^3)$ and $\mathbb{E}(\mathcal{A}) = \mu/(\mu-\lambda)^2$, the result is proved. □

Combining Lemmas 6 and 7 yields the first order term indicated in Theorem 2.

Second order term

To compute the second order term in the power series expansion in ε of $\mathbb{E}(\Delta^\varepsilon)$, three cases have to be considered :

— $t_1^+ \leq B$ or $t_2^+ \leq B$: one or two additional departures occur in a busy period ;
— $t_1^- \leq B$ or $t_2^- \leq B$: one or two departures are canceled ;
— $t_1^+ \leq B$ and $t_1^- \leq B$: one additional departure takes place and another one is canceled.

It is not difficult to show that any event involving a third jump yields a term of the order ε^3 in the expansion of the mean bit rate. Due to the space constraints, we prove a proof of a part of the expansion. The complete proofs of the expansion can be found in Antunes et al [**?**].

In the case that two additional departures occur during $\widetilde{B}^\varepsilon$, the difference between the areas of the busy periods due to the first additional jump is given by the two first terms on the right hand side of the following equation

$$\begin{aligned}\mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\{t_1^+ \leq B, t_2^+ \leq t_1^+ + B_{L(t_1^+)-1}\}}\right) &= \mathbb{E}\left((B - t_1^+)\mathbf{1}_{\{t_1^+ \leq B, t_2^+ \leq t_1^+ + B_{L(t_1^+)-1}\}}\right) \\ + \mathbb{E}(\bar{\mathcal{A}}_{B_1})\mathbb{P}(t_1^+ \leq B, t_2^+ \leq t_1^+ + B_{L(t_1^+)-1}) &+ \mathbb{E}\left(B_{L(t_2^+)-1}\mathbf{1}_{\{t_1^+ \leq B, t_2^+ < t_1^+ + B_{L(t_1^+)-1}\}}\right) \\ + \mathbb{E}(\bar{\mathcal{A}}_{B_1'})\mathbb{P}(t_1^+ \leq B, t_2^+ < t_1^+ + B_{L(t_1^+)-1}) &+ o(\varepsilon^2), \quad (18)\end{aligned}$$

which follows by the same arguments stated for only one additional departure.

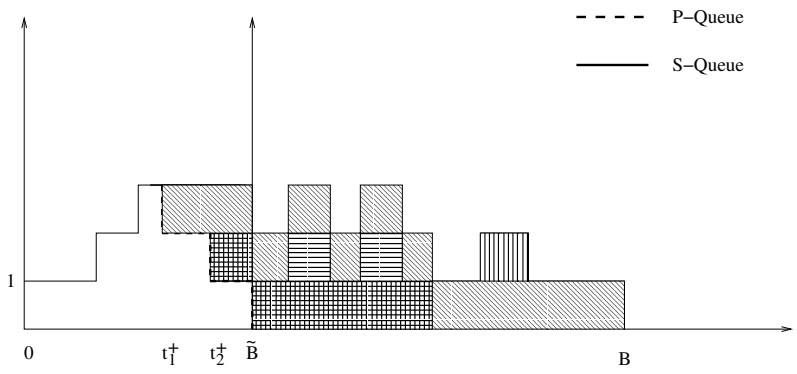


FIG. 2 – Two additional departures

Due to the second additional jump, the difference Δ^ε increases by the sum of two disjoint areas. The first one is given by $B_{L(t_2^+)-1}$ and represent the distance from the second additional jump until the first time the S-Queue with less than one customer is empty. Note that by conditioning on the event $\{t_1^+ \leq B, t_2^+ \leq t_1^+ + B_{L(t_1^+)-1}\}$, at the stopping time $\tilde{B^\varepsilon}$, a new busy period B_1' starts with the same distribution as B. Thus, the second area is given by the area of the sub-busy periods (i.e. periods where $L > 1$) in B_1'. Therefore,

$$\mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\mathcal{E}_+}\right) = \mathbb{E}\left(\Delta^\varepsilon \mathbf{1}_{\{t_1^+ \leq B, t_2^+ \geq t_1^+ + B_{L(t_1^+)-1}, t_1^- \geq t_1^+ + B_{L(t_1^+)-1}\}}\right) + E\left(\Delta^\varepsilon \mathbf{1}_{\{t_1^+ \leq B, t_2^+ \leq t_1^+ + B_{L(t_1^+)-1}\}}\right) + o(\varepsilon^2) \tag{19}$$

$$= \mathbb{E}\left((B - t_1^+)\mathbf{1}_{\{t_1^+ \leq B\}}\right) + \mathbb{E}\left(B_{L(t_2^+)-1}\mathbf{1}_{\{t_1^+ \leq B, t_2^+ < t_1^+ + B_{L(t_1^+)-1}\}}\right) + \mathbb{E}(\bar{\mathcal{A}}_{B_1})\mathbb{P}(t_1^+ < B) + \mathbb{E}(\bar{\mathcal{A}}_{B_1'})\mathbb{P}(t_1^+ \leq B, t_2^+ < t_1^+ + B_{L(t_1^+)-1}) + K(\varepsilon) + o(\varepsilon^2). \tag{20}$$

where $K(\varepsilon)$ is a term which is not expressed here for sake of simplicity. The following proposition gives the expansion of some of the terms of Equation (20). The other expansions are done much in the same way (with various complications).

Proposition 29. *The following expansions hold*

$$\mathbb{P}(t_1^+ \leq B) = \varepsilon \frac{\mathbb{E}[p^+(X(0))]}{\mu - \lambda} - \varepsilon^2 \mathbb{E}\left(\int_0^B (B - v)\, \mathbb{E}\left[p^+(X(0))p^+(X(v))\right]\, dv\right) + o(\varepsilon^2),$$

$$\mathbb{P}(t_1^+ < B, t_2^+ \leq t_1^+ + B_{L(t_1^+)-1}) = \varepsilon^2 \rho \mathbb{E}\left(\int_0^B (B - v)\, \mathbb{E}\left[p^+(X(0))p^+(X(v))\right]\, dv\right) + o(\varepsilon^2).$$

Démonstration. Since $\mathbb{P}(t_1^+ \leq B) = \mathbb{E}\left(1 - \exp\left(-\varepsilon \int_0^B p^+(X(s))\, ds\right)\right)$, the expansion in power series of ε has the first term $\mathbb{E}(p^+(X(0)))/(\mu - \lambda)$ and second term

$$\frac{1}{2}\mathbb{E}\left(\left(\int_0^B p^+(X(s))ds\right)^2\right) = \mathbb{E}\left(\int_{0\leq u\leq v\leq B} \mathbb{E}\left(p^+(X(0))p^+(X(v-u))\right) dudv\right)$$
$$= \mathbb{E}\left(\int_0^B (B-v)\,\mathbb{E}\left[p^+(X(0))p^+(X(v))\right]\, dv\right)$$

by stationarity of the process $(X(t))$. The first expansion is proved.

The event $\{t_1^+ \leq B\,, t_2^+ < t_1^+ + B_{L(t_1^+)-1}\}$ occurs only when t_1^+ and t_2^+ are in a sub-busy period $[s_{i-1} + E_i, s_i]$, for some $i \in \{1, \ldots, H\}$, (a period where L is always > 1). The variables E_i are i.i.d. exponential with parameter λ, $B_1^i = s_i - s_{i-1} - E_{i-1}$ has the same distribution as B and H is geometrically distributed with parameter $\lambda/(\lambda + \mu)$. The probability that the first two additional jumps are in the i-th sub-busy period, is

$$\mathbb{E}\left(\int_{s_{i-1}+E_i}^{s_i} \varepsilon p^+(X(u)) e^{-\varepsilon \int_0^u p^+(X(s))\, ds}\left(1 - e^{-\varepsilon \int_u^{s_i} p^+(X(s))\, ds}\right)\, du\right)$$
$$= \varepsilon^2 \mathbb{E}\left(\int_{s_{i-1}+E_i}^{s_i} p^+(X(u)) \int_u^{s_i} p^+(X(s))\, ds\, du\right) + o(\varepsilon^2)$$
$$= \varepsilon^2 \mathbb{E}\left(\int_{0\leq u\leq v\leq B} \mathbb{E}_\nu\left(p^+(X(0))p^+(X(v-u))\right)\, du\right) + o(\varepsilon^2),$$

which gives the second expansion. □

The complete expansion is now detailed.

Proposition 30. *The coefficients of ε^2 in the expansion of $\mathbb{E}(\Delta^\varepsilon)$ are given by*

$$a_+ = -\frac{\rho}{\mu(1-\rho)}\mathbb{E}\left(\int_0^B (B-v)\,\mathbb{E}\left[p^+(X(0))p^+(X(v))\right]\, dv\right)$$
$$- \frac{1-\rho}{2}\mathbb{E}\left(\int_0^B (B-u)^2 \mathbb{E}\left[p^+(X(0))p^+(X(u))\right]\, du\right). \quad (21)$$

$$a_- = -\frac{1}{\mu^3(1-\rho)}\mathbb{E}\left(\sum_{1\leq i<j\leq N} p^-(X(D_i))p^-(X(D_j))\right)$$
$$- \frac{1}{\mu^2}\mathbb{E}\left(\sum_{i=1}^{N}\sum_{j=1}^{N'} p^-(X(D_i))p^-(X(B+D'_j))\left(B_1 - D'_j + \frac{\mu}{(\mu-\lambda)^2}\right)\right) \quad (22)$$

$$a_{\pm} = \frac{1}{\mu}\mathbb{E}\left(\sum_{i=1}^{N}\int_0^B p^-(X(D_i))p^+(X(s))\,ds\left(\frac{\mu}{(\mu-\lambda)^2}+B-D_i\right)\right)$$
$$+\frac{1}{\mu}\mathbb{E}\left(\sum_{i=1}^{N}\int_0^{B_1} p^-(X(D_i))p^+(X(B+s))\left(B_1-s+\frac{\lambda}{(\mu-\lambda)^2}\right)\,ds\right)$$
$$-\frac{1}{\mu}\mathbb{E}\left(\sum_{i=1}^{H}\sum_{k=1}^{N_i}\int_0^{A_i} p^+(X(u))p^-(X(D_i^k))\left(\frac{\lambda}{(\mu-\lambda)^2}+A_i-u\right)\,du\right) \quad (23)$$

where H is geometric distributed with parameter $\lambda/(\mu+\lambda)$, $(N_i, D_1^i, \ldots D_{N_i}^i)$ denotes respectively the number of departures and the departures times in a busy period B_1^i, and $A_i = B_1^i + E_0 + \sum_{k=i+1}^{H}(E_k + B_1^k)$ where (E_i) are i.i.d is exponentially distributed with parameter $\mu+\lambda$ and (B_1^i) are i.i.d with the same distribution as B.

Quatrième partie

Sous Projet 6

Mesure de SLA et tarification

Annexe A

Introduction

La première partie de ce chapitre décrit la solution mise en place sur le réseau national RENATER-3 dans le but d'observer le respect de SLAs par des mesures actives. Ces types de mesures doivent répondre à de sévères règles comme nous l'avions étudié lors de la deuxième année du projet Métropolis (précision, synchronisation, multiplicité des services, mesures de bout-en-bout, etc.).

La deuxième partie aborde un nouvel aspect de la tarification par l'analyse de la nature statistique du trafic IP et de son impact sur les performances des différents services de l'Internet. Cette vision conduit à remettre en question l'efficacité des mécanismes de QoS existants. De même, le fait que la fonction principale de la tarification dans un réseau d'opérateur est le retour sur investissement nous mène à douter de la pertinence de tout système de contrôle de congestion basé sur les prix. Pour palier ces inconvénients il est proposé une architecture alternative orientée flot. Un routeur appelé Cross-protect, associant contrôle d'admission par flot et un ordonnancement de type fair queueing, permet de réaliser les garanties de performance nécessaires sans avoir à distinguer explicitement des classes de service ni à réserver des ressources dans le cadre de prétendus contrats de trafic.

Annexe B

Chapitre 1 : Déploiement de sondes de mesures actives sur un backbone

B.1 Introduction

Ce chapitre fait suite à l'étude de mesures de SLAs abordée lors de la deuxième année du projet Métropolis. Cette étude avait pour but d'évaluer les différentes solutions de mesures actives afin de contrôler les SLAs mis en place sur un backbone. RENATER a déployé au cours de l'année 2004 des sondes de mesures actives sur la partie métropolitaine du réseau. C'est cette solution qui est décrite.

B.2 Rappels

B.2.1 Les métriques

Le réseau RENATER-3 offre à la communauté d'enseignement et de recherche les services opérationnels suivants : IPv4, IPv6, MPLS, les classes de services (au nombre de 4) et le multicast IPv4. Les SLAs (non encore définis actuellement) doivent donc être respectés pour chacun de ces services. Les métriques correspondantes qui ont été retenues sont celles définies par le groupe IPPM de l'IETF, ainsi que des métriques RTP[1].
Prenons par exemple la mesure du delai unidirectionnel : la mesure doit être effectuée séparément pour le service IPv4 et pour le service IPv6 mais aussi en fonction des classes de servives ou du service MPLS. Une même métrique est donc utilisée pour presque chacun des servives.

B.2.2 Problématiques

Lors de la précédente étude, nous nous étions confrontés à plusieurs problématiques :

[1] *Real Time Protocol*

– La précision des mesures : sur un backbone national les délais unidirectionels entre les POPs[2] sont de l'ordre de quelques millisecondes. La précision de la mesure doit donc être d'environ $100\mu s$.
– La synchronisation : une telle précision ne permet pas l'utilisation du protocole NTP[3] pour les mesures. L'emploi du GPS sur les sondes de mesures et quasi inévitable. Cela pose le problème physique de l'installation de l'antenne dans les POPs ainsi que le problème financier.
– La fiabilité du système d'exploitation : l'étude menée a démontré que l'estempillage des paquets sondes devait se faire au plus bas niveau, c'est à dire si possible au niveau de la carte réseau (possibilité de modification du driver) pour éviter les latences dues aux interruptions systèmes.

B.3 La solution mise en place

Le GIP RENATER a choisi au cours de l'année 2004 de déployer des sondes QoSMetrics. Ces dernières répondent à toutes les problèmatiques auxquelles nous pouvons être confrontées lors de la mise en place d'une telle solution.

B.3.1 Caractéristiques

Les principales caractéristiques des sondes QoSMetrics sont les suivantes :
– intégration d'un système GPS sur les sondes avec également une carte PCI avec un oscillateur compensé en température,
– utilisation du protocole QTP[4] lors de la perte du signal GPS ou du non-emploi du GPS,
– sondes non-équipées de disques durs (boot sur mémoire flash),
– estampillage au niveau de la carte réseau,
– gestion des protocoles IPv4, IPv6, Multicast (et bientôt MPLS).

Les tests peuvent être configurés en mode continu ou en "burst", et la taille des paquets est également paramétrable ainsi que le protocole de la couche transport.
La solution QoSMetrics a en outre été choisie car elle propose un système de mesures de bout-en-bout. Après avoir téléchargé une applet, l'administrateur d'un site RENATER peut effectuer une suite de mesures avec la sonde de son choix. Les résultats sont enregistrés dans la base de données du système mais également envoyés par mail à l'utilisateur. Ces mesures permettent d'avoir des données à partir du poste, qui peut poser problème, jusqu'au coeur du réseau national. Outre des mesures IPPM, l'applet donne des informations sur le poste client comme le système d'exploitation utilisé et la taille des buffers du protocole TCP. Ces données, qui ne sont pas toujours disponible facilement, sont essentielles car elles permettent dès la première phase d'investigation d'éliminer le poste de la liste des causes possibles du problème.

[2] *Point Of Presence*
[3] *Network Time Protocol*
[4] *QoSMetrics Time Protocol*, d'une précision bien supérieure à NTP

B.3.2 Architecture

L'emplacement des sondes sur le backbone est décrite sur la figure B.1. En attendant une demi-douzaine de sondes supplémentaires au cours du premier semestre 2005, les sondes ont été réparties dans des POPs stratégiques au niveau du routage et en fonction des projets éventuels nécessitant de telles mesures au niveau du réseau pour garantir les SLAs.

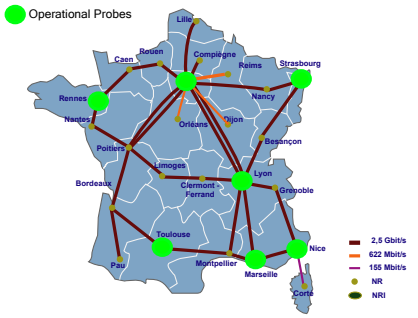


FIG. B.1 – Emplacement des sondes sur RENATER-3

Le type de raccordement des sondes varie en fonction des POPs comme vous pouvez le constater sur la figure B.2. Certaines sont raccordées directement sur le routeur, d'autres sont raccordées sur un switch lui-même raccordé en *uplink* sur le routeur. La sonde de Paris est particulière puisque dans les locaux du GIP RENATER (donc comme dans un site client). Dans ce dernier cas les paquets sondes traversent un équipement de plus, qui plus est un équipement d'un réseau local, et les résultats dépendants de cette sonde sont légérement supérieurs aux autres (section B.3.3).

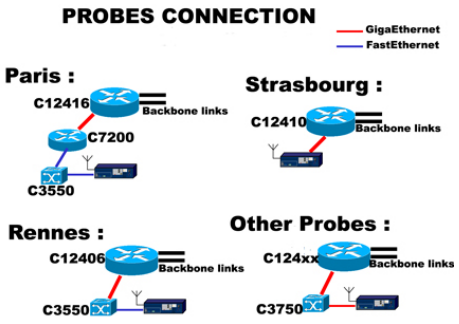


FIG. B.2 – Raccordement des sondes

B.3.3 Les résultats

Les tableaux B.3 et B.4 représentent les résultats obtenus à partir d'un scénario "*full mesh*" pour les deux protocoles IPv4 et IPv6. Chacune des sondes envoie des paquets vers toutes les autres en mode continu. Chaque minute la moyenne, le maximun et le minimum sont collectés par le serveur central. Les deux tableaux ont été construits à partir des valeurs moyennes.

Delay (ms)	Nice	Strasbourg	Rennes	Marseille	Lyon	Paris	Toulouse
Nice	/	12.01	9.20	1.40	5.49	8.66	3.62
Strasbourg	11.58	/	8.19	8.84	6.07	3.85	9.35
Rennes	8.71	8.14	/	7.33	7.25	5.30	5.18
Marseille	1.41	9.24	7.83	/	2.72	6.08	2.25
Lyon	5.55	6.56	7.71	2.80	/	3.37	3.32
Paris	8.39	3.77	5.26	5.65	2.88	/	5.23
Toulouse	3.56	9.76	5.61	2.18	3.24	5.60	/

Jitter (ms)	Nice	Strasbourg	Rennes	Marseille	Lyon	Paris	Toulouse
Nice	/	0.00	0.01	0.00	0.00	0.12	0.00
Strasbourg	0.01	/	0.02	0.01	0.00	0.09	0.01
Rennes	0.01	0.01	/	0.01	0.01	0.09	0.01
Marseille	0.00	0.00	0.01	/	0.00	0.04	0.00
Lyon	0.01	0.00	0.01	0.00	/	0.10	0.01
Paris	0.04	0.04	0.05	0.04	0.04	/	0.04
Toulouse	0.00	0.00	0.01	0.00	0.01	0.09	/

Pkts Loss	Nice	Strasbourg	Rennes	Marseille	Lyon	Paris	Toulouse
Nice	/	0	0	0	0	0	0
Strasbourg	0	/	0	0	0	0	0
Rennes	0	0	/	0	0	0	0
Marseille	0	0	0	/	0	0	0
Lyon	0	0	0	0	/	0	0
Paris	0	0	0	0	0	/	0
Toulouse	0	0	0	0	0	0	/

FIG. B.3 – Délais, gigues et pertes de paquets IPv4

On peut observer sur les tableaux B.3 et B.4 que les mesures effectuées pour IPv4 et IPv6 sont sensiblement les mêmes (à quelques dizaines de microsecondes près). Certains délais sont toutefois différents (Rennes-Nice), ce phénomène s'explique par le routage différent entre le deux protocoles pour ce chemin (passage par Paris en IPv6 et par Toulouse en IPv4). Cette assymétrie qui n'est pas normale devrait être corrigée d'ici peu.
La mesure de la gigue démontre la stabilité des délais à travers le backbone. Ces mesures sont inférieures à $10\mu s$ sauf pour Paris mais cela s'explique par l'emplacement de la sonde.

B.4 Conclusion

La solution mise en place sur le réseau RENATER-3 est aujourd'hui opérationnelle et d'autres réseaux ont déployé des solutions similaires : DFN (le réseau allemand) ainsi que GÉANT (le réseau européen de la recherche). La mesure du respect des SLAs est devenu aujourd'hui une des préoccupations des opérateurs et même si l'on est encore dans la première phase de nombreux opérateurs ont déployé de telles solutions de mesures (surtout en Asie).

Delay (ms)	Nice	Strasbourg	Rennes	Marseille	Lyon	Paris	Toulouse
Nice	/	12.01	9.21	1.40	5.49	8.67	3.62
Strasbourg	11.59	/	8.20	8.84	6.07	3.86	9.35
Rennes	12.78	8.16	/	7.35	7.27	-	5.19
Marseille	-	-	-	/	-	-	-
Lyon	5.55	6.56	7.72	2.80	/	3.40	3.32
Paris	8.42	3.81	5.30	5.69	2.91	/	5.27
Toulouse	3.56	9.76	5.61	2.18	3.24	5.36	/

Jitter (ms)	Nice	Strasbourg	Rennes	Marseille	Lyon	Paris	Toulouse
Nice	/	0.00	0.01	0.00	0.00	0.12	0.00
Strasbourg	0.01	/	0.02	0.01	0.00	0.09	0.01
Rennes	0.01	0.01	/	0.01	0.01	-	0.01
Marseille	-	-	-	/	-	-	-
Lyon	0.00	0.00	0.01	0.00	/	0.09	0.01
Paris	0.04	0.04	0.05	0.04	0.04	/	0.04
Toulouse	0.00	0.00	0.01	0.00	0.01	0.07	/

Pkts Loss	Nice	Strasbourg	Rennes	Marseille	Lyon	Paris	Toulouse
Nice	/	0	0	0	0	0	0
Strasbourg	0	/	0	0	0	0	0
Rennes	0	0	/	0	0	-	0
Marseille	-	-	-	/	-	-	-
Lyon	0	0	0	0	/	0	0
Paris	0	0	0	0	0	/	0
Toulouse	0	0	0	0	0	0	/

FIG. B.4 – Délais, gigues et pertes de paquets IPv6

Annexe C

Chapitre 2 : Trafic Internet, QoS et tarification : vers une architecture orientée flot

C.1 Introduction

L'intégration (ou la convergence) des différents services de voix, vidéo et données dans un même réseau IP constitue un objectif majeur de l'évolution actuelle des réseaux de télécommunications. Dans cet article nous évaluons les perspectives de la réalisation de cette convergence en exploitant les différents mécanismes des architectures de QoS normalisées comme Intserv, Diffserv et MPLS. Nous évaluons ces mécanismes à la lumière de ce que l'on sait sur la nature du trafic. Cette évaluation nous amène à remettre en cause leur efficacité et à proposer une solution alternative basée sur une architecture orientée flot.
Le réseau *best effort* surdimensionné permet de satisfaire la majorité des exigences des utilisateurs et présente l'avantage d'avoir un coût d'exploitation relativement faible. Cependant, cette solution présente plusieurs inconvénients :

- l'absence de traitement sélectif oblige l'opérateur à sécuriser le service de tous les utilisateurs alors que seulement quelques uns parmi eux ont une exigence forte de disponibilité,
- les paquets des applications temps réel peuvent subir un délai éventuellement important dû aux rafales de paquets de données
- la qualité de service est vulnérable au comportement éventuellement malicieux de certains utilisateurs.

La tarification des services d'un réseau commercial doit garantir le retour sur investissement de l'opérateur en restant suffisamment simple et transparente. Pour limiter les coûts, et donc les prix, il importe d'optimiser le dimensionnement des ressources et de mettre en oeuvre des mécanismes de contrôle de trafic dont l'exploitation est aussi simple que possible. Pour remplir ce double objectif, il est nécessaire de bien comprendre la relation trafic-performance qui relie demande (volume et caractéristiques du trafic), capacité (débit des liens et son partage entre les différents services), et performance (temps de réponse, taux de perte,...).

C'est l'analyse approfondie de cette relation qui nous a conduit à remettre en cause l'efficacité des mécanismes des architectures QoS normalisées. Il s'avère, en effet, que les performances du réseau sont excellentes pour tous dans les conditions normales de charge. Par contre, dès que la demande dépasse la capacité, suite à une panne, une anomalie ou une erreur de prévision, les performances se dégradent rapidement. Les mécanismes de QoS, dans ces cas exceptionnels, agissent en protégeant la performance des classes de trafic les plus prioritaires. En revanche, la qualité de service des autres classes de trafic se dégrade de façon très significative et certaines applications ne peuvent plus être supportées.
Cette analyse nous permet de définir une architecture alternative où le contrôle de surcharge est plus efficace. On préconise une architecture orientée flot où la qualité de service est maintenue grâce au contrôle d'admission flot par flot : on n'admet un nouveau flot que si la qualité de service des flots en cours est préservée. En associant le contrôle d'admission à un ordonnancement original de type fair queueing, il devient possible de garantir les performances des applications voix, vidéo et données sans être obligé de distinguer des classes de service ni de réserver explicitement des ressources.
Dans la section suivante, on décrit les caractéristiques essentielles du trafic IP. Nous abordons ensuite, à la section C.3, la relation trafic-performance pour le trafic *streaming* (applications audio et vidéo), le trafic élastique (applications de transfert de documents) et le mélange des deux types de trafic. La section C.4 évoque la question essentielle pour un opérateur d'un schéma de tarification. L'architecture orientée flot est présentée dans la section C.5 et nous formulons quelques conclusions à la section C.6.

C.2 Le trafic Internet

Dans cette section on présente un bref aperçu des principales caractéristiques du trafic ayant un impact sur la réalisation des garanties de QoS.

C.2.1 Variations du trafic

La figure C.1 représente les variations à long terme du trafic d'un lien du réseau dorsal. On observe clairement une période d'activité de pointe pendant laquelle la demande de trafic atteint approximativement la même valeur à chaque jour ouvré. Le réseau doit être dimensionné pour satisfaire aux exigences des utilisateurs sous cette charge maximale.

Pour établir la relation trafic-performance entre demande, capacité et performances, on supposera que le trafic dans la période de pointe peut être modélisé par un processus stochastique stationnaire. Ce processus est extrêmement complexe au niveau paquet [20]. Une modélisation à l'échelle flot s'avère plus appropriée car la QoS expérimentée par les utilisateurs s'apprécie à ce niveau. En outre, les caractéristiques du processus en question s'en trouvent simplifiées.

C.2.2 Flots et sessions

On entend par le terme "flot" l'ensemble des paquets rapprochés dans le temps relatifs à une même instance d'application et observés sur un élément du réseau. Cette définition plutôt vague est suffisante pour comprendre la nature du trafic IP ; une définition plus précise est nécessaire pour

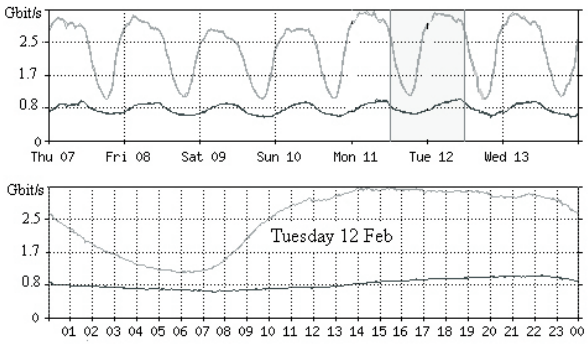


FIG. C.1 – Demande de trafic hebdomadaire et journalière d'un lien OC192

identifier un flot dans la pratique.
Généralement, les flots appartiennent à des sessions. Une session consiste en une succession de flots séparés par des périodes de silence appelées temps de réflexion. La session est liée à une activité prolongée d'un utilisateur telle que la navigation sur le Web, la consultation de mails ou la pratique d'un jeu en réseau. Une caractéristique essentielle, pour la modélisation, est l'indépendance supposée entre les différentes sessions. Pour une population importante d'utilisateurs, on obtient alors un processus Poissonnien d'arrivées de sessions. Ce fait est confirmé par des résultats empiriques et est reconnu comme un des rares invariants du trafic IP [12].

C.2.3 Flots élastiques et streaming

On distingue, en fonction de leurs critères de QoS, deux types de flots : les flots streaming et les flots élastiques. Les flots streaming sont produits par les applications audio et vidéo temps réel. Leur qualité exige un délai par paquet et un taux de perte négligeables. Les flots élastiques, quant à eux, proviennent des applications de transfert de documents tel que les pages Web, les mails ou les fichiers MP3. Le critère de QoS associé est lié au temps de réponse, ou de façon équivalente, au débit moyen réalisé au cours d'un transfert.

C.2.4 Caractérisation des applications à débit variable

La figure C.2 représente le débit en octets par trame d'une séquence vidéo codée en MPEG-4 (extraite de [11]). Le débit varie sur des échelles temporelles multiples exhibant ainsi un comportement dit auto-similaire. Une des conséquences de cette variabilité est la difficulté de caractériser un flot vidéo de manière simple en termes des paramètres d'un mécanisme de contrôle de trafic. En particulier, le mécanisme de seau percé (ou leaky bucket), qui est adopté dans la plupart des architectures de QoS, ne permet pas une caractérisation adéquate [23].

Si un flot élastique (suivant notre définition) est simplement caractérisé par la taille du document à transférer, l'agrégation de plusieurs flots, constituant par exemple tout le trafic d'une entreprise, produit un processus qui est aussi variable que celui présenté sur la Figure C.2 [20]. Encore une

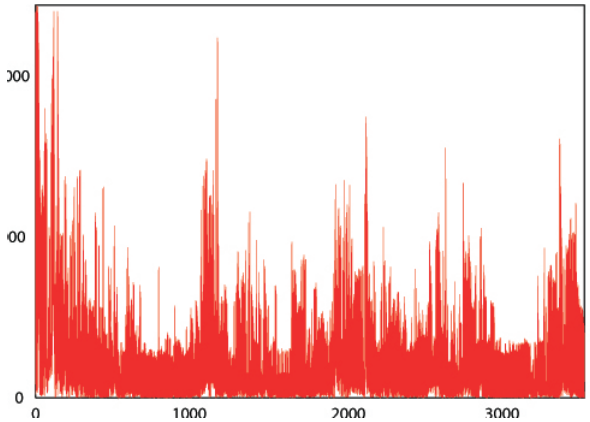


FIG. C.2 – Trace vidéo MPEG4 de [11]

fois, décrire ce genre de trafic d'une manière synthétique pour l'allocation des ressources s'avère quasiment impossible.

C.3 La relation trafic-performance

Comprendre la relation trafic-performance est nécessaire aussi bien pour la planification du réseau que pour la conception de mécanismes efficaces de contrôle de trafic.

C.3.1 Performance du trafic streaming

Dans ce paragraphe on suppose que les flots *streaming* se partagent un lien spécialisé.

Flots à débit constant

Si les flots sont à débit constant, les performances au niveau flot sont celles d'un réseau à commutation de circuits multidébit. La qualité de service est garantie grâce à l'application d'un contrôle d'admission : un nouveau flot n'est admis que si une certaine condition d'admission est respectée.

Il est commode dans ce cas d'appliquer une même condition d'admission à tous les flots indépendamment de leur débit. De ce fait, un flot quelconque se présentant dans le système est bloqué dès que la capacité disponible est insuffisante en supposant un débit maximum pour ce flot. On assure ainsi un taux de blocage identique pour chaque classe de débit. Le contrôle se trouve simplifié car il est inutile que l'utilisateur signale au réseau le débit requis.

Le phénomène d'économie d'échelle de la commutation de circuits est bien connu : les réseaux sont plus efficaces lorsqu'ils multiplexent un grand nombre de flots, chacun ne nécessitant qu'une faible proportion de la capacité d'un lien. L'opérateur a donc un intérêt économique à limiter le débit maximum pour lequel la qualité de service est garantie. Cette limitation s'applique, quelle que soit le modèle de QoS mis en œuvre.

Les paquets subissent des délais variables dans les files d'attente des routeurs produisant un phénomène de gigue qui pourrait altérer la qualité des applications. Nos recherches indiquent cependant que ce phénomène reste contrôlable et les délais suffisamment faibles simplement en limitant la

charge maximum du lien à environ 90% de la capacité [8].

Flots à débit variable

En réalité, la majorité des flots streaming ne sont pas à débit constant mais à débit variable exhibant éventuellement un comportement auto-similaire avec des variations extrêmes (cf. C.2). Supposons pour simplifier que les flots puissent être assimilés à des fluides possédant un débit instantané bien défini. Dans ce cas, on distingue clairement les notions de multiplexage avec et sans buffer : le multiplexage *avec* buffer permet d'absorber un excédant momentané de trafic par rapport à la capacité du lien C ; le multiplexage sans buffer nécessite que le taux d'arrivée global reste inférieur à la capacité C.

Afin d'illustrer des mécanismes de contrôle d'admission et leur performance, on considère un cas d'étude introduit dans l'article [10]. Un certain nombre de flots identiques partagent un lien de capacité C. Ils ont un débit crête $\rho = 1,5$ Mbit/s avec des variations de débit bornées par les paramètres d'un *leaky bucket* : capacité $b = 95$ Kbits et débit $r = 150$ Kbit/s. Les autres caractéristiques des sources ne sont pas spécifiées dans [10].

Pour effectuer nos comparaisons en fonction des paramètres des sources et non pas des paramètres du *leaky bucket*, on suppose que ces dernières sont déterminées pour assurer une faible probabilité de non conformité de 10^{-6}. On suppose en outre que les sources sont de type *on/off*, avec une distribution exponentielle des durées on et off. Une source particulière respectant ces critères se caractérise par un débit moyen $m = 50$ Kbit/s et une durée moyenne d'activité de 3 ms (4500 bits).

Notons le rapport 3 entre le taux de fuite du *leaky bucket* et le débit moyen. Il s'agit d'un ordre de grandeur typique même s'il est assez arbitraire dans la mesure où on aurait pu choisir d'autres caractéristiques de trafic satisfaisant la probabilité de non conformité supposée. Le rapport serait notamment beaucoup plus important si nous avions choisi un flot à variation de débit auto-similaire comme celui de la figure C.2.

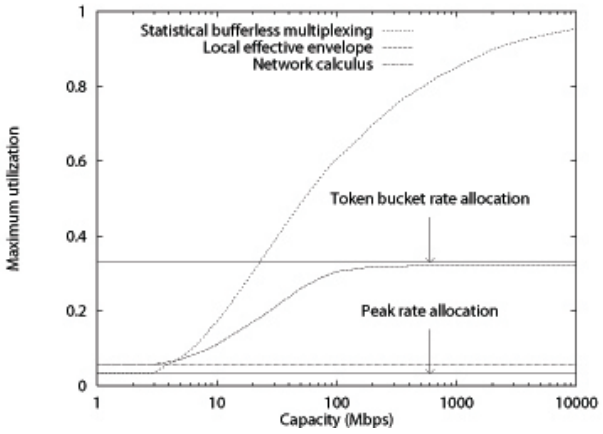


FIG. C.3 – Taux d'utilisation réalisable selon le critère choisi de contrôle d'admission.

La figure C.3 compare l'utilisation réalisable avec quatre algorithmes de contrôle d'admission en fonction de la capacité du lien lorsque le délai maximum est limité à 50 ms :

1. une allocation basée sur le débit crête,
2. l'application du *network calculus* déterministe (utilisant les paramètres du *leaky bucket* et le débit crête) afin d'éviter rigoureusement tout dépassement du délai maximum [19],

3. l'application d'une relaxation stochastique du *network calculus* décrite dans [10] : les sources de trafic sont supposées "pire cas" par rapport aux paramètres de trafic mais indépendantes et l'on respecte une probabilité de dépassement du délai maximum de 10^{-6},
4. l'application d'un multiplexage sans buffer en supposant que le débit moyen m soit connu et en respectant une probabilité de dépassement de débit de 10^{-6}.

La comparaison entre l'allocation basée sur le débit crête, le *network calculus* et le *network calculus* stochastique est présentée dans [10] : l'hypothèse raisonnable d'indépendance des sources permet un gain considérable, surtout dans le cas où la capacité du lien est importante. La différence entre la troisième et la quatrième approche est également très significative. Elle illustre nos remarques à la section C.2 sur l'inadéquation du *leaky bucket* comme descripteur de trafic. Le gain relatif d'utilisation d'un lien de grande capacité est de r/m.
Comme m n'est pas connu au préalable, le gain apporté par un multiplexage sans buffer dépend de la mise en œuvre d'un contrôle d'admission efficace basé sur la mesure. Les approches évoquées dans [13] et [15] et leur évaluation semblent justifier cette hypothèse, surtout lorsque le rapport C/m est important.
On obtiendrait un gain d'utilisation supplémentaire en effectuant un multiplexage statistique avec buffer. L'inconvénient de cette approche est que les performances dépendent alors de toutes les caractéristiques du trafic des sources, dont les propriétés éventuelles d'auto-similarité. Ce type de multiplexage s'avère donc très difficile à mettre en œuvre.

C.3.2 Performance du trafic élastique

On suppose que les flots élastiques partagent la bande passante de manière *équitable*. L'équité est souvent citée comme objectif mais n'est réalisée qu'approximativement dans la pratique. L'hypothèse de partage équitable permet de simplifier les modèles utilisés et de mettre plus clairement en évidence les facteurs importants pour la performance.
En supposant que l'arrivée des sessions suit un processus de Poisson, la performance d'un lien à partage équitable est très bonne tant que la demande de trafic est inférieure à la capacité [4]. En l'absence de limitations externes, le débit moyen réalisé par un flot est alors égale à la capacité résiduelle moyenne, $C - A$, où C est la capacité du lien et A la demande totale (taux d'arrivée des flots multiplié par la taille moyenne des flots). Le débit réalisé sur un chemin du réseau est déterminé principalement par le lien ayant la capacité résiduelle la plus faible [7]. Ceci correspond, habituellement, au débit de la ligne d'accès de l'utilisateur de sorte que les autres liens du réseau sont pratiquement transparents pour les flots élastiques.
Si le réseau est transparent en charge normale, l'impact de la congestion est dévastateur en cas de surcharge [9]. Lorsque la demande dépasse la capacité, le débit réalisé par flot diminue rapidement tandis que le nombre de flots en cours continue à croître. Dans la pratique, cette croissance est limitée par l'impatience des utilisateurs (ou des applications) qui abandonnent certains flots dont le débit devient trop faible. Au lieu de se fier ainsi à l'impatience pour stabiliser les performances, il nous semble plus judicieux de limiter proactivement la demande en appliquant un contrôle d'admission. Le critère de rejet d'un nouveau flot devrait faire en sorte qu'en charge normale ($A < 0.9C$, par exemple), la probabilité de blocage soit négligeable tandis qu'en surcharge ($A > C$), on assure un débit satisfaisant pour les flots admis. Il s'avère qu'un tel critère consiste à maintenir le débit par flot supérieur à environ 1% de la capacité du lien [3].
En fait, un débit de 1% de la capacité est typiquement bien plus grand que la limite d'acceptabilité des utilisateurs (il est même plus grand que la plupart des débits d'accès). Cependant, il est important

de noter que la relaxation du seuil d'admission n'apporterait aucun gain. En effet, la capacité du lien est utilisée (presque) entièrement en surcharge et la probabilité de blocage est approximativement $(A-C)/A$, quelque soit le *seuil d'admission*. Il est donc préférable de choisir un seuil relativement élevé qui permette aux flots qui ne sont pas limités ailleurs en débit de compléter leur transfert plus rapidement.

C.3.3 Intégration des flots streaming et élastiques

Tant que les paquets des flots streaming sont servis en priorité dans les files d'attente, les flots streaming et élastique peuvent partager les mêmes liens de transmission. L'intégration est bénéfique pour les deux types de trafic :

- les flots *streaming* voient un lien à faible charge subissant ainsi des délais et des pertes très faibles
- les flots élastiques se partagent toute la capacité du lien non utilisée par les flots *streaming* et par conséquent réalisent un meilleur débit.

Un contrôle d'admission est indispensable pour parer au problème de surcharge. Un nouveau flot (streaming ou élastique) est rejeté si la bande passante disponible pour un flot élastique descend en dessous d'un seuil d'admission de l'ordre de 1% de la capacité du lien, ou bien si la gigue des flots streaming devient trop importante. L'application de la même condition d'admission pour tous les flots permet d'égaliser les probabilités de blocage et facilite le contrôle car il n'y a pas besoin que les utilisateurs signalent les caractéristiques spécifiques de leurs flots.

C.4 Les garanties de QoS

On évalue la faisabilité des garanties de QoS à la lumière des considérations précédentes sur les caractéristiques du trafic et la relation trafic-performance. Par garantie de QoS, on entend le type de garantie envisagée dans un SLA (*Service Level Agreement*) où des critères de performance sont assurés pour un trafic dont les caractéristiques sont précisées a priori. Ce genre de contrat de trafic est proposé dans les mécanismes d'Intserv et de Diffserv ainsi que dans certaines applications de MPLS.

C.4.1 Le contrat de trafic

La notion de contrat de trafic implique les trois actions suivantes :

- les utilisateurs spécifient leur trafic en termes de caractéristiques intrinsèques et de localisation et déclarent leurs critères de performance,
- en fonction de cette spécification *à priori*, le réseau applique un contrôle d'admission en acceptant un nouveau contrat seulement si les critères de performance de ce contrat ainsi que ceux des contrats déjà établis sont satisfaits,
- afin de s'assurer que les utilisateurs respectent leur contrat, soit leur trafic est policé à l'entrée du réseau, soit les ressources qui leur sont allouées sont contrôlées par des mécanismes d'ordonnancement.

C.4.2 Les contrats pour micro-flots

Dans *Intserv*, les contrats de trafic peuvent être établis pour chaque flot individuellement. Cette solution est souvent écartée pour des raisons d'extensibilité (ou *scalability*), Dans cet article on se focalise sur d'autres difficultés liées à la nature du trafic et à la faisabilité des garanties de performance.
La première difficulté réside dans le choix des descripteurs de trafic. Les premières normes ATM [16] observaient qu'un descripteur de trafic devrait être :

1. compréhensible par les utilisateurs,
2. utile pour l'allocation des ressources,
3. vérifiable par le réseau.

Malheureusement, ces trois propriétés ne sont guère compatibles. Par exemple, les paramètres d'un *leaky bucket* sont vérifiables par conception mais il est difficile pour un utilisateur de les choisir (notamment pour un flot comme celui de la figure C.2) et, d'après la discussion du paragraphe C.3.1, ils se révèlent peu exploitables pour l'allocation des ressources.
Par ailleurs, il est très difficile d'atteindre des objectifs de performance précis quant au délai et à la perte de paquets. Les garanties déterministes envisagées dans la classe *Guaranteed Service* de *Intserv* n'ont pas grand sens si le *network calculus* conduit à un surdimensionnement exagéré (cf. paragraphe C.3.3). En pratique, les garanties statistiques ne sont réalisables qu'en cas de multiplexage sans buffer. Elles dépendent alors de mesures en temps réel et non pas d'un descripteur *a priori* du trafic.
Notons enfin que l'obligation de déclarer leurs paramètres de trafic constitue une lourde contrainte pour les utilisateurs. Cette contrainte n'est pas nécessaire dans le cadre d'un multiplexage sans buffer avec un contrôle d'admission basé sur la mesure.

C.4.3 Les contrats pour les tunnels

Les contrats de trafic sont fréquemment établis pour un agrégat de flots routés dans un tunnel. Ici encore, l'utilisateur rencontre des difficultés pour spécifier son trafic via un descripteur tel que le leaky bucket. Le débit d'un agrégat de trafic présente des fluctuations aléatoires extrêmes (variations auto-similaires) qu'il est notoirement difficile de décrire.
L'expérience des réseaux ATM ou à relais de trames (*Frame Relay*), où ce genre de contrat est employé, montre que les utilisateurs ont tendance à surestimer leur demande. Une méthode de contrôle d'admission courante consiste alors à sur-réserver la capacité des liens plusieurs fois (le facteur est typiquement de 5 ou plus). Notons que ceci constitue une forme assez imprécise de contrôle d'admission basé sur la mesure, le facteur de sur-réservation étant estimé à partir d'observations du trafic réel.

C.4.4 Les contrats de trafic non localisés

Au lieu d'utiliser un ensemble de contrats de trafic point à point, les clients VPN préfèrent spécifier le trafic total entrant ou sortant des nœuds de leur réseau. Il n'y a donc aucune indication sur le volume de trafic offert à un lien donné à l'intérieur du réseau. L'opérateur doit encore appliquer une certaine forme de contrôle d'admission basé sur la mesure. Dans ce cas, les paramètres de trafic déclarés sont encore moins pertinents pour la réservation des ressources que dans le cas des tunnels.

C.4.5 La différenciation de service

Les contrats non localisés sont également envisagés dans le modèle *Diffserv*. Les utilisateurs précisent un descripteur pour le trafic de chaque classe de service. L'opérateur doit appliquer un contrôle d'admission afin de respecter les garanties de QoS par classe sans savoir la répartition du trafic en fonction des différentes destinations possibles.
La classe EF (*Expedited Forwarding*) permet d'obtenir des garanties statistiques grâce au traitement prioritaire réservé à ses paquets et à l'application d'un multiplexage sans buffer. Les descripteurs de trafic déclarés sont alors peu utiles car l'allocation des ressources doit encore une fois être basée sur des mesures du trafic.
Le groupe de PHB (*Per-Hop Behaviour*) AF (*Assured Forwarding*) est censé permettre une différenciation de qualité de service selon quatre niveaux distincts. Cependant, il reste à démontrer comment il est possible de réaliser cette différenciation de manière suffisamment fiable. On peut douter qu'une méthode satisfaisante puisse jamais exister.

C.5 Tarification et QoS

Dans cette section on considère la relation entre tarification et qualité de service. La tarification doit assurer le retour sur investissement du fournisseur de réseau. En outre, dans certaines propositions, elle est utilisée comme mécanisme de contrôle de congestion.

C.5.1 Le retour sur investissement

Le retour sur investissement constitue l'objectif premier d'un opérateur de réseau. Les prix des différents composants d'un service (installation, abonnement, usage,...) doivent permettre de récupérer les frais d'investissement et de fonctionnement (*Capex* et *Opex*). En particulier, il est nécessaire de répartir les coûts liés au trafic parmi les différents utilisateurs.
Le coût d'un flot IP donné est difficile à évaluer. Une tarification en fonction du coût marginal n'est guère envisageable car celui-ci est en général négligeable. Il s'agit plutôt de définir une règle pour partager les coûts globaux de manière convenable entre les différents flots. Dans le contexte du réseau téléphonique, pour évaluer les frais d'interconnexion entre opérateurs, on utilise couramment la notion de coût moyen incrémental à long terme [2].
D'après cette approche, même le trafic écoulé par des ressources qui seraient libres par ailleurs (dont le coût marginal est donc nul) entraîne un coût et donc est susceptible d'être facturé. Une hypothèse raisonnable serait de considérer que le coût d'un flot soit proportionnel au volume transféré. Le coût pourrait également dépendre d'autres caractéristiques de trafic du flot ou du fait qu'il soit de type élastique ou streaming. Cependant, ces considérations sont d'une importance secondaire vu leur faible impact sur le dimensionnement nécessaire (d'après les relations trafic-performance de la section C.3).

C.5.2 La discrimination de prix

Le coût n'est pas le seul facteur qui détermine la tarification. En particulier, il est efficace du point de vue économique de pratiquer une discrimination de prix lorsqu'il y a plusieurs segments

de marché distincts avec des valorisations différentes du même service. Divers dispositifs peuvent être utilisés pour réaliser cette discrimination. Les compagnies aériennes en fournissent la parfaite illustration. Le confort propre à la classe affaire justifie une différence de prix avec la classe touriste mais la différence pratiquée dépasse largement la disparité des coûts respectifs d'une place de chaque classe. D'autres discriminations de prix sont pratiquées au sein de la classe touriste, notamment par le biais des tarifs pour un voyage qui chevauche un week-end. Cette différenciation permet aux particuliers de payer moins cher que les voyageurs d'affaires pour exactement la même qualité de service.
Dans l'Internet, l'emploi de la discrimination de prix est souvent identifié au besoin d'offrir différentes classes de QoS. Malheureusement, vue la nature de la relation trafic-performance (performance excellente en charge normale et très mauvaise en surcharge), il est quasiment impossible de créer des classes de service qui soient les équivalentes des classes affaire et touriste des compagnies aériennes. La différence de qualité n'est manifeste que dans des conditions de trafic exceptionnelles ; le choix des classes premium offre donc plutôt une garantie de disponibilité qu'un niveau supérieur de qualité de service.
Fixer le prix d'un contrat de trafic est un problème difficile. Sachant que, pour satisfaire ces contrats, on emploie une forme de contrôle d'admission basé sur la mesure (cf. section C.4), le coût d'un flot dépend du volume de trafic réellement émis et en aucun cas des paramètres déclarés dans le contrat. On doit donc s'interroger sur la pérennité à long terme d'un schéma de facturation qui est basé sur la valeur de ces paramètres.
La distinction entre trafic élastique et trafic *streaming* pourrait éventuellement constituer une clef pour la discrimination des prix. Notons cependant que les utilisateurs n'ont pas systématiquement une valorisation plus forte pour les services audio et vidéo par rapport aux services de transfert des données. En outre, le coût de transport (supposé proportionnel au volume de trafic) est pratiquement le même pour les deux types de trafic.
D'autres clés pour la discrimination de prix sont sans doute plus acceptables que des garanties de QoS qui restent nécessairement vagues. Par exemple, le débit d'un modem DSL est un facteur déterminant dans la tarification Internet actuelle. La tarification pourrait également dépendre de différents regroupements de services (la notion économique de *bundling*).

C.5.3 La tarification de la congestion

La majeure partie des recherches effectuées dans le domaine de la tarification est liée au contrôle de congestion et non pas au retour sur investissement. L'exemple le plus connu de tarification de la congestion est le " *smart market*" proposé par MacKie-Mason et Varian [21]. Dans le *smart market*, les utilisateurs incluent une enchère dans chaque paquet. En cas de congestion, les utilisateurs présentant les enchères les plus faibles sont rejetés en premier. Les paquets acceptés sont facturés en fonction de l'enchère la plus élevée parmi les paquets rejetés. On démontre qu'un tel schéma est économiquement optimal dans le sens où le prix est fixé à la valeur qui permet à un maximum d'utilisateurs de profiter de la ressource partagée et où ces utilisateurs sont ceux qui en ont la plus forte utilité.
D'après cet exemple il apparaît clairement que la tarification de la congestion ne tient pas compte du retour sur investissement. Quand le réseau n'est pas en congestion il n'y a pas de facturation ; de ce fait, un réseau bien dimensionné n'apporte aucun revenu avec le *smart market*.
Une approche plus pragmatique de la tarification de la congestion a été introduite par Shenker et al. [24] : le réseau offre plusieurs classes de services avec des tarifs qui augmentent en fonction de leur

niveaux de qualité de service ; les utilisateurs ajustent leur facture en choisissant une classe de service plus ou moins prioritaire en cas de congestion. Cette proposition constitue une des motivations de la création des classes différenciées du groupe AF de l'architecture Diffserv.
La proposition d'un "Internet auto-géré" par Kelly [17] utilise un système particulier de tarification de la congestion. Des notifications explicites de congestion (marques ECN) sont utilisées dans les protocoles de contrôle de congestion pour signaler une congestion imminente. Dans le schéma de Kelly, chaque marque est munie d'un coût élémentaire, le coût d'un flot étant proportionnel au nombre de marques reçues. En l'absence de congestion, aucune marque n'est émise et le coût des flots est nul. Dans le cas contraire, les utilisateurs reçoivent un flux de marques dont l'intensité est proportionnelle à leur propre débit d'émission. Ceux qui accordent une utilité importante à leur transfert émettent à haut débit et paient le prix fort. Les autres peuvent s'abstenir ou ralentir leur débit en attendant une période de moindre congestion.
Malgré une abondante littérature sur ce genre d'approche, on peut émettre de sérieuses réserves sur l'utilisation par un opérateur de la tarification de la congestion. Généralement, les ressources réseau ne sont pas rares. L'opérateur peut facilement augmenter la capacité des liens, et il le fera avant qu'une éventuelle congestion n'ait lieu si le retour sur investissement est assuré. Les utilisateurs pourraient interpréter toute congestion comme étant un signe de mauvaise gestion ; ils trouveraient alors aberrant de devoir dépenser plus d'argent à cause de cela.
Même si l'opérateur cherche à dimensionner son réseau pour éviter toute congestion, des moments de surcharge se produiront inévitablement, notamment à cause de pannes ou d'erreurs de prévision. Plutôt que de gérer de tels événements par une tarification dynamique, il nous semble bien plus raisonnable de préserver l'efficacité du réseau en mettant en œuvre un contrôle d'admission préventif au niveau flot.
L'expérience montre qu'en matière de tarification, les utilisateurs ont une nette préférence pour la transparence et un penchant certain pour l'absence de risques [22, 1]. Ainsi il est peu probable, ne serait-ce que pour ces raisons, qu'ils accepteraient le caractère imprévisible de la tarification de la congestion.

C.6 Une architecture orientée flot

Les considérations précédentes sur la nature du trafic IP, son impact sur la performance, la faisabilité des garanties de QoS et le choix d'un schéma de tarification, nous conduisent à remettre en question l'efficacité des mécanismes de QoS proposés. L'un des messages principaux de cette section est d'attirer l'attention sur le fait qu'il est nécessaire d'implémenter une architecture alternative que nous appelons architecture orientée flot (*flow-aware networking*).

C.6.1 Identification d'un flot

Le niveau flot constitue la granularité appropriée pour le contrôle du trafic. Il s'agit de l'objet identifiable le plus fin qui puisse être assimilé à un service de transport fourni par le réseau. L'application d'un contrôle d'admission à ce niveau permet de protéger la qualité de service des flots en cours. Sa mise en œuvre nécessite une définition plus précise de ce qu'est un flot.
Un choix idéal serait d'utiliser le champ *flow label* dans l'entête des paquets IPv6 en l'associant à l'adresse IP source et/ou à l'adresse IP destination. En sélectionnant librement la valeur du *flow label*, l'utilisateur pourrait lui-même décider quelle entité devrait constituer un "flot" pour les besoins

de son application. Par exemple, tous les éléments d'une même page Web pourraient porter le même identifiant de flot. En IPv4, le quintuplet constitué des adresses IP, du protocole et des numéros de ports, pourrait être utilisé avec cependant un moindre degré de flexibilité.

C.6.2 Contrôle d'admission implicite au niveau flot

Considérons un lien emprunté par des flots élastiques et des flots *streaming* avec une gestion par file prioritaire, tel que décrit au paragraphe C.3.3. Les utilisateurs identifient leurs flots comme étant élastiques ou *streaming* et on applique un contrôle d'admission pour préserver la qualité des flots en cours. Les conditions d'admission sont les mêmes pour tout nouveau flot et doivent permettre de préserver le débit moyen des flots élastiques et d'éviter un délai non négligeable pour les paquets des flots *streaming*.
On pourrait imaginer que le contrôle d'admission soit difficile à mettre en œuvre surtout pour un lien de grande capacité. En fait, la mise en œuvre envisagée se contente d'un descripteur minimal de trafic (le débit crête maximal d'un flot streaming) et ne nécessite pas de signalisation [3]. Les nouveaux flots sont identifiés "au vol" :

- on maintient une liste des flots en cours et l'identificateur de flot de chaque paquet y est comparé,
- si le paquet appartient à un flot déjà présent dans la liste, il est directement transmis,
- sinon il s'agit d'un nouveau flot qui est donc soumis à la procédure de contrôle d'admission,
- si les conditions d'admission sont satisfaites, le flot est rajouté à la liste et le paquet est transmis, -sinon le paquet est rejeté, la perte du premier paquet signifiant le rejet du flot.

C.6.3 Le routeur Cross-protect

Malgré les avantages apportés par l'architecture orientée flot décrite ci-dessus, celle-ci présente tout de même quelque inconvénients :

- il est nécessaire de distinguer explicitement les flots streaming et élastiques,
- le débit crête des flots streaming doit être contrôlé (police ou conditionnement des flots),
- le niveau d'équité du partage de la bande passante nécessite la collaboration des utilisateurs.

Un développement récent appelé Cross-protect [18] permet de pallier ces insuffisances.
Un routeur Cross-protect associe le contrôle d'admission au niveau des interfaces d'entrée à un algorithme original de fair queueing au niveau des interfaces de sortie, comme illustré sur la figure C.4.
L'algorithme PFQ (Priority Fair Queueing) s'est inspiré de Start-time Fair Queueing (SFQ) [14]. Le partage est équitable (c'est à dire, non pondéré) ; on adjoint un service prioritaire pour les paquets des flots dont le débit d'arrivée est inférieur au débit équitable courant au sens max-min. Ainsi, les flots streaming dont le débit est inférieur au débit équitable sont traités dans les conditions d'un multiplexage sans buffer, et leurs paquets ne subissent qu'un faible délai. Le contrôle d'admission permet de maintenir le débit équitable suffisamment élevé pour assurer la qualité de service des applications audio et vidéo envisagées. Tout flot streaming dont le débit crête est supérieur au débit équitable subira des pertes de paquet et sera contraint de s'adapter.

Le nom *Cross-protect* provient du fait que les mécanismes de contrôle d'admission et d'ordonnancement PFQ se protègent mutuellement. En effet, le contrôle d'admission limite le nombre de

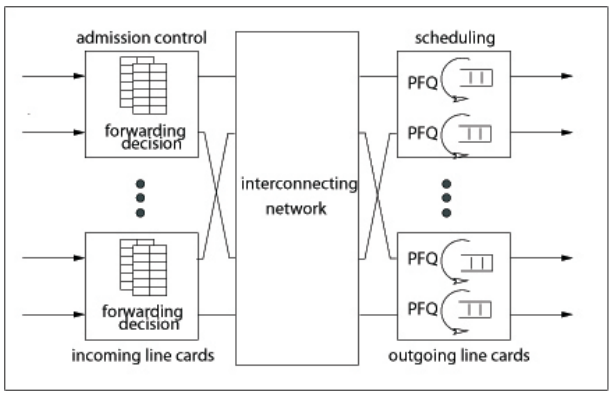


FIG. C.4 – Un routeur *Cross-protect*

flots traités par l'algorithme d'ordonnancement[1] tandis que ce dernier fourni les mesures de congestion nécessaires au contrôle d'admission. *Cross-protect* permet également de protéger les flots streaming et élastiques face à d'éventuels utilisateurs mal intentionnés.
Le principal avantage de *Cross-protect* par rapport à la solution esquissée au paragraphe précédent est l'absence de toute distinction explicite entre classes de service. On garde la simplicité actuelle de l'exploitation d'un réseau best effort.

C.6.4 Tarification et architecture orientée flot

La tarification dans un réseau orienté flot est beaucoup plus simple que pour les architectures de QoS classiques. Une tarification basée sur le volume pourrait être appliquée car tous les flots admis reçoivent une bonne qualité de service et sont donc susceptibles d'être facturés.
Le contrôle d'admission pourrait être appliqué de manière sélective en privilégiant une certaine classe de flots ou certains utilisateurs. Cette différenciation, au niveau de l'accès au service, constituerait une clé utile pour la discrimination de prix. Par exemple, les flots best effort pourraient être bloqués dès que la bande passante disponible descend en dessous de 2% de la capacité du lien, tandis que les flots premium ne seraient rejetés qu'à partir d'un seuil de 0,5%. Ainsi, le trafic premium ne serait quasiment jamais bloqué même en cas de surcharge globale due à une panne, par exemple [5].

C.7 Conclusion

L'analyse des caractéristiques du trafic IP et de son impact sur les performances des applications nous conduit à remettre en question les mécanismes de QoS généralement proposés. On doute également de l'efficacité de tout système de tarification dynamique où le prix est fonction de la congestion. La tarification doit surtout assurer le retour sur investissement en restant simple et transparente pour les utilisateurs.
Cette analyse nous conduit à proposer une architecture alternative orientée flot. Cette architecture permet de satisfaire les contraintes de QoS des flots sans avoir à distinguer différentes classes de

[1] Ce nombre est de l'ordre de quelques centaines quelle que soit la capacité du lien [18].

service, ni à négocier des contrats de trafic. Cette simplification se réalise grâce à l'association d'un contrôle d'admission basé sur la mesure et d'un nouvel algorithme d'ordonnancement par flot. La protection mutuelle des mécanismes de contrôle d'admission et d'ordonnancement, ajoutée à la protection de la performance des flots contre des utilisations malicieuses, nous conduisent à appeler le routeur proposé *Cross-protect*.
L'architecture orientée flot pourrait être introduite progressivement en équipant les routeurs un par un. L'architecture Cross-protect pourrait également être utilisée en commun avec les mécanismes existant de Diffserv et de MPLS afin de fournir des garanties supplémentaires au trafic *best effort*. Enfin, l'architecture orientée flot permettrait d'adopter un système de tarification proche de celui du réseau téléphonique où les utilisateurs paient en fonction du volume de trafic qu'ils génèrent sans distinction des applications sous-jacentes.
Cette architecture ne nécessite pas de normalisation car les mécanismes proposés fonctionnent avec les protocoles existants de l'Internet best effort. C'est un avantage important. L'identification des flots pourrait cependant être améliorée en adoptant une convention pour l'utilisation du champ *flow label* de l'en-tête IPv6. On envisage d'utiliser deux bits de ce champ pour signifier que, pour identifier un flot, le label devrait être associé à l'adresse IP source, l'adresse IP destination, les deux ou aucune. Cette possibilité laisserait à l'utilisateur une souplesse intéressante en définissant ce qui doit être considéré comme une entité de trafic (par exemple, tous les éléments d'une page Web pourraient appartenir à un même flot).
Un problème majeur restant est de convaincre les constructeurs de routeurs du bien fondé des arguments esquissés dans ce chapitre, afin qu'ils acceptent de mettre en œuvre les mécanismes préconisés. Les intérêts de l'opérateur et du constructeur dans la définition du futur Internet ne sont pas forcément convergents, le dernier ayant peu de motivation à reconnaître les inconvénients de l'architecture de QoS des routeurs actuels.

Bibliographie

[1] J. Altmann, K. Chu. A proposal for a flexible service plan that is attractive to users and Internet providers. Proceedings of Infocom 2001.

[2] W. J. Baumol, J. G. Sidak, *Toward Competition in Local Telephony*, The MIT Press, Cambridge, 1994.

[3] N. Benameur, S. Ben Fredj, S. Oueslati-Boulahia, J. Roberts. Quality of service and flow-aware admission control in the Internet, In Computer Networks, Vol 40, pages 57-71, 2002.

[4] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié, and J.W. Roberts. Statistical bandwidth sharing : A study of congestion at flow level. In *ACM SIGCOMM*, pages 111–122, 2001.

[5] S. Ben Fredj, S. Oueslati, J. Roberts. Measurement-based admission control for elastic traffic. in J. Moreira et al. (Eds) Teletraffic Engineering in the Internet Era. Proceedings of ITC 17, Elsevier, 2001.

[6] D. Bertsekas, R. Gallager. *Data Networks*, Prentice Hall, 1992

[7] T. Bonald, A. Proutière. On performance bounds for balanced fairness. Performance Evaluation, Vol 55, No 1-2, pages 25-50, 2004.

[8] T. Bonald, A. Proutière, and J.W. Roberts. Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding. In *Proceedings of IEEE INFOCOM*, pages 1104–1112, 2001.

[9] T. Bonald, J. Roberts. Congestion at flow level and the impact of user behaviour. Computer Networks, Vol 42, 521-536, 2003.

[10] R.R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn. Satistical Service Assurance for Traffic Scheduling Algorithms. *JSAC*, 18(12) :2651–2664, December 2000.

[11] F.H.P. Fitzek, M. Reisslein. MPEG-4 and H.263 video traces for network performance evaluation. IEEE Network Magazine, Volume : 15 Issue : 6 , Pages : 40-54, Nov.-Dec. 2001

[12] S. Floyd and V. Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4) :392–403, August 2001.

[13] R.J. Gibbens, F.P. Kelly, and P.B. Key. A Decision-Theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13(6) :1101–1114, August 1995.

[14] P. Goyal, H. Vin, H. Cheng. Start-time Fair Queueing : A scheduling algorithm for integrated services packet switching networks. IEEE/ACM Transactions on Networking, Vol 5, No 5, Oct 1997.

[15] M. Grossglauser and D. Tse, "A Time-Scale Decomposition Approach to Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, Vol 11, No 4, August 2003.

[16] ITU-T. Traffic control and congestion control in B-ISDN. Recommendation I.371, Geneva, 2000.

[17] F. P. Kelly. Models for a self-managed internet. *Philosophical Transactions of the Royal Society*, A358 :2335–2348, 2000.

[18] A. Kortebi, S. Oueslati, J. Roberts. Cross-protect : implicit service differentiation and admission control. Proceedings of IEEE Workshop on High Performance Switching and Routing, Phoenix, USA, April 2004.

[19] J. Y. Le Boudec and P. Thiran. *Network Calculus*. Springer Verlag LNCS 2050, June 2001.

[20] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1) :1–15, 1994.

[21] J. K. MacKie-Mason and H. Varian. *Pricing the Internet*, chapter Public Access to the Internet. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[22] A. M. Odlyzko. Internet pricing and the history of communications. *Computer Networks*, 36 :493–517, 2001.

[23] A.R. Reibman, A.W. Berger. Traffic descriptors for VBR video teleconferencing over ATM networks. IEEE/ACM Transactions on Networking, Vol 3, No 3, 329-339, 1995.

[24] S. Shenker, D. D. Clark, D. Estrin, and S. Herzog. Pricing in Computer Networks : Reshaping the Research Agenda. *ACM Computer Communication Review*, 26 :19–43, April 1996.