Contrôle de congestion et gestion du trafic à partir de mesures Application pour l'optimisation de la qualité de service dans l'Internet

Nicolas Larrieu

Contrôle de congestion et gestion du trafic à partir de mesures

La métrologie n'est appliquée dans la recherche, l'ingénierie et la conception des réseaux Internet que depuis le début des années 2000, mais cette approche est de plus en plus populaire et tend à se généraliser. Ses principes consistent à étudier, caractériser, analyser et modéliser le trafic existant sur les liens de l'Internet, afin de comprendre les principes qui régissent le comportement du réseau par rapport à un trafic qui s'avère encore méconnu. En particulier, garantir la qualité de service (QdS) dans l'Internet est un problème essentiel aujourd'hui. La métrologie du trafic Internet, et notamment son analyse montre que les mécanismes de transport actuels (TCP) introduisent des propriétés de LRD (Long Range Dependence) qui se traduisent par une grande variabilité du trac et obligent à surdimensionner les ressources de communication. L'objectif de cet ouvrage est de présenter un protocole de transport qui réduit cette LRD afin d'optimiser l'utilisation des ressources de communication. Ainsi, les nouveaux mécanismes protocolaires et architecturaux de l'Internet pourront être parfaitement adaptés aux besoins des utilisateurs et aux contraintes du trafic.

Nicolas Larrieu, Docteur en Réseaux et Télécommunications de l'Institut des Sciences Appliquées (INSA), est enseignant-chercheur à l'Ecôle Nationale de l'Aviation Civile (ENAC), Toulouse, depuis septembre 2006. Ses travaux de recherche portent sur la sécurisation des communications aéronautiques entre avions ainsi que l'infrastructure réseau sol.

978-3-8416-2007-1

Ingénierie et métrologie réseau

Larrieu

Nicolas Larrieu

Contrôle de congestion et gestion du trafic à partir de mesures

Application pour l'optimisation de la qualité de service dans l'Internet

paf

Nicolas Larrieu

Contro^le de congestion et gestion du trafic à partir de mesures

Nicolas Larrieu

Contro^le de congestion et gestion du trafic à partir de mesures

Application pour l'optimisation de la qualité de service dans l'Internet

Presses Académiques Francophones

Impressum / Mentions légales
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Information bibliographique publiée par la Deutsche Nationalbibliothek: La Deutsche Nationalbibliothek inscrit cette publication à la Deutsche Nationalbibliografie; des données bibliographiques détaillées sont disponibles sur internet à l'adresse http://dnb.d-nb.de.

Coverbild / Photo de couverture: www.ingimage.com

Verlag / Editeur:
Presses Académiques Francophones
ist ein Imprint der / est une marque déposée de
OmniScriptum GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland / Allemagne
Email: info@presses-academiques.com

Herstellung: siehe letzte Seite /
Impression: voir la dernière page
ISBN: 978-3-8416-2007-1

Zugl. / Agréé par: INSA de Toulouse

Résumé

La métrologie n'est appliquée dans la recherche, l'ingénierie et la conception des réseaux Internet que depuis peu de temps (le début des années 2000), mais cette approche est de plus en plus populaire et devrait tendre à se généraliser. Ses principes consistent à étudier, caractériser, analyser et modéliser le trafic existant sur les liens de l'Internet, afin de comprendre les principes qui régissent le comportement du réseau par rapport à un trafic qui s'avère méconnu. En particulier, garantir la qualité de service (QdS) dans l'Internet est un problème essentiel aujourd'hui. La métrologie du trafic Internet, et notamment son analyse montre que les mécanismes de transport actuels (TCP) introduisent des propriétés de LRD (Long Range Dependence) qui se traduisent par une grande variabilité du trafic et obligent à surdimensionner les ressources de communication [20]. L'objectif de cette thèse est de trouver des protocoles de transport qui réduisent cette LRD afin d'optimiser l'utilisation des ressources de communication. Ainsi, les nouveaux mécanismes protocolaires et architecturaux de l'Internet pourront être parfaitement adaptés aux besoins des utilisateurs et aux contraintes du trafic [Lar03a]. Finalement, c'est le processus de recherche et d'ingénierie des réseaux qui sera modifié en lui ajoutant une phase métrologique en amont permettant de collecter des données et des connaissances sur l'existant, qui permettront ensuite de concevoir et mettre en œuvre de nouveaux réseaux optimisés. Ainsi, ce travail de thèse présente une nouvelle approche pour l'Internet, dont l'objectif est d'améliorer la gestion du trafic, la QdS et plus généralement, les services réseaux. Cette approche, appelée MBN (Measurement Based Networking), repose principalement sur l'utilisation de techniques de métrologie actives et passives qui permettent de mesurer en temps réel différents paramètres du réseau et de son trafic pour ainsi réagir très rapidement et très précisément à des évènements spécifiques se produisant dans le réseau (apparition de congestion par exemple). Nous illustrerons en particulier l'approche MBN au travers du développement d'un mécanisme de contrôle de congestion orienté mesures intitulé MBCC (Measurement Based Congestion Control) et nous l'évaluerons au travers de simulations NS-2 [Lar05c]. Nous montrerons, en particulier, comment ce nouveau mécanisme permet d'améliorer les caractéristiques du trafic ainsi que la QdS dans l'Internet, malgré la complexité et la variabilité du trafic actuel. Enfin, dans la dernière partie de ce travail de thèse nous présenterons comment l'approche MBN et le mécanisme MBCC, en particulier, peuvent garantir une QdS robuste, i.e. capable de fournir la QdS demandée en toutes circonstances, notamment en présence d'attaques de DdS (Déni de Service) [Owe04f]. En effet, nous utiliserons l'architecture MBA (Measurement Based Architecture) basée sur des mesures du trafic en temps réel pour s'adapter aux ruptures légitimes ou illégitimes du trafic s'y produisant en continu. En particulier, nous démontrerons que le mécanisme de congestion MBCC associé à MBA, conçu pour générer des trafics réguliers et optimaux, rend l'Internet plus robuste que TCP face aux attaques de DdS.

Mots-clés : métrologie du trafic Internet, Internet de nouvelle génération, qualité de service, robustesse réseau, protocole de transport, contrôle de congestion.

Abstract

Internet measurement has only been used for research, engineering and design of Internet networks since few years (since the beginning of years 2000), but it is more and more popular and spreads rapidly. It deals with studying, characterizing, analyzing and modeling traffic on the different Internet links in order to understand network behaviors when facing traffics which are largely unknown at this time. In particular, guarantying QoS (Quality of Service) for the Internet is currently one of the most challenging issues. Internet traffic monitoring and analysis show that current transport mechanisms (TCP) introduce Long Range Dependence (LRD) properties which induce very high traffic variability and force to largely over-provision communication resources [20]. This thesis aims at designing new communication protocols and architectures able to reduce the traffic LRD in order to optimize the use of communication resources. Then, new protocol and architectural mechanisms could be perfectly suited to users' needs and traffic constraints [Lar03a]. Finally, this is the research and network engineering process which will be modified by integrating a new initial monitoring stage for collecting data and some knowledge on existing networks and traffic ; it will help, in a second time, to design and deploy new optimized networks. Thus, this PhD work deals with a new approach for the Internet, aiming at improving traffic management, QoS and more generally network services. This approach, called Measurement Based Networking (MBN), is built on the use of active and passive monitoring techniques to evaluate in real time different network parameters and analyze its traffic in order to react very quickly and accurately to specific events arising in the network (for instance, congestion events). We will illustrate, in particular, the MBN approach by designing a new measurement based congestion control mechanism (MBCC) which will be evaluated thanks to NS-2 simulations [Lar05c]. We will show, in particular, how this new mechanism can improve traffic characteristics as well as Internet QoS, despite the complexity and variability of current Internet traffics. Finally, in the last part of this PhD work, we will present how the MBN approach and MBCC can guaranty a robust QoS, i.e. are able to provide the requested QoS in all circumstances, including in the presence of Denial of Service (DoS) attacks [Owe04f]. Indeed, we will use the MBA architecture, based on real time traffic measurements, to adapt to normal or abnormal disruptions which arise in the network. In particular, we will demonstrate that the MBCC congestion mechanism, designed for generating regular and optimized traffic, makes the Internet more robust than TCP when facing DoS attacks.

Keywords : Internet measurement, next generation Internet, quality of service, network robustness, transport protocol, congestion control.

Remerciements

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS-CNRS), dirigé au cours de mon séjour successivement par MM. Jean Claude Laprie et Malik Ghallab, que je tiens à remercier cordialement pour leur accueil. J'adresse également mes plus sincères remerciements à MM. Michel Diaz et Jean-Pierre Courtiat, directeurs de recherche au CNRS et responsables du groupe Outils et Logiciels pour la Communication, pour m'avoir accepté au sein de leur équipe. Je suis également très reconnaissant pour le temps et le travail accordés par l'ensemble des membres du jury de ma thèse : M. Christophe CHASSOT, maître de conférences à l'INSA de Toulouse, M. Serge FDIDA, professeur à l'université Pierre et Marie Curie ; M. Olivier FESTOR, directeur de recherche au LORIA, M. Fabrice GUILLEMIN, ingénieur recherche et développement à France Télécom R&D ; M. Guy LEDUC, professeur à l'EPFL. En particulier, je remercie MM S. FDIDA et G. LEDUC d'avoir accepté d'être les rapporteurs de cette thèse. Je tiens à remercier Philippe OWEZARSKI, mon directeur de thèse, qui m'a apporté bien plus qu'un encadrement scientifique. Il a su par son soutien me donner confiance et me pousser à m'améliorer au cours de ces trois années et demi de collaboration. Il m'a offert un contexte de travail de très bonne qualité qui m'a permis d'explorer le domaine des réseaux de télécommunication dans les meilleures conditions. Je lui suis également reconnaissant pour tous ses gestes d'encouragement. Je veux remercier également toutes les personnes directement liées à mes travaux : Augustin SOULE (actuellement doctorant au LIP6) sans qui le déploiement de la plateforme METROPOLIS aurait été beaucoup plus difficile, Laurent BERNAILLE (doctorant du LIP6) pour sa maîtrise de l'outil TD de la société QoSMOS. Je remercie également les stagiaires que j'ai co-encadrés avec mon directeur de thèse pour le travail fourni et la bonne humeur qu'ils ont apportés : MM. Hubert MARTIN-DEIDIER, Nadhem MARSIT et Yu ZHANG. Merci également à toute l'équipe OLC pour son accueil et sa bonne humeur : Thierry G. et Thierry V., Pascal, Olivier, Nico R., Yann... Je remercie aussi toutes les personnes du LAAS qui m'ont permise d'avoir un cadre de travail agréable : Mmes A. Bergez, A. Evrard, C. Moulin, J. Penavayre et G. Briand ainsi que MM. C. Berty, D. Daurat, E. Le Denmat, B. Meunier et P. Pichon. Ils ont tous contribué à m'offrir de bonnes conditions matérielles de travail. Merci aussi à tous les autres amis que j'ai côtoyés au cours de ces trois années soit directement au LAAS (Samuel, Vincent...) soit à «l'extérieur du laboratoire» mais dont le soutient régulier m'a aidé à surmonter des moments parfois difficiles : Vincent, Blaise, Seb, Franck, Greg, Pascal, Cyril, Manu, Lolo, Olivier, Pierre, Sophie, Sandrine, Fabien, Mag, Nours, Marie, Yannick, Laure... Je souhaite enfin remercier tous mes proches (en particulier mes parents et ma soeur Sophie) qui ont été soumis à rude épreuve lors de l'accompagnement de mes travaux et qui ont toujours su trouver les mots justes pour m'encourager. Un grand merci !

Table des matières

Glossaire

- ACI : Action Concertée Incitative
- ADSL : Asymetric Digital Subscriber Line
- AMP : Active Measurement Project
- ATM : Asynchronous Transfer Mode
- CA : Congestion Avoidance
- CAIDA : Cooperative Association for Internet Data Analysis
- CBQ : Class-Based Queueing
- CBR : Constant Bit Rate
- CCDF : Complement of Cumulative Distribution Function
- CCID : Congestion Control IDentifier
- CL : Controlled Load
- CNIL : Commission Nationale Informatique et Liberté
- CNRS : Centre National de la Recherche Scientifique
- CSMA / CA : Carrier Sense, Multiple Access with Collision Avoidance
- CSMA / CD : Carrier Sense, Multiple Access with Collision Detection
- DCCP : Datagram Congestion Control Protocol
- DdS : Déni de Service
- DDdS : Déni de Service Distribué
- DiffServ WG : Differentiated Services Working Group
- DoS : Denial of Service
- DRR : Deficit Round Robin
- DSCP : DS CodePoint
- DVMRP : Distance Vector Multicast Routing Protocol
- E-NEXT : Emerging Networking Experiments and Technologies
- ECMN : Explicit Congestion Notification
- ERF : Extensible Record Format
- FAI : Fournisseur d'Accès Internet
- FTP : File Transfert Protocol
- GET : Groupe des Ecoles des Télécommunications
- GS : Guaranteed Service
- HTML : Hyper Text Markup Language
- HTSPN : Hierarchical Time Stream Petri Net
- HTTP : Hyper Text Transfert Protocol
- IETF : Internet Engineering Task Force
- INRIA : Institut National de Recherche en Informatique et en Automatique
- INSA : Institut National des Sciences Appliquées

– IntServ WG : Integrated Services Working Group
– IPPM : IP Performance Metrics
– ISP : Internet Service Provider
– IST : Information Society Technologies
– LAAS : Laboratoire d'Analyse et d'Architecture des Systèmes
– LBNL : Lawrence Berkeley National Laboratory
– LIP6 : Laboratoire d'Informatique de Paris VI
– LRD : Long Range Dependence ou dépendance longue mémoire
– MBA : Measurement Based Architecture
– MBCC : Measurement Based Congestion Control
– MBN : Measurement Based Networking
– METROPOLIS : Métrologie pour l'Internet et les Services
– METROSEC : Métrologie pour la Sécurité et la Qualité de Service
– MIB : Management Information Base
– MINC : Multicast-based Inference of Network-internal Characteristics
– M-JPEG : Motion Joint Photographic Experts Group
– MOME : Monitoring and Measurement
– MPEG : Motion Picture Experts Group
– MSP : Measurement Signaling Protocol
– MTU : Maximum Transmission Unit
– NAM : Network AniMator
– NLANR : National Laboratory for Applied Network Research
– NNTP : Network News Transport Protocol
– NSF : National Science Foundation
– OCxMON : Optical Carrier Monitoring
– P2P : Pair-à-Pair
– PHB : Per Hop Behavior
– PIM : Protocol Independent Multicast
– POP : Point of Presence
– QdS : Qualité de Service
– QoS : Quality of Service
– RAM : Random Access Memory
– R&D : Recherche et Développement
– RED : Random Early Detection
– RENATER : Réseau National pour l'Enseignement et la Recherche
– RIPE : Réseaux IP Européennes
– RSVP : Resource reSerVation setup Protocol
– RTP : Real-Time Transport Protocol
– RTSP : Real-Time Streaming Protocol
– RTT : Round Trip Time
– SACK : Selective ACKnowledgement
– SCP : Secure CoPy
– SCTP : Stream Control Transmission Protocol
– SFQ : Start-time Fair Queuing
– SLA : Service Level Agreement
– SMTP : Simple Mail Transfert Protocol

– SNMP : Simple Network Management Protocol
– SRM : Scalable Reliable Multicast
– TCL : Tool Command Language
– TCP/IP : Transport Control Protocol / Internet Protocol
– TFRC : TCP-Friendly Rate Control
– TO : Time Out
– UCB : University of California, Berkeley
– UINC : Unicast INC
– USC / ISI : University of Southern California / Information Sciences Institute
– v.a. : variable aléatoire
– VBR : Variable Bit Rate
– VTHD : Vraiment Très Haute Vitesse
– WWW : World Wide Web

Introduction

L'Internet est en train de devenir le réseau universel pour tous les types d'informations, du transfert simple de fichiers binaires jusqu'à la transmission de la voix, de la vidéo ou d'informations interactives en temps-réel. L'Internet se doit donc de fournir de nouveaux services adaptés aux applications Internet et aux données qu'elles transmettent. De plus, l'Internet croît très rapidement, en taille (nombre d'utilisateurs, d'ordinateurs connectés, etc.) et en complexité, en particulier à cause de la nécessité d'offrir de nouveaux services et d'optimiser l'utilisation des ressources de communication pour améliorer la qualité de service offerte aux utilisateurs. A cause de la complexité grandissante de l'Internet, l'évolution de ce réseau global est indissociable d'une parfaite connaissance et compréhension des caractéristiques du trafic. Par conséquent, le développement d'outils et de techniques de métrologie réseau pour capturer le trafic Internet ainsi que de méthodologies pour analyser ses caractéristiques est aujourd'hui un sujet d'ingénierie et de recherche de premier plan. La métrologie – au sens littéral "la science des mesures" – est en train d'apparaître dans de nombreux domaines du monde des réseaux comme :

- la caractérisation et la modélisation du trafic,
- l'analyse du trafic et du réseau (comportement),
- l'ingénierie des trafics,
- l'optimisation de la QdS et des performances,
- la tarification,
- la sécurité,
- l'administration de réseau...

Parmi les domaines d'étude et de recherche concernés par la métrologie, celui de la QdS dans l'Internet n'est certainement pas le moindre. En effet, les propositions faites dans la communauté Internet, ces dix dernières années, pour offrir des QdS différenciées et garanties n'ont pour l'heure pas complètement abouti et continuent à se heurter à de nombreuses difficultés liées à la complexité des interconnexions de réseaux et à leur hétérogénéité de ressources. En particulier, définir et quantifier la QdS dans l'Internet reste un problème non résolu. Les approches issues du monde de la téléphonie et basées sur des métriques simples comme le débit, le délai ou le taux de perte restent insuffisantes pour modéliser entièrement et finement les caractéristiques et performances du trafic Internet. La modélisation du trafic dans son ensemble est une tâche à réinventer. Les travaux les plus récents ont essayé de décrire la variabilité du trafic Internet, dont la dynamicité est à la base des difficultés rencontrées par les chercheurs et les ingénieurs réseaux pour mettre au point des techniques de garantie de la QdS. Ces travaux de modélisation sont parvenus à montrer que le trafic Internet est très loin des modèles simples Poissonniens et Markoviens utilisés dans le monde de la téléphonie, et que les modèles qui représentent mieux le trafic Internet sont des modèles ayant des propriétés d'auto-similarité

ou de dépendance longue mémoire (LRD) [100]. La métrologie des réseaux de l'Internet doit permettre d'apporter une réponse à ces questions concernant le (ou les) modèle(s) de trafic de l'Internet qui font aujourd'hui défaut. En particulier, la méconnaissance du trafic Internet est vraisemblablement à la base des difficultés rencontrées pour la mise en œuvre de mécanismes de garantie de QdS, car il était alors impossible de confronter des solutions théoriques à des conditions réalistes de trafic.

Ainsi, la métrologie de l'Internet n'est appliquée dans la recherche, l'ingénierie et la conception des réseaux Internet que depuis le début des années 2000, mais cette approche est de plus en plus populaire et devrait tendre à se généraliser. Le principe, développé dans le chapitre 1, consiste à étudier, caractériser, analyser et modéliser le trafic existant sur les liens de l'Internet, afin de comprendre les principes qui régissent le comportement du réseau par rapport à un trafic qui s'avère méconnu. C'est donc le processus de recherche et d'ingénierie des réseaux qui se trouve modifié par l'ajout d'une phase métrologique en amont permettant de collecter des données et des connaissances sur l'existant, pour permettre, ensuite, de concevoir et mettre en œuvre de nouvelles architectures protocolaires optimisées.

Ainsi, notre premier axe de recherche, détaillé dans le chapitre 2, a été consacré à la caractérisation du trafic Internet, une phase de travail préliminaire et primordiale pour la compréhension à la fois des phénomènes observables dans le réseau et des mécanismes qui les régissent. Ces différentes études, détaillées dans le chapitre 2, ont été menées dans le cadre du projet METROPOLIS labellisé par le RNRT en déployant une plate-forme de métrologie passive qui permet la collecte et l'analyse de traces de trafic sur le réseau RENATER. Ces études ont été aussi conduites dans le cadre du réseau d'excellence européen E-NEXT financé par la communauté européenne. Nous avons, en particulier, analysé des trafics prélevés dans des situations topologiques différentes : au niveau du cœur de réseau et au niveau des liens d'accès à ce dernier. Les premiers résultats ont fait apparaître des caractéristiques très différentes pour les deux types de traces. En effet, nous montrerons dans la suite que le trafic en bordure est fortement variable (pour des échelles de temps diverses), il présente de la corrélation (par exemple au niveau des arrivées de paquets) et dans certains cas de la dépendance longue mémoire. A l'inverse, plus on analyse le trafic en cœur de réseau, plus les caractéristiques d'oscillation et de variabilité observables en bordure s'atténuent pour tendre vers un comportement sensiblement Poissonien. D'autre part, ces travaux mettront en évidence une transformation du profil du trafic Internet au cours des dernières années. En particulier, la taille des flux échangés sur ce réseau est de plus en plus importante, en grande partie à cause de l'explosion du trafic P2P. Ces applications, par les oscillations à long terme qu'elles induisent, créent de la LRD dans le réseau. Cette LRD ainsi que ces oscillations sont très néfastes pour la QdS du réseau étant donné qu'elles provoquent des phénomènes importants de congestion et un niveau de service très instable pour les utilisateurs. De plus, vous verrons par la suite que les oscillations à long terme créent des propriétés de non-stationnarité dans le trafic, la valeur moyenne du trafic changeant fréquemment de façon significative. En effet, il est aisé d'observer des phénomènes de non stationnarité dans le trafic sur des périodes longues (plusieurs heures). Ces différences sont dûes aux alternances jour / nuit ou encore à la présence de pauses dans la journée (pause déjeuner par exemple).

Ces travaux de caractérisation, nous ont permis, de plus, de disposer du maximum d'informations concernant le comportement des mécanismes de contrôle de congestion (TCP notamment) dans les périodes de surcharge ou de congestion du réseau (à l'origine de l'apparition des pertes). Ainsi, nous pourrons par la suite, proposer de nouveaux mécanismes de contrôle de

congestion qui réduiront (voire annuleront) les perturbations (i.e. les oscillations) introduites dans le trafic et de ce fait, rendre la QdS des services Internet plus stable (cf. chapitre 3). Pour y arriver, les mécanismes traditionnels de contrôle de congestion de TCP seront supprimés et remplacés par de nouveaux qui, au lieu d'être chaotiques, convergeront vers un comportement stable. Si le phénomène oscillatoire est annulé (ou très fortement réduit), cela se traduira par une optimisation de l'utilisation des ressources de communication, car il ne sera plus nécessaire de sur-dimensionner les réseaux.

Ainsi, l'analyse des caractéristiques du trafic Internet a mis en évidence sa variabilité et son instabilité très importante. Dès lors, les solutions avancées, à l'heure actuelle, pour améliorer les caractéristiques du trafic Internet, par leur caractéristique statique, sont sous optimales face à l'instabilité très importante du trafic Internet. En effet, la proposition de l'utilisation du mécanisme TFRC (telle qu'elle sera détaillée à la fin du chapitre 1), bien qu'elle va mettre en évidence une diminution et le lien existant entre le degré de variabilité et le niveau de dépendance à long terme du trafic, traduira aussi, un niveau d'utilisation globale du réseau légèrement plus faible que celui obtenu avec le protocole TCP. En effet, un mécanisme de contrôle de congestion comme TFRC (implanté sur les hôtes à la périphérie du réseau), n'est pas conscient de l'état du réseau à tout instant et ainsi, pour ne pas être dangereux pour le réseau, se doit d'être moins agressif que les mécanismes actuels implantés dans TCP. La solution pour disposer d'un mécanisme à la fois respectueux de l'état du réseau et efficace en terme d'utilisation des ressources disponibles réside dans la modification du comportement des entités d'extrémités qui doivent évoluer de mécanismes statiques vers des mécanismes dynamiques. En effet, elles doivent être capables de s'adapter à la variabilité ainsi que la dynamicité du trafic et des ressources disponibles. Pour cela, ces dernières doivent être conscientes en temps réel de l'évolution des différents paramètres du réseau et capables d'intégrer ces modifications dans leurs algorithmes de réaction.

Cette connaissance de la dynamique du trafic peut être acquise par les équipements de métrologie qui permettent en temps réel de disposer d'informations sur l'état du réseau et sur son évolution au cours du temps. La suite de notre travail de thèse s'est donc orientée sur la définition d'une approche de gestion des comportements du réseau basée sur les résultats de mesure. Une première proposition d'architecture, intitulée "Measurement Based Networking" (MBN) a donc été définie (cf. chapitre 3). Les applications de cette nouvelle approche pour la gestion du réseau sont très importantes. En effet, les informations métrologiques peuvent servir à la définition de nouveaux mécanismes protocolaires, de nouvelles techniques de tarification, de gestion de la QdS... Ainsi, la suite de notre travail a donc porté sur l'exploitation des techniques de métrologie réseau en temps réel afin de définir une architecture orientée mesure (appelée dans la suite MBA) et l'ensemble des mécanismes nécessaires permettant de mieux adapter les mécanismes du réseau aux fréquents changements mesurés dans le trafic. Nous verrons, plus particulièrement, un composant essentiel de cette architecture, le protocole MSP, qui permet d'informer en temps réel tous les acteurs du réseau de l'évolution des caractéristiques du trafic. Bien sur, la délivrance d'information sur l'état du réseau passe par le déploiement d'un système distribué de mesure temps réel que nous avons défini, mis en œuvre et validé en simulation . Etant donné que les exemples d'applications de l'approche MBN sont nombreux, nous avons décidé de l'illustrer au travers d'un nouveau mécanisme de contrôle de congestion orienté mesures (appelé dans la suite MBCC). Il s'agit d'une approche pour mieux contrôler le trafic qui devra être capable de limiter le nombre de congestions et de pertes dans le réseau mais aussi de pouvoir améliorer la régularité du trafic et l'utilisation des ressources

du réseau.

Comme nous le verrons par la suite, l'approche MBN et le mécanisme MBCC vont permettre de lisser le trafic et ainsi de réduire l'impact de ruptures dans le trafic même celles liées aux attaques. Par conséquent, dans la dernière partie de notre travail de thèse, nous aborderons le domaine de la sécurité des réseaux informatiques, en presentant une application du mécanisme MBCC pour améliorer la robustesse de la QdS réseau dans le cas extrême d'un trafic d'attaques (i.e. comportant des flux de déni de service). Ce sera l'objet du chapitre 4 dans lequel nous quantifierons précisément l'apport de MBCC par rapport à TCP lors qu'ils sont confrontés à ce type de trafic extrèmement variable et instable.

Ainsi, au travers de ce travail de thèse synthétisé dans le présent document, nous allons montrer les possibilités offertes par la métrologie dans plusieurs axes de recherche des réseaux : le contrôle de trafic, la QdS ainsi que la lutte contre les attaques de DdS ; tout cela dans le but de fournir des services Internet stables.

Chapitre 1

Les caractéristiques du trafic Internet et leurs évolutions

Le réseau Internet subit depuis quelques années une explosion considérable en terme d'augmentation du nombre d'utilisateurs et de la quantité du trafic qu'il doit transporter, en même temps qu'une mutation au niveau de ses usages comme nous l'avons mentionné en introduction. Ainsi, il est nécessaire d'opérer une évolution technologique du réseau de façon à le rendre capable de transporter les paquets des différents types d'informations proposées par toutes les applications utilisant l'Internet avec des QdS adéquates. Cette mutation technologique est encore aujourd'hui problématique, et il est très difficile de trouver de nouvelles architectures et de nouveaux protocoles capables d'offrir des mécanismes universels pour la gestion de la QdS dans l'Internet. Les premières tentatives se sont soldées par des échecs, notamment par un manque de connaissance globale de tous les protocoles et mécanismes de l'architecture TCP/IP, ainsi qu'un manque de connaissance des comportements des utilisateurs et des trafics qu'ils génèrent. Ainsi, nous allons voir dans la suite de ce chapitre comment la métrologie peut se positionner comme un outil adéquat pour aider à dissiper une partie de cette méconnaissance des caractéristiques du trafic et des comportements réseaux et utilisateurs.

1.1 Métrologie de l'Internet : un nouvel outil pour la recherche en réseaux

La métrologie, ou science des mesures, est une activité en plein essor dans le domaine des réseaux IP. Les opérateurs réseaux utilisent des techniques de métrologie depuis la mise en place des premiers réseaux de communication, mais cette discipline n'a jusqu'à présent jamais été utilisée comme elle aurait dû l'être. Pour l'instant, les opérateurs utilisent la métrologie, souvent passive et en ligne (à partir de SNMP et de ses MIB), pour faire de la supervision du réseau ainsi que du trafic qui circule dessus. Mais ce type de solution ne permet pas une analyse du trafic très fine. En effet, le protocole SNMP associé aux MIB ne permet pas de considérer des granularités d'observation inférieures à quelques minutes. Or nous verrons dans la suite de ce chapitre qu'il est nécessaire de considérer l'ensemble des échelles (des plus fines aux plus larges) pour une bonne analyse du trafic. Dès lors, le besoin d'outils de métrologie plus performants (capables de considérer à la fois des échelles d'analyse inférieures et supérieures à la seconde) devient nécessaire.

Avec l'essor du réseau Internet, la métrologie devient la pierre angulaire de nombreuses activités autant au niveau de la recherche en réseau que de sa conception, sa mise en place ou encore sa gestion. En effet, la métrologie recouvre maintenant des domaines d'étude comme :

- La classification des flux et du trafic, soit pour pouvoir trier les flux en fonction de la qualité de service qu'ils requièrent, soit, par rapport à des problèmes de routage, pour pouvoir les encapsuler dans des "trunks" [1] de trafic et leur faire tous globalement emprunter la même route optimale.
- Le dimensionnement des réseaux qui permet de mettre en place des capacités suffisantes pour assurer en permanence un service de qualité adapté à tous les utilisateurs.
- L'analyse des mécanismes du réseau, et ce autant aux niveaux des mécanismes des routeurs, des algorithmes de routage et des mécanismes de transport assurant le contrôle de flux, d'erreur, de congestion... Cette analyse permet de comprendre comment tous ces mécanismes interagissent entre eux, et de régler de façon fine les différents paramètres mis en jeu. D'un point de vue théorique, les résultats d'analyse permettent la conception de nouveaux mécanismes et protocoles.
- L'échantillonnage qui consiste à déterminer les moments et les endroits stratégiques à observer afin d'obtenir une vision globale (à partir d'un ensemble d'informations partielles) du réseau et du trafic.
- La modélisation de trafic qui permet de représenter et comprendre le trafic actuel, et ensuite d'adapter les paramètres du réseau aux caractéristiques du trafic pour, dans un deuxième temps, pouvoir réaliser de la prédiction de trafic et ainsi s'adapter, par exemple, aux caractéristiques de l'Internet du futur.
- La tarification et les SLA [2] qui permettent de définir des coûts de service en relation avec les ressources consommées, par exemple.

1.1.1 Les principes de la métrologie réseau

De nombreux projets de recherche en métrologie sont en cours, en particulier conduits par des opérateurs Internet et des laboratoires d'études et de recherche partout dans le monde. Ces projets peuvent être répartis en deux grandes classes : ceux fondés sur les mesures actives et ceux reposant sur les mesures passives décomposées elles-mêmes en mesures passives en ligne et hors ligne. Chacune de ces deux classes permet de mieux comprendre le comportement à la fois du réseau (observation des taux de perte, des délais...) et des applications (réactions en temps réel des applications aux pertes dans le réseau, du taux de transmission utile...) et de mettre en lumière les interactions entre les applications et le réseau.

Métrologie active : principe et exemples

Le principe des mesures actives consiste à engendrer du trafic dans le réseau pour l'étudier et observer les effets des équipements et protocoles – réseaux et transport – sur le trafic : taux de perte, délai, RTT... Cette première approche possède l'avantage de prendre un positionnement orienté utilisateur. Les mesures actives restent le seul moyen pour un utilisateur

1. Il s'agit d'un très gros volume d'information représentant un ensemble de flux appartenant à des machines différentes dans le réseau mais possédant des liens communs au niveau de l'itinéraire suivi par les paquets d'information qui les constituent.

2. Ce sont les critères définis entre l'opérateur et son client qui stipulent les conditions normales d'utilisation du réseau et des QdS qui seront fournies ainsi que les sanctions encourues lors du non respect de ces critères.

de mesurer les paramètres du service dont il pourra bénéficier. En revanche, l'inconvénient majeur de cette approche est la perturbation introduite par le trafic de mesure qui peut faire évoluer l'état du réseau et ainsi fausser la mesure. De nombreux travaux menés actuellement abordent ce problème en essayant de trouver les profils de trafic de mesures qui minimisent les effets du trafic supplémentaire sur l'état du réseau. C'est par exemple le travail en cours au sein du groupe IPPM de l'IETF [99] [10] [11] [12]. Les mesures actives "simples" sont tout de même majoritaires dans l'Internet pour lequel de nombreux outils de test, de validation et / ou de mesure sont disponibles. Parmi eux, on peut citer les très célèbres *ping* et *traceroute*.

- *Ping* permet de vérifier qu'un chemin est valide entre deux stations et de mesurer certains paramètres comme le RTT ou le taux de perte.
- *Traceroute* permet de voir apparaître l'ensemble des routeurs traversés par les paquets émis jusqu'à leur destination et donne une indication sur les temps de passage en chacun de ces nœuds. L'un des projets les plus simples en théorie était le projet Surveyor [71] de la NSF aux Etats-Unis qui reposait sur l'utilisation de *Ping*, amélioré par la présence d'horloges GPS sur les machines de mesure. L'objectif était d'étudier les délais de bout en bout et les pertes dans l'Internet. Plusieurs projets ont actuellement pour sujet les mesures actives.
- Le projet NIMI (initié par Vern Paxson aux Etats-Unis) [103] a pour objectif le déploiement d'une infrastructure nationale (au niveau des Etats-Unis) de mesures actives. Cette infrastructure est flexible et permet le recueil de diverses mesures actives. Elle a été utilisée durant les deux ou trois années passées pour plusieurs campagnes de mesures, dont la détermination d'une matrice de distance dans Internet. L'infrastructure NIMI s'est aussi étendue en Europe, notamment en Suisse.
- Initialement en Europe, le projet RIPE [105], tente de déployer une infrastructure semblable à celle de NIMI. Cette infrastructure s'est depuis étendue au reste du monde. Par rapport à NIMI, RIPE fournit des services à ses clients : RIPE se propose de réaliser des études qui peuvent être demandées par ses clients, en plus des services classiques d'accès à des statistiques globales d'utilisation des chemins des réseaux de recherche surveillés.
- Le projet MINC [4] [31] était un client du projet NIMI. Il utilisait la diffusion de sondes actives par le biais du multicast pour inférer la structure interne du réseau et les propriétés sur tous les liens d'interconnexion ainsi traversés. En allant plus loin, c'est la tomographie qui est au centre de ce projet qui se focalise sur certains aspects dynamiques du trafic, comme les propriétés du routage, les pertes et les délais. Toutefois, comme le multicast n'est pas un service disponible partout, et comme il a été montré que le trafic dans l'Internet n'est pas symétrique, l'exploitation du multicast reste parfois difficile bien que son intérêt dans cette tâche ait été démontré. Aussi, le projet UINC a vu le jour et tente de reproduire le travail de MINC en unicast.
- Le projet Netsizer [85] de Telcordia (ex Bellcore) a pour objectif de mesurer la croissance de l'Internet, les points durs de congestion, les délais... Pour cela, depuis une ensemble de stations situées chez Telcordia, un programme teste la présence sur le réseau de toutes les adresses IP existantes et met à jour suivant les résultats une carte de l'Internet. Un des gros problèmes de ce projet reste la difficulté de représentation de ces informations.
- Le projet Européen INTERMON [80], qui se situe dans le prolongement du projet européen AQUILA [15], se propose de développer un ensemble d'outils de bout en bout pour permettre une caractérisation de la QdS dans les réseaux à grande échelle (Internet en particulier). L'idée est de pouvoir ainsi faciliter la définition de SLA entre les clients et

les opérateurs Internet Européens. En particulier, un des objectifs de ce projet est de pouvoir détecter plus facilement quand les paramètres du contrat de service sont rompus et quelles sont les causes de ces dysfonctionnements : par exemple un simple non respect par les clients des paramètres définis dans le SLA ou à l'inverse des disfonctionnements réseaux plus problématiques pour l'opérateur. Dans ce dernier cas, l'utilisation d'outils basés sur *traceroute* est privilégiée pour détecter par exemple les changements et ruptures dans les routes Internet.

- Le projet américain AMP du NLANR [13] est un projet débuté il y a maintenant 5 ans. Ce fut un projet novateur dans le domaine de "l'active probing". Les résultats collectés par mesures actives ont permis d'étudier l'impact de l'augmentation de la charge du trafic dans le réseau et ainsi améliorer sa conception, sa capacité ainsi que sa disponibilité à long terme.
- Le projet MOME [82] est une action coordonnée de l'IST et d'un programme de l'union européenne visant à offrir une plateforme commune pour l'échange d'outils et de connaissance. Il doit permettre la coordination des activités menées dans le cadre de la métrologie et de la mesure des réseaux IP entre les différents projets et les autres partenaires européens. Ainsi, la plateforme fournit des informations pour tester l'interopérabilité des outils de mesures des différents projets. Enfin, une base de données qui contient les résultats des mesures collectées est mise à disposition.

Métrologie passive : principes et exemples

Les projets de mesures passives sont apparus beaucoup plus tardivement que les projets de mesures actives car ils nécessitent des systèmes de capture et d'analyse du trafic en transit relativement avancés. Le principe des mesures passives est d'analyser le trafic et d'étudier ses propriétés en un ou plusieurs points du réseau. L'avantage des mesures passives est qu'elles ne sont absolument pas intrusives et ne changent rien à l'état du réseau lorsqu'on utilise des solutions matérielles dédiées (par exemple sur la base de cartes DAG [42] présentées dans la suite de ce manuscrit). De plus, elles permettent des analyses très avancées. En revanche, il est très difficile de déterminer le service qui pourra être offert à un client en fonction des informations obtenues en métrologie passive.

D'autre part, les systèmes de métrologie passive, peuvent se différencier en fonction du mode d'analyse des traces. Ainsi, le système peut faire une analyse en-ligne ou hors-ligne. Dans le cadre d'une analyse en-ligne, toute l'analyse doit être effectuée dans le laps de temps correspondant au passage du paquet dans la sonde de mesure. Une telle approche, temps-réel, permet de faire des analyses sur de très longues périodes et donc d'avoir des statistiques significatives. Par contre, la complexité maximale pour ces analyses reste très limitée à cause du faible temps de calcul autorisé. Une analyse hors-ligne oblige, à l'inverse, la sonde à sauvegarder une trace du trafic pour analyse ultérieure. Une telle approche demande ainsi d'énormes ressources ce qui représente une limitation pour des traces de très longue durée. Par contre, une analyse hors-ligne permet des analyses extrêmement complètes et difficiles, capables d'étudier des propriétés non triviales du trafic. De plus, comme les traces sont sauvegardées, il est possible de faire plusieurs analyses différentes sur les traces, et de corréler les résultats obtenus pour une meilleure compréhension des mécanismes complexes du réseau.

L'endroit idéal pour positionner des sondes de mesures passives est indéniablement dans les routeurs. CISCO a ainsi développé le module Netflow [84] pour ses routeurs ; celui scrute

le trafic en transit, et génère régulièrement des informations statistiques sur ce trafic. Netflow a ainsi été utilisé dans de nombreux projets présentés ci-après. L'expérience montre toutefois que les performances de Netflow restent limitées (code écrit en Java et interprété ainsi que échantillonage du trafic en 1/N), et que l'influence sur les performances du routeur est non négligeable.

- Le premier projet connu (le projet AT&T Netscope [52]) a débuté il y a 7 ans et repose sur le système Netflow de CISCO. Ce projet de mesures passives en ligne a pour but d'étudier les relations entre le trafic transitant en chaque nœud du réseau et les tables de routage des routeurs. L'objectif est d'utiliser ces résultats pour améliorer les politiques et décisions de routage, afin d'équilibrer au mieux la charge dans les différents liens du réseau, et ainsi améliorer la qualité de service perçue par chaque utilisateur. C'est de la tomographie réseau permettant de trouver ensuite des politiques adéquates d'ingénierie des trafics.
- Vern Paxson d'ACIRI a également conduit un projet de mesures passives en ligne dont l'objectif était de proposer un modèle pour les arrivées de flux et de paquets sur les liens de l'Internet. Ce travail [100] achevé depuis 1995 a été une référence dans le domaine de la caractérisation du trafic Internet. Cependant, aujourd'hui, avec l'apparition de nouvelles applications qui n'existaient pas à l'époque et avec les changements dans la façon d'utiliser l'Internet, ce travail doit être reconduit.
- De façon plus générale, le laboratoire CAIDA à San Diego [32], Californie, est spécialisé dans l'étude du trafic Internet et mène un projet dont l'objectif est d'étudier sur le long terme l'évolution du trafic, avec l'apparition des nouvelles applications comme les jeux, le commerce électronique, etc. D'autre part, ce projet étudie aussi les variations de trafic en fonction du moment de la journée, du jour de la semaine, de la période de l'année, etc. Pour ce faire, le système de métrologie reposait, à l'origine, sur les modules OC3MON [14] et OC12MON qui permettaient de traiter le trafic de liens IP / ATM dont les capacités respectent respectivement les normes OC3 (155 Mbps) et OC12 (622 Mbps). A l'heure actuelle, leur système permet de capturer le trafic des liens allant jusqu'à 2,5 Gbps (liens OC48). Enfin, pour l'analyse statistique, CAIDA a développé la suite logicielle CoralReef [36] qui est complémentaire des systèmes OCxMON.
- Ensuite, SPRINT a démarré il y a presque 5 ans un des projets les plus ambitieux du moment basé sur des mesures passives hors ligne. Ainsi, SPRINT enregistre des traces complètes de tous les entêtes de tous les paquets qui transitent en certains points de son réseau IP. Cette granularité microscopique permet d'approfondir les analyses que l'on peut faire dans la compréhension des interactions qui existent entre tous les flux, les mécanismes des routeurs, etc. A noter que le système IPMON de SPRINT, repose sur la carte DAG [42] conçue par l'université de Waikato [3] en Nouvelle Zélande et qui se charge d'extraire les entêtes des paquets, de les estampiller suivant une horloge GPS [43] et de les stocker sur un disque dur.
- Enfin, le projet français METROPOLIS [79] dans lequel s'inscrit ce travail a commencé depuis novembre 2001. Nous lui consacrons le paragraphe suivant.

3. Ce projet a donné naissance à une "jeune pousse", ENDACE, qui est maintenant en charge du développement des nouvelles cartes DAG.

1.1.2 Le projet METROPOLIS

Les études en métrologie des réseaux Internet ont été initiées à la fin des années 80 aux Etats Unis. En France, bien que cette discipline bénéficie d'un engouement considérable, elle restait très peu formalisée, parcellaire et dispersée entre plusieurs laboratoires universitaires et industriels. Le projet METROPOLIS dans le cadre duquel s'inscrit une partie de ce travail de thèse fut l'une des premières tentatives dans ce domaine en France. Il avait pour objectif de fédérer plusieurs laboratoires universitaires et de l'industrie. Ainsi, les différents partenaires étaient le LAAS, le LIP6, France Télécom R&D, l'INRIA, le GET, Eurécom et RENATER. Ce projet a été labellisé par le RNRT, a commencé en novembre 2001 et s'est achevé en février 2005. Il aborde les thèmes d'étude suivants :

- La classification du trafic et le dimensionnement du réseau ;
- L'analyse du réseau (protocoles, routeurs) ;
- La modélisation du trafic et de ses propriétés ;
- La définition de procédure de tarification et de mise en place de SLA.

La particularité de ce projet est qu'il combinait les approches de métrologie actives et passives, ce qui permettait de corréler ces deux types de mesures. Cette combinaison a apporté un avantage considérable par rapport aux autres projets. Son deuxième point fort résidait dans la diversité des réseaux sur lesquels ont été effectuées les mesures. Les réseaux étudiés étaient :

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

Néanmoins, les résultats présentés dans la suite de ce manuscrit ne s'appuient pas sur des traces collectées sur le réseau VTHD. D'autre part, nous utiliserons d'autres traces publiques (dont la liste est détaillée dans l'annexe E), à titre de comparaison pour vérifier s'il n'existe pas de spécificité française dans le trafic généré et les usages pratiqués sur le réseau Internet français.

Dans le cadre de Métropolis, il a été demandé à plusieurs responsables des réseaux académiques français l'autorisation de déployer les équipements de mesure sur le réseau qu'ils opèrent. L'objectif était de pouvoir déployer ces sondes sur des réseaux de natures différentes, soit sur le réseau de cœur, sur les réseaux régionaux et à la sortie des laboratoires. A l'issue du projet, 3 sondes DAG ont été déployées. Ainsi, comme le montre la figure 1.4, deux sondes en Fast-Ethernet ont été positionnées à la sortie de deux grands laboratoires que sont le LAAS (figure 1.1) et le LIP6 (figure 1.2). La troisième sonde en Giga-Ethernet a, quant à elle, été positionnée à la sortie du réseau de Jussieu sur RAP (figure 1.3), permettant ainsi d'avoir accès aux traces du trafic d'un réseau à très haut débit. Le détail des équipements passifs déployés dans le projet Metropolis est présenté dans l'annexe A.

Ce choix de positionnement a été aussi stratégique par rapport au positionnement des sondes de métrologie passive macroscopique et de sondes de métrologie active dont un exemplaire de chacune d'entre-elles a été placé également au LAAS et au LIP6 (figure 1.4). Ainsi, il a été possible, pour un même trafic de corréler les analyses micro- et macroscopiques (par l'intermédiraire des sondes QoSMOS, cf. annexe A pour le détail de cet équipement), ainsi que les mesures actives et passives, et ce en plusieurs points du réseau sur leur chemin entre Toulouse et Paris.

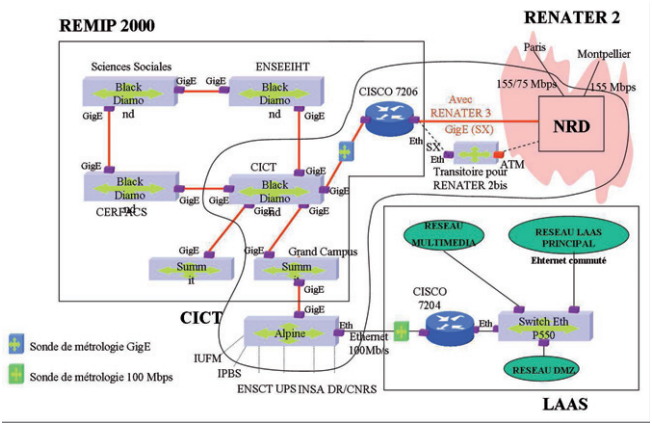


FIGURE 1.1 – Schéma de déploiement sur la plate-forme toulousaine

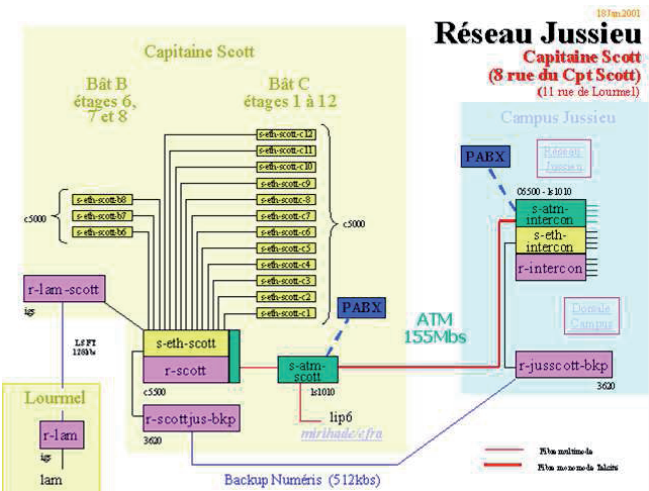


FIGURE 1.2 – Schéma de déploiement au LIP6

1.2 Trafic Internet : principes et notions associées

1.2.1 Notions mathématiques associées

Les premières études métrologiques sur le trafic Internet menées partout dans le monde ont globalement montré que ce dernier est particulièrement instable, à cause des propriétés d'auto-

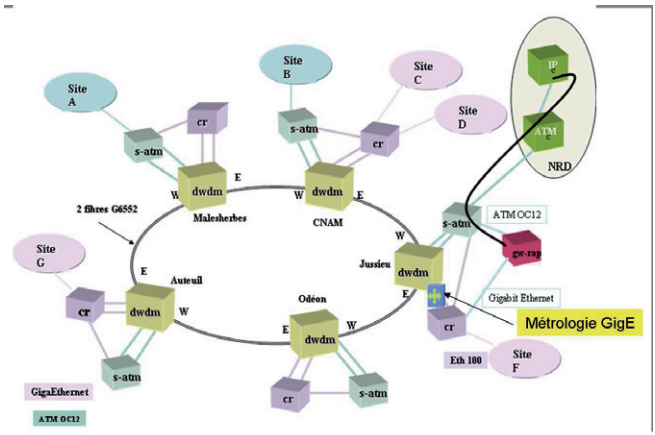


FIGURE 1.3 – Schéma de déploiement à Jussieu

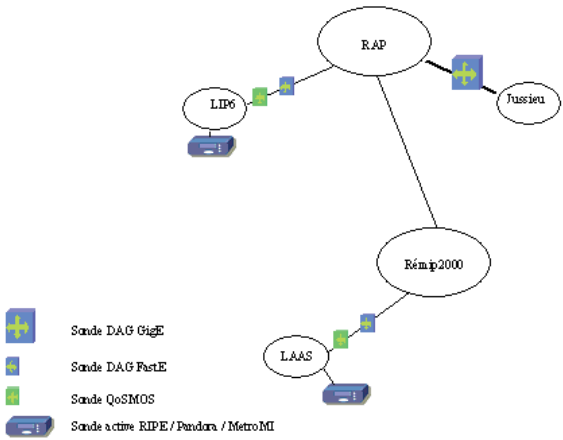


FIGURE 1.4 – Carte de déploiement de quelques sondes DAG, QoSMOS et MetroMI

similarité et de dépendance à long terme appelée aussi (LRD) [75]. Il est aussi montré que la distribution à queue lourde est très impliquée dans ces propriétés [137]. Avant de détailler dans la suite de ce chapitre toutes ces caractéristiques du trafic Internet, il est nécessaire d'introduire les notions mathématiques associées aux différents comportements observés dans

le réseau.

Fonction d'auto-corrélation

Avant de présenter cette fonction mathématique, il faut définir les notions d'indépendance et de corrélation :

– X, Y sont deux v.a. indépendantes ssi

$$P(X < x \cap Y < y) = P(X < x)P(Y < y) \tag{1.1}$$

– X, Y sont deux v.a. décorrélées ssi

$$E(XY) = E(X).E(Y) \tag{1.2}$$

En pratique, on dispose de N mesures. La fonction d'auto-covariance se calcule comme la fonction de covariance entre deux séries. La seconde série est ici la même que la première mais décalée d'un nombre K d'éléments. La fonction d'auto-covariance c_K s'écrit alors :

$$c_K = \sum_{k=0}^{K}(\frac{1}{N-k}\sum_{t=0}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})) \tag{1.3}$$

où $\bar{x}$ représente la moyenne de la série de points.

– Interprétation des résultats

Une caractéristique des lois et des distributions que l'on peut calculer à partir des traces réseaux est d'avoir une fonction d'auto-corrélation spécifique. En effet, elle traduit une corrélation persistante dans le temps [4] ainsi que la présence de dépendance à long terme entre les objets analysés (les paquets TCP la plupart du temps). Ainsi, il est nécessaire de pouvoir calculer de façon systématique la fonction d'auto-covariance d'une série représentant ses caractéristiques d'auto-corrélation. Pour ce qui est de l'interprétation de l'auto-corrélation d'une série, on considère que la covariance est nulle lorsque la corrélation empirique (donné par le graphique de $\frac{Autocov(K)}{Autocov(0)}$) est contenue entre $\frac{2}{\sqrt{n}}$ et $-\frac{2}{\sqrt{n}}$ (n étant le nombre de points sur lesquels on calcule la fonction d'auto-corrélation). Il s'agit de l'application du théorème de la limite centrale avec des hypothèses gaussiennes. Le niveau de confiance est alors de 95 % (cf. [40]). La courbe représentant l'auto-corrélation s'analyse en ayant au préalable calculé l'intervalle de confiance dans lequel la fonction doit se situer. Si la courbe dépasse cet intervalle, il existe de la corrélation. C'est une première information pour mettre en évidence dans un deuxième temps la présence de LRD dans la série analysée. En effet, on parle de LRD dans la série quand la corrélation reste présente pour une valeur de K grande.

Q-Q Plot d'une série

La représentation Quantile-Quantile (Q-Q) d'une série est une technique graphique pour déterminer si deux ensembles de données viennent de populations possédant une distribution commune. Le Q-Q Plot est la représentation de la fonction quantile d'une première série de données en fonction de la fonction quantile d'une deuxième série de données. La fonction

4. La fonction d'auto-corrélation ne se situe pas dans l'intervalle de confiance pour K grand (supérieur à 5000 : cette borne a été définie de manière empirique grâce aux calculs qui sont détaillés dans [Lar02]).

quantile d'une variable aléatoire est la fonction réciproque de sa fonction de répartition. Quand cette fonction de répartition est strictement croissante, sa fonction réciproque est définie sans ambiguïté. Mais une fonction de répartition reste constante sur tout intervalle dans lequel la variable aléatoire ne peut pas prendre de valeurs. C'est pourquoi on introduit la définition suivante : soit X une variable aléatoire à valeurs dans R, et F_x sa fonction de répartition ; on appelle fonction quantile de X la fonction, notée Q_x, de $]0,1[$ dans R, qui à $u \in\ ]0,1[$ associe :

$$Q_x(u) = inf\{x|F_x(x) \geq u\} \tag{1.4}$$

La première bissectrice est aussi tracée dans une représentation Q-Q Plot. Il s'agit d'une ligne de référence. Si les deux ensembles de données ont la même distribution, les points sont répartis approximativement le long de cette ligne de référence. Au contraire, plus la représentation s'éloigne de la ligne de référence, plus les deux ensembles de données comportent des distributions différentes. En pratique, la fonction quantile de la série de points est souvent comparée au quantile d'une série de points qui suit une loi exponentielle. Si le nuage de points suit la première bissectrice (cf. Figure 1.5) alors la distribution de la série de points peut être approchée par une loi exponentielle ; au contraire, si le nuage de points s'éloigne fortement de la première bissectrice, la distribution ne peut être approchée par une loi exponentielle. Dans ce dernier cas (cf. Figure 1.6), on est en présence d'une "queue lourde". Ce phénomène implique qualitativement la présence de dépendance longue dans la série des points analysés. Pour évaluer quantitativement ce phénomène de dépendance longue, on doit utiliser une méthode d'évaluation du facteur de Hurst (voir la section 1.3.3).

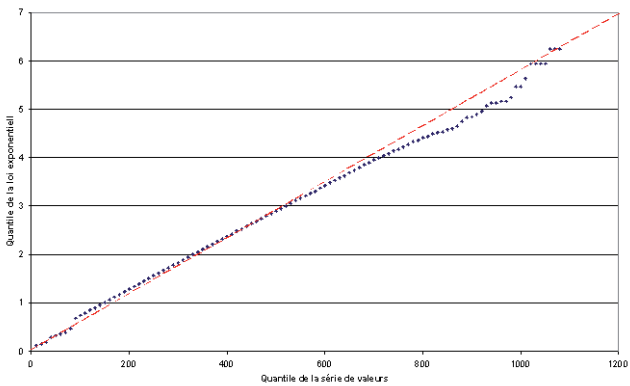


FIGURE 1.5 – Série de points dont la distribution suit une loi exponentielle

Processus à dépendance longue (LRD)

Un processus à dépendance longue ou à mémoire longue signifie que la dépendance entre deux variables du processus ne diminue pas trop rapidement avec l'éloignement temporel. La définition mathématique introduite par [37] est présentée ci-dessous :

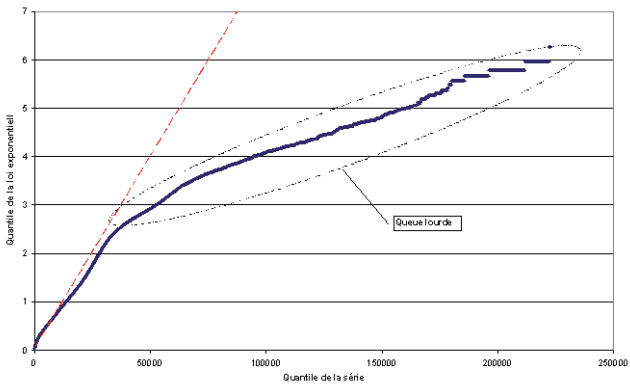


FIGURE 1.6 – Série de points dont la distribution ne suit pas une loi exponentielle

Soit $X = X_t$ un processus stochastique (à covariance) stationnaire à temps discret, on dit que X_t est à mémoire longue s'il satisfait les propriétés suivantes :

1. $\sum_{t=0}^{\infty} \rho(t) = \infty$ (ρ est la fonction d'auto-corrélation),
2. La densité spectrale S est singulière [5] à l'origine,
3. $m\ var X^{(m)} \to \infty$ quand $m \to \infty$.

Un processus à dépendance longue possède la propriété suivante :

$$\rho(t) \overset{t\to\infty}{\longmapsto} ct^{-\beta} \text{ où } 0 < \beta < 1 \text{ (ρ est la fonction d'auto-correlation).} \tag{1.5}$$

Ainsi la fonction d'auto-correlation décroît hyperboliquement.

La dépendance à long terme a été découverte en premier par Hurst qui la définit comme un processus ayant une fonction d'auto-corrélation non sommable (première propriété) et caractérisée par un paramètre H, défini par la formule $H = 1 - \frac{\beta}{2}$ [6], et appelé paramètre de Hurst. En 1993, Leland, Taqqu, Willinger et Wilson ont mis en évidence la LRD pour des séries temporelles de paquets Ethernet. Depuis, une multitude de travaux et d'articles ont traité de la dépendance à long terme du trafic Internet (voir [137], [129], [96], et [46]). Nous y reviendrons dans les sections suivantes.

Distribution à décroissance lente

Plusieurs travaux de recherches (voir [137], [129], [96] et [46]) ont démontré que la distribution à queue lourde pour certaines caractéristiques du trafic (distribution des tailles de fichiers, des durées de transfert...) est l'une des principales causes de LRD du trafic Internet.

5. Une fonction f est dite singulière en un point a si elle n'est pas explicitement définie en ce point (à cause par exemple d'une division par zéro si $x = a$ ou dans le cas d'une fonction définie sur un ensemble topologiquement ouvert, d'un point a qui est à la frontière de l'ensemble de définition de la fonction – C'est le cas de la fonction $ln(x)$ lorsque $x = 0$).

6. β étant le coefficient de la fonction d'auto-corrélation [22]

Une distribution est à queue lourde si sa fonction de distribution a la propriété suivante :

$$P[X > x] \overset{t\to\infty}{\longmapsto} x^{-\alpha}, \alpha \in\ ]0, 2[\tag{1.6}$$

En d'autres termes, la forme asymptotique de la distribution à queue lourde suit une loi exponentielle avec α inférieur à 2.

On l'appelle "à queue lourde" car, comparée à la distribution exponentielle et la distribution normale, une variable aléatoire qui suit une distribution à queue lourde peut montrer pour des très grandes valeurs de X une probabilité $P[X]$ supérieure à celle obtenue pour une distribution exponentielle équivalente. Cette variable a une variance infinie si $\alpha \in]0, 2[$ et une moyenne infinie si $\alpha \in]0, 1[$.

Processus auto-similaire

L'auto-similarité est une notion très importante dans la caractérisation du trafic Internet. En effet, la nature du trafic de données en général, celui d'Internet plus particulièrement, présente un aspect auto-similaire. Il s'agit de la manifestation du phénomène suivant : la structure des variations d'amplitude du signal analysé (par exemple le nombre d'octets transférés par unité de temps) se reproduit de manière similaire quelle que soit la finesse temporelle avec laquelle il est représenté. Ainsi, le comportement d'un trafic auto-similaire est à l'opposé de celui d'un trafic poissonnien, dont les variations d'amplitude sont filtrées au fur et à mesure que l'on augmente la taille de la fenêtre d'observation [100]. Il existe différentes définitions mathématiques de l'auto-similarité. La suivante concerne les processus à temps continu :

Un processus X(t) est dit auto-similaire de paramètre $H \in R$, si et seulement si pour tout $c > 0$, $c^H X(t)$ et $X(ct)$ possèdent les mêmes distributions jointes à tous les ordres. Ainsi pour tout entier n, t_1, ... , t_n, x_1 , ... , x_n :

$$P(X(t_1) \leq x_1, ..., X(t_n) \leq x_n) = P(X(ct_1) \leq c^H x_1, ..., X(ct_n) \leq c^H x_n) \tag{1.7}$$

Cette définition signifie que si l'on modifie l'échelle sur laquelle on observe le processus par un facteur positif c et que l'on "zoome" le même processus par ce facteur élevé à la puissance H, alors l'allure des deux processus obtenus est la même. Par conséquent, il n'y a pas une stabilisation vers une moyenne comme dans le cas du processus de Poisson.

1.2.2 Paramètres caractéristiques du trafic Internet

Les différents niveaux d'analyse : paquet, flux et session

L'analyse des caractéristiques du trafic Internet s'effectue commodément en se plaçant à un niveau de représentation selon trois entités de trafic, correspondant à trois échelles de temps différentes et, quoique de manière assez grossière, à trois niveaux (couches) de la pile protocolaire des réseaux de données :

– Les *paquets* forment l'entité de trafic la plus fine que l'on considère dans les réseaux de données, le paquet étant l'unité élémentaire traitée par la couche réseau des protocoles. Les paquets sont a priori de longueur variable dans un réseau IP et leur processus d'apparition est très complexe, en raison notamment de la superposition de services de nature très diverse et de l'interaction des couches protocolaires (dispositifs de contrôle de flux et de retransmission sur perte de paquets, comme avec TCP [24] par exemple). Le

trafic au niveau paquet possède dans de nombreux cas (trafic d'un réseau Ethernet par exemple [138]) des caractéristiques d'auto-similarité, laquelle rend très ardue l'évaluation de ses performances à ce niveau. Les échelles de temps décrivant le processus des paquets sont la microseconde et la milliseconde, en fonction des ordres de grandeur du débit de transmission des liens.

- Les *flux* constituent une entité de trafic intermédiaire que l'on pense être la mieux adaptée pour effectuer les études d'ingénierie du trafic IP. Ils correspondent à des transferts plus ou moins continus de séries de paquets associés à une même instance d'une application donnée. Les flux de type streaming (notion définie dans le paragraphe suivant) sont associés à des communications audio/vidéo (téléphonie sur IP, vidéoconférence) ou encore à des téléchargements en temps réel de séquences vidéo. Les flux de trafic élastique (notion définie dans le paragraphe suivant) sont créés par le transfert d'un fichier, d'un message, d'un objet (ou document) au sein d'une page HTML... Un flux correspond donc plus ou moins à la couche transport de la pile protocolaire Internet ; mais pas complètement puisque cette notion n'est pas nécessairement équivalente à celle d'une connexion TCP, par exemple, comme on le verra par la suite avec la notion de *flots*. On peut estimer que les flux ont une durée s'étendant de quelques secondes à quelques minutes, voire quelques heures.
- Les *flots* sont constitués de l'ensemble des flux qui ont des propriétés communes. Par exemple, les flux bi-directionnels échangés entre deux machines sur le réseau par une application particulière ou encore l'ensemble des paquets générés par un protocole particulier sur le réseau entre deux machines spécifiques. On parle par exemple de *flot* HTTP ou FTP.
- Au plus haut niveau, on peut tenter de définir la notion de *session* dans le but de se rapprocher des périodes d'activité des utilisateurs (transposition de la notion d'appels considérée en téléphonie à commutation de circuits). Pour le trafic streaming, ce niveau ne se distingue guère de celui des flux, du moins temporellement, puisque ce dernier correspond déjà à des communications ou des appels. S'agissant du trafic élastique, les sessions peuvent être associées à des connexions Telnet, FTP, ou à des envois de messages électroniques. La notion de session est (encore) plus floue au sujet des connexions de type WWW selon le protocole HTTP : on peut par exemple la définir comme étant la durée de transfert d'une page HTML dans son ensemble (comportant plusieurs objets à transférer) ou d'une suite de pages associées à une même consultation. Les sessions sont générées par la couche application des réseaux et l'ordre de grandeur de leur durée se situe entre quelques minutes et quelques heures.

Caractéristiques des niveaux paquet et flux

Un réseau de type Internet est dit multi-services : il a pour vocation de transporter un grand nombre de types de services possédant des caractéristiques de trafic différentes et éventuellement des contraintes de Qualité de Service différenciées. Cependant, dans le souci d'une modélisation simplifiée autant que pour les besoins opérationnels de gestion du réseau, l'on recherche plutôt une classification grossière des différents types de trafic. La plupart des auteurs s'accordent généralement pour distinguer deux grandes classes de trafics de télécommunications dans les réseaux à haut débit [106] :

1. *Trafic "streaming"*

Le trafic de type streaming, dont la durée et le débit ont une réalité intrinsèque bien que éventuellement variable. Souvent associé à la notion de services orientés connexion, son intégrité temporelle doit être préservée par le réseau. Le délai de transfert des données de même que sa variation, la gigue, doivent être contrôlables, tandis qu'un certain degré de perte de paquets peut être toléré. Les flux de trafic streaming sont typiquement produits par les services téléphoniques et vidéo (vidéoconférence ou téléchargement en ligne et en temps réel de séquences).

2. *Trafic "élastique"*

   Le trafic dit élastique, ainsi nommé car son débit peut s'adapter à des contraintes extérieures (bande passante insuffisante par exemple) sans pour autant remettre en cause la viabilité du service. Cette classe de trafic est essentiellement engendrée par le transfert d'objets numériques par nature (par opposition au transfert en mode numérique d'informations analogiques à la source) tels que des pages Web (application HTTP), des messages électroniques (e-mail, application SMTP) ou des fichiers de données (application FTP). Le respect de leur intégrité sémantique est indispensable mais les contraintes de délai de transfert sont moins fortes. Cette intégrité sémantique est la plupart du temps assurée par le protocole de transport (TCP) et ne constitue donc pas un élément de performance sur lequel l'opérateur de réseau peut agir ; en revanche, le maintien d'un certain débit effectif minimum de transfert des documents est un objectif de QdS.

Le trafic de type élastique est actuellement largement majoritaire sur les réseaux IP : on constate couramment [121] des proportions supérieures à 95% en volume (octets) et à 90% en nombre de paquets pour le trafic TCP, protocole sous lequel fonctionnent la plupart des applications mentionnées ci-dessus. Des proportions similaires ont été observées récemment sur le réseau de France Télécom.

Les travaux de ces dernières années sur la modélisation du trafic des paquets dans l'Internet ont montré qu'une caractéristique importante du trafic Internet est son aspect auto-similaire (ou possédant des dépendances à long terme). Cette auto-similarité – voire cette multi-fractalité – et son extrême variabilité à toutes les échelles de temps est caractéristique du principe même de la commutation de paquets qui induit des transmissions en rafales [106] [122]. Ce comportement se caractérise notamment par une décroissance lente, par exemple sous forme de loi puissance, sous exponentielle ou à queue lourde [69], de la fonction d'auto-corrélation du nombre de paquets transférés par unité de temps (typiquement 100 ms) : les processus auto-similaires sont des cas particuliers des processus à dépendance à long terme (LRD). Même pour les applications "stream", la caractéristique de LRD se retrouve dans les transferts de séquences vidéo à débit variable (VBR selon la terminologie ATM) [23], probablement due à la variabilité des paramètres de transmission liés au codage des trames (MPEG par exemple), à la dynamique des images, etc.

Concernant le trafic de type élastique, l'identification du processus des paquets tel qu'il est offert au réseau est particulièrement délicate. En effet, les dispositifs de correction d'erreur et de perte génèrent la retransmission de paquets supplémentaires et les mécanismes de contrôle de flux (TCP notamment) régulent les débits de transmission [24]. Les analyses de trafic doivent donc se contenter des données de trafic effectivement mesurées sur des liens, compte tenu de ces retransmissions et régulations. Le caractère auto-similaire du trafic TCP a été largement étudié. En complément de ce qui a été dit au paragraphe précédent, notons les tentatives d'explication : aux échelles de temps supérieures à un délai de transmission typique (RTT,

de l'ordre de 100 ms), le comportement auto-similaire serait dû à l'extrême variabilité de la taille des documents transférés (la loi de distribution est de type "à queue lourde", comme la loi de Pareto) ; tandis que les caractéristiques multi-fractales aux échelles de temps inférieures seraient provoquées par les mécanismes de contrôle de congestion du protocole TCP [54] [55]. C'est aussi la conclusion à laquelle [39] est arrivé lorsqu'il a analysé le trafic HTTP (dominant dans le trafic Internet). De la même manière, [75] a montré que le trafic Internet peut être représenté par un processus ON/OFF dont la distribution des durées des périodes ON est à queue lourde. Nous reviendrons en détails sur les causes de l'auto-similarité du trafic dans la section suivante.

Caractéristiques du niveau session

Le processus des demandes de communication, qu'elles soient de type streaming ou de type élastique, a toutes les raisons de pouvoir être considéré comme poissonnien (dans la mesure où l'on peut admettre que la population source est de taille quasi-infinie). C'est l'un des principaux invariants communément reconnus en modélisation du trafic Internet [57]. En effet, au niveau des sessions utilisateurs, le processus d'arrivée résulte de la superposition d'un nombre élevé de demandes élémentaires indépendantes entre elles (à opposer à la dépendance inter-flux au sein d'une même session). Des campagnes d'observation remontant au début de la précédente décennie, époque où le trafic Internet était essentiellement dominé par les applications FTP, Telnet ou SMTP, ont montré que les arrivées de session obéissaient correctement à un processus de Poisson [100], bien que ces diverses applications aient des modes de fonctionnement très différents (au niveau des connexions TCP générées par exemple).

Dans le but de caractériser les sessions Web, des observations indirectes ont été effectuées [53] par la collecte d'informations relatives aux appels par modem à destination d'un ISP : instants d'arrivée, taille et durée des appels. Bien que ces demandes de connexion ne contiennent pas uniquement des sessions Web, ces données ont été utilisées pour tenter de caractériser les processus liés à ces dernières. L'analyse, uniquement qualitative, montre que le processus d'arrivée est cohérent avec un processus poissonnien, du moins avec un processus de renouvellement (pour lequel les intervalles inter-arrivées sont indépendamment et identiquement distribués). En fait, peu de travaux ont eu pour objet de valider rigoureusement le caractère poissonnien du processus d'arrivée des sessions, probablement en raison du caractère naturel, presque évident, de l'hypothèse, mais aussi certainement à cause de la difficulté d'identifier des sessions générées par les utilisateurs à partir de traces réelles. Citons seulement l'article récent [89] qui obtient sur des données collectées aux Bell Labs (donc sur un réseau local) un excellent comportement, aux premier et second ordres, du modèle de représentation des arrivées de session par un processus de Poisson.

La loi des durées de session (ou des longueurs), quant à elle, possède des caractéristiques de distribution à décroissance lente, quoiqu'elle soit difficile à identifier comme indiqué plus haut. Que ce soit dans [100] ou dans [53], donc avec ou sans présence prépondérante des sessions Web, la loi des durées de session (ou de taille exprimée en Koctets) possède une queue de distribution caractéristique d'une loi de Pareto de moyenne finie, mais de variance infinie, significative d'une grande variabilité. D'un point de vue qualitatif, les paramètres quantitatifs des modèles de représentation étant bien entendu différents, les caractéristiques statistiques des durées de session sont dans l'ensemble similaires à celles des longueurs de flux.

1.2.3 Eléments de modélisation du trafic Internet

L'objectif de la modélisation du trafic est de donner un modèle réaliste des arrivées des flux, des paquets et des pertes. Cette action est indispensable, car les informations sur le trafic donneront les informations nécessaires pour la conception, le dimensionnement, la gestion et l'opération d'un réseau. D'autre part, elle donnera aussi les tendances d'évolution du réseau et de ces mécanismes, et permettront par exemple, de concevoir des simulateurs permettant de confronter les nouveaux protocoles de la recherche à des trafics Internet réalistes.

L'échec de la modélisation poissonnienne

Jusqu'à présent, les ingénieurs et chercheurs en réseau utilisaient le modèle poissonnien pour modéliser les processus d'arrivée du trafic. Or ce modèle, même s'il a été utilisé pour modéliser le trafic téléphonique, est incapable de représenter les rafales et les relations de dépendance qui existent entre les flux, les paquets et les pertes dans l'Internet. La figure 1.7 illustre l'écart à différentes échelles temporelles entre un trafic comportant un caractère auto-similaire (colonne de gauche) et un trafic poissonnien simulé (colonne de droite). Chaque graphique représente l'évolution du débit au cours du temps. On peut noter que dans les deux cas, les courbes de trafic ne se lissent pas avec la même vitesse lorsque la granularité de l'observation augmente (les graphiques de la première ligne possèdent la granularité la plus fine et la granularité augmente avec les lignes qui suivent ci-dessous). En effet, bien que l'amplitude des oscillations décroît lorsque la granularité d'observation est plus importante, il est clair que pour une granularité d'observation importante (1 seconde par exemple), l'amplitude des oscillations du trafic Internet est plus importante que celle du trafic poissonien. Cette particularité est le signe de la présence de LRD dans le trafic Internet (il existe des phénomènes oscillatoires qui dépassent la seconde).

Qualitativement, cela signifie qu'il est indispensable de surdimensionner les liens et les tailles des files d'attente dans les routeurs pour prendre en compte cette variance. Quantitativement, les tailles des files d'attentes dans les routeurs doivent être déterminées en fonction de la LRD du trafic. Ainsi, l'évaluation du facteur de Hurst d'un processus auto-similaire devrait aider à quantifier, comme nous le verrons dans la suite, le niveau de surdimensionnement nécessaire pour un fonctionnement optimal du réseau.

Cas particuliers

Dans le paragraphe précédent, nous avons parlé du trafic Internet comme un trafic qui n'est pas poissonien. Mais, en fait, dans la communauté scientifique, il existe des débats qui défendent cette théorie. Le papier [135] illustre que le trafic Internet dans le réseau de cœur devient de plus en plus poissonnien au fur et à mesure de l'augmentation de la capacité des liens observés et de l'agrégation des trafics. D'autres observations dans le cadre du projet METROPOLIS rejoignent cette théorie. Il est apparu que le trafic dans les réseaux de cœur suit un modèle poissonnien.

Au cours de notre travail de thèse, nous avons pu faire l'analyse d'une trace collectée sur un lien dans un réseau du cœur de l'opérateur France Télécom (cf. annexe E.1 pour le détail des propriétés de la trace analysée). La fonction d'auto-corrélation des arrivées de paquets (cf. figure 1.8) montre que 95.57 % des points sont dans l'intervalle gaussien $(-0.002, +0.002)$. Ceci signifie que la série a très peu de dépendance à long terme. Pour vérifier si la loi d'arrivée

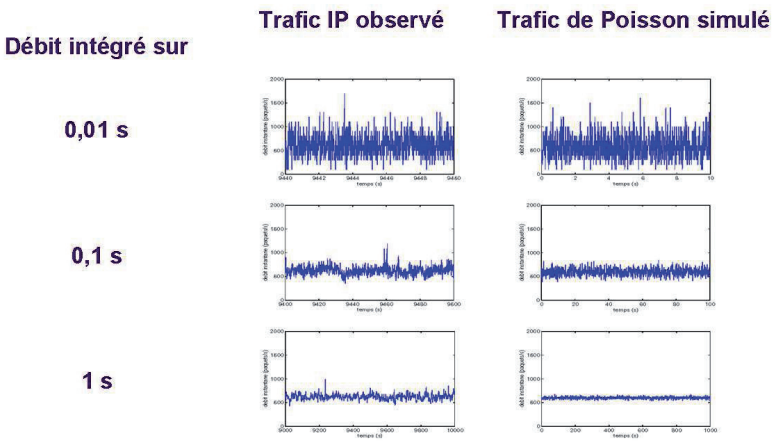


FIGURE 1.7 – Comparaison entre les oscillations observables dans un trafic Internet et un trafic poissonnien

des paquets suit une loi de Poisson, nous utilisons l'outil Q-Q Plot pour comparer les deux lois (cf. figure 1.9). Le résultat confirme aussi la constatation de [135].

Mais il faut remarquer que ces observations sont toutes effectuées sur les trafics de réseaux de cœur. Face à la croissance constante du trafic Internet et aux besoins de plus en plus forts des utilisateurs en termes de garantie de service et de performance, les opérateurs ont pour la plupart choisi de sur-dimensionner leur réseau d'un facteur allant généralement de 2,5 à 3 (et parfois plus). Ce choix a la propriété de repousser les phénomènes de congestion en bordure du réseau de l'opérateur. C'est donc dans les réseaux d'accès et de bordure que le trafic a les caractéristiques les plus complexes et les plus néfastes pour les performances globales du réseau. C'est pour cela que dans notre travail, nous nous focalisons sur des traces recueillies sur les réseaux de bordure et d'accès de l'opérateur RENATER.

Tentatives majeures de modélisation du trafic Internet

La représentation du trafic Internet a fait l'objet de nombreuses mesures et analyses (voir Allman et Paxson [7] et Paxson et Floyd [100] par exemple). Il s'agit de modéliser le processus d'arrivée des paquets aux différents points du réseau : routeurs sur la frontière, nœuds internes ou encore les routeurs de sortie ("Egress-routers").

Plusieurs types de représentation sont envisageables.

1. *Modèles à base d'enveloppe de trafic*

   Cette méthode consiste à déterminer a priori une enveloppe de trafic qui conduit à définir des cas pires, "worst case". Des valeurs maximales pour le débit crête et la longueur de rafale par exemple sont fixées, il s'agit de déterminer les algorithmes qui assurent un

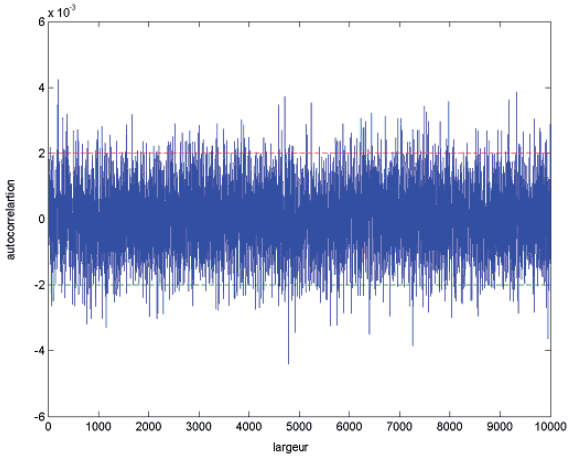


FIGURE 1.8 – Fonction d'auto-correlation des arrivées de paquet

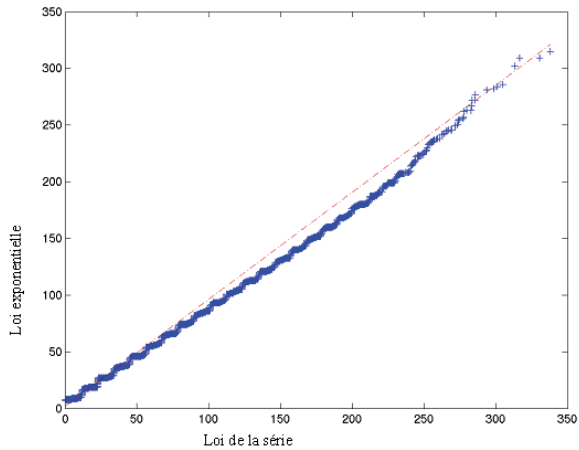


FIGURE 1.9 – Q-Q Plot de la loi d'arrivée et de la loi exponentielle

niveau de qualité de service donné avec des trafics complètement arbitraires respectant les contraintes de débit. Ce type de problématique conduit généralement à des algorithmes

qui sous-utilisent le réseau, voir par exemple l'article d'El Walid et al. [48] pour le contrôle d'admission.

2. *Modèles statistiques*

   Les trafics sont définis par leurs statistiques. Ces modèles présentent l'avantage de représenter correctement un grand nombre de types de trafic. De plus, des résultats qualitatifs précis peuvent être obtenus (ou du moins, envisagés) à partir des modèles mathématiques associés.

   (a) *Modélisation de niveau paquet (microscopique) :* trafics fractals. Si cette description est très populaire depuis les travaux de Leland et al. [75], elle n'a pas donné lieu, pour l'instant, à de réels développements algorithmiques spécifiques. Il est très difficile d'obtenir des résultats qualitatifs (sur les taux de débordements des mémoires tampons par exemple) avec ce type de modèle. Voir dans ce domaine les travaux de Norros sur le mouvement brownien fractionnaire [86] [87]. Actuellement, les travaux dans ce domaine se concentrent principalement sur la définition et l'estimation des paramètres caractérisant ces trafics.

   (b) *Modélisation de niveau flux (macroscopique) :* trafics dont les rafales ont lieu de façon aléatoire et la taille d'une rafale a une queue de distribution à décroissance lente (i.e. beaucoup de longues rafales et très grande variabilité de la taille de celles-ci). D'autre part, la plupart des travaux de caractérisation de trafic ont mis en évidence la présence de la LRD dans le trafic, que ce soit au niveau des flux ou au niveau des paquets. Ils ont également révélé la présence des queues lourdes dans les distributions des arrivées des flux et des paquets. D'un autre côté, plusieurs travaux de recherches considèrent la présence de queues lourdes dans la distribution des tailles de fichiers, des tailles de flux des arrivées des paquets et des flux comme la cause principale de la dépendance longue-mémoire ([137] [129] [96] [46].

   (c) *Modélisation des processus de perte :* si les modèles basés sur des chaînes de Markov (comme le modèle de Gilbert par exemple) ne peuvent pas représenter globalement le trafic au niveau des paquets sur Internet [101], ils peuvent cependant décrire convenablement certaines arrivées de rafales de paquets ou encore les processus de débordement des mémoires tampons des routeurs [102].

3. *Modèles basés sur le comportement du protocole TCP*

   La plupart des études du protocole TCP concernent le cas d'une très longue connexion en supposant que le taux de perte t_p est très faible. Il s'agit de déterminer quel est le débit d'une telle connexion. Le résultat classique dans ce domaine est que la taille de la fenêtre de congestion est de l'ordre de $\frac{1}{\sqrt{t_p}}$ quand t_p devient petit. Si (W_n) est la suite des tailles de fenêtres de congestion successives, divers modèles ont été utilisés pour représenter cette suite. L'intérêt d'une telle approche est de pouvoir faire de la prédiction sur le débit des flux dans l'Internet ou encore du dimensionnement réaliste d'un réseau en associant à ces modèles des informations sur les caractéristiques des flux (en terme de nombre et de durées par exemple).

   (a) *Les mesures et les simulations* présentées par Madhavi et Floyd [76] ainsi que Floyd [58] ont mis en évidence des résultats similaires au début des années 1990.

   (b) *Les modèles auto-régressifs a priori*

On suppose qu'il existe des suites i.i.d. (An), (Bn) telles que $W_{n+1} = A_n W_n + B_n$, cette dépendance linéaire permet de déterminer la plupart des caractéristiques de la connexion à l'équilibre. C'est l'hypothèse faite dans les travaux d'Altman et al. [6] [8] et Baccelli et Hong [17] [18].

(c) *Les approximations par chaîne de Markov finie*

Cela concerne entre autres les travaux de Padhye et al. [94], Vojnovi et al. [132] et Vojnovi et Le Boudec [133]. La formule du débit de Padhye et al. est souvent reprise dans la littérature.

(d) *Les modèles fluides*

La variable W est supposée à valeurs continues et l'évolution de cette variable est gouvernée par une équation différentielle déterministe perturbée par un processus de Poisson. Le résultat principal est celui Ott et al. [91], voir aussi Altman et al. [6].

(e) *Les approximations de champs moyens*

Il s'agit ici de considérer qu'un routeur est traversé par un très grand nombre de connexions et de supposer que deux connexions prises séparément sont quasiment indépendantes. Cette approximation permet d'écrire une équation différentielle déterministe de l'évolution d'une connexion donnée. C'est l'optique d'Adjih et al. [5].

(f) *Les modèles de perte de paquets*

Pour ces modèles, seul le processus de perte est la composante aléatoire d'une connexion TCP. Dumas et al. [47] considère le comportement de la connexion quand les pertes de paquets sont faibles et indépendantes. Guillemin et al. [64] [65] étudie la connexion TCP avec le processus de perte mis en évidence par les mesures de Paxson (pertes corrélées). Voir aussi Bolot [27] et Yajnik et al. [140]. De façon remarquable, les constantes dans les mesures de Floyd, du modèle de Ott et de l'approximation de champs moyens d'Adjih et al. sont obtenues par des théorèmes limites rigoureux (voir aussi pour détails Altman et al. [6]). D'autres aspects d'une connexion TCP ont été envisagés par Brown [29] (quand les temps d'aller et retour des différentes connexions ne sont pas identiques) et par Kelly [72].

1.3 Analyse des phénomènes de LRD dans le trafic

1.3.1 Tendances d'évolution du trafic

Les sections précédentes ont clairement mis en évidence le foisonnement dans la littérature d'études des propriétés de LRD du trafic. Dans cette section, nous analyserons en détail ce phénomène en nous basant sur les données réelles dont nous disposons dans le cadre du projet METROPOLIS (cf. annexe E pour détails).

Commençons cette partie par décrire l'évolution de la distribution du trafic par application mesurée dans l'Internet ces dernières années. La figure 1.10 illustre cette distribution mesurée en mai 2000 sur le réseau SPRINT (cf. annexe E.6). La grande proportion représentée par le trafic HTTP (plus de 75 % du trafic Internet) est remarquable. On note aussi que les principales applications standards sont représentées : web, web sécurisé, courriel, ftp ou news. Cependant, de nouvelles applications émergentes (à cette époque) sont présentes :

flux de streaming multimédia (comme MediaPlayer ou RealAudio) ou jeux distribués en réseau (comme Quake). Néanmoins, la caractéristique la plus importante de ce trafic reste son élasticité et ses contraintes temporelles de QdS qui ne sont pas importantes (pour la grande majorité de ses applications).

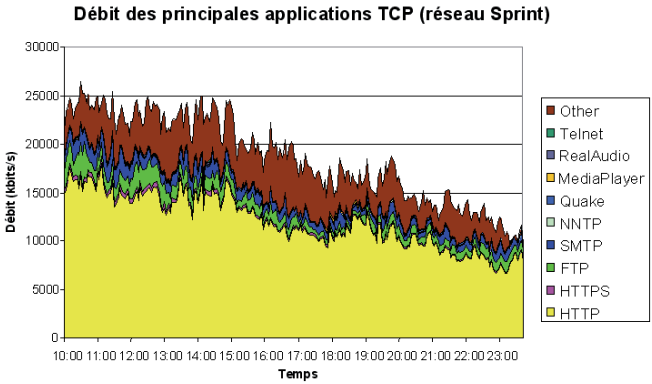


FIGURE 1.10 – Répartition du trafic sur le réseau SPRINT (mai 2000) — les applications sont classées dans le même ordre sur la légende et dans le graphique

Trois mois plus tard, la distribution était à peu près la même à l'exception d'une nouvelle application, la deuxième en partant du haut sur le graphique de la figure 1.11, qui est devenue, en quelques semaines, l'une des applications majeures de l'Internet. Il s'agissait de Napster, affublée du terme «application tueuse» car, en l'espace de trois mois, elle représenta entre 20 et 30 % du trafic. Ce type d'application, Peer-to-Peer (P2P), connut, au fil du temps, un succès de plus en plus important auprès des utilisateurs, représentant ainsi une part de plus en plus importante du trafic au sein de l'Internet. Bien que Napster eut quelques déboires avec la justice américaine, elle a ouvert la voie à tout un ensemble d'applications P2P comme Gnutella, E-donkey, Morpheus et d'autres.

Avènement de nouveaux usages dans l'Internet

En effet, trois ans plus tard, le trafic P2P n'a cessé d'augmenter et à l'heure actuelle, sur certains liens du réseau Renater, il peut représenter la même proportion que le trafic HTTP (cf. figure 1.12). Bien sûr, Napster a été remplacé par Kazaa ou E-donkey. Une telle augmentation du trafic P2P a irrémédiablement eu un impact sur les caractéristiques du trafic global. En particulier, à cause de la nature des fichiers échangés (la plupart du temps de la musique ou des films) qui sont comparativement beaucoup plus longs que les flux du trafic web, le trafic majoritaire quelques années auparavant.

En fait, cette augmentation du trafic P2P couplée à la présence du trafic classique induit les caractéristiques suivantes :

1. Le trafic Internet est toujours composé de milliers de petits flux appelés souris (imputables principalement au trafic web ainsi qu'au trafic de contrôle P2P),

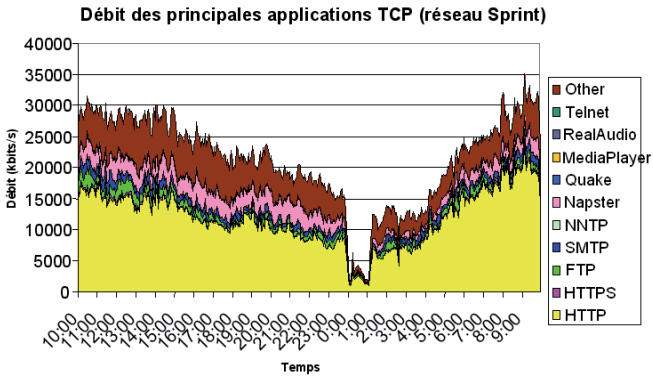


FIGURE 1.11 – Répartition du trafic sur le réseau SPRINT (août 2000) — les applications sont classées dans le même ordre sur la légende et dans le graphique

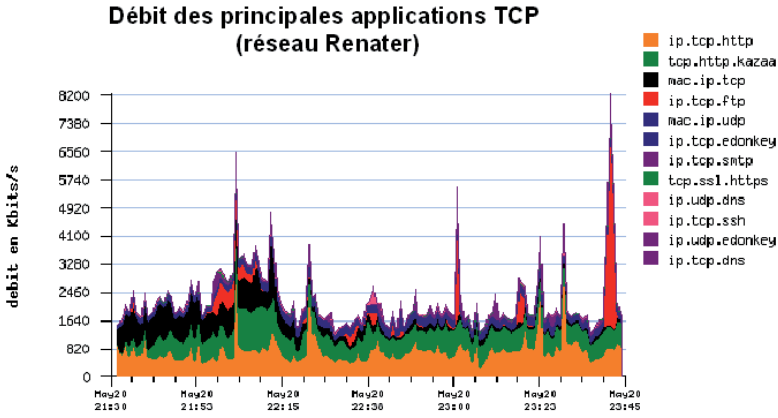


FIGURE 1.12 – Répartition du trafic sur le réseau RENATER (mai 2003) — les applications sont classées dans l'ordre inverse sur la légende et dans le graphique

2. Un nombre de flux éléphants qui ne cesse d'augmenter.

A tel point que la distribution de la taille des flux dans l'Internet change de façon importante. Ce phénomène a été analysé depuis le début des années 2000 et les résultats sont présentés dans la figure 1.13. La distribution exponentielle (celle possédant la queue la moins lourde),

sert de référence car elle est proche du modèle de Poisson [7]. Nous pouvons voir sur cette figure que la proportion de très longs flux a augmenté de façon très importante depuis l'an 2000. Si en 2000, cette distribution n'était pas très éloignée d'une exponentielle, ce n'est plus du tout le cas à l'heure actuelle. Au contraire, elle dispose d'une queue très lourde et est très éloignée de la distribution exponentielle.

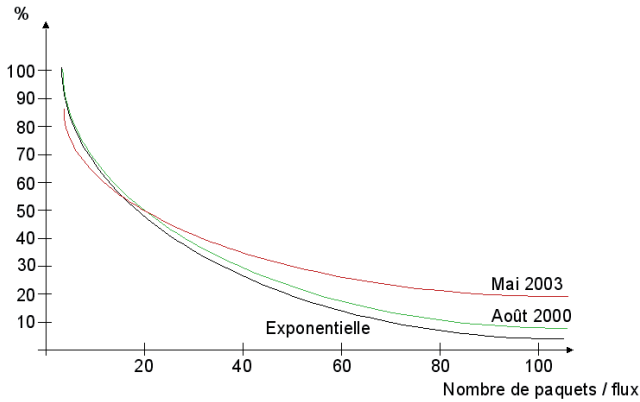


FIGURE 1.13 – Evolution de la distribution des tailles de flux dans l'Internet entre 2000 et 2003

Evolution de la variabilité du trafic

Ainsi, à l'heure actuelle, il y a de plus en plus de longs flux, ce qui a un impact sur le profil global du trafic. Comme nous l'avons vu dans la figure 1.7, il est facile de remarquer que le trafic Internet ne se lisse pas aussi vite que le trafic poissonnien. L'analyse a montré que ce résultat est fortement dû aux éléphants présents dans le trafic Internet. En effet, la transmission d'éléphants crée dans le trafic l'arrivée d'une grande vague de données qui a la particularité de durer un temps relativement long (plus d'une seconde [8]). C'est pour cela que l'on observe cette différence entre les deux types de trafic : la transmission des éléphants induit des variations persistantes dans le trafic actuel.

Rôle de la dépendance à long terme dans le trafic

Pour donner une vue concrète des problèmes de la LRD sur le trafic, analysons son impact sur un paramètre primordial dans le réseau : les pertes. La figure 1.14 (a) décrit un "leacky bucket" qui présente une analogie avec un routeur Internet comprenant un tampon, un lien entrant et un lien sortant. Ainsi, lorsque des vagues de données arrivent dans le trafic (cf. figure 1.14 (b)), si l'objectif est de fournir un service n'introduisant pas de délais ou de pertes

7. Ce modèle est utilisé comme référence dans la majorité des cas lorsqu'il s'agit de réaliser des simulations ou des évaluations de performance en réseau.

8. Les flux web sont traditionnellement transmis en moins d'une seconde dans l'Internet actuel.

supplémentaires, il est impératif de sur-dimensionner le lien. La seconde caractéristique apparaît lorsqu'une vague arrive faisant augmenter le niveau dans le tampon (cf. figure 1.14 (c) par rapport à la figure 1.14 (b)). Mais lorsque la persistance des oscillations augmente (cf. figure 1.14 (d)), et c'est le cas avec le trafic Internet actuel, l'arrivée d'une vague de très grande amplitude peut entraîner un débordement du tampon et induire des pertes (cf. figure 1.14 (e)).

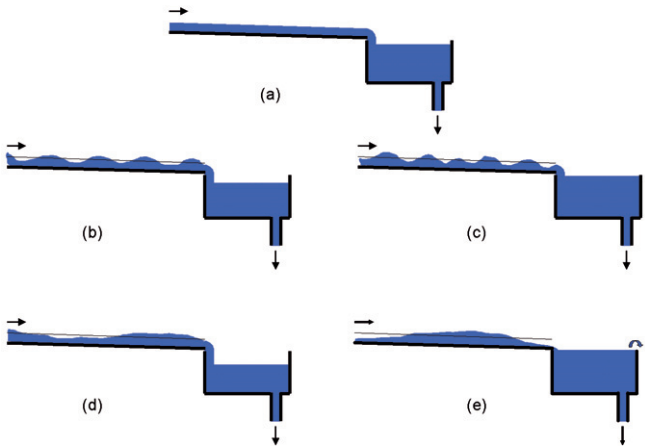


FIGURE 1.14 – Illustration de l'impact de la LRD du trafic sur le processus de perte

Etant donné que les connexions TCP utilisées pour transmettre les flux éléphants plus volumineux durent plus longtemps, la dépendance qui existe entre les paquets d'une même connexion se propage ainsi sur des échelles de temps plus longues. C'est ce phénomène que l'on nomme LRD. On lui attribue plusieurs causes dont la principale est imputable aux mécanismes de contrôle de congestion de TCP (le protocole dominant de l'Internet [96]). Parmi tous les mécanismes de TCP, il est évident que celui basé sur un contrôle en boucle fermée introduit de la dépendance à court terme, étant donné que les acquittements dépendent de l'arrivée d'un paquet, et que l'émission de tous les paquets suivants de la connexion sont conditionnés par cet acquittement. De la même façon, les deux mécanismes de TCP ("slow-start" et "congestion avoidance") introduisent de la dépendance à plus long terme entre les paquets de différentes fenêtres de congestion. Ainsi, en généralisant ces observations, il est évident que tous les paquets TCP d'une connexion sont dépendants les uns des autres. En plus, l'augmentation des capacités des liens de l'Internet, en permettant la transmission de flux de plus en plus longs, augmente le phénomène de LRD. C'est pourquoi on observe dans la figure 1.7, la persistance d'un comportement oscillatoire dans le trafic Internet qui reste très marqué même avec une granularité d'observation importante (1 s).

D'autre part, étant donné que le phénomène de dépendance de TCP se propage dans le trafic par l'intermédiaire des flux (i.e. les connexions TCP) [129], l'augmentation de la taille

des flux induit une augmentation de la portée de la dépendance qui peut atteindre des échelles très importantes. Ainsi, une oscillation au temps t induit alors d'autres oscillations à d'autres instants qui peuvent être potentiellement très éloignés de t. D'autre part, il est évident que les éléphants, en raison de leur durée de vie très importante dans le réseau et des grandes capacités de ce dernier (la plupart du temps les liens étant sur-dimensionnés), ont le temps d'atteindre de grandes valeurs pour leur fenêtre de contrôle de congestion. Ainsi, une perte induit pour le flux qui la subit une importante diminution, suivie par une importante augmentation de son débit. L'augmentation de la taille des flux favorise donc les oscillations avec une forte amplitude et un phénomène de dépendance à long terme.

Bien sûr, les oscillations sont très néfastes pour une utilisation optimale des ressources globales du réseau étant donné que la capacité libérée par un flux subissant une perte ne peut pas être immédiatement utilisée par un autre (en raison de la phase de slow-start notamment). Ceci se traduit par un gaspillage de ressources et induit une diminution de la QdS globale du réseau. Plus le trafic oscille, moins les performances sont importantes [97].

Ainsi, la nature auto-similaire du trafic Internet est très nuisible à la QdS. Dans la théorie des files d'attente, si la loi d'arrivée est auto-similaire, alors la longueur moyenne de la file d'attente est supérieure à celle nécessitée par une loi de Poisson. Plus le degré d'auto-similarité est élevé, plus longue sera la file d'attente, et plus rapidement la longueur de la file va croître (ce paramètre étant fonction de la croissance du taux d'utilisation du réseau). Cela veut dire que le routeur a une plus forte probabilité d'être engorgé si le trafic réseau est auto-similaire. En fait, plus le degré d'auto-similarité du trafic est élevé, plus grand doit être le tampon nécessaire pour éviter des pertes éventuelles. La figure 1.15 illustre la fonction entre la taille de la file d'attente, le taux d'utilisation du réseau et le facteur de Hurst. D'autre part, si l'on considère un point de vue utilisateur, il est clair que plus le trafic devient auto-similaire plus la QdS perçue de bout en bout par ce dernier sera instable étant donné la variabilité induite par la propriété d'auto-similarité dans le trafic actuel. Nous allons détailler ce dernier point dans la section qui suit.

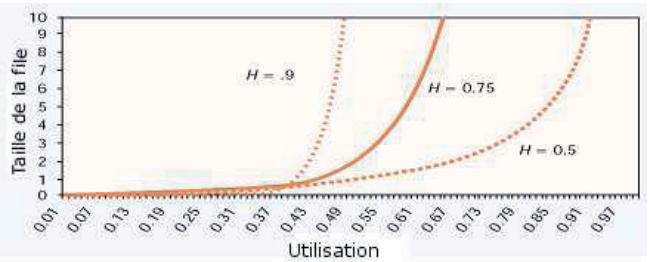


FIGURE 1.15 – Croissance de la file d'attente

Illustration de l'impact de la LRD sur la QdS du réseau

Pour éviter la dégradation de performance de réseau, l'opérateur doit augmenter la taille du tampon de routeur. Mais l'augmentation du tampon augmente aussi le délai de retransmission du routeur et aussi les RTT de transmission. Par conséquent, les flux durent plus longtemps et

introduisent plus de LRD. Dans [97], les auteurs ont mesuré les performances du réseau sous différentes conditions d'auto-similarité. Ils ont montré que lorsque le degré d'auto-similarité du trafic est élevé, le délai de retransmission augmente proportionnellement à la taille du tampon. Ceci pose un problème difficile relatif au dimensionnement du réseau pour maintenir une certaine QdS dans un environnement auto-similaire.

Face à la croissance constante du trafic Internet et aux besoins de plus en plus forts des utilisateurs en termes de garantie de service et de performance, les opérateurs doivent surdimensionner leur réseau selon les caractéristiques du trafic pour éviter la dégradation des performances. L'analyse de l'évolution du facteur de Hurst du trafic devrait aider à quantifier le niveau de surdimensionnement nécessaire pour un fonctionnement optimal du réseau.

Traditionnellement dans un pareil cas, les opérateurs réseaux décident d'augmenter la taille des files d'attente. Mais ce n'est pas la bonne solution car ceci induit dans un premier temps, une augmentation de la valeur des délais et des RTT et dans un deuxième, une diminution des performances de TCP. En effet, la transmission des flux dure plus longtemps, ce qui allonge la portée de la LRD et introduit donc plus de perturbations dans le réseau. Comme la taille des flux ne peut être réduite [9], l'unique solution est de modifier les comportements protocolaires pour diminuer la portée des vagues générées par les applications. C'est une des orientations de recherche qui va être abordée dans la suite de ce manuscrit.

1.3.2 Les causes possibles de l'auto-similarité du trafic Internet

De nombreuses recherches ont été menées depuis quelques années pour déterminer les causes possibles de l'auto-similarité et de la dépendance à long terme du trafic. Voici les principales conclusions auxquelles elles ont abouti.

La distribution des tailles flux

Comme déjà mentionné précédemment, dans [77] l'auteur a montré qu'un trafic auto-similaire peut être généré par superposition d'un grand nombres de sources ON/OFF dont les périodes ON et OFF sont à queue lourde.

Dans la modélisation ON/OFF (cf. figure 1.16), les sources ON/OFF correspondent aux ordinateurs individuels. La période ON correspond à la période de transmission et la période OFF correspond à l'absence de transmission de la machine ou de l'utilisateur.

Dans [96], les auteurs montrent en simulation que dans un tel modèle, si la distribution des tailles de fichiers (ou durée de transmission) est à queue lourde, alors le trafic superposé est auto-similaire. De plus, même si la distribution des tailles des fichiers n'est pas à queue lourde, mais si la distribution des durées de silence (OFF time) est à queue lourde, alors le trafic global est aussi auto-similaire. Par contre, il est à noter que la contribution pour les durées OFF est moins importante que celle des durées ON.

[38] a fait une analyse sur le trafic d'un réseau de bordure, il montre que la distribution de temps de transmission (ou tailles des fichiers) est à queue lourde, donc que le trafic Internet est auto similaire. Cette analyse a été menée en 1995, à l'époque, le trafic HTTP était dominant dans le trafic global, mais ce n'est plus le cas aujourd'hui. En effet, le trafic P2P occupe actuellement une très grande proportion dans les réseaux commerciaux, et il est composé de flux de très longue durée par rapport aux flux HTTP. Donc, si la distribution des temps de

9. C'est une caractéristique résultant du comportement des utilisateurs de l'Internet.

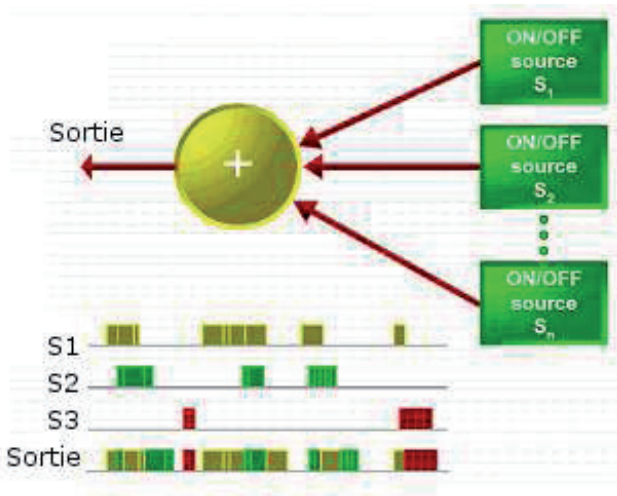


FIGURE 1.16 – Superposition des source ON/OFF

transmission était déjà à queue lourde, alors la distribution d'aujourd'hui devrait mettre en évidence une amplification de ce phénomène. Nous étudierons en détail cette propriété du trafic actuel dans le chapitre suivant.

Les mécanismes de contrôle de congestion du protocole TCP

De nombreuses études ont montré que les mécanismes de TCP engendrent des modèles de trafics complexes, auto-similaires ou même des comportements chaotiques. Dans [67], Liang et al. utilisent une modélisation Markovienne et des simulations pour montrer que, lorsqu'un flux TCP traverse un réseau congestionné, il montre des caractéristiques d'auto-similarité pour, par exemple, ses séries de débits au cours du temps. En effet, les mécanismes de congestion de TCP ("Slow-Start" et "Congestion Avoidance") sont tels qu'ils créent une forte dépendance entre les paquets et les pertes de paquets.

[61] a confirmé ce résultat et indiqué que lorsque la probabilité de perte est élevée, le mécanisme TO (Time Out [10]) contribue à la génération de l'auto-similarité dans le réseau, quand la probabilité de perte est réduite, CA contribue plus à la génération de l'auto-similarité.

De plus, [130] montre que quand un flux TCP traverse un routeur engendrant un goulot d'étranglement (voir la figure 1.17), si le trafic du fond est auto-similaire, alors le flux devient auto-similaire. Ceci veut dire que le phénomène de LRD peut se propager partout dans l'Internet.

10. Le mécanisme TO de TCP considère comme perdu tout paquet ou ensemble de paquets qui n'ont pas été acquittés par le récepteur dans un laps de temps inférieur à la valeur du TO. Ainsi, lors de l'expiration de ce timeur, l'émetteur renvoie automatiquement à destination du récepteur la séquence de paquets perdus.

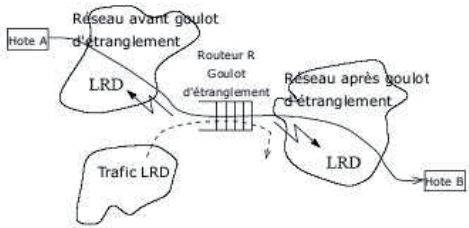


FIGURE 1.17 – Propagation de la LRD entre flux par la traversée de goulots d'étranglements

Les autres causes possibles

En plus des deux causes ci-dessus qui sont les plus discutées dans la littérature, il existe d'autres causes possibles considérées comme plus "marginales" tels que les comportements humains (par exemple : le "thinking time" (OFF time) qui est à queue lourde), le comportement des applications...

En particulier, les politiques régissant les routeurs sont aussi considérées comme responsables de la présence de LRD dans l'Internet. Par exemple, il a été montré [111] que certaines politiques de destruction des paquets, RED en particulier, pouvaient permettre dans certaines conditions de réduire la corrélation entre les pertes et donc la LRD par rapport à la politique classique de type DropTail. Il est également établi que la traversée successive de routeurs ou l'agrégation de trafics au niveau de ces derniers jouent un rôle dans ce phénomène [126].

1.3.3 Mise en évidence de la LRD dans le trafic global

Méthode de mesure de la LRD : la décomposition en ondelettes

Nous avons montré jusqu'à présent des résultats qualitatifs concernant la nature du trafic Internet. Il est donc maintenant nécessaire de quantifier sa nature oscillatoire et ses caractéristiques de LRD. Pour cela, nous avons utilisé une analyse en ondelettes [1] sur les traces de trafic dont nous disposons. La fonction en ondelettes a été sélectionnée car elle représente au mieux une oscillation (la preuve est visuelle sur la figure 1.18). Le principe de cette analyse consiste à extraire du trafic toutes les ondelettes possibles. Pour cela, nous utilisons plusieurs fonctions en ondelettes chacune de fréquences différentes afin d'obtenir les différentes échelles oscillatoires.

Les fonctions avec les périodes les plus larges représentent les plus longues vagues, c'est à dire celles générées par les flux les plus longs. Etant donné que l'auto-similarité du trafic Internet peut intuitivement être perçue comme la répétition d'un même processus mais pour des échelles temporelles différentes, la méthode d'analyse en ondelettes du trafic est parfaitement adaptée pour la mise en évidence de ce type de comportement dans le trafic car elle représente sous la forme d'un seul graphique (cf. exemple de la figure 1.19) toute la variabilité du processus analysé pour l'ensemble des échelles temporelles que l'on peut extraire. D'autre part, le trafic purement auto-similaire est très rare à l'heure actuelle dans l'Internet (comme nous le verrons par la suite, on met en évidence en majorité deux lois d'échelle distinctes quelles

que soientt les traces que l'on peut analyser), ainsi les graphiques que nous allons obtenir laisserons apparaître des invariants pour une ou plusieurs plages de fréquences considérées. Dans le cas où cette plage s'étendra uniquement sur les échelles les plus grandes du processus (i.e. les fréquences les plus petites) on ne parlera plus de processus strictement auto-similaire mais de processus mettant en évidence uniquement de la LRD.

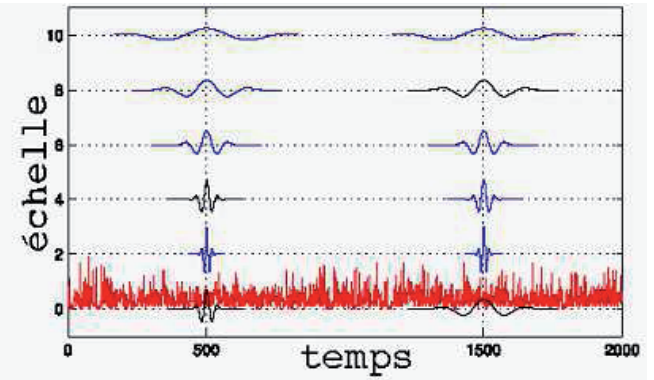


FIGURE 1.18 – Analyse en ondelettes de la LRD du trafic Internet (granularité 1 ms)

Une première mesure globale de la LRD du trafic

Ces calculs de LRD ont été réalisés sur les traces capturées sur le réseau du LAAS-CNRS (cf. annexe E.3 pour les détails concernant ces traces). La courbe de la figure 1.19 a été obtenue en utilisant l'outil LDEstimate [1] qui estime la LRD qui se manifeste dans le trafic à toutes les échelles temporelles (cf. annexe D pour les détails d'utilisation de cet outil). Son principe repose sur l'utilisation de la méthode des ondelettes pour permettre d'évaluer le paramètre de Hurst en considérant le second ordre statistique d'une série de données. Cette méthode qui est l'une des plus élaborées a été implémentée sous la forme d'un outil Matlab par Messieurs Veitch [127] et Abry [1]. Le résultat produit par cet outil est une représentation graphique des lois qui régissent le niveau de dépendance du trafic à différentes échelles temporelles. Il est obtenu grâce à une analyse en ondelettes du trafic Internet et représente le niveau de variabilité des oscillations en fonction de la granularité d'observation. Le facteur de Hurst (caractéristique des processus auto-similaires qui se retrouvent dans le trafic Internet, cf. [98]) est obtenu directement sur la courbe de LRD en mesurant sa pente. Si le facteur de Hurst est compris entre $0,5$ et 1, on met en évidence la présence de LRD dans le processus analysé : plus la valeur est proche de 1 plus la LRD est forte. A l'inverse, si le facteur est inférieur à 0,5, il n'y a pas de LRD.

La figure 1.19 montre un comportement différent pour deux échelles temporelles (appelé phénomène de "bi-scaling", nous reviendrons sur cette notion par la suite). Il est à noter que ce phénomène a été mis en évidence sur un grand nombre de traces de trafic de part le monde et cela quelle que soit la méthode de caractérisation de la LRD considérée. La frontière entre

ces deux niveaux de LRD se trouve autour de l'octave 8 et met en évidence des niveaux de LRD différents pour les échelles de temps courtes et longues, ceci se traduisant par différentes lois de puissance. Pour les petites échelles (octaves < 8) [11], c'est à dire les paquets proches les uns des autres, la dépendance est peu marquée. Par contre, pour les échelles plus grandes (octaves > 8), c'est à dire des paquets appartenant à des fenêtres de congestion consécutives, la dépendance est beaucoup plus importante.

Evidemment, ce phénomène existe pour l'ensemble des fenêtres de congestion d'un même flux. Ainsi, la présence dans le trafic de flux très longs introduit un phénomène de dépendance à très long terme qui est visible sur la figure 1.19 pour les octaves très grandes (supérieurs à 12). Ce niveau de LRD dans le trafic devient un problème majeur étant donné que chaque oscillation se produisant à un temps t peut se reproduire à n'importe quel temps t' qui est dépendant de t (en raison de la LRD qui existe entre les paquets échangés par le biais des protocoles traditionnels : ici TCP sur les longs flux). Il est intéressant de noter que nos expériences ont montré que le coude présent sur la courbe de LRD correspondait à la taille moyenne des flux (nous reviendrons en détails sur ce résultat dans la section et le chapitre suivant), la partie droite de la courbe correspondant donc à l'impact des flux éléphants.

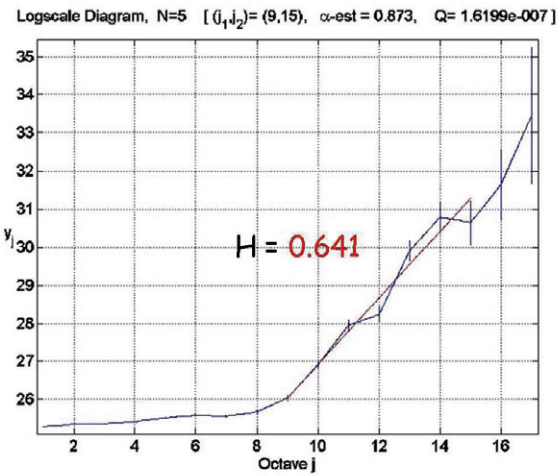


FIGURE 1.19 – Evaluation de la LRD dans le trafic Internet

Mesure de la LRD du trafic aux niveaux paquet et flux

Il est nécessaire d'analyser plus finement le trafic pour mettre en évidence la propriété de LRD, pour cela nous considérons différemment les niveaux paquet et flux et nous nous intéressons à la corrélation et la dépendance que l'on peut y mesurer.

11. Avec la granularité considérée, cette échelle correspond aux durées inférieures à 2^8 ms.

1. *Analyse au niveau flux*

   Pour analyser la distribution d'arrivées des flux (générée en suivant une granularité de 10 μs), nous avons eu recours à l'outil Q-Q Plot. Dans notre cas, nous avons comparé notre série des inter-arrivées avec une loi exponentielle. La figure 1.20 illustre la présence d'une queue lourde dans la distribution des inter-arrivées des flux. En effet, la figure fait apparaître un éloignement visible de la fonction quantile par rapport à la ligne de référence. Ceci prouve que les deux séries de données (les inter-flux TCP et les inter-arrivées exponentielles) comportent des distributions différentes.

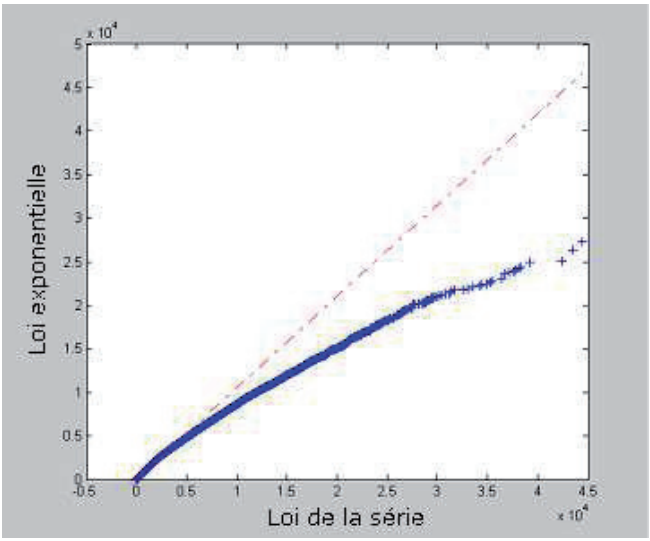


FIGURE 1.20 – Représentation Q-Q Plot de la loi d'arrivée des flux TCP

Pour mesurer quantitativement l'auto-similarité (ou LRD) dans la série, nous recourons à l'outil LDestimate. La mesure du facteur de Hurst par la méthode des ondelettes donne le résultat suivant : $H = 0.775$ (figure 1.21). Ce facteur supérieur à 0.5 indique la présence de la LRD dans le trafic des flux. De plus, cette valeur nous permettra ultérieurement de comparer quantitativement le degré de dépendance longue introduite par chaque type de flux.

Nous soulignons ici que la pente de la courbe est plus forte pour les octaves entres 6 et 13 que pour les échelles les plus courtes (octaves 2 à 6). Cela montre que la dépendance à long terme est beaucoup plus importante que la dépendance à courte terme.

2. *Analyse au niveau paquet*

   La fonction d'auto-corrélation illustrée par la figure 1.22 se caractérise par un comportement complexe (décroissance lente) où toutes les valeurs sont situées en dehors des bornes gaussiennes (0.0012). Cette corrélation entre les paquets prouve la présence de

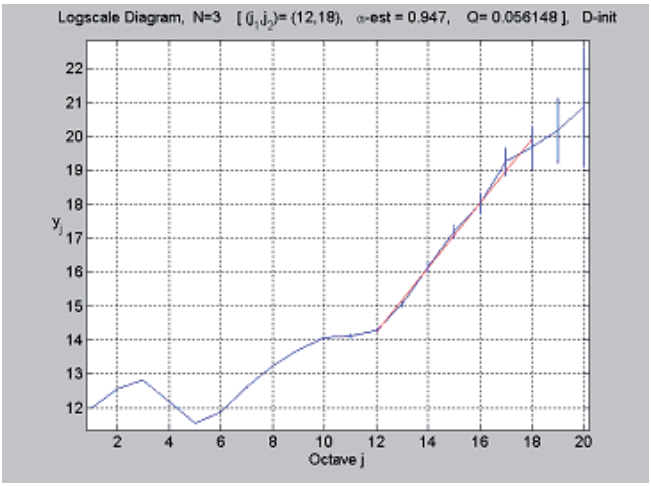


FIGURE 1.21 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série d'inter-paquets (granularité 100 μs)

dépendance entre les paquets. La mesure du facteur de Hurst par la méthode des ondelettes donne le résultat suivant : $H = 0.887$ (figure 1.23). Ce facteur supérieur à 0.5 indique la présence de la LRD dans les séries. Il est intéressant de noté que la LRD est plus présente dans le trafic lorsqu'on considère de façon globale le niveau paquet que le niveau flux. Nous verrons dans le chapitre suivant que ce résultat peut être affiné lorsqu'on analyse de façon différente les flux selon leur durée par exemple ou encore leur volume.

Mise en évidence de la caractéristique de "bi-scaling" du trafic Internet

Nous venons de mettre en évidence un plus grand degré de LRD dans le trafic au niveau paquet. Nous allons à présent analyser plus finement les caractéristiques de cette LRD. Pour cela, nous considérons à nouveau les figures 1.19, 1.21, nous voyons que l'on distingue deux régimes distincts : le premier concerne les petites échelles de l'octave 1 à l'octave 12. Le second régime concerne les octaves les plus élevés (octave 12 à 18). Nous observons sur ce diagramme un alignement des points de la courbe pour les octaves entre 12 et 18. Cet alignement indique la présence d'une forte LRD représentée par la droite et exhibe un comportement en loi de puissance. Pour les octaves entre 1 et 12, un tel alignement n'existe pas ou est très faible. Nous pouvons observer le même phénomène dans presque tous les diagrammes LDestimate présentés dans ce rapport. Ce phénomène indique deux régimes de dépendance : une pour la dépendance qui existe à court terme (par exemple pour les paquets proches temporellement) et l'autre pour les échelles temporelles plus élevées. Il est à noter que ce dernier phénomène de dépendance pour les grandes échelles de temps que l'on nomme LRD est la source de

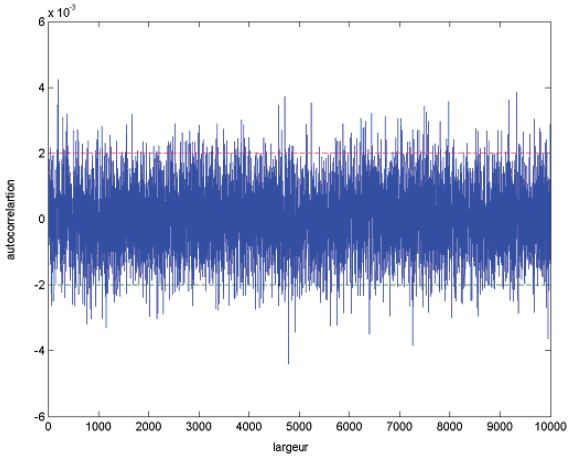


FIGURE 1.22 – Fonction d'auto-corrélation de la loi d'arrivées des paquets

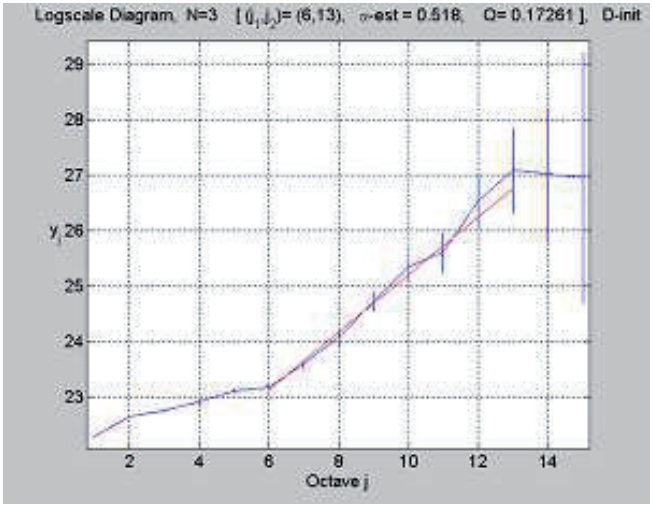


FIGURE 1.23 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série d'inter-flux des flux TCP (granularité 100 ms)

dégradation des performances de l'Internet actuel [138]. En effet, la dépendance à court terme serait elle aussi pénalisante pour le réseau mais elle est très faible dans l'ensemble des traces que nous avons analysées, c'est pourquoi nous nous focaliserons sur le long terme dans la suite de ce manuscrit et sur l'analyse du phénomène de LRD.

La forte dépendance longue peut s'expliquer par la structure de contrôle à boucle fermée des mécanismes de contrôle de congestion de TCP pour lesquels l'émission d'un paquet d'une fenêtre de contrôle de congestion dépend de la réception d'un acquittement associé à la réception d'un paquet de la fenêtre de congestion précédente. Naturellement, ce phénomène existe pour des fenêtres de congestion consécutives, mais aussi pour toutes les fenêtres de congestion d'un même flux.

Dans la figure 1.21, le point tournant est à l'octave 12. Si nous prenons en compte que la granularité des mesures réalisées est de 100 μs, alors en terme de temps, le point se situe à $2^{12}\mu s \simeq 409ms$. Nous avons développé un programme pour obtenir l'approximation de la moyenne des durées inter-fenêtre (ou inter RTT) de la trace. La valeur moyenne se situe vers 386 ms. Ce résultat est donc très cohérent avec notre interprétation. Le phénomène de LRD apparait donc pour les flux les plus longs, c'est à dire ceux dont la connexion s'étend sur plusieurs RTT.

1.3.4 Etude quantitative de la relation existant entre oscillations et LRD dans le trafic Internet

Evaluation de l'impact de TFRC sur la QdS

Les deux parties précédentes viennent de mettre en évidence les phénomènes oscillatoires et de LRD du trafic Internet (au niveau de réseaux d'accès à RENATER, plus généralement appelés réseaux de collecte dans la terminologie RENATER). Ces observations et analyses, associées à la littérature dans le domaine amènent à penser que la LRD est un bon moyen de caractériser la variabilité du trafic, en particulier dans sa persistance. Toutefois, ce problème n'a, à notre connaissance, jamais été vraiment abordé dans la littérature existante. Aussi, l'expérience qui va être décrite dans la suite a pour objectif, sur un exemple, de montrer l'existence de ce lien entre les deux aspects variabilité et LRD. Pour ce faire, l'expérience menée s'est proposée de comparer au travers de simulations NS le trafic réel avec le même trafic re-simulé (le principe de l'expérience est donné dans les paragraphes suivants et plus de détails sur la méthode de rejeu sont disponible dans l'annexe C), mais pour lequel le mécanisme de contrôle de congestion du protocole de transmission TCP a été remplacé par TFRC [Lar03a] [57]. Comme nous allons le voir plus loin, l'objectif de TFRC par rapport à TCP est de fournir des sources de trafic beaucoup plus lisses et régulières, c'est-à-dire des sources qui ne présentent pas ou peu d'oscillations. Ce mécanisme a été aussi défini pour permettre un meilleur transfert du trafic généré par les applications de streaming dans l'Internet qui nécessitent naturellement un maximum de régularité pour leur débits d'émission et de réception. L'objectif est donc de montrer que lorsque l'on emploie TFRC, et donc quand on génère un trafic régulier et lisse, la LRD qui apparaît dans le réseau est très réduite par rapport au cas où TCP est utilisé.

1. *Les principes du mécanisme TFRC*
   TFRC a pour objectif d'offrir aux applications des débits en émission lisses et réguliers avec des variations très douces ; dans tous les cas plus douces que celles de TCP. En utilisant un tel mécanisme de contrôle de congestion à la transmission des flux éléphants,

c'est-à-dire à la part la plus importante du trafic comme nous le verrons dans le chapitre suivant, nous espérons pouvoir réduire sensiblement les oscillations du trafic. Un éléphant est un flux qui contient plus de 100 paquets de données échangés au cours de la connexion. Cette notion de typologie des flux sera présentée en détails dans le chapitre qui suit, l'objectif de cette section n'étant pas d'analyser la contribution des différents types de flux sur la LRD du trafic mais de mettre en évidence le lien qui existe entre LRD et oscillations dans le trafic Internet. Le débit en émission de chaque source TFRC est calculé sur la base d'informations provenant du récepteur, notamment le taux de perte observé lors du dernier RTT. Ce débit est calculé une fois par RTT et dépend du taux de perte mesuré par le récepteur et du RTT mesuré par l'émetteur [59] [57] en utilisant l'équation 1.8 :

$$X = \frac{s}{R * \sqrt{2 * b * \frac{p}{3}} + (t_{RTO} * (3 * \sqrt{3 * b * \frac{p}{8}}) * p * (1 + 32 * p^2)))} \tag{1.8}$$

où :
- X est le debit d'émission en octets/seconde,
- s est la taille des paquets en octets,
- R est le RTT en secondes,
- p est le taux de perte (entre 0 et 1.0), c'est à dire le nombre d'événements de pertes divisé par le nombre de paquets transmis,
- t_{RTO} est le timeout de retransmission de TCP en secondes,
- b est le nombre de paquets acquittés par un seul acquittement TCP.

Dans TFRC, un événement de perte est considéré si au moins une perte apparaît au cours de la période d'un RTT. Cela signifie que plusieurs pertes intervenant dans le même RTT font partie du même événement de perte. Ainsi, le modèle de dépendance des pertes de l'Internet est cassé car la plupart des pertes dépendantes apparaissent dans le même RTT (liées à une congestion sporadique). La récupération des pertes va ainsi être facilitée et beaucoup plus efficace qu'avec TCP lequel n'est pas très efficace pour récupérer plusieurs pertes en séquence ou corrélées. Cette approche utilise en fait les résultats de [141] qui présente une analyse et un modèle pour le processus de perte des liens de l'Internet.

2. *Description de l'expérience*

   Cette expérience a pour objectif de fournir une étude comparative des caractéristiques globales du trafic suivant que les éléphants sont transmis en utilisant TCP ou TFRC. Cette expérience vise aussi à fournir des résultats dans un environnement réaliste. Pour cela, elle se base sur des traces de trafic capturées par les équipements de métrologie passive DAG [Owe04d] [35]. Ainsi, les flux identifiés dans les traces de trafic originales sont rejoués dans NS-2 avec les mêmes dates de démarrage relatives et en respectant les autres caractéristiques des flux (taille des paquets, taille des flux, etc. voir annexe C pour détails sur la méthode). D'autre part, l'environnement de simulation a été construit de façon à ce que la mise en forme des paquets soit faite de façon cohérente avec ce qui s'est passé dans la réalité. Ainsi, les files d'attente et capacités des liens de l'environnement de simulation sont dimensionnées de façon à respecter les taux et modèles de pertes observés sur la trace réelle.

   De même, les délais de chacun des liens sont fixés de façon à respecter les RTT mesurés

pour les flux. Enfin, les sources de trafic sont positionnées pour respecter entre les flux re-simulés les contentions qui sont apparues dans la réalité. Pour plus d'informations quant à la méthode de re-simulation, le lecteur pourra se reporter à [Lar04b] [Owe04a]. Dans cet environnement de simulation et pour le compte de notre expérimentation, les flux éléphants sont donc transmis dans le simulateur en utilisant TFRC alors que les autres flux utilisent TCP New Reno [12]. Dans la suite, l'étude comparative va donc porter sur la trace originale d'une part et sur la trace simulée d'autre part dans laquelle les flux éléphants sont transmis en utilisant TFRC.

Par rapport au thème de cette étude comparative qui vise à étudier les effets de TFRC sur le caractère oscillant du trafic, les paramètres qui vont être évalués sont les paramètres traditionnels de débit, mais aussi des paramètres statistiques du trafic comme la LRD, et quelques paramètres mesurant le niveau de variabilité du trafic. Pour cela, nous utilisons un coefficient de stabilité (SC) qui est défini par le quotient :

$$\text{Coefficient de Stabilité (SC)} = \frac{\text{trafic moyen échangé}}{\text{écart type du trafic échangé } (\sigma)} \tag{1.9}$$

3. *Impact de TFRC sur la QdS des flux*

   La figure 1.24 présente le trafic dans les deux cas d'étude soit le cas réel et le cas simulé (avec TFRC). Visuellement, il apparaît clairement qu'en utilisant TFRC pour transmettre les éléphants à la place de TCP, le trafic global est bien plus lisse et régulier, et que tous les grands pics de trafic que l'on peut voir sur le trafic réel ont disparu du trafic simulé avec TFRC.

   Les résultats quantitatifs sont présentés dans le tableau 1.1. Ils confirment que la variabilité du trafic dans le cas du trafic réel (utilisant TCP pour transmettre les éléphants) est bien plus importante par rapport au cas simulé dans lequel les éléphants sont générés avec le protocole TFRC (pour l'écart type σ, nous avons calculé que $\sigma(\text{trafic réel}) = 157.959 \text{ ko} \gg \sigma(\text{trafic simulé}) = 102.176 \text{ ko}$). De la même façon, le coefficient de stabilité est plus faible dans le cas réel (SC = 0.521) par rapport au cas simulé (SC = 0.761).

   En ce qui concerne le débit global, nous avons mesuré des débits assez proches dans les deux cas ($\text{Débit(trafic réel)} = 82.335 \text{ ko} \approx \text{Débit(trafic simulé)} = 77.707 \text{ ko}$). Ce résultat est excellent pour TFRC qui n'a pas les mêmes capacités que TCP pour consommer rapidement une grande quantité de ressources [Owe03a], et même si TFRC est donc moins agressif que TCP, il est capable d'atteindre quasiment le même niveau de performance que TCP. Ceci confirme l'importance de la stabilité du trafic pour obtenir des performances de haut niveau et optimisées pour les réseaux de communication [97].

   En ce qui concerne la LRD, la figure 1.25 montre que dans le cas simulé la propriété de bi-scaling est sensiblement réduite et la courbe, même pour les grandes octaves a une pente peu marquée. Cela signifie que toutes les formes de dépendance, et en particulier

12. TCP New Reno a été choisi car c'est encore aujourd'hui la version du protocole TCP la plus utilisée dans l'Internet. Pour augmenter encore le réalisme des simulations, il serait intéressant de rejouer les flux courts avec la version de TCP qui a été utilisée dans la trace originale. Mais déterminer cette information à partir d'une trace passive est impossible pour la plupart des flux courts : seuls ceux qui subissent un grand nombre de pertes peuvent donner suffisamment d'informations pour déterminer la version de TCP qui a été utilisée. De toute manière, l'erreur reste minime car sur des souris, les mécanismes modifiés d'une version à l'autre de TCP n'interviennent que très rarement. Les modifications successives de TCP ont surtout été faites pour augmenter la performance de TCP pour la transmission de flux longs.

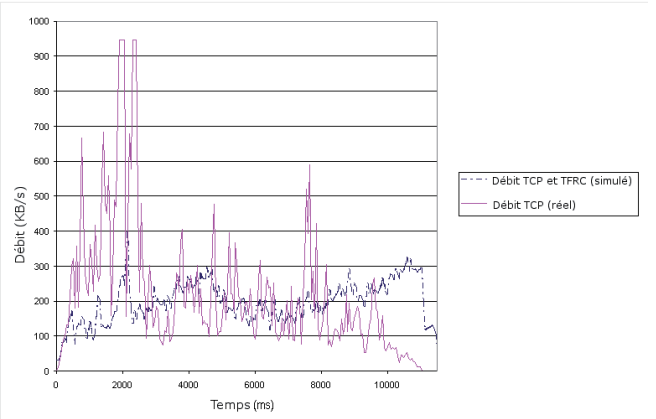


FIGURE 1.24 – Evolution du débit au cours du temps

Protocole	Débit moyen (kB)	σ du débit(kB)	SC
TCP New Reno (NR) : cas réel	82.335	157.959	0.521
TCP NR & TFRC : cas simulé	77.707	102.176	0.761

TABLE 1.1 – Caractérisation du débit pour les protocoles TCP et TFRC

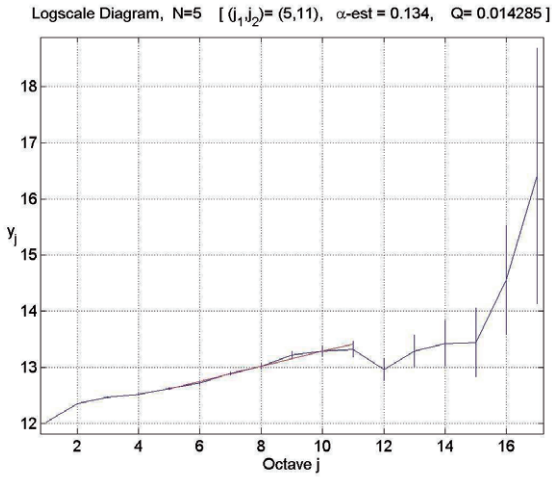


FIGURE 1.25 – Evaluation de la LRD pour le trafic simulé incluant des éléphants TFRC

celles à long terme ont été réduites de façon drastique. Les valeurs pour la LRD qui s'exprime à l'aide du facteur de Hurst sont : H(trafic réel) = 0.641 et H(trafic simulé) = 0.194 (ce qui dans ce dernier cas est non significatif, mais marque bien l'absence de LRD). Un tel résultat confirme deux aspects de notre proposition :

- TFRC permet de lisser le trafic lié à chaque flux individuellement, ainsi que le trafic global sur un lien ;
- La LRD est un paramètre significatif qui permet de qualifier et quantifier les lois d'échelles et de dépendance des oscillations.

La LRD : une métrique caractéristique de la QdS

Sans en donner une preuve formelle irréfutable, cette expérience a permis de mettre en évidence le lien étroit qui existe entre la caractéristique oscillante du trafic et la LRD. En effet, à partir du moment où on utilise pour transmettre les flux éléphants [13] un protocole qui ne crée pas d'oscillations (TFRC) et qui brise le modèle de dépendance lors de la récupération des pertes, la LRD disparaît quasiment du trafic.

Ce résultat d'analyse est important car il donne un outil pour caractériser qualitativement et quantitativement un des phénomènes caractéristiques du trafic Internet, qui est de plus un élément dégradant de la performance du réseau. Surtout, il permet de donner des directions de recherche pour trouver des parades à ce phénomène, en particulier concernant les protocoles de transport et leurs mécanismes de contrôle de congestion.

13. Ces flux représentent, comme nous le verrons dans le chapitre suivant, l'essentiel de la charge du trafic.

1.4 Conclusion

Ce premier chapitre avait pour objectif d'introduire les concepts de base de notre travail de thèse relatif à la métrologie du réseau Internet et à la caractérisation du trafic. En particulier, il a mis en évidence la complexité des propriétés du trafic Internet actuel qui fait apparaître d'importants phénomènes oscillatoires que l'on peut "résumer" par la caractéristique de LRD. En particulier, dans la dernière section nous avons mis en évidence de façon expérimentale comment cette propriété de LRD avait un impact prépondérant sur le caractère oscillatoire du trafic et comment cette caractéristique du trafic (quantifiable par l'intermédiaire du facteur de Hurst) peut être prise en compte pour analyser la QdS fournie par le réseau. Dans le chapitre suivant, nous allons nous intéresser aux caractéristiques des trafics Internet actuels en considérant tour à tour différents points de collecte et en introduisant de nouvelles méthodologies pour permettre une analyse approfondie des caractéristiques de LRD du trafic réseau. L'objectif final est de pouvoir mieux connaître les propriétés du trafic de façon à pouvoir ensuite mieux agir sur la QdS dans l'Internet.

Chapitre 2

Evaluation de l'impact des caractéristiques du trafic Internet sur la QdS du réseau

2.1 Principales caractéristiques du trafic Internet actuel

2.1.1 Méthodes de caractérisation du trafic Internet

De la nécessité de décomposer le trafic

Il est très difficile de modéliser le trafic dans son ensemble. Premièrement, les lois d'arrivées des paquets et des flux ne sont pas régies par des lois simples (par exemple une loi de Poisson). Comme nous l'avons vu dans le chapitre précédent, elles montrent des propriétés d'auto-similarité, de LRD ou encore de multi-fractalité. Deuxièmement, il peut exister un ou deux régimes de dépendance dans le trafic (phénomène de "bi-scaling" introduit dans le chapitre précédent). Même si des modèles pour le trafic Internet peuvent être proposés, ils ne sont pas du tout simples et ne décrivent pas tous les aspects du trafic. La complexité du trafic nous pousse à essayer de décomposer le trafic en plusieurs classes qui peuvent être modélisées par des modèles plus simples (loi de puissance, loi de Poisson, loi Markovienne...). L'objectif de ce travail est donc de trouver des classes de trafic qui ont un comportement unique.

Dans le chapitre précédent, nous avons mis en évidence la présence de mémoire longue dans les processus d'arrivées des paquets et des flux de nos traces. Nous savons par ailleurs que la LRD s'étudie plus spécialement sur les grandes échelles temporelles. Ceci a été à l'origine d'une réflexion qui stipule qu'il serait plus adéquat d'étudier la LRD dans les flux les plus longs, c'est à dire ceux contenant le plus grand nombre de paquets. De plus, plusieurs études montrent que l'émergence de nouvelles applications mettant en œuvre des transferts de fichiers de plus en plus volumineux ont changé la composition du trafic et ont rendu les flux les plus longs largement majoritaires en volume de trafic [95], [70], [Lar02]. Toutes ces remarques nous ont poussés à décomposer le trafic, dans un premier temps, en deux classes. On parle dans la littérature liée à ce domaine de flux souris (connexions contenant un nombre de paquets très faible) et de flux éléphants (connexions possédant un grand nombre de paquets) [95]. Comme nous le verrons dans la partie suivante, les flux éléphants représentent la plus grande portion du trafic malgré leur faible nombre alors que les flux souris largement majoritaires ne

représentent qu'un faible taux de la quantité d'informations qui circule sur l'Internet.

Nous verrons cependant dans les parties suivantes qu'une décomposition de cet ordre, relativement simple, ne permet pas d'extraire assez de propriétés intéressantes dans le trafic. Il nous a donc été nécessaire de pousser plus avant cette décomposition en considérant le type d'application qui générait les flux souris ou éléphants. C'est l'objet de la décomposition par famille applicative. Nous allons donc mettre en évidence dans les sections suivantes les différentes propriétés du trafic à l'aide de ces deux types de décomposition.

Décomposition "souris vs. éléphants"

L'étude effectuée sur nos traces du réseau Renater, illustrée par la figure 2.1, montre que les flux éléphants (dont le nombre de paquets est supérieur à 100 [1]) représentent 80 % du volume total du trafic alors qu'en nombre de flux ils n'atteignent même pas les 3 %. Les flux souris quant à eux représentent plus de 87 % du nombre de flux et à peine 4 % du volume total du trafic. Il apparaît donc sur la figure 2.1 qu'un grand nombre de petits flux (ou "souris") représente une très faible proportion de la bande passante totale. Ce résultat s'inverse lorsqu'on considère les grands flux (ou "éléphants"). En effet, seuls quelques très longs flux (plus de cent paquets) véhiculent plus des trois quarts du trafic total du réseau.

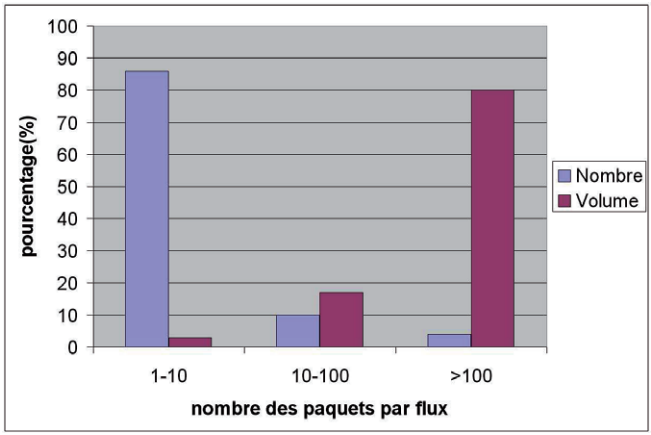


FIGURE 2.1 – Taille des flux TCP vis à vis de la bande passante

Décomposition par famille applicative

1. *Notion de "famille"*

1. Cette borne maximum a été choisie de façon empirique. Cependant, il n'est pas nécessaire de l'augmenter étant donné que l'on prend déjà en compte la très forte majorité du trafic en volume lorsque un flux éléphant est défini de la sorte.

Une famille de trafic est composée de tous les trafics générés par des applications du même type. La notion de “famille” est plus grande que celle de “protocole applicatif”. Certains protocoles sont classés dans la même famille parce que les trafics qu'ils génèrent ont presque les même caractéristiques. Par exemple, on classifie tous les trafics générés par les applications Peer-to-Peer (Kazaa, E-donkey, Bittorrent...) dans la famille “Peer-to-Peer” parce que toutes ces applications sont utilisées pour échanger des fichiers entre les utilisateurs. On classifie tous les trafics utilisés par les lecteurs streaming (Realplayer, WindowMediaPlayer...) dans la famille “Streaming” parce que les flux liés à ces applications sont tous de durées très longues et possèdent un débit plutôt stable et avec des contraintes temporelles fortes.

Décomposer le trafic en différentes familles applicatives n'est pas une nouvelle méthode en soi. Mais la plupart de telles décompositions sont basées uniquement sur le numéro de port utilisé dans l'entête TCP. Cette méthode prend en compte le numéro de port utilisé par le serveur (dans le cas d'une application client-serveur) pour écouter les connexions entrantes de ses clients. Par exemple, un serveur Web écoute les requètes HTTP d'un navigateur Web sur le port 80. Ainsi classifier le trafic par numéro de port revient à considérer le volume de trafic échangé au cours du temps pour l'ensemble des numéros de ports qui sont connus dans l'Internet. Classifier les flux par numéro de port est une méthode simpliste et rapide. Le filtrage par numéro de port ne nécessite pas beaucoup de puissance de calcul et on peut facilement trouver sur l'Internet une liste de ports actualisée qui décrit l'utilisation de chacun d'eux. C'est ce qui est généralement fait aujourd'hui.

Mais avec l'apparition des applications Peer-to-Peer et d'autres nouvelles applications personnelles, la décomposition par numéro de port est de moins en moins sûre. En effet, aujourd'hui, pour des raisons de sécurité, beaucoup d'administrateurs systèmes ont choisi de fermer les ports “non bien connus” (numéros supérieurs à 1024). Dans le cas de blocage, les utilisateurs peuvent utiliser un port non bloqué pour échanger les fichiers. Généralement, les ports non bloqués sont les ports bien connus : Web en 80, FTP en 21, Telnet en 23... Mais la fermeture de ces ports limite aussi les services du serveur. Ainsi, de plus en plus de services sont configurables pour utiliser les ports bien connus. Un bon exemple est le Windows Media Services de Microsoft qui peut offrir un service de streaming video/audio via le port 80. Dès lors, un trafic autrefois identifié comme trafic Web parce que échangé sur le port 80 ne doit plus être considéré comme tel à l'heure actuelle.

De la même façon les applications P2P ne respectent pas non plus “la règle du jeu” pour les numéros de port. Toutes les applications P2P permettent à l'heure actuelle aux utilisateurs de configurer le numéro de port à utiliser. Dès lors, le trafic Peer-to-Peer affecte considérablement la fiabilité de la décomposition par numéro de port. Dans notre analyse sur la trace de Jussieu, 11.8 % de trafic Peer-to-Peer utilise le port 80. Nous savons, d'autre part, que la nature du trafic P2P est très différente de celle du trafic Web : le trafic P2P transporte généralement des fichiers très gros (de quelques méga octets à quelques Giga octets) alors que le trafic Web transporte les fichiers de page web qui ne sont pas, en général, volumineux. Nous ne pouvons donc pas obtenir les caractéristiques précises de ces deux familles de trafic si nous les mélangeons.

2. *Principes de la décomposition du trafic en famille avec l'outil “Traffic Designer”*

Comme nous avons mis en évidence les limites de la décomposition applicative par numéro de port, nous devons donc classifier les flux selon leur contenu syntaxique. Nous avons utilisé pour cela l'outil "Traffic Designer" développé par la société QoSMOS [2] [120] (cf. Annexe A) qui peut faire l'analyse sémantique de chaque flux passant dans le réseau et fournir ainsi une liste de tous les flux utilisant une application considérée. Nous nous servons de cette liste pour faire la décomposition de la trace. La méthode de classification du trafic utilisée en fonction de l'application utilise une méthode de détection qui reconnait la famille applicative de chaque flux en se basant sur les premiers paquets échangés par l'application considérée. A l'aide de cette méthode (et contrairement aux méthodes "classiques" d'analyse des numéros de port) nous sommes capables de reconnaître un trafic échangé sur un numéro de port qui n'est pas traditionnellement dédié à cette application. D'autre part, il est ainsi possible de détecter tout le trafic généré par une application dont les numéros de port changent de façon dynamique. Par exemple, si une application P2P utilise le port 80 (port utilisé en standard par les applications Web), nous pouvons la détecter et classifier ce trafic comme P2P et non du trafic Web.

2.1.2 Caractéristiques simples du trafic en fonction du type de réseau

Après avoir présenté les différentes méthodes de décomposition utilisées dans notre travail, nous allons mettre en évidence les différences existant entre les trafics d'un réseau académique et d'un réseau commercial. Ce travail préliminaire nous permettra dans un deuxième temps de procéder à une décomposition plus avancée du trafic du réseau de Jussieu (le réseau académique considéré plus haut) à l'aide des deux méthodes de décomposition (souris / éléphants et par famille applicative) introduites précédemment. Seules les questions de volume par application sont donc traitées dans cette première partie, les paramètres plus avancés, comme le niveau de LRD, seront considérés par la suite dans les sections 2.1.3 et 2.2.1.

Ainsi, dans cette section, nous présentons une répartition temporelle de l'évolution de la distribution du trafic ainsi que des résultats cumulatifs. Tous ces résultats de classification sont présentés selon trois métriques : octets, paquets et flux (i.e. connexions). De plus, cette section présentera la distribution de la taille des flux pour toutes les applications détectées dans le trafic analysé. Cette information est de première importance étant donné que les travaux de [96] et [97] ont montré que les changements dans la distribution de la taille des flux, qui possède une queue de plus en plus lourde, a un rôle prépondérant sur l'évolution de la nature du trafic, en faisant apparaître des propriétés d'auto-similarité ou tout au moins de dépendance à long terme [Owe04b]. Il est ainsi très intéressant de pouvoir observer si ce type d'information reste valable pour tous les types de trafic (réseau académique : Man de Jussieu ; ou réseau opérateur : plaque ADSL de F&T). Nous avons considéré le trafic du réseau de Jussieu par rapport à celui du LAAS pour le comparer au réseau F&T car les débits sont à peu près équivalents sur les deux liens analysés : Gigabit Ethernet dans les deux cas, cf. annexe E pour détails (le réseau du LAAS représentant un trafic beaucoup moins important en terme de volume, sur une technologie Fast Ethernet).

Nom	Nombre de Flux	% en nb de flux	% en volume
TCP	1,745,870	75.2 %	92 %
UDP	492,732	21.2 %	7.4 %
ICMP	84,269	3.6 %	0 %
ESP	28	0 %	0 %
GRE	15	0 %	0.6 %
IPv6	7	0 %	0 %
IGMP	7	0 %	0 %
PIM	2	0 %	0 %
HOPOPT	1	0 %	0 %

TABLE 2.1 – Nombre de flux pour chacun des principaux protocoles de l'Internet

Exemple de caractérisation du trafic d'un réseau d'accès à Renater (MAN de Jussieu)

Nous avons capturé plusieurs traces de trafic sur le réseau du campus de Jussieu et elles ont toutes sensiblement montré les mêmes quantités d'octets, paquets et flux ainsi que la même distribution de trafic selon l'application considérée. Nous avons donc arbitrairement sélectionné l'une d'elles, et cette section présente la répartition et les graphiques camemberts obtenus par l'intermédiaire de notre méthode de classification applicative. Les données de la trace de Jussieu (cf. annexe E.4) sont globalement caractérisées comme suit (la contribution en terme de nombre de flux pour les différents protocoles est détaillée dans le tableau 2.1.2) :

- Date de capture : lundi 1er octobre 2004 ;
- Emplacement de la sonde de capture : réseau du campus de Jussieu ;
- Heure de départ : 14H50 ;
- Durée de la capture : 3600 secondes ;
- Nombre total de paquets : 80.437.378 ;
- Nombre total de flux : 2.322.931.

Etant donné que nous voulions analyser les flux entiers de façon à obtenir une distribution de la taille des flux par application la plus précise possible, nous avions besoin de nous focaliser sur les flux TCP qui démarraient et finissaient dans la trace. Un flux est défini comme terminé si son dernier "flag" TCP fait apparaître le message de fin de connexion (soit par l'intermédiaire d'une séquence de paquets FIN - ACK, soit de façon plus abrupte avec un paquet RST) ou si aucun paquet n'a été vu pour ce flux depuis au moins 4 minutes. Comme indication, les flux débutés avant la trace représentent 0,5 % des flux TCP (20 % du volume total), tandis que les flux encore actifs quand la trace se termine réprésentent 0,6 % de tous les flux (10 % du volume total). Contrairement à la différence en terme de nombre de flux qui est négligeable, la différence en terme de volume est relativement importante. Ce filtrage explique pourquoi le débit du trafic semble être faible au début de la trace dans les graphiques. Les figures 2.2 et 2.3 présentent les différentes répartitions et graphiques en camembert qui résultent du processus de classification du trafic par application. La figure 2.2 présente pour l'ensemble des applications la quantité de trafic total en octets, paquets et flux. La même chose est présentée dans la figure 2.3 mais en ne considérant que les familles d'applications. Nous rappelons qu'une famille d'application est constituée de toutes les applications qui ont le même objectif : par exemple, Kazaa et E-donkey appartiennent à la même famille P2P.

2. QoSMOS est une "jeune pousse" issue du laboratoire LIP6 à Paris.

Les figures 2.4 et 2.5 représentent l'évolution du trafic au cours du temps en octets / s à la fois pour les différentes applications et leurs familles associées. Le même type d'information est disponible dans les figures 2.6 et 2.7 mais pour le nombre de paquets / s et le nombre de flux / s.

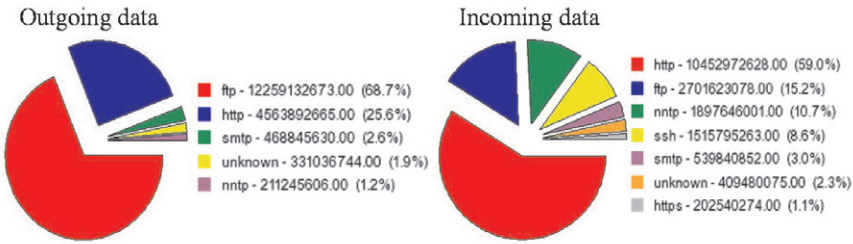


FIGURE 2.2 – Quantité globale de données par application (données sortantes et entrantes). Ces graphiques représentent la distribution du trafic par application en octets.

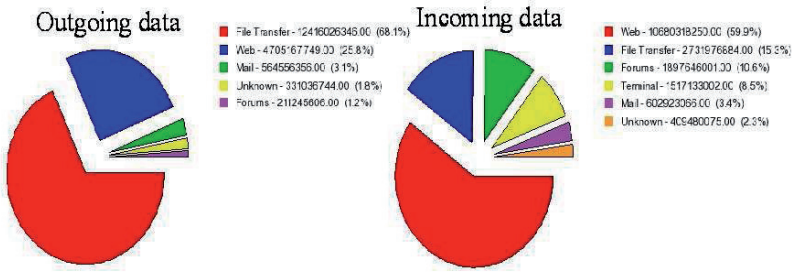


FIGURE 2.3 – Quantité globale de données par famille d'applications (données sortantes et entrantes). Ces graphiques représentent la distribution du trafic par application en octets.

La figure 2.8 représente la distribution de la taille des flux pour chaque application dans un diagramme log-log. L'objectif est donc d'évaluer si les distributions des tailles de flux sont à queue lourde ou non. Ce paramètre est très important étant donné que le degré de queue lourde des distributions à un impact sur la LRD et l'auto-similarité du trafic et agit aussi par conséquent sur les problèmes de performances du réseau. En fait, la figure 2.8 confirme dans les grandes lignes que les applications qui génèrent le plus de flux éléphants possèdent les distributions des tailles de flux les plus lourdes, comme [96] en a déjà fait la démonstration. En particulier, il est important de noter que les distributions les plus à queue lourde ne sont pas celles des applications P2P ou FTP, mais celles des applications orientées flux (basées sur RTSP), les applications terminal et les applications de chat. En étendant cette analyse, il apparait que le débit du trafic généré par chaque application est aussi un paramètre très

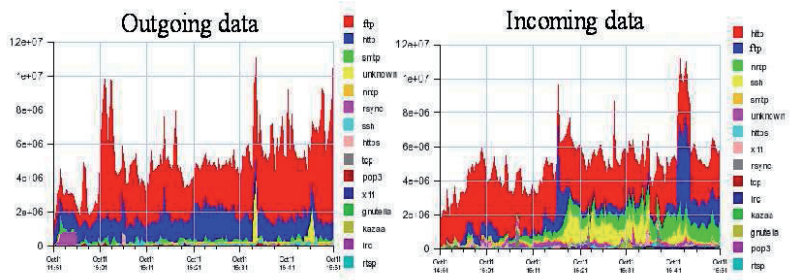


FIGURE 2.4 – Répartition du débit par application en octets / s (données entrantes et sortantes). Ces différentes répartitions représentent l'évolution du trafic octet au cours du temps pour les principales applications présentent dans le trafic de Jussieu – les applications sont classées dans le même ordre sur la légende et dans le graphique

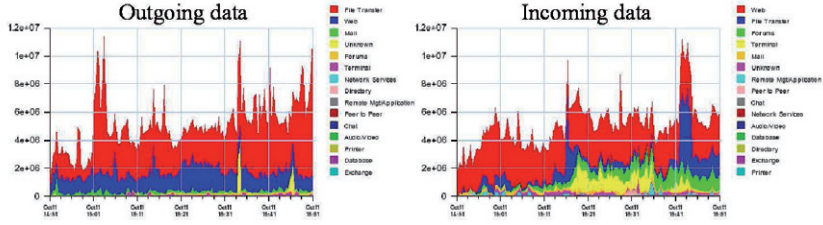


FIGURE 2.5 – Répartition du débit par famille d'applications en octets / s (données entrantes et sortantes). Ces différentes répartitions représentent l'évolution du trafic octet au cours du temps pour les principales familles d'applications présentes dans le trafic de Jussieu – les applications sont classées dans le même ordre sur la légende et dans le graphique

important pour expliquer la création de LRD dans le réseau (et non pas seulement la taille des flux). Ce résultat dépasse le cadre de cette analyse préliminaire et sera présenté dans la section 2.2.1 de ce chapitre.

Exemple de caractérisation du trafic d'une plaque ADSL de France Télécom

Cette section présente maintenant le même type d'analyse que celle menée pour les traces de Jussieu, mais cette fois sur une trace France Télécom collectée sur une plaque ADSL parisienne. Les caractéristique globales (cf. annexe E.2) sont les suivantes (la contribution en terme de nombre de flux pour les différents protocoles est détaillée dans le tableau 2.1.2) :

– Date de capture : mercredi 15 octobre 2004 ;
– Emplacement de la sonde de capture : dérivation d'un lien à haut débit connectant

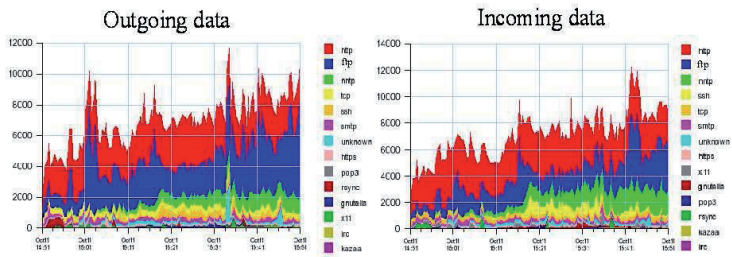


FIGURE 2.6 – Répartition du débit par application en paquets / s (données entrantes et sortantes). Ces différentes répartitions représentent l'évolution du trafic paquet au cours du temps pour les principales applications présentes dans le trafic de Jussieu – les applications sont classées dans le même ordre sur la légende et dans le graphique

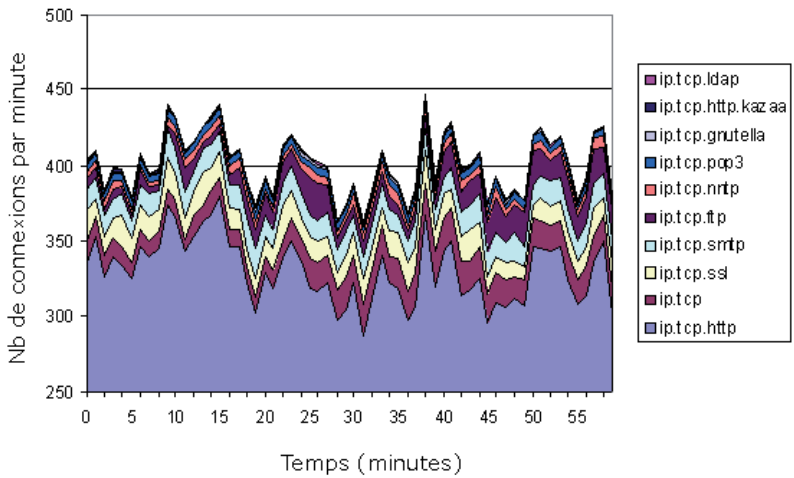


FIGURE 2.7 – Répartition du débit par application en flux / s (données entrantes et sortantes). Ces différentes répartitions représentent l'évolution du nombre de nouveaux flux au cours du temps pour les principales applications présentes dans le trafic de Jussieu : les flux ip.tcp représentent les flux où aucune donnée applicative n'a été échangée (c'est le cas lors de tentatives d'ouvertures de connexion infructueuses) – les applications sont classées dans le même ordre sur la légende et dans le graphique.

différentes plaques ADSL de la région parisienne de Fontenay aux Roses (seul le trafic

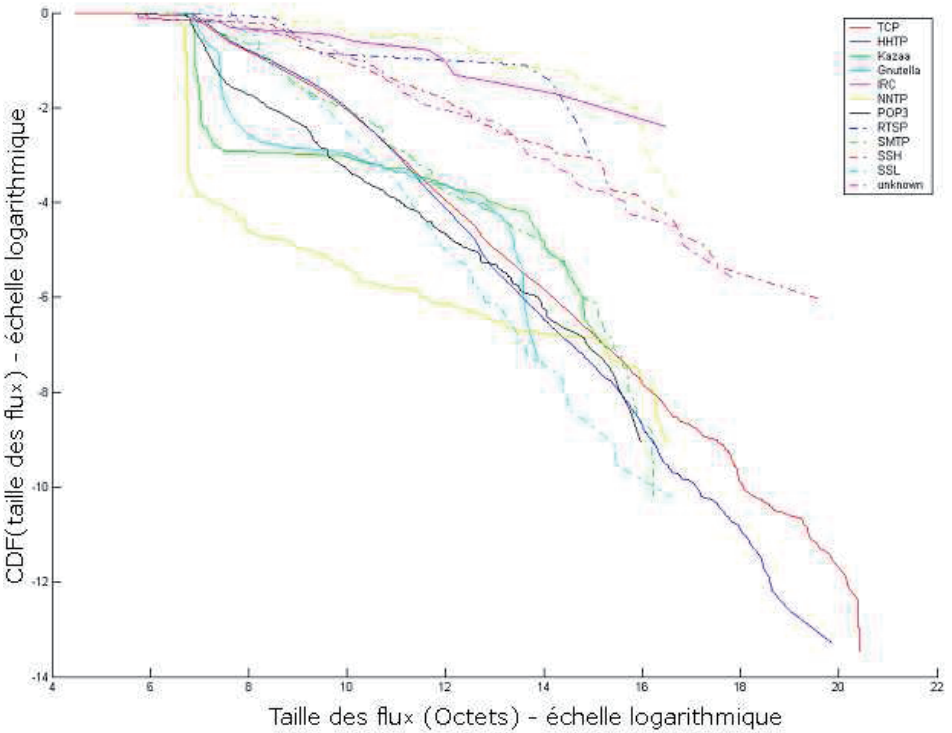


FIGURE 2.8 – Distribution de la taille des flux par application dans la trace de Jussieu

du lien descendant est analysé) ;
- Heure de départ : 19H01 ;
- Durée de la capture : 1300 secondes ;
- Nombre total de paquets : 134.434.541 ;
- Nombre total de flux : 9.636.105.

Les figures 2.9 et 2.10 illustrent la contribution en volume total des différentes applications et familles d'applications définies dans la section précédente. Il apparait clairement sur ces résultats que les applications P2P ont une contribution significative sur la charge totale. C'est la principale différence entre un trafic académique et un réseau commercial. Dans un trafic de campus, comme nous l'avons observé sur les traces du réseau de Jussieu, le trafic P2P est pratiquement inexistant. Dans un réseau commercial, le P2P a un impact beaucoup plus important. Nous verrons ceci plus précisément quand nous présenterons les distributions des tailles de flux pour les principales applications.

Nom	Nombre de flux	% en nombre de flux	% en volume
TCP	4,805,502	49.87 %	87.85 %
UDP	4,709,060	48.87 %	11.95 %
ICMP	121,378	1.26 %	0.04 %
ESP	91	0 %	0.14 %
GRE	38	0 %	0.03 %
IPv6	26	0 %	0 %
AH	10	0 %	0 %

TABLE 2.2 – Nombre de flux en fonction de principaux protocoles de l'Internet

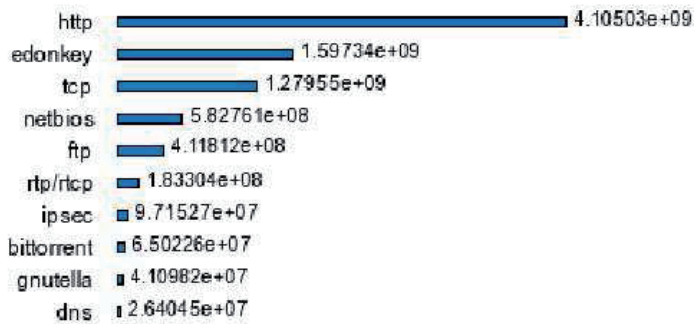


FIGURE 2.9 – Quantité total de données par applications (données entrantes). Cet histogramme représente la distribution du trafic par applications en octets.

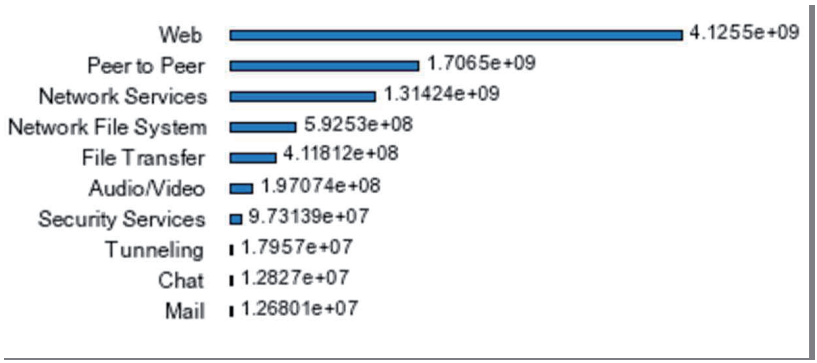


FIGURE 2.10 – Quantité total de données par famille d'applications (données entrantes). Cet histogramme représente la distribution du trafic par familles d'applications en octets.

Les figures 2.11, 2.12 et 2.13 représentent le débit pour les applications et les familles d'applications. Ces figures montrent que le débit global est stationnaire dans l'ensemble ; il n'y a pas une déviation évidente dans le processus du débit. On peut aussi observer que la proportion de P2P est en augmentation ce qui implique que sa contribution à la charge globale du réseau devient de plus en plus significative. Plusieurs traces de trafic montrent que le trafic P2P domine en fin de journée (i.e. entre 21H00 et 00H00). La figure 2.14 présente le débit en nombre de flux / s, elle illustre que la situation est stable pendant la période d'observation.

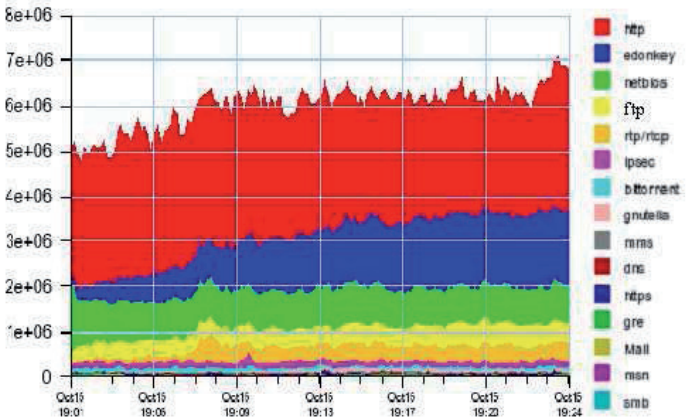


FIGURE 2.11 – Répartition du débit par application en octets / s (données entrantes). Cette répartition réprésente l'évolution du trafic octet au cours du temps en fonction des principales applications présentes dans le trafic France Télécom – les applications sont classées dans le même ordre sur la légende et dans le graphique.

La figure 2.15 représente les distributions des tailles de flux pour les différentes applications. Il est intéressant de noter que les distributions sont sensiblement différentes de celles obtenues pour le trafic du campus de Jussieu. Dans le diagramme log-log, les distributions ne peuvent pas être facilement approximées par des droites. Une analyse plus poussée [20], a montré que la queue des distributions peut être approximée par des distributions Weibull. De plus, les distributions des tailles de flux pour les applications P2P mettent en exergue un comportement bimodal. En examinant plus précisément la composition du trafic P2P, nous allons observer (ce résultat va être illustré dans les sections suivantes et est également présenté dans [20]) qu'un grand nombre de flux P2P sont relativement petits. Ces flux sont générés par le trafic de signalisation des réseaux P2P (recherche de fichier ou gestion de l'anneau de P2P). En fait, pour une analyse plus fine, il est nécessaire de séparer les petits flux (souris) des grands flux (éléphants). En effet, la plupart des protocoles P2P, en particulier E-donkey, segmentent les longs flux en plus petits (appelés "chunks"), qui peuvent être téléchargés de façon asynchrone et en parallèle par les clients. Ces deux notions vont être approfondies, dans les sections qui suivent, relatives à la décomposition souris vs. éléphants et une décomposition applicative plus

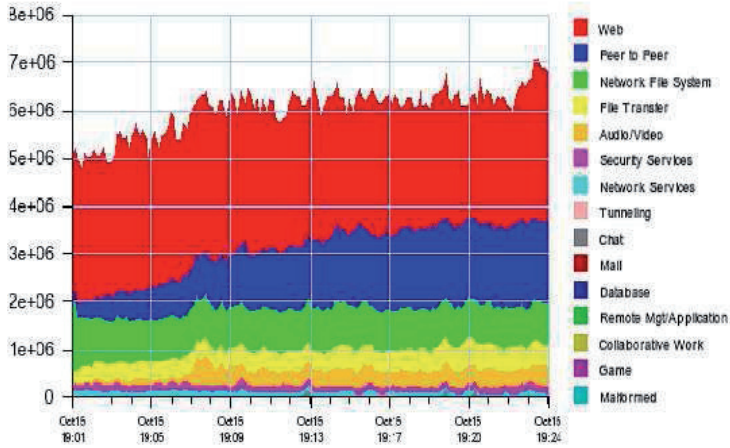


FIGURE 2.12 – Répartition du débit par famille d'applications en octets / s (données entrantes). Cette répartition réprésente l'évolution du trafic octet au cours du temps en fonction des principales familles d'applications présentes dans le trafic France Télécom – les applications sont classées dans le même ordre sur la légende et dans le graphique.

détaillée du trafic de Jussieu.

Conclusion

Cette section a présenté une analyse de la distribution du trafic par application dans deux réseaux différents : un réseau académique et un réseau commercial. Pour les répartitions des distributions de trafic par application, nous avons proposé l'utilisation d'une méthodologie basée sur la collecte de traces de trafic par l'intermédiaire de sondes DAG, et parallèlement sur la classification du trafic en utilisant les boîtiers TD de la société QoSMOS, qui est basée sur une reconnaissance applicative des paquets. Nous avons mis en évidence des différences majeures dans l'utilisation des deux réseaux. Par exemple, le réseau Renater est dédié à l'éducation et à la recherche, et exclut au maximum les transferts P2P qui ne sont pas en relation avec une utilisation "éducation" ou "recherche" du réseau. Autant que possible, les administrateurs réseaux des campus universitaires font de leur mieux pour mettre en œuvre cette politique. De plus, étant donné que le campus de Jussieu accueille un des principaux serveurs FTP français, qui réplique de nombreux serveurs dans le monde (GNU, Linux...), l'application qui est responsable de la plus grande proportion du trafic est FTP. A l'opposé, il est clair que la principale proportion du trafic dans les réseaux commerciaux (comme le réseau de France Télécom) est composé de trafic P2P.

Ces premiers résultats d'analyse de traces font donc déjà apparaître des différences de caractéristiques entre les différents réseaux considérés. L'invariant qui ressort de ces résultats

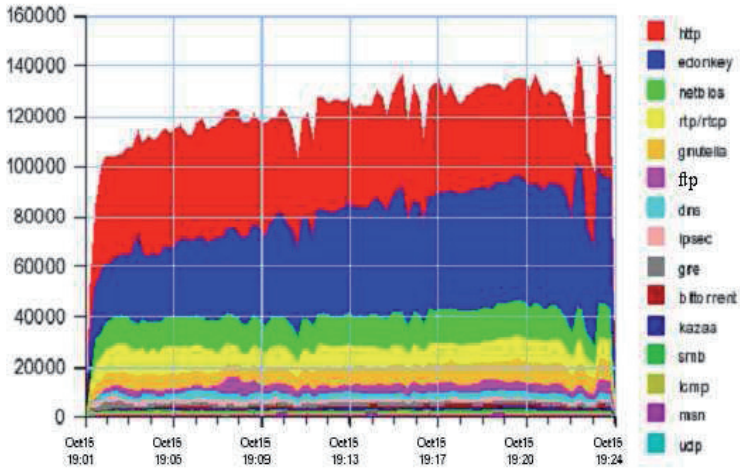


FIGURE 2.13 – Répartition du débit par application en paquets / s (données entrantes). Cette répartition réprésente l'évolution du trafic paquet au cours du temps en fonction des principales applications présentes dans le trafic France Télécom – les applications sont classées dans le même ordre sur la légende et dans le graphique.

concerne la forte proportion de flux très longs et la queue lourde très prononcée de la distribution de la taille de ces flux. Comme nous l'avons déjà résumé dans les sections précédentes, ce paramètre est à l'origine du comportement à dépendance longue du trafic Internet actuel. Il semble donc fort à propos dans les deux sections à venir de se focaliser sur ces deux caractéristiques majeures de l'Internet actuel : la forte proportion de flux longs et dans un deuxième temps les différentes applications qui sont génératrices de tous ces flux éléphants dans le réseau. L'objectif reste de pouvoir quantifier précisément quels sont les plus forts contributeurs à la LRD du réseau. Ce résultat nous permettra dans un deuxième temps (cf. chapitre suivant) de pouvoir mieux agir sur ce paramètre de dégradation des performances du réseau.

2.1.3 Caractéristiques du trafic selon la décomposition « souris vs. éléphants »

Analyse au niveau flux

1. *Etude du trafic souris*
   Les flux souris ont été définis comme des flux contenant un petit nombre de paquets. [95] considère traditionnellement une limite de 10 paquets pour la taille maximale d'un flux souris. Un exemple de ces flux est donné par les connexions ouvertes pour télécharger différents objets appartenant à une page web. Il est intéressant de modéliser les arrivées

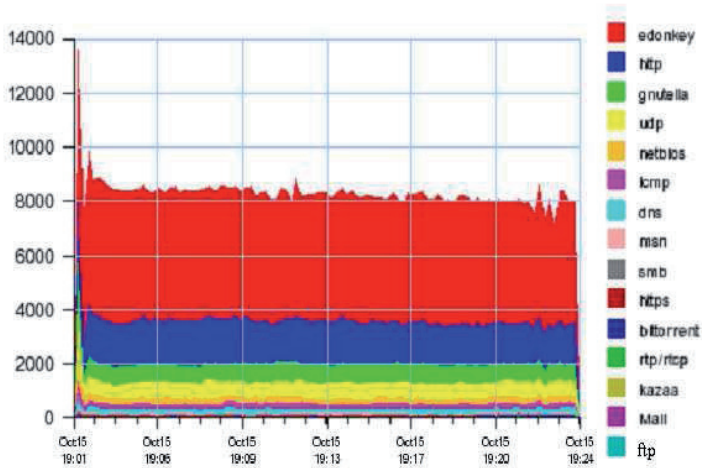


FIGURE 2.14 – Répartition du débit par application en flux / s (données entrantes). Cette répartition réprésente l'évolution du trafic en termes de nouveaux flux pour les principales applications présentes dans le trafic France Télécom – les applications sont classées dans le même ordre sur la légende et dans le graphique.

de ces flux car ils sont très liés au niveau applicatif et donnent une idée plus précise des comportements des utilisateurs et des applications. Nous avons fait l'analyse de la LRD sur les inter-arrivées des flux souris. La mesure du facteur de Hurst par la méthode des ondelettes (voir la figure 2.16) donne le résultat suivant : $H = 0.805$. Ceci montre que la LRD est très forte dans cette série de données. L'analyse Q-Q Plot (cf. figure 2.17) implique aussi que la distribution d'inter-arrivées des flux souris est à queue lourde. Le trafic web est majoritairement composé de flux souris correspondants aux transferts d'objets graphiques, d'images, de texte, de logo, d'Applets Java, etc. Concrètement, les résultats précédents illustrent que lorsqu'un utilisateur navigue sur le Web, le protocole HTTP 1.0 ouvre autant de connexions que d'objets dans la page visitée et il crée ainsi de la dépendance entre chaque connexion ouverte pour télécharger chaque élément d'une même page, qui sont en majorité des flux souris.

2. *Etude du trafic éléphant*

   Les mêmes analyses ont été effectuées sur le trafic des éléphants. Le facteur de Hurst est de 0.57 (cf. figure 2.18). Ce résultat met en évidence que la LRD est très faible pour cette partie du trafic. L'analyse Q-Q Plot (cf. figure 2.19) nous donne un résultat qui va dans le même sens : la loi d'arrivées des inter-flux éléphants est très proche de la loi exponentielle. Il est à noter que le trafic éléphant correspond à des transferts de gros fichiers. Il s'agit donc de téléchargement par des utilisateurs de fichiers audio ou vidéo par exemple. Ces flux dépendent essentiellement du comportement des utilisateurs et

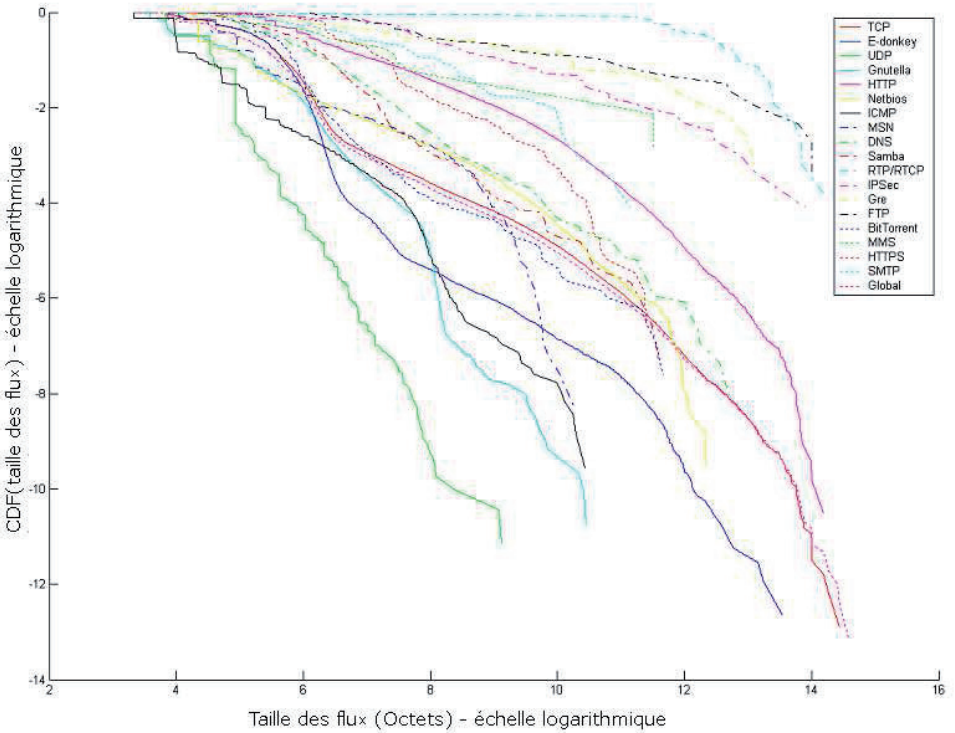


FIGURE 2.15 – Distribution des tailles de flux pour chaque application dans la trace F&T

sont généralement exécutés en tâches de fond. Il est donc logique que les résultats pour ce type de trafic soient très proches d'un trafic poissonnien.

Analyse au niveau paquet

Après avoir étudié la loi d'inter-arrivées des flux, nous nous sommes intéressés aux paquets composant chacun de ces flux. Les figures 2.21 et 2.20 montrent la mesure du facteur de Hurst sur la série d'inter-arrivées des paquets. Le résultat nous montre que :

- les arrivées de paquet sont à dépendance longue ;
- les paquets éléphants ont une LRD plus forte que les paquets souris ($H = 0.718$ pour les souris et $H = 0,972$ pour les éléphants).

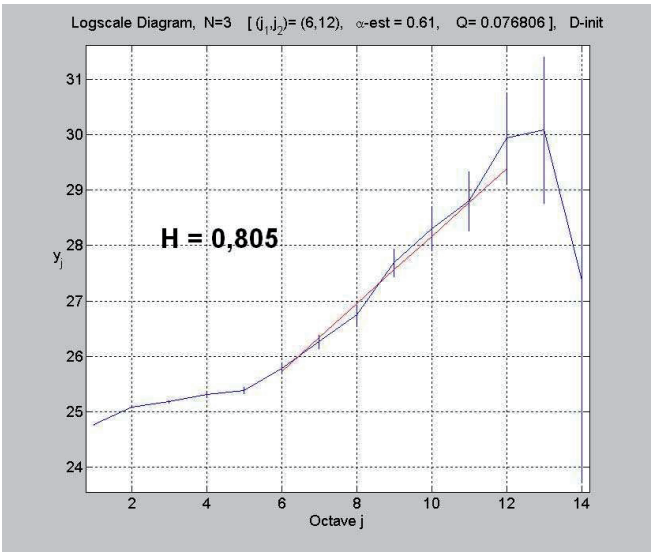


FIGURE 2.16 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série d'inter-arrivées des flux souris

Conclusion concernant la décomposition entre flux souris et éléphant

A l'issue de ce travail de caractérisation, nous avons observé les phénomènes suivants :

- la dépendance est bien marquée au niveau flux pour la classe des souris ;
- la dépendance au niveau paquet pour les flux souris est moins forte que les flux éléphant étant donné que les flux souris sont des flux très courts pour lesquels les phénomènes de corrélation et de dépendance à long terme sont beaucoup moins prononcés que pour les flux éléphants (ceci est dû en grande partie au mécanisme de contrôle de congestion de TCP qui entraîne beaucoup plus de dépendance entre les paquets d'une même fenêtre de congestion dans le cas de l'échange d'un flux éléphant que d'un flux souris, la fenêtre de congestion étant beaucoup plus grande dans le premier cas) ;
- la LRD au niveau paquet pour les souris est quand même très élevée (H=0,718) dans cette trace ;
- les arrivées des flux éléphants semblent être assimilables à un processus de Poisson et la LRD au niveau des arrivées des flux éléphant n'est pas importante par rapport à celle des flux souris ;
- l'analyse au niveau paquet des flux éléphants met en évidence plus de dépendance à long terme pour cette classe par rapport à la LRD présente dans le trafic global. Ceci est lié à la présence très importante des paquets des flux éléphants qui sont majoritaires en volume dans le trafic Internet et qui apportent une très forte contribution à la LRD globale du trafic.

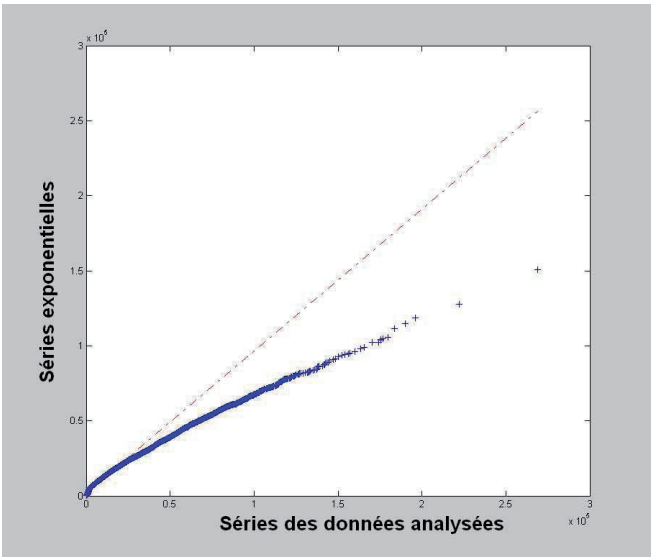


FIGURE 2.17 – Représentation Q-Q Plot de la loi d'arrivée des flux souris

2.1.4 Les limites d'une décomposition souris / éléphants

Cette section qui se termine a successivement mis en évidence une méthode de caractérisation du trafic basée sur sa décompostion en différentes classes de trafic et illustré au travers d'un exemple l'intérêt et les possibilités offertes par cette méthode. En effet, le trafic Internet est par nature difficile à caractériser dans sa globalité, en particulier à cause de la complexité des caractéristiques traditionnellement mises en évidence lors des phases d'analyse simple. La méthode proposée dans cette section permet de casser cette complexité en se focalisant sur des sous-ensembles du trafic qui possèdent des caractéristiques plus intéressantes car plus simples. Un exemple de ces caractéristiques a été développé au travers de la caractérisation du trafic en deux classes, souris et éléphants, qui a permis de mettre en évidence par exemple un comportement des flux éléphants assimilable à un processus poissonnien et à l'inverse de faire ressortir les problèmes de corrélation et de dépendance à long terme au niveau des flux souris ou des paquets des flux éléphants. Cette caractéristique complexe mesurable au niveau des paquets des flux éléphants était néanmoins attendue pour cette partie du trafic. En effet, le grand nombre de paquets composant un flux éléphant favorise la création du phénomène de LRD pour cette partie du trafic comme nous l'avons déjà développé dans le chapitre 1 au travers de la figure 1.14.

Ainsi, en décomposant le trafic global en deux classes, nous avons bien mis en évidence la diversité des caractéristiques des classes de flux : les arrivées des flux souris sont dépendantes à long terme alors que celles des flux éléphants sont faiblement dépendantes. Cette constatation

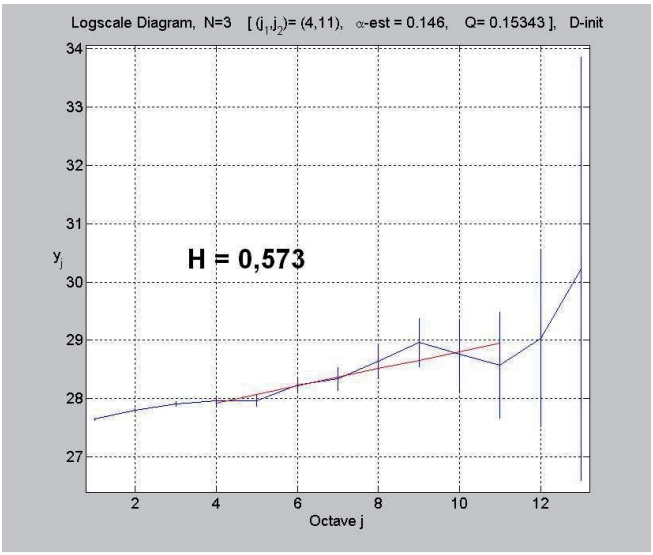


FIGURE 2.18 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série d'inter-arrivées des flux éléphants

est très importante pour la modélisation. Elle nous permet d'imaginer une approche visant à trouver un modèle pour chaque type de flux en tenant compte des caractéristiques spécifiques à chaque classe. Mais même si nous pouvions modéliser parfaitement les différentes classes de trafic, beaucoup de questions restent encore en suspens : d'où viennent les différences que l'on peut observer entre les deux classes de trafic ? Au niveau applicatif, quel est le plus grand contributeur à la LRD globale ? Et au delà, comment diminuer la LRD en vue d'améliorer la QdS dans le réseau. Pour permettre d'apporter des réponses aux questions précédentes, dans la partie suivante, nous allons aborder un deuxième type de décomposition : la décomposition par famille qui va permettre d'isoler de façon plus précise les caractéristiques des différents flux souris et éléphants en réalisant une distinction sur le type d'application qui génère ces flux.

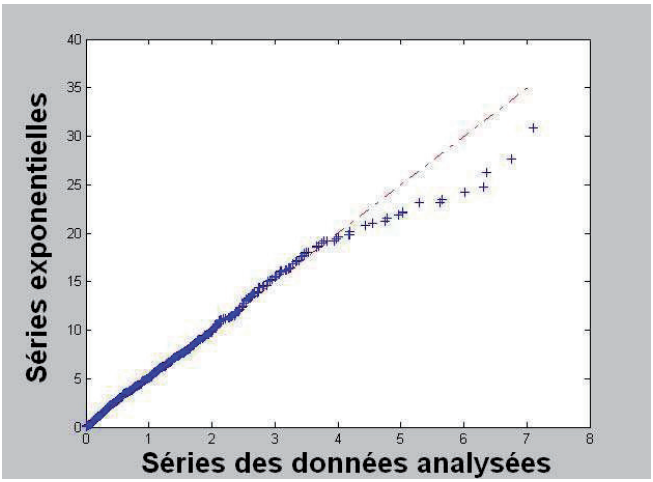


FIGURE 2.19 – Représentation Q-Q Plot de la loi d'arrivée des flux éléphants

2.2 Relation entre taille des flux, LRD, QdS et performances du réseau

2.2.1 Etude de l'impact de la famille d'application sur la variabilité du trafic

Définition des « grandes » familles applicatives

Dans la trace de Jussieu (cf. annexe E.4), nous avons isolé quelques grandes familles du trafic :

- Web : tous les trafics utilisant le protocole HTTP ;
- FTP : tous les trafics utilisant le protocole FTP ;
- Terminal : les trafics Telnet, SSH. La proportion de ce type de trafic est considérable dans les réseaux académiques mais très minoritaire dans les réseaux commerciaux ;
- Streaming : le trafic correspondant à l'échange des flux vidéo / audio en temps réel ;
- P2P : trafic très important dans les réseaux commerciaux mais plutôt faible dans les réseaux académiques ;
- Unknown : tous les flux que TDPlayer ne peut pas identifier sont classifiés dans la famille Unknown.

Il y a trois causes principales pour lesquelles TDPlayer ne peut pas identifier un flux :

1. le flux est sur un nouveau protocole de transport que TDPlayer ne connaît pas : c'est à dire le flux est dans un protocole qui n'est pas (encore) supporté par l'outil QoSMOS ;
2. le début de flux n'est pas dans la trace ; dès lors TDPlayer ne peut pas faire l'analyse sémantique correctement pour ce flux ;

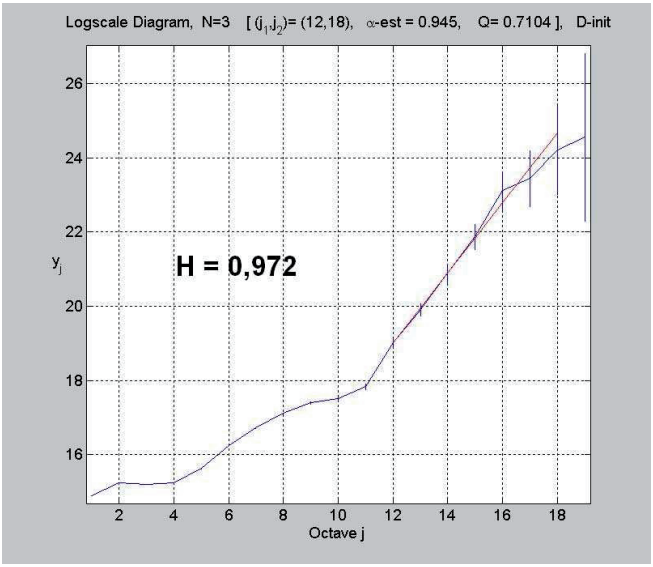


FIGURE 2.20 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série d'inter-arrivées des paquets éléphants

3. le flux est généré par les applications non standards conçues par la recherche et en cours d'expérimentation.

Caractéristiques du trafic en fonction de la famille applicative

En utilisant TDPlayer, nous pouvons isoler les différentes familles de trafic. Nous rappelons que la trace de Jussieu a été collectée sur un lien d'accès à Renater qui est un réseau d'enseignement et recherche. A partir de la figure 2.22, on peut voir que la proportion de la famille terminal et celle de FTP est grande tandis que la famille Peer-to-Peer est minoritaire. Ceci représente la différence entre les réseaux commerciaux et les réseaux académiques qui a été mise en évidence dans la section 2.1.2. La famille Unknown occupe la plus grande proportion. L'origine de la famille Unknown a été expliquée dans le paragraphe précédent.

L'existence de la famille Unknown est un peu gênante pour la décomposition parce que les caractéristiques d'une famille inconnue n'ont aucun intérêt. Nous allons donc mener une analyse plus détaillée de cette classe du trafic pour en extraire un maximum d'informations.

1. *Distribution des tailles de flux*

   Nous utilisons le diagramme de CCDF pour comparer les queues des distributions par famille. La figure 2.23 représente la CCDF des tailles de flux par famille pour la trace de Jussieu. La queue du trafic global est légèrement plus lourde que la famille Web et moins lourde que toutes les autres familles. Nous allons voir dans les paragraphes qui

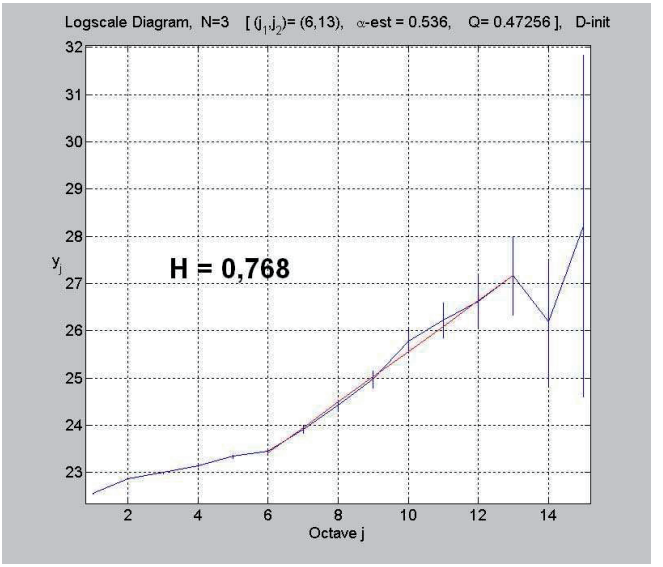


FIGURE 2.21 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série d'inter-arrivées des paquets souris

suivent que le niveau de queue lourde des différentes familles est en relation directe avec le niveau de LRD du trafic considéré.

2. *Analyse de la LRD par famille*

   Nous allons analyser les caractéristiques de chaque famille. Les analyses sont toutes réalisées sur la trace de Jussieu.

   (a) *Trafic TCP global*

   La mesure du facteur de Hurst par la méthode des ondelettes donne le résultat suivant : $H = 0.855$ (figure 2.24). La courbe met en évidence le phénomène de bi-scaling avec le point de variation de la loi de puissance (bi-scaling) situé à l'octave 8, c'est-à-dire à 256 ms (2^8 ms) en terme de temps. La pente de la courbe entre l'octave 10 et 17 montre qu'il y a une forte LRD pour les grandes échelles de temps.

   (b) *Trafic Peer-to-Peer*

   Le trafic P2P est la famille majoritaire sur les réseaux commerciaux. Bien que la proportion de trafic P2P ne soit pas très grande dans la trace de Jussieu, l'analyse sur cette famille nous donne quand même des résultats intéressants. La mesure du facteur de Hurst (figure 2.25) par la méthode des ondelettes donne le résultat suivant : $H = 1.01$ [3]. Ceci indique une LRD très important. Comme nous l'avons

3. En théorie, la valeur du facteur de Hurst ne doit pas dépasser 1. Les valeurs supérieures à 1 sont dues à

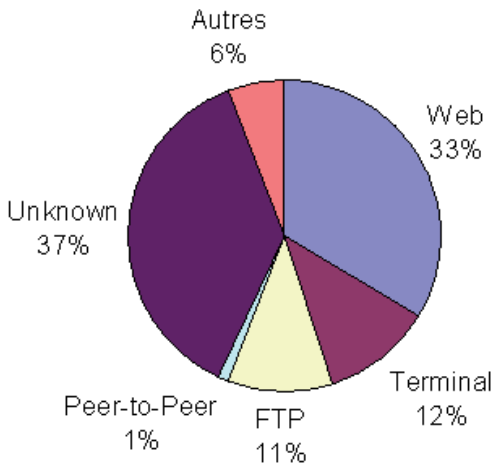


FIGURE 2.22 – Composition (en volume) de la trace de Jussieu

constaté dans le paragraphe précédent, les flux P2P sont normalement des flux longs et volumineux. Une LRD très grande était donc attendue.

Parmi les applications P2P, les plus utilisées sont E-donkey, Kazaa, Bittorrent, Morpheus et Gnutella. Bien qu'elles soient toutes appelés P2P, les mécanismes de transfert qu'elles implémentent sont différents (nous approfondirons cette notion dans la suite de la section). Ainsi, les caractéristiques du trafic qu'elles vont générer, devraient être, elles aussi, différentes. Dans la trace de Jussieu, nous avons observé la présence de trafic Kazaa et E-donkey. La figure 2.26 compare les queues de la distribution des tailles de flux. Il est évident que la queue de E-donkey est bien moins lourde que celle de Kazaa.

Par conséquent, le trafic de E-donkey (figure 2.27) devrait avoir une LRD moins forte que celui de Kazaa (figure 2.28). L'analyse par la méthode des ondelettes confirme notre constatation.

La différence entre le trafic Kazaa et E-donkey peut s'expliquer par le fait que E-donkey établit plusieurs flux, au lieu d'un seul. Ainsi, pour transférer un fichier volumineux, chaque flux transfère un segment du fichier. Ce mécanisme réduit la présence des très longs flux et ainsi réduit la LRD du trafic. Aujourd'hui, les applications P2P sont de plus en plus utilisées. Par conséquent, dans les réseaux

une imprecision de l'outil LDEstimate lorsque le calcul de la LRD se fait sur un nombre réduit de points. Or dans le cas des différentes familles applicatives, certains flux ne sont pas très présents dans la trace globale, néanmoins cette valeur élevée de H traduit et permet de conclure sur le niveau de LRD très important pour le trafic considéré.

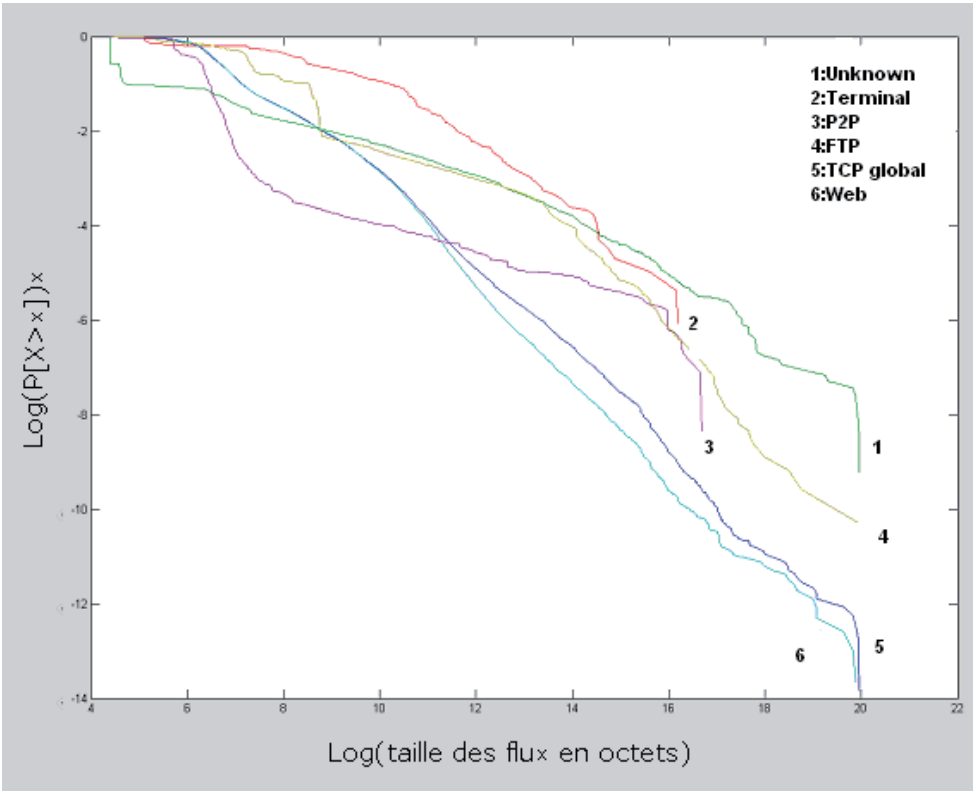


FIGURE 2.23 – CCDF (log-log) des tailles de flux par famille

commerciaux, surtout sur les plaques ADSL, le trafic P2P occupe une très grande proportion du trafic global. Ainsi, le trafic P2P peut introduire une LRD importante dans le trafic global.

(c) *Trafic Terminal*

Les flux de terminaux Telnet ou SSH sont aussi des flux longs mais pas forcément volumineux. Une connexion Telnet qui transfère les commandes d'un utilisateur n'est pas un gros flux en terme de volume. Mais la LRD existe néanmoins pour ce type d'application. Dans [100], les auteurs illustrent la LRD de la connexion Telnet. Aujourd'hui, SSH est plus utilisé que Telnet parce que SSH fournit une connexion sécurisée. Il est très souvent utilisé pour établir un tunnel sécurisé pour le transfert d'informations critiques. Ainsi des gros volumes de données peuvent être véhiculés dans des flux SSH. Ceci implique que la LRD peut être très élevée. C'est exactement

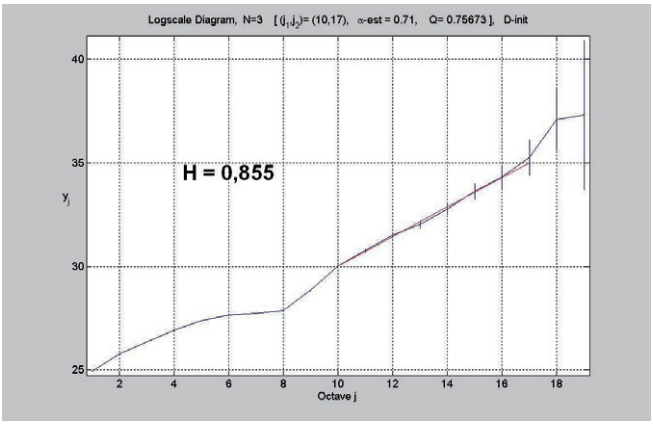


FIGURE 2.24 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit du trafic global (granularité 1 ms)

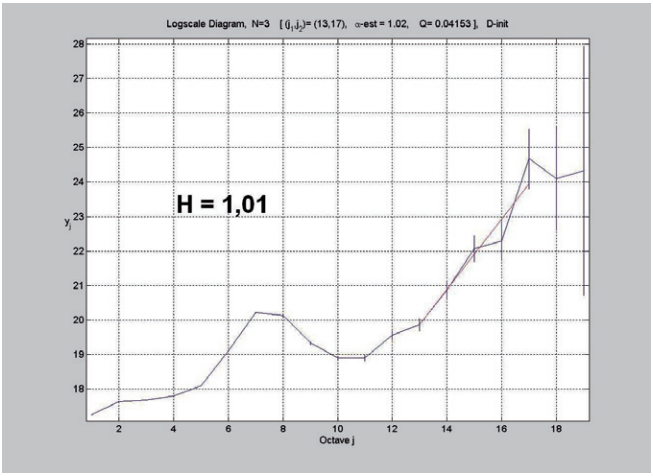


FIGURE 2.25 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit du trafic P2P (granularité 1 ms)

le cas dans la trace de Jussieu, les gros flux SSH sont très présents. En effet, les trois flux les plus gros de cette famille transfèrent 4 Giga octets de données en 50 minutes ! La mesure du facteur de Hurst par la méthode des ondelettes montre que

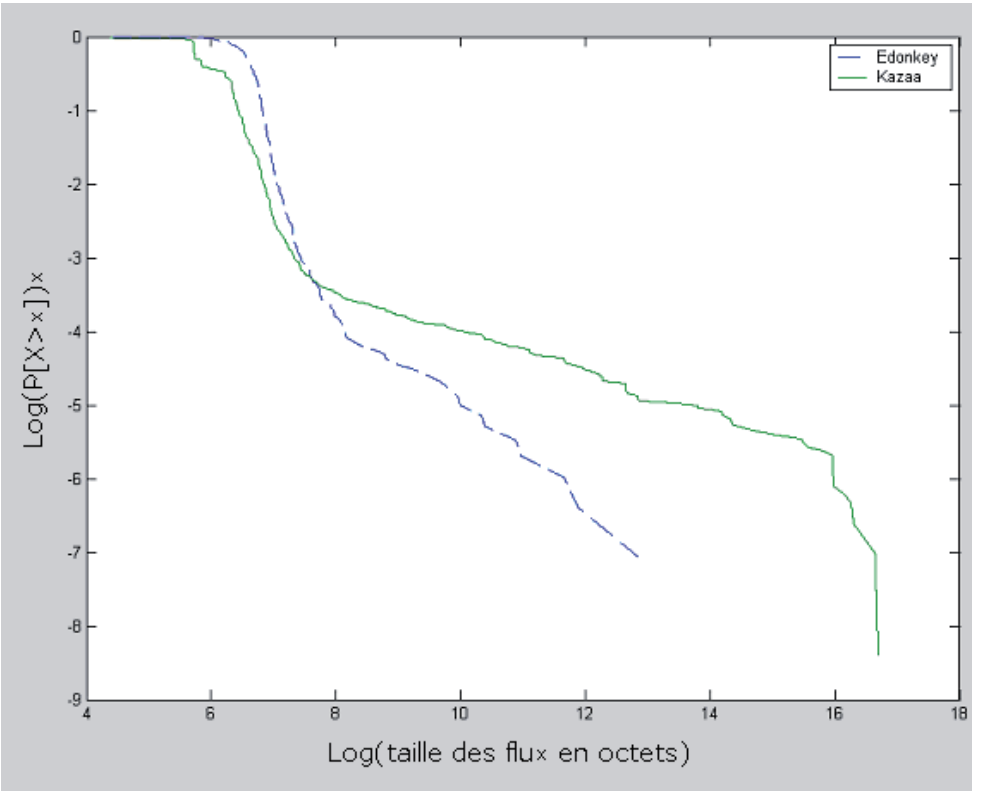


FIGURE 2.26 – CCDF (log-log) des tailles de flux de E-donkey et Kazaa

la LRD de la famille Terminal est très élevée : $H = 1,15$. (figure 2.29).

(d) *Trafic WEB*

Tout le trafic de protocole HTTP est classifié dans la famille WEB. C'est une grande famille autant sur les réseaux commerciaux que sur les réseaux d'enseignement et recherche. [38] a montré que le trafic Web était à queue lourde et auto-similaire en 1995. La majorité des flux WEB sont des flux utilisés pour transférer des fichiers d'image et des textes HTML. Les gros flux ne sont pas très présents dans cette famille. Ainsi, la LRD de cette famille devrait normalement être moins élevée par rapport aux familles où les gros flux sont majoritaires. La mesure du facteur de Hurst par la méthode des ondelettes montre que la LRD de la famille est quand même élevée : $H = 0,904$, plus élevée que la LRD globale ($H = 0,71$). Ceci implique la présence de gros flux dans cette classe du trafic.

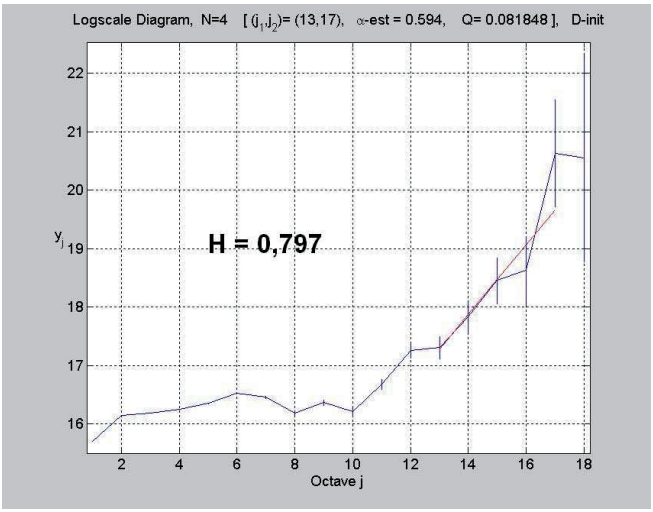


FIGURE 2.27 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit du trafic E-donkey (granularité 1 ms) — Paramère Hurst : 0,797

Comme nous le savons, le protocole HTTP est conçu pour transporter les fichiers Web (texte HTML et images). Normalement ces fichiers ne sont pas très volumineux pour la rapidité d'affichage de pages Web. Dans ces conditions, pourquoi le trafic HTTP montre-t-il quand-même une LRD si élevée dans cette trace ? Pour trouver son origine, nous isolons les gros flux (plus de 1 méga octets) dans cette famille qui pourrait influencer les caractéristiques globales.

Le résultat est le suivant :

i. Le nombre des flux HTTP supérieurs à 1 méga est de 679 alors que le nombre total des flux http est de 1,167,579.

ii. Le volume total des flux HTTP supérieurs à 1 méga est de 5.31 Giga octets pour un volume total des flux HTTP de 14.3 Giga octets . Ces 679 flux, soit 0,00058 % des flux dans cette famille, représentent 37,3 % du volume total.

iii. La mesure du facteur de Hurst par la méthode des ondelettes montre que la LRD du trafic de ces gros flux HTTP est très élevée : $H = 1,01$ (figure 2.31), plus élevée que la LRD globale de cette famille ($H = 0,904$).

iv. Certains flux HTTP sont très gros. Le volume des dix flux les plus gros est de 2,56 Giga octets.

Nous pouvons en conclure que les gros flux HTTP augmentent la LRD globale de la famille HTTP. Etant donné la taille des fichiers web en général relativement faible (quelques dizaines d'octets à quelques Kilo octets), seuls les téléchargements réalisés par l'intermédiaire du protocole HTTP peuvent générer des flux aussi gros.

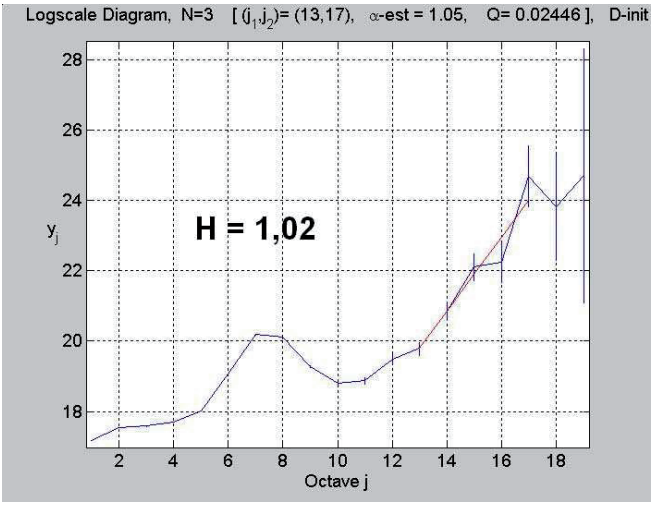


FIGURE 2.28 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit du trafic Kazaa (granularité 1 ms) — Paramère Hurst : 1,02

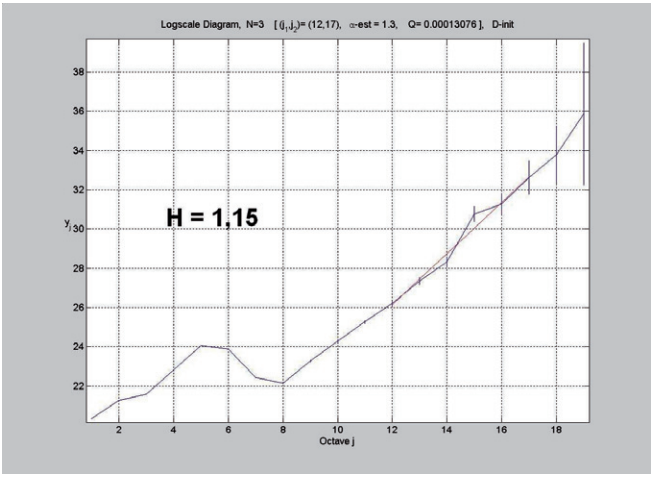


FIGURE 2.29 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit du trafic Terminal (granularité 1 ms) — Paramère Hurst : 1,02

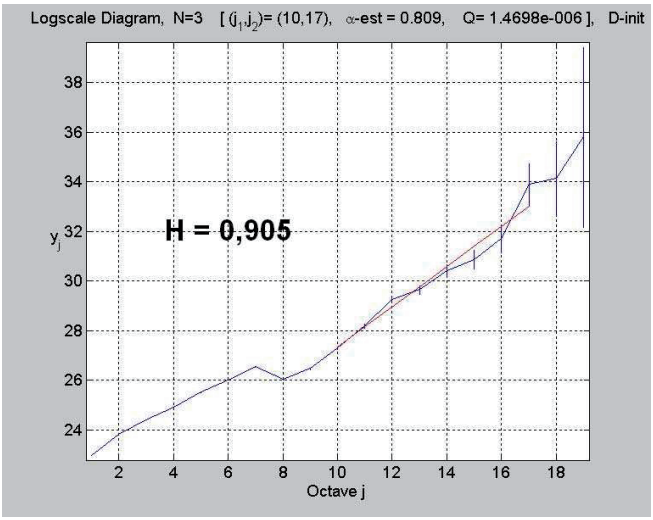


FIGURE 2.30 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit du trafic Web (granularité 1 ms) — Paramère Hurst : 0,904

(e) *Trafic Unknown*

La famille Unknown est une famille spéciale. Elle est composée de tous les flux non identifiables par TDPlayer. Nous avons déjà proposé trois causes possibles pour les flux inconnus dans la section précédente. De la figure 2.23, nous pouvons voir que la distribution des tailles de flux de la famille Unknown a une des queues les plus lourdes. Nous avons isolé et trié les flux les plus gros de cette famille. Le résultat nous montre que les 100 flux les plus gros ont contribué à 84,16 % (13,46 Giga octets sur 16 Giga octets) de volume total de la famille. C'est pourquoi la queue de la distribution est si lourde. De plus, l'étude détaillée des paquets composant ce trafic [4] montre que la majorité appartient au protocole NNTP, il s'agit donc de transferts effectués lors des mises à jour des serveurs de news de l'Internet, ce qui explique le volume très important des données échangées par ces flux.

Conclusion sur la décomposition par famille applicative

La décomposition par famille nous permet d'isoler et analyser les trafics générés par les différentes applications. En comparant les caractéristiques des différentes familles de flux, nous pouvons connaître la contribution de chaque famille à la LRD globale. Selon notre analyse, les familles dont les gros flux sont les plus présents ont une LRD plus élevée.

4. Il s'agit de considérer, à l'aide du logiciel Ethereal, le contenu de chacun des paquets de cette classe de trafic.

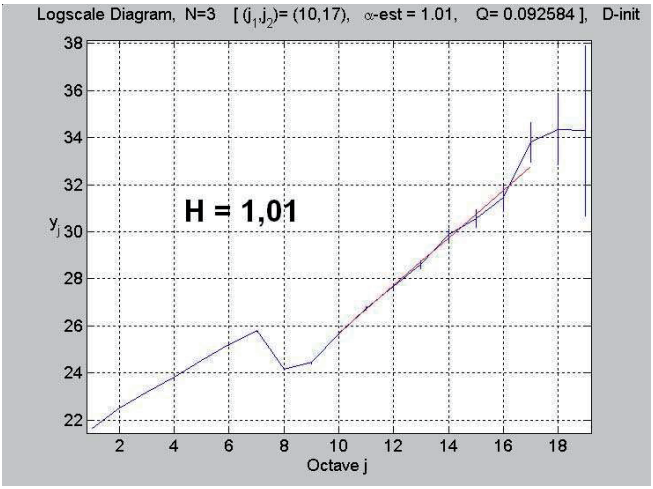


FIGURE 2.31 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit des flux http supérieur à 1 méga octets (granularité 1 ms)

Mais nous pouvons aussi observer que, dans cette trace, les différentes familles n'ont pas de différences radicales : toutes les familles montre une LRD élevée. Ainsi, il apparaît que la famille des flux n'est pas le facteur décisif si nous voulons trouver la "source" de LRD globale dans cette trace. Dans la partie suivante, nous allons donc analyser la LRD en considérant un niveau de raffinage supérieur : le type de comportement applicatif qui est à l'origine de la génération de ces différents flux.

2.2.2 Etude de l'impact de la taille des flux sur le niveau de LRD du trafic

Dans la partie précédente, nous avons essayé de trouver quelles familles introduisent le plus de LRD dans le trafic global. Le résultat sur la trace de Jussieu nous montre que la famille d'application n'apparaît pas comme un facteur capital pour la LRD. Toutes les familles engendrent une LRD élevée à cause des gros flux qu'elles contiennent. Pour certaines familles d'applications, les gros flux sont directement liés aux différentes utilisations du réseau. Nous prenons en compte par le terme "utilisation", la méthode à l'origine de la génération des flux dans le réseau. Il s'agit de considérer si les utilisateurs génèrent des données pendant un laps de temps très important ou non (définition de la notion de durée des flux ci-après) ou à l'inverse si le paramètre important n'est pas la durée du transfert mais plutôt la quantité d'informations qu'ils génèrent pendant celui-ci (définition de la notion de volume du flux dans la section ci-après). Dans cette partie, nous examinons donc l'impact des flux éléphants sur le trafic global dont la création est liée aux différentes utilisations du réseau. En particulier, nous examinons deux relations : LRD et durée des flux, ainsi que LRD et volume des flux. Nous voulons savoir, de la durée ou du volume des flux éléphants, quel est le facteur le plus

important pour la LRD. Il est à noter que les analyses sont toutes réalisées sur la trace Jussieu.

Impact de la durée des flux échangés

Nous avons isolé les flux de différentes durées et nous les avons regroupés en trois groupes : durées supérieures à 60 secondes (groupe 1), 600 secondes (groupe 2) et 3 000 secondes (groupe 3). La mesure du facteur de Hurst par la méthode des ondelettes montre que : l'augmentation des durées de flux n'augmente pas la LRD ! Dans la figure 2.32, la LRD du premier groupe est même plus élevée que celle des deux autres groupes (figures 2.33 et 2.34).

Pour mieux comparer les trois groupes, nous calculons, pour chaque groupe, la moyenne des débits de tous les flux :

– groupe 1 : Volume : 29.6 Go — Hurst : 0.826 — Moyenne des débits : 34.74 Koctets/S ;
– groupe 2 : Volume : 24.5 Go — Hurst : 0.803 — Moyenne des débits : 16.51 Koctets/S ;
– groupe 3 : Volume : 13.3 Go — Hurst : 0.814 — Moyenne des débits : 12.83 Koctets/S.

Il est à noter que :

– les débits moyens des groupes 2 et 3 sont inférieurs à celui du groupe 1 ;
– les débits moyens des trois groupes sont tous très faibles.

Ce résultat nous montre que, dans cette trace, la durée des flux n'est pas un facteur décisif pour la LRD du trafic, surtout quand le débit est faible.

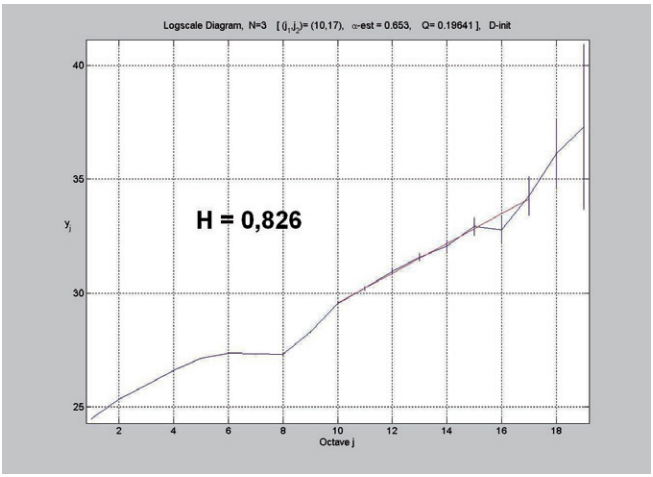


FIGURE 2.32 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit des flux http pour le groupe 1 — Hurst : 0.826

Impact du volume des flux échangés

Nous faisons les mêmes analyses pour deux groupes de flux en ne considérant maintenant que le paramètre volume :

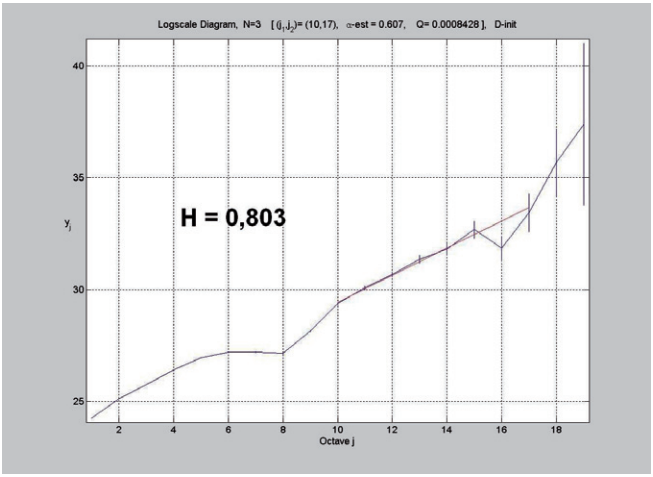


FIGURE 2.33 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit des flux http pour le groupe 2 — Hurst : 0.803

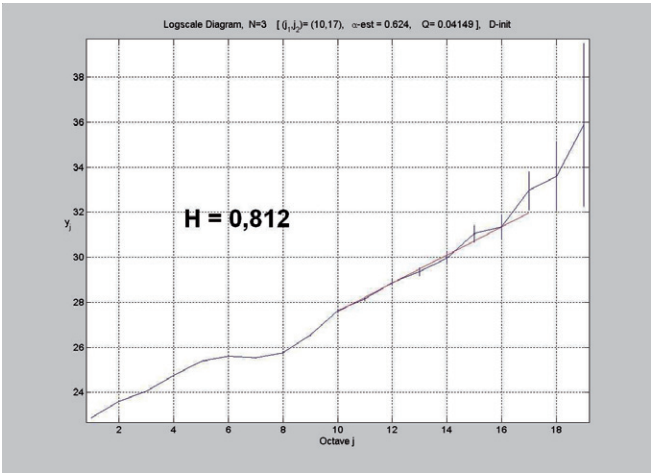


FIGURE 2.34 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit des flux http pour le groupe 3 — Hurst : 0.814

– groupe 4 : les flux dont le volume est supérieur à 10 Mega octets ;

	Volume (%)	Nombre de flux (%)	Niveau de LRD (paramètre de Hurst)
Trafic Web total	14,4 GOctets (100 %)	1.167.579 (100 %)	H = 0,905
Trafic des plus gros flux Web (taille > 1 MOctet)	5,31 GOctets (37,3 %)	679 (0,00058 %)	H = 1,011
Trafic des 100 plus gros flux Web	3,99 GOctets (27,7 %)	100 (0,000086 %)	H = 1,201

TABLE 2.3 – Détails de la contribution à la LRD du trafic en fonction de la taille des flux pour la famille Web

– groupe 5 : les flux dont le volume est supérieur à 100 Mega octets.

Nous observons que la moyenne des débits pour les groupes 4 et 5 (figures 2.35 et 2.36) est beaucoup plus élevée que celle des groupes 1, 2 et 3 (figures 2.32, 2.33 et 2.34) :

– groupe 4 : Volume : 26,7 Go — Hurst : 0,860 — Moyenne des débits : 464 Koctets / s ;
– groupe 5 : Volume : 19,5 Go — Hurst : 1,08 — Moyenne des débits : 327.5 Koctets / s.

Il est à noter que :

– les débits moyens des deux groupes sont beaucoup plus importants que dans le cas de la décomposition précédente ;
– le niveau de LRD et le volume semblent varier dans le même sens.

Selon les résultats précédents, il apparait donc que l'augmentation en volume des flux augmente bien le niveau de LRD du trafic.

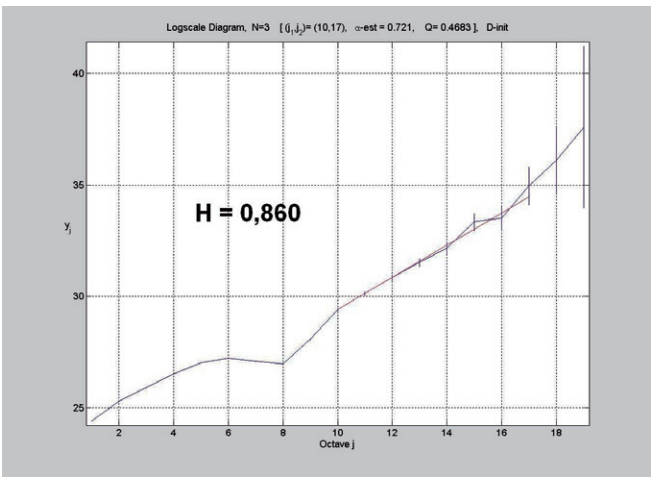


FIGURE 2.35 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit des flux http pour le groupe 4 — Hurst : 0,860

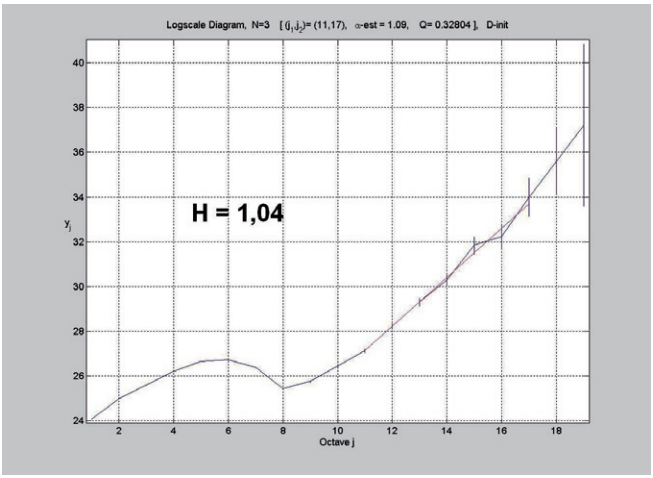


FIGURE 2.36 – Diagramme logscale obtenu par la méthode des ondelettes appliquée à la série de débit des flux http pour le groupe 5 — Hurst : 1,08

Conclusion pour les flux de longue durée et les flux de gros volume

Dans la trace de Jussieu, nous avons observé que :
- la durée des flux a beaucoup moins d'influence sur la LRD du trafic que le volume des flux ;
- le débit des flux a une influence certaine sur la LRD du trafic. En outre, plus le débit est faible, moins la durée n'a d'influence sur la LRD ;
- le volume des flux a plus d'influence sur la LRD du trafic. Plus les flux sont volumineux, plus la LRD du trafic est grande.

Origine des flux volumineux

Dans le paragraphe précédent, nous avons constaté que les flux volumineux sont les grands contributeurs à la LRD globale du trafic. Nous avons isolé les plus gros flux dans cette trace pour voir leurs origines. Leurs principales sources sont :

1. *Transfert de donnée via un tunnel sécurisé par le protocole SSH.*
   Normalement, les flux SSH et Telnet sont des flux longs au niveau des durées mais peu importants en volume. Mais l'utilisation d'un tunnel sécurisé (dans le cas du protocole SCP par exemple) peut générer des très gros flux en terme de volume à haut débit qui vont faire augmenter la LRD globale de cette partie du trafic.
2. *Téléchargement des gros fichiers par le protocole HTTP.*
   Le rôle du protocole HTTP est de télécharger les fichiers HTML et les images. Il n'est pas conçu pour télécharger des fichiers d'un gros volume (FTP est conçu pour ça). Mais

aujourd'hui, HTTP est de plus en plus utilisé dans les services web comme protocole de téléchargement anonyme. Plus les flux sont volumineux, plus la LRD est importante pour ce type de trafic.

3. *Téléchargement de gros fichiers par le protocole FTP.*
   Le protocole FTP est conçu pour le téléchargement avec identification d'utilisateur. Dans les réseaux d'enseignement et de recherche, FTP est le principal protocole de téléchargement.
4. *Echange de données entre les serveurs de news par le protocole NNTP.*
   Il est à noter que 90 % en volume de la famille Unknown est constitué de flux très volumineux échangés via le protocole NNTP.
5. *Echange de données entre les applications Peer-to-Peer.*
   Les applications P2P sont, dans la plupart des cas, utilisées pour échanger des fichiers de musiques et films dont les tailles varient de quelques Mega octets à quelques Giga octets. Les flux P2P sont normalement très gros et leurs LRD est très élevée. La proportion du trafic P2P n'est pas très grande dans cette trace, mais selon nos observations, le trafic P2P est une famille majoritaire dans les réseaux commerciaux.

2.3 Conclusion

Dans ce chapitre, nous avons étudié les différents paramètres qui sont à l'origine de la propriété de LRD dans le trafic. Nous avons d'abord observé que les flux éléphants (par définition des flux longs qui contiennent un grand nombre de paquets) étaient les plus gros contributeurs à ce phénomène dans le trafic Internet. Nous avons voulu dans un deuxième temps, affiner ce résultat en considérant les différentes applications qui étaient à l'origine de cette LRD parmi l'ensemble des flux éléphants. Cette étude n'a pas fait ressortir une application en particulier, étant donné que chaque application Internet est capable dans certaines configurations d'utilisation de générer des flux très longs. Nous avons donc dû, dans un troisième temps, introduire une métrique pour caractériser la longueur des flux : soit en terme de durée de connexion, soit en termes de volume d'information échangé. Nous nous sommes ainsi aperçus, que ce sont les flux les plus "gros" (en terme d'octets échangés) qui génèrent le plus de LRD. Le paramètre de durée du flux n'est que secondaire.

Ce résultat est très important étant donné que nous avons vu dans le chapitre précédent, le rôle primordial de TCP dans la génération de la LRD (par le fonctionnement en boucle fermée de son mécanisme de contrôle de congestion en particulier). En effet, plus les flux seront volumineux, plus le mécanisme TCP aura des difficultés à les transférer sans connaître de période de congestion. Ainsi, cette étude pose les bases en termes de caractérisation de la LRD nécessaire pour pouvoir dans la suite de ce manuscrit agir sur les mécanismes de contrôle de congestion de façon à mieux transférer ces flux volumineux et ainsi diminuer la caractéristique de LRD dans l'Internet. A l'aide des résultats de caractérisation du trafic apportés par nos études métrologiques, nous allons pouvoir, dans le chapitre qui suit, faire le point sur les actions à privilégier pour pouvoir améliorer la QdS du réseau et la régularité du trafic.

Chapitre 3

De l'utilisation des mesures de trafic pour améliorer les performances de l'Internet

Dans les chapitres précédents, nous avons successivement mis en évidence la présence de LRD dans le trafic Internet et nous avons démontré que cette caractéristique pouvait être utilisée comme une métrique pour quantifier le niveau d'oscillation du trafic Internet et par la même le niveau de performance et de QdS offerte par le réseau. Nous allons maintenant dans ce chapitre nous appuyer sur ce qui a été présenté jusqu'à présent pour introduire le cœur de la contribution de notre travail de thèse. Nous allons en particulier décrire comment toutes les connaissances apportées par nos différentes études métrologiques vont nous permettre de pouvoir proposer une nouvelle approche au niveau transport [1] pour améliorer la gestion des flux éléphants dans l'Internet et comment cette approche se positionne comme une alternative performante aux approches traditionnelles de gestion de la QdS dans l'Internet (cf. section 3.3). A l'issue du travail présenté dans le chapitre précédent, nous pouvons maintenant considérer un flux éléphant comme un flux échangeant un grand nombre de paquets mais dont le volume d'informations échangé au total est aussi très important (cf. le rôle des flux éléphants volumineux sur la LRD du trafic mis en évidence à la fin du chapitre 2). Il s'agit donc d'un flux constitué toujours d'un grand nombre de paquets, chaque paquet étant, de plus, volumineux en nombre d'octets.

Dès lors, il est nécessaire de définir plus précisément qu'elles sont ces approches qui sont traditionnellement suivies pour garantir la QdS réseau à l'heure actuelle dans l'Internet. C'est l'objet de la section 3.1.1 qui suit. A l'issue de cette section, nous définirons comment notre approche se positionne en complément de ce qui se fait actuellement dans l'Internet (cf. section 3.2.2). En effet, elle permet d'exploiter les outils de métrologie de l'Internet et l'ensemble des résultats qu'ils fournissent sur l'état et l'évolution du trafic et du réseau. La suite du chapitre sera relative à la présentation et à l'évaluation de notre approche de gestion des flux éléphants dans l'Internet.

1. Nous avons en effet mis en évidence dans le chapitre 1 que cette couche était parfaitement adéquate pour agir efficacement sur le niveau de variabilité du trafic.

3.1 La gestion de la QdS dans l'Internet

3.1.1 Les différentes métriques traditionnelles pour caractériser la QdS

Besoins en fiabilité

Les medias continus (audio et vidéo) ont comme caractéristiques d'être plus ou moins redondants. Ainsi deux images successives d'une transmission vidéo comportent généralement peu de différences. De cette redondance résulte la possibilité que des pertes d'information (image) soient acceptables du point de vue de l'utilisateur final. Il apparaît donc ici pour les medias les plus importants d'une application multimédia (audio et vidéo) une contrainte de fiabilité du transfert des données non plus totale mais partielle, la perte de certaines informations pouvant être acceptable. Notons cependant que l'expression d'une contrainte de fiabilité partielle est à coupler avec la façon dont sont codées les données audio et vidéo. En effet, certains codages (MPEG par exemple) introduisent une dépendance entre les images qui peut rendre indécodables plusieurs images consécutives en cas de perte de l'une d'entre elles, plus importante que les autres. A l'inverse, un codage de type M-JPEG n'introduisant aucune dépendance entre les images, une contrainte de fiabilité exprimée en termes d'un pourcentage maximum de pertes admissibles et d'un nombre maximum de pertes consécutives s'avère alors valide.

Besoins temporels

Les applications de diffusion différée de medias continus traitent leurs données de la même façon que s'il s'agissait de medias discrets. Deux types d'applications multimedias présentent des contraintes temporelles : les applications de diffusion en temps réel de medias continus et les applications multimédias interactives. Les contraintes temporelles s'expriment généralement par le biais de deux paramètres : le délai de transit des données et la gigue.

- *Délai*
  Pour les applications interactives (telle que la visioconférence), et à un degré moindre pour les applications de diffusion en temps réel (telles que le streaming audio ou vidéo), afin que la communication se déroule comme si elle avait lieu localement, il faut que les données soient transmises en un temps inférieur au seuil de perception humain lié au media considéré. Il apparait ainsi une contrainte sur le délai de bout en bout du transfert des données.
- *Gigue*
  Les medias continus (tels que l'audio et la vidéo) présentent des contraintes temporelles dues à leur caractère isochrone. Ces contraintes s'expriment en terme de régularité dans l'arrivée des données (on suppose que la source des données émet à un débit correspondant au débit idéal de présentation). Cette régularité s'exprime par une contrainte sur le temps inter-arrivées des données, c'est-a-dire sur la différence entre les dates d'arrivée de deux données successives. Cette contrainte est appelée la "gigue". La date d'arrivée d'une donnée étant calculée par :

$$t_{\text{réception}} = t_{\text{émission}} + dt_{min} + \delta dt \tag{3.1}$$

  où :
  - dt_{min} désigne le temps de transmission optimal (sans attente dans le réseau) ;

– δdt désigne le temps d'attente dans le réseau.
On peut alors exprimer la gigue par :

$$t_{\text{inter réception}} = t_{\text{inter émission}} + \delta dt_2 - \delta dt_1 \tag{3.2}$$

où
– δdt_1 et δdt_2 représentent les temps d'attente dans le réseau pour deux données successives.

Besoins en débit

Les besoins des applications en termes de débit sont très variables. Certaines, comme les applications Web ou mail ne requièrent que quelques KiloOctets de bande passante pour les flux qu'elles échangent. D'autres, au contraire, sont beaucoup plus exigentes. Bien sûr, c'est cette dernière catégorie qui tend à se généralier car de plus en plus d'applications récentes nécessitent une bande passante importante. On peut citer les applications de diffusion en temps réel comme par exemple les chaînes de télévision sur Internet. Dans ce dernier cas, il est d'ailleurs primordial de pouvoir fournir le service le plus stable et le plus régulier possible de façon à ce que l'utilisateur à l'extrémité du réseau reçoive son flux multimédia avec un niveau de qualité le plus régulier possible.

3.1.2 Les approches traditionnelles visant à garantir la QdS dans l'Internet

Afin de répondre aux besoins des applications à QdS, des études ont d'abord été entreprises sur les protocoles de transport afin d'en augmenter les performances et les fonctionnalités. Dans un deuxième temps, des études ont été menées au niveau IP afin de fournir aux paquets véhiculés un service différent (et meilleur) que le best effort actuel, qui n'offre aucune garantie. Dans un premier paragraphe, nous présentons tout d'abord les propositions issues de deux groupes de travail de l'IETF pour offrir des services IP améliorés : le groupe de travail IntServ et le groupe de travail DiffServ ; nous présentons ensuite les nouveaux protocoles de Transport SCTP et DCCP conçus pour étendre les fonctionnalités et/ou les services des protocoles TCP et UDP.

Couche IP : IntServ

Le groupe IntServ propose d'offrir des garanties de QdS par flux et a défini deux types de services en plus du best effort : le CL (Controlled Load) et le GS (Guaranteed Service). Le CL propose un service de bout en bout exprimable de façon qualitative en termes de bande passante : il assure que la transmission se fera comme sur un réseau peu chargé (pas de congestion). Le GS propose un service exprimable de façon quantitative en terme de bande passante et de délai de transit maximal : il garantit que tous les paquets d'un même flux arriveront (aux erreurs dues au médium physique près) en un temps borné défini par l'application qui utilise le service. Afin de réserver les ressources dans le réseau (bande passante et mémoire tampon) nécessaires à l'obtention de ces services, l'approche IntServ nécessite l'utilisation d'un protocole de réservation de ressources : RSVP [28]. Ce protocole propage la demande de réservation à tous les routeurs sur le chemin des données (de façon dynamique afin de s'adapter aux changements de route). Chaque routeur est en charge d'accepter ou non la réservation en

tenant compte des ressources disponibles localement et de la caractérisation du trafic fournie avec la réservation.

La limite principale de cette approche concerne la surcharge induite pour les routeurs traversés par ce type de flux. En effet, chacun d'eux doit stocker une information relative aux machines de bout en bout qui échangent ces données pour pouvoir les traiter comme prioritaires par rapport au reste du trafic. Avec le grand nombre de flux émis en temps réel dans l'Internet à l'heure actuelle, la charge CPU et les limitations en mémoire RAM des routeurs deviendraient rapidement impossibles à supporter si l'approche IntServ était déployée à large échelle, ce qui entrainerait une rupture du service et un possible effondrement des différents routeurs traversés par ces flux. Cette limitation porte sur l'impossibilité d'une mise à l'échelle du service IntServ dans l'ensemble de l'Interet. A l'heure actuelle, IntServ n'est ainsi déployé qu'à l'échelle d'un seul domaine et quand ce dernier comporte un nombre de noeuds limité. Pour pallier cette limitation technique, une approche parallèle (c'est à dire sans signalisation) a été proposée, elle est détaillée ci-après.

Couche IP : DiffServ

Il s'agit de l'approche développée par le WG DiffServ. L'idée de base des solutions DiffServ est de fournir une QdS différenciée aux paquets traversant un réseau tout en repoussant (le plus possible) la complexité du traitement en bordure du réseau afin de ne pas surcharger le cœur du réseau. De plus, afin d'éviter le problème de passage à l'échelle inhérent aux solutions IntServ, le choix a été fait de traiter un nombre limité d'agrégats (paquets IP n'appartenant pas nécessairement à un même flux) plutôt que des flux individuels. De plus, toujours pour contourner le problème du passage à l'échelle, l'approche DiffServ ne génère aucun trafic de signalisation pour éviter de surcharger les routeurs de cœur. Dans les paragraphes suivants, nous présentons d'abord une notion importante pour les solutions DiffServ, la notion de domaine, puis nous décrirons les principes généraux de ces solutions, avant de présenter l'architecture DiffServ de fourniture de QdS au niveau IP.

- *La notion de domaine*
  Tel que nous l'avons vu dans le chapitre précédent, l'Internet est constitué d'une interconnexion de différents réseaux. Cependant plusieurs de ces réseaux sont souvent rassemblés sous une même autorité administrative (par exemple, dans les grandes entreprises, les centres de recherches, les universités...) ; [25] désigne par domaine, un ensemble de nœuds (hôtes et routeurs) administrés de façon homogène. Dans un domaine, on distingue les nœuds internes et les nœuds frontières : les premiers ne sont entourés que de nœuds appartenant au domaine alors que les seconds sont connectés à des nœuds frontières d'autres domaines.
- *SLA : Service Level Agreement*
  Pour un utilisateur (une personne ou une organisation qui loue les services d'un ISP pour accéder au réseau), l'utilisation d'une architecture à services différenciés implique la signature d'un contrat avec le fournisseur d'accès Internet : ce contrat s'appelle le SLA. Contrairement à ce qui se passe avec RSVP, ce contrat est signé avant toute connexion au réseau, et non à l'établissement d'une quelconque session (établir une session RSVP revient en effet à passer un contrat avec les routeurs intermédiaires, qui garantissent certaines propriétés du transport de données tant que le trafic respecte un certain profil). Ce contrat peut contenir les informations suivantes :

  - le trafic que l'utilisateur peut injecter dans le réseau fournisseur (en termes de volume de données, de débit moyen, d'hôtes sources ou destinations...)
  - les actions entreprises par le réseau en cas de dépassement de trafic (rejet, surtaxe...),
  - la QdS que le fournisseur s'engage à offrir au trafic généré ou reçu par l'utilisateur (ou les deux). Celle-ci peut s'exprimer notamment en termes de délai, de bande passante, de fiabilité ou de sécurité. Pour le moment, seuls des contrats statiques, c'est-à-dire peu susceptibles de changer dans le temps, sont étudiés. Après signature du SLA, l'utilisation des services DiffServ est transparente pour l'utilisateur, l'architecture DiffServ ayant été conçue pour fonctionner avec les applications déjà existantes.
- *PHB : Per Hop Behavior*
  Du point de vue du réseau, l'implantation de l'architecture nécessite un découpage du réseau en domaines (au sens domaine Internet). Tous les nœuds (hôtes et routeurs) d'un domaine implémentent les mêmes classes de service et les mêmes comportements vis-à-vis des paquets des différentes classes (PHB). Un comportement inclut le routage, les politiques de service des paquets (notamment la priorité de passage ou de rejet en cas de congestion) et éventuellement la mise en forme du trafic entrant dans le domaine. Les nœuds internes ne doivent pas conserver d'états en mémoire (contrairement à ce qui apparaît dans l'architecture IntServ), ils ne font que transmettre les paquets selon le comportement défini pour leur classe. Ce comportement est donc purement local et ne tient pas compte d'un état global du réseau. Les nœuds frontières se chargent de marquer les paquets selon le code réservé à chaque classe, comme nous allons le voir dans la partie suivante. Dans la documentation, les termes de caractérisation du contrat sont volontairement informels afin de rendre la spécification la plus ouverte possible.
- *Architecture à services différenciés*
  L'architecture proposée par le groupe DiffServ [25] est basée sur un modèle simple dans lequel le trafic entrant dans un réseau est conditionné, puis assigné à une classe de comportement, au niveau du point d'entrée de ce réseau. Chaque classe est identifiée par un code unique : le DSCP. A l'intérieur du réseau, les paquets sont acheminés selon le comportement associé au code de la classe à laquelle ils appartiennent. Examinons à présent les éléments clefs dans un environnement multi domaines du réseau à services différenciés. Les services différenciés sont mis en œuvre grâce à un conditionnement du trafic entrant et un acheminement des paquets selon un comportement par nœud (PHB). Le conditionnement du trafic intervient aux nœuds frontières d'un domaine (à l'entrée de celui-ci) afin d'assurer que le trafic entrant soit conforme aux règles spécifiées dans le SLA et de le préparer aux traitements par PHB à l'intérieur du domaine. On distingue deux cas d'entrée d'un paquet dans un domaine : quand le paquet passe par un nœud frontière d'un domaine vers un nœud frontière d'un autre domaine et quand le paquet est généré par l'hôte source. En fait, on réunit les deux cas en considérant l'hôte source comme un domaine à un seul nœud (donc obligatoirement un nœud frontière). Le conditionnement est réalisé par un conditionneur de trafic, pouvant contenir les éléments suivants :
  - le classifieur sélectionne des paquets dans le trafic en se basant sur le contenu d'une partie de leur en-tête (cette classification peut être faite sur le DSCP seulement : classification par agrégation des comportements), ou sur n'importe quelle combinaison d'un ou plusieurs champs de l'en-tête du paquet (adresse source ou destination, type de protocole), et de l'en-tête de niveau transport (TCP ou UDP) tels que les numéros

de port source ou destination. Le rôle d'un classificateur est d'extraire certaines caractéristiques des paquets et de les transmettre aux autres éléments du conditionneur ;
– le contrôleur détermine les paquets qui respectent le profil associé à leur classe et ceux qui sont hors profil. Selon qu'ils sont dans le profil ou hors profil, les paquets sont traités différemment (en fonction de ce qui est spécifié dans le SLA). Les actions peuvent être :

  1. *pour un paquet respectant le profil :*
     – de le laisser passer sans autre conditionnement (cas où les domaines utilisent deux ensembles de comportements et de marquage identiques) ;
     – de marquer les paquets selon un nouveau DSCP (si le paquet n'était pas encore marqué ou si les deux domaines utilisent des marquages et des comportements différents) ;
  2. *pour les paquets hors profil :*
     – le retardement de ceux-ci jusqu'à ce qu'ils respectent le profil (shaping) ;
     – le rejet des paquets ;
     – le marquage avec un DSCP spécial ;
     – le déclenchement d'une action de compte rendu.

– le rôle du marqueur est donc de marquer l'en-tête des paquets qui lui sont transmis avec le DSCP correspondant à leur classe s'ils sont dans le profil ou selon un DSCP spécifique sinon ;
– le rôle de l'écarteur est de retenir des paquets hors profil jusqu'à ce qu'ils soient dans le profil. Un écarteur possède une mémoire de taille finie : des paquets sont donc détruits s'il y a saturation de celle-ci.

Les limites des solutions DiffServ ont principalement trait :
– au manque de finesse dans le paramétrage des services, dû à la nécessité de limiter le nombre d'agrégats dans le réseau. En effet, seuls trois types de service différents sont définis dans l'approche DiffServ, on peut néanmoins imaginer de compléter le nombre de classes de service pour affiner l'approche, il n'est pas possible de toutefois multiplier à l'infini les classes étant donné que le problème de surcharge des routeurs intermédiaires se rencontrerait de la même façon qu'il a été mis en évidence pour l'approche IntServ.
– à la difficulté d'un accord entre administrateurs des différents domaines. En effet, les différents opérateurs sont en concurrence les uns envers les autres. Dès lors, il n'est pas dans leur intérêt de rendre transparent leur politique de gestion des services fournis à leurs clients. Ainsi, fournir une même qualité de service de bout en bout de l'Internet (i.e. en traversant différents domaines) s'avère à l'heure actuelle une tâche délicate voire même impossible.
– à la limitation de la réactivité des services qui sont proposés. En effet, toutes les classes de service reposent sur la définition de contrats de SLA eux mêmes basés sur des métriques de QdS statiques (délai, pertes...) et ne prennent pas en compte le caractère variable des caractéristiques du trafic Internet, une solution à cette limitation qui sera abordée dans la section 3.2 repose sur l'utilisation de technique de mesure temps réel dans le résau pour pouvoir réagir au plus tôt et de façon la plus précise aux évolutions se produisant dans le réseau.

Couche Transport (étendue)

De nombreux travaux se sont attachés à proposer de nouveaux protocoles de Transport pour enrichir les fonctionnalités et/ou les services des protocoles UDP et TCP. Les trois paragraphes suivants présentent deux de ces protocoles, SCTP et DCCP, actuellement développés à l'IETF.

– *SCTP*

Le protocole SCTP [115], est un protocole de transport à fiabilité totale se déployant sur un service paquet de niveau réseau sans connexion, offert par exemple par le protocole IP. Il est unicast et orienté session, une session étant définie comme une association établie entre deux hôtes. Dans le cas où un hôte dispose de plusieurs adresses IP, les adresses sont échangées lors de l'établissement de la session (on appelle "multi-homing" le fait que plusieurs adresses IP puissent correspondre à une session). Un mécanisme de contrôle d'erreur est implémenté dans SCTP et permet de détecter les pertes, la rupture de séquences, la duplication ou la corruption de paquets. Un schéma de retransmission est utilisé pour corriger ces erreurs. SCTP utilise le principe de SACK [78] pour la confirmation de la réception des données. Les retransmissions sont faites après expiration d'un timer ou sur interprétation du SACK. Au contraire de TCP, SCTP est orienté message (ce qui le rapproche par cet aspect de UDP). Chaque paquet contient un en-tête commun et une partie donnée (contenant soit des données utilisateurs soit des données de contrôle). En fait, bien qu'il soit orienté message, plusieurs données peuvent être contenues dans le même paquet, mais seront délivrées à l'application avec le format des messages initiaux. SCTP offre un service de multiplexage/démultiplexage entre flux : une application multimédia peut être découpée en plusieurs flux pouvant avoir chacun des schémas de remise des données différents. C'est donc un protocole d'ordre total au sein d'un flux et n'offrant aucune garantie sur l'ordre inter-flux, ce qui permet au protocole de délivrer les données d'un flux même si des pertes ou des déséquencements sont détectés sur un autre flux. Le type de contrôle de flux et de congestion est négocié à l'établissement de la connexion. Ces mécanismes sont construits sur la base des algorithmes de TCP : le récepteur informe l'émetteur de sa taille de buffer et la taille de la fenêtre de congestion est contrôlée au cours de la connexion SCTP. Les mécanismes de "slow-start, congestion avoidance, fastrecovery et de fast-retransmit" sont les mêmes que ceux de TCP mais ils utilisent les paquets SCTP comme unités d'acquittement. SCTP peut intéresser les applications désirant un service de transport à ordre partiel. Cependant, le fait que SCTP offre un service totalement fiable entraîne une incompatibilité avec les applications multimédias ayant des contraintes en terme de débit, de délai ou de gigue. Une extension [116] de SCTP permet d'offrir un service à fiabilité partielle temporisée. Ce dernier concept signifie que l'utilisateur peut spécifier une durée de vie à son message. Mais ce service n'est pas adapté aux applications à temps contraint présentant des données applicatives spécifiques, comme par exemple les données des images I, P et B d'un flux vidéo MPEG.

– *DCCP*

Le protocole DCCP [73], offre un service de transport non fiable pour des flux en datagrammes (donc type UDP) mais intégrant plusieurs mécanismes de contrôle de congestion, ce qui permet aux applications utilisant habituellement UDP de ne pas avoir à implémenter le leur. Le but de DCCP est d'offrir l'efficacité d'UDP à certaines applica-

tions tout en respectant les autres flux (TCP) du réseau. Les mécanismes de contrôle de congestion sont négociés pour les deux sens de la connexion entre les hôtes au moyen d'un identifiant appelé CCID. Plusieurs mécanismes sont disponibles, parmi lesquels un contrôle de congestion TCP-like utilisant une fenêtre de congestion et un algorithme TFRC TCP-Friendly Rate Control [68]. DCCP peut être utilisé par toutes les applications présentant des contraintes temporelles et qui sont capables de s'adapter aux fluctuations de débit imposées par les mécanismes de contrôle de congestion. Cela dit, même si DCCP apporte un plus par rapport à TCP en termes de mécanisme de contrôle de congestion (implémentation de TFRC), il reste encore basé sur une solution de bout en bout statique. Cette solution a été évaluée dans le premier chapitre de ce manuscrit dans lequel nous avons mis en évidence que même si la régularité du trafic était améliorée avec l'utilisation de TFRC, la performance globale restait légèrement en deça de ce que pouvait mettre en œuvre TCP dans le réseau. La solution pour améliorer à la fois la régularité du trafic et les performances obtenues dans le réseau repose sur l'utilisation d'informations temps réel sur l'état du réseau de façon à réagir au plus près aux fluctuations de bande passante disponible ou encore aux phénomènes de congestion se produisant dans le réseau. En effet, comme nous l'avons détaillé dans les chapitres précédents, la variabilité du trafic Internet est très importante, ce qui se traduit par une très grande dynamique des ressources disponibles qu'il faut prendre en compte pour permettre une meilleure gestion du réseau et de la QdS que l'on peut y mettre en œuvre. Ainsi, notre proposition reposant sur les principes précédemment énoncés va maintenant être présentée en détail dans la section suivante.

3.2 Actions à mener pour une amélioration de la gestion de la QdS dans l'Internet

3.2.1 Prise en compte des caractéristiques du trafic Internet actuel

Comportements oscillants et variabilité du trafic

Garantir la QdS consiste à fournir le service demandé dans toutes les circonstances, y compris les plus difficiles. Parmi ces dernières, le niveau de QdS dans l'Internet est particulièrement sensible à un grand nombre de «ruptures» dues à un volume important de trafics inattendus qui peut être légitime dans le cas de la diffusion d'un évènement populaire sur le réseau, une défaillance technique ou encore un comportement illicite d'un utilisateur du réseau (attaque de déni de service par exemple). Les ruptures de trafic incluent plus généralement tous les évènements qui peuvent provoquer un changement important dans les caractéristiques du trafic et qui peuvent affecter la QdS dans le réseau. Dans ce contexte, le développement de méthodologies pour permettre des mesures globales dans le réseau est devenu un enjeu important. Ces méthodes sont essentielles pour permettre de détecter et de réagir à ces ruptures.

Absence de stationnarité du trafic

Evidemment, ces variations importantes dans le profil du trafic ont un impact sur ses propriétés de stationnarité. Cette remarque justifie le développement de mécanismes efficaces permettant de garantir une QdS stable au cours du temps pour tous les utilisateurs. La

non-stationnarité du trafic, définie comme un changement dans la moyenne du débit, a donc des répercutions importantes sur la QdS du réseau. De plus, les études récentes sur le trafic Internet, ont mis en évidence une versatilité forte du trafic dont les caractéristiques sont à la fois très différentes d'un lien à l'autre et évoluent aussi très rapidement au cours du temps [114] : quelques exemples de ruptures que l'on peut rencontrer dans l'Internet sont présentées dans les figures 3.1, 3.2, 3.3 et 3.4. Lorsqu'on parle de ruptures, les plus simples concernent celles se produisant pendant la journée, la semaine ou encore le mois. En effet, sur la figure 3.1, on peut observer les fluctuations qui se produisent à différentes échelles temporelles et avec différentes amplitudes en fonction de la fenêtre d'intégration que l'on considère. Ces variations s'apparentent à des modifications régulières et relativement prévisibles dans le profil du débit réseau analysé. A l'inverse la figure 3.2, illustre l'apparition d'une variation non attendue sur un lien qui est le résultat du reroutage de près de 90 % du trafic suite à une intervention de maintenance sur le lien spécifiquement analysé : ce reroutage vers d'autres liens entraîne donc un effondrement du trafic utile pendant quelques heures. D'autres variations peuvent s'analyser sur des échelles temporelles beaucoup plus courtes (cf. figure 3.3), on appelle généralement ces phénomènes, "flash-crowd" [2], qui correspondent à une augmentation très importante mais relativement courte dans le temps du trafic légimite, par exemple dans le cas d'un téléchargement massif d'un évènement populaire sur le réseau. Enfin, un dernier type de rupture que l'on peut mettre en évidence dans le réseau est détaillé dans la figure 3.4. Il s'agit de la manifestation d'une attaque de DdS sur un lien de l'Internet. La notion d'attaque sera abordée en détail dans le dernier chapitre de cette thèse. Néanmoins, sans rentrer dans les détails, on peut observer que pour mesurer une rupture dans le trafic il faut parfois ne pas considérer une métrique standard (ici le débit) mais des métriques secondaires (ici le nombre de paquets par unité de temps) de façon à mettre en évidence la variabilité du trafic qui peut être masquée lorsqu'on ne considère que le niveau octet mais qui apparaît par contre au niveau paquet. Ce phénomène est dû à la typologie de l'attaque utilisée qui génère un grand nombre de paquets de petite taille, ce qui ne se traduit pas au niveau octet par un changement du débit au cours du temps mais par contre modifie très fortemement le profil du nombre de paquets au cours du temps.

Notre proposition qui est présentée dans la suite de ce chapitre introduit ainsi une nouvelle solution pour gérer le réseau en prenant en compte les propriétés de non-stationnarité du trafic, les fortes variations que l'on peut mesurer sur un même lien ou au contraire les caractéristiques très différentes du trafic d'un lien à l'autre observables dans le réseau et que nous venons de détailler dans cette section.

Topologie complexe de l'Internet

Un autre problème important est relatif à la difficulté de mise en œuvre des mécanismes de QdS de bout en bout face à l'hétérogénéïté de la topologie et de la structure administrative de l'Internet. Ce point est illustré par la figure 3.5. L'Internet est généralement défini comme une interconnexion de réseaux. C'est évidemment vrai mais incomplet. En effet, l'Internet doit être de plus en plus vu comme un réseau global découpé en différents domaines ou Systèmes Autonomes (SA), indépendant administrativement et gérés de façon autonome. Chaque réseau de chaque SA propose donc différents niveaux de service et de QdS à ses usagers. Ce phénomène devient de plus en plus important avec la prolifération de nouveaux types de réseaux

2. La traduction française "foule subite" n'est pas très usitée.

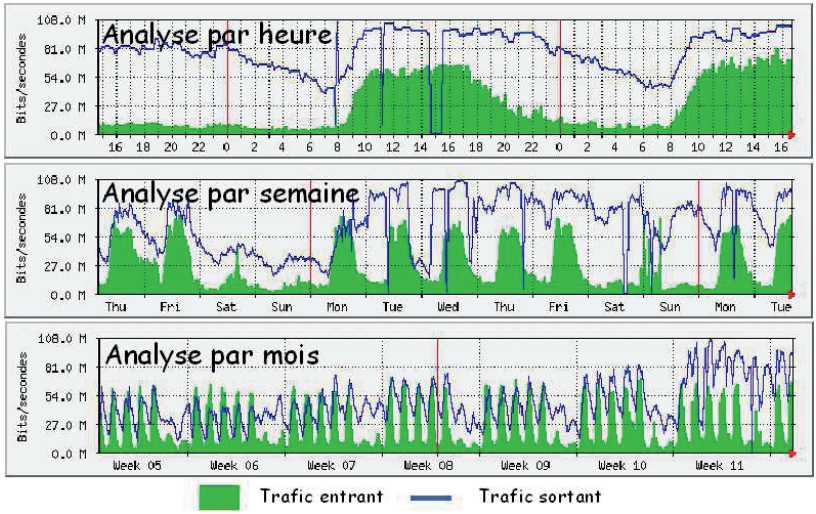


FIGURE 3.1 – Exemples de ruptures quotidienne, hebdomadaire ou mensuelle dans le trafic Internet

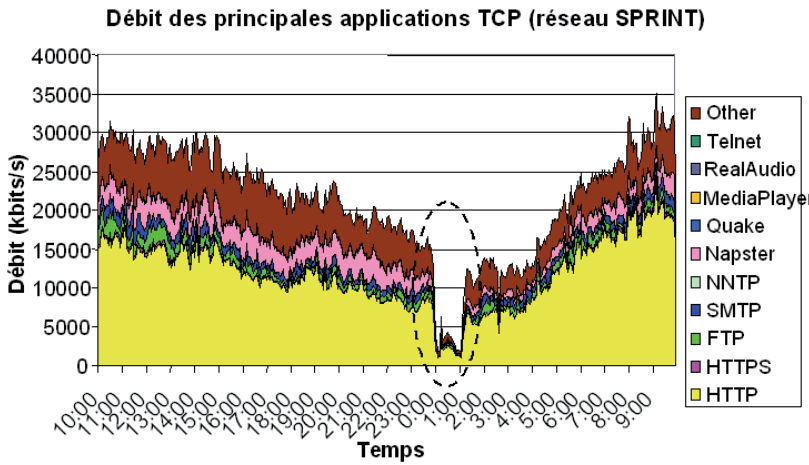


FIGURE 3.2 – Illustration d'une rupture dans le trafic produite par la fenêtre de maintenance d'un opérateur télécom

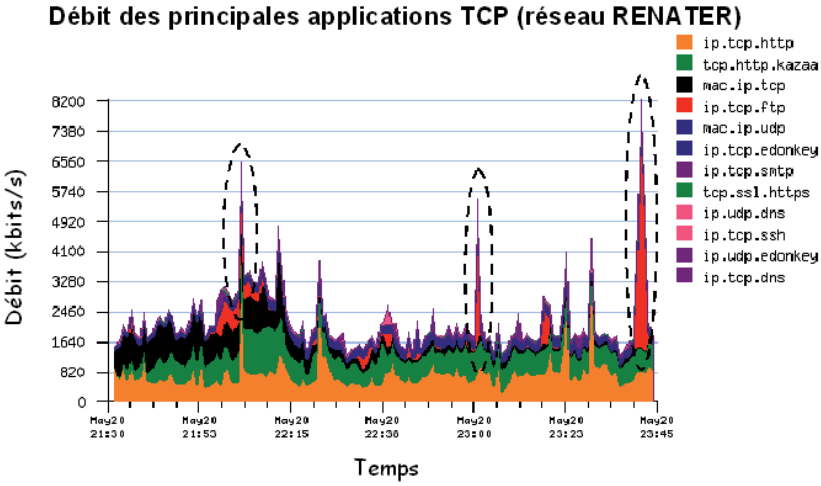


FIGURE 3.3 – Illustration d'une rupture dans le trafic générée par une foule subite ("flash crown")

reposant sur des technologies sans-fil (WIFI, GPRS, UMTS, etc.) ou satellite qui proposent des niveaux de QdS très différents. Dans un tel contexte, assurer de la QdS de bout en bout est particulièrement ardu étant donné que le niveau de service obtenu par un utilisateur sera minoré par le domaine ayant les prestations les plus basses parmi tous les domaines traversés sur le chemin entre la source et la destination. En particulier, les liens de "peering" sont souvent sous-dimensionnés et à l'origine d'une diminution importante de la QdS et des performances dans la communication de bout en bout [88]. Dans un tel contexte, améliorer la QdS de bout en bout devrait nécessiter la mise en place d'une infrastructure globale et de procédures de gestion commune pour éviter les différences entre SA. Une telle hypothèse ne peut que rester utopique tant la compétition «économique» entre opérateurs et fournisseurs d'accès est importante. Ainsi, trouver un accord global entre tous ces acteurs pour définir comment échanger le trafic Internet est totalement invraisemblable. La QdS de bout en bout est donc toujours un problème à considérer avec une vision multi-domaine.

3.2.2 L'objectif d'amélioration de la QdS et des performances du réseau

Il est apparu clairement dans le chapitre précédent que les gros flux sont les principaux contributeurs à la LRD global du trafic. Pour réduire l'impact des gros flux sur le niveau de performance du réseau, nous pouvons imaginer les propositions suivantes.

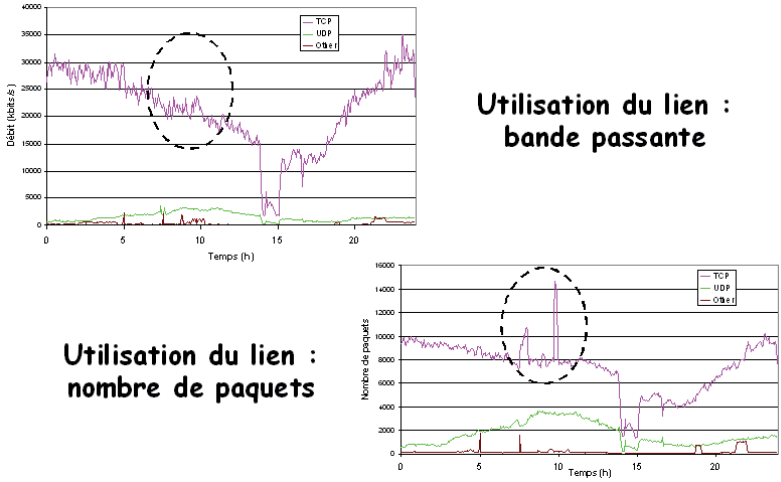


FIGURE 3.4 – Illustration d'une rupture dans le trafic produite par une attaque de déni de service

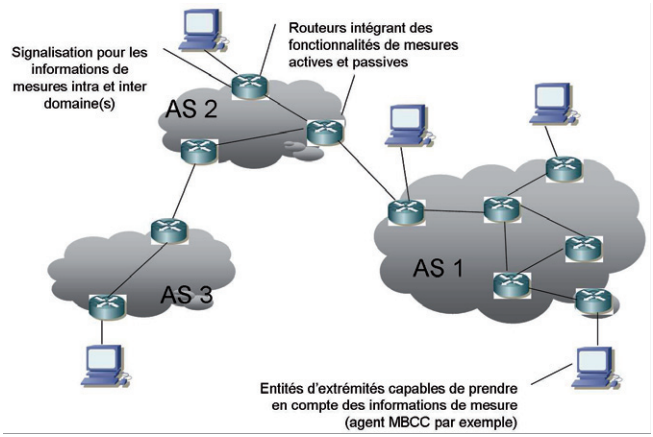


FIGURE 3.5 – Illustration de l'hétérogénéïté de la topologie actuelle de l'Internet

Réduire le nombre des gros flux au niveau applicatif

Comme nous avons montré dans le précédent chapitre, le trafic E-donkey a une LRD moins importante que le trafic Kazaa parce que l'applications E-donkey utilise plusieurs flux au lieu

d'un flux unique pour transférer un fichier. Si toutes les applications P2P populaires adoptent cette approche, la LRD dans le trafic P2P va être réduite considérablement, de même que la LRD du trafic global dans les réseaux commerciaux. Malheureusement, il est clair qu'on ne peut pas modifier toutes les applications existantes qui génèrent des flux sur le réseau. Par exemple, les applications multimédia ou de services interactifs ne peuvent pas adopter cette approche. Ainsi, celle-ci reste destinée principalement aux applications Peer-to-Peer qui génèrent une très grande proportion de trafic sur les réseaux commerciaux. Dès lors, réduire les gros flux est une approche intuitive mais plutôt limitée voire même impossible dans l'Internet actuel et ce, à cause des contraintes de fonctionnement de la majorité des applications existantes. Il est donc nécessaire d'apporter une solution en considérant un niveau de réduction de la LRD plus bas dans les couches de l'Internet, le niveau transport en particulier, comme nous allons le présenter ci-après.

Améliorer le mécanisme de TCP au niveau protocolaire

Les mécanismes de TCP sont à l'origine de la majorité de la LRD du trafic réseau. L'amélioration des mécanismes de TCP changera complètement les caractéristiques du trafic de l'Internet. L'objectif est de trouver un nouveau mécanisme de contrôle de congestion qui génère un trafic plus régulier et lisse. Dans le cadre des projets METROPOLIS et E-NEXT, des études présentées dans le chapitre précédent ont été faites pour mesurer l'impact du protocole TFRC [56] sur les caractéristiques du trafic. TFRC est un nouveau mécanisme de contrôle de congestion visant à offrir un trafic lisse pour les applications de "streaming". Un résultat est développé à la fin du chapitre 1 et présenté dans le papier [Lar03a]. Il est montré que TFRC réduit considérablement la LRD du trafic global et offre un trafic beaucoup moins oscillatoire. Bien que TFRC ne soit pas conçu, initialement pour remplacer l'ensemble des transferts réalisés par TCP (à l'origine uniquement les transferts multimédia), il nous montre que modifier le mécanisme de contrôle de congestion de TCP pour réduire la LRD est une démarche tout à fait efficace.

Lisser le profil du trafic des flux éléphants

En s'inspirant des résultats de la caractérisation métrologique présentée au cours du chapitre 2 ainsi que des deux points précédents, il semble donc important d'apporter plus de stabilité aux flux "éléphants" (pour rappel nous considérons les flux générant un grand nombre de paquet volumineux en taille). En effet, ils vont engendrer des phénomènes de dépendance très longue mémoire entre tous les paquets d'un même flux, à cause de leur taille. Nous proposons donc pour optimiser la QdS perçue par les utilisateurs d'utiliser une typologie différente de ce qui est fait habituellement, et de choisir une différenciation des traitements pour les flux courts et les flux longs, en mettant en place un service "éléphant" qui s'adressera à l'ensemble des très longs flux volumineux (comme nous les avons définis au début de ce chapitre). Il s'agit donc pour ces flux de lisser le débit au cours du temps (c'est à dire supprimer le comportement oscillatoire observable sur de grandes échelles de temps). En effet, ce type de flux correspond de plus en plus à des flux ayant des contraintes temporelles, générés par des applications comme des visioconférences, de la vidéo à la demande, de la téléphonie sur IP, c'est à dire des applications qui ont besoin de services dits garantis, comme le service EF de DiffServ par exemple. Pour accroître la régularité des flux "éléphants", nous allons tester dans le chapitre suivant l'apport d'un nouveau mécanisme de contrôle de congestion (MBCC) qui a été conçu

pour fournir un service adapté aux applications multimédia, en essayant d'éviter les variations brutales de débit qui surviennent avec TCP lors des reprises de pertes. Cette approche sera bien sûr rendue possible par les nouvelles connaissances relatives aux caractéristiques du trafic apportées par les différentes études que nous venons de présenter dans le chapitre précédent.

Une nouvelle approche pour améliorer la QdS : lisser le comportement des flux éléphants

Notre contribution se positionne comme une solution complémentaire des architectures précédemment énoncées (IntServ et DiffServ). Il s'agit de proposer une meilleure QdS à l'ensemble des flux transitant sur un même support, sans faire d'hypothèse sur un quelconque traitement privilégié que l'on accorderait à une partie du trafic (flux EF par exemple dans l'approche DiffServ). En effet, nous souhaitons par notre approche optimiser la QdS pour la plus grande portion du trafic (les éléphants représentent en effet plus de 90 % du trafic en volume, cf. chapitre 2) et non pour quelques flux identifiés dans le trafic. Les travaux de recherche récents en réseau, basés sur l'utilisation de techniques de métrologie, ont permis d'améliorer les connaissances sur le trafic Internet. De plus, ces études ont montré que la variabilité du trafic est problématique pour la stabilité et les performances du réseau [97]. En particulier, nous avons vu dans le chapitre précédent que les applications P2P, utilisées massivement pour échanger des volumes d'information très importants (albums musicaux ou films) sont en train de modifier les caractéristiques du trafic Internet, dégradent ses performances et la QdS qu'il offre [Owe04b]. Ces applications, par les oscillations à long terme qu'elles induisent, créent de la dépendance longue mémoire (LRD) dans le réseau. Cette LRD ainsi que ces oscillations sont très néfastes pour la QdS du réseau étant donné qu'elles provoquent des phénomènes importants de congestion et un niveau de service très instable pour les utilisateurs [138]. De plus, les oscillations à long terme créent des propriétés de non-stationnarité dans le trafic, la valeur moyenne du trafic changeant fréquemment de façon significative.

Ainsi, nous sommes convaincus qu'il est possible en définissant un mécanisme de contrôle de congestion adaptatif (i.e. capable de connaître précisément l'état de congestion du réseau à un moment donné) de réduire la variabilité du trafic TCP actuel et de fait, d'améliorer la QdS perçue de bout en bout par un plus grand nombre d'utilisateurs (en termes de perte, de gigue ou encore de RTT). Cette modification du mécanisme de contrôle de congestion utilisé dans l'Internet sera rendu possible par l'utilisation d'une architecture de mesure distribuée capable de renseigner les différents acteurs du réseau (hôtes d'extrémité, de bordure ou de cœur de réseau) sur l'état et l'évolution des caractéristiques du trafic au cours du temps. Notre contribution va maintenant être détaillée par la présentation de notre approche pour la gestion du réseau s'appuyant sur les mesures.

3.3 L'approche MBN pour la gestion des réseaux de l'Internet

3.3.1 Présentation de l'approche MBN

Dans cette section, on se propose donc d'exploiter les techniques de métrologie réseau en temps réel afin de définir une architecture orientée mesure (appelée dans la suite MBA pour Measurement Based Architecture) et l'ensemble des mécanismes nécessaires permettant de mieux adapter les mécanismes du réseau aux fréquents changements mesurés dans le trafic

(appelée MBN pour Measurement Based Networking). Un composant essentiel de cette architecture est le protocole MSP (Measurement Signaling Protocol) qui permet d'informer en temps réel tous les acteurs du réseau de l'évolution des caractéristiques du trafic. Ce protocole devra réaliser un compromis entre le besoin en rapidité pour permettre de délivrer des informations sur l'état du réseau en temps réel et le souci d'éviter sa surcharge afin de lui permettre ainsi de fonctionner dans des réseaux de grande taille (section 3.3.2). Les exemples d'applications de l'approche MBN sont nombreux. Dans le présent chapitre, nous allons démontrer comment MBN et le nouveau mécanisme de contrôle de congestion orienté mesures (appelé dans la suite MBCC pour Measurement Based Congestion Control) qui exploite cette approche peut se poser comme solution au problème de variabilité du trafic Internet présenté dans les deux premiers chapitres. Il devra être capable de limiter le nombre de congestions et de pertes dans le réseau mais aussi d'améliorer la régularité du trafic et l'utilisation des ressources du réseau. Dès lors, la partie 3.3.2 présente les principes de fonctionnement de MBCC et de MSP et leur évaluation conjointe en simulation (NS-2). Cette dernière se déroule en deux parties. La première étape (section 3.3.3) permettra de définir quels sont les paramètres optimaux de MSP pour concilier performance (vitesse), passage à l'échelle et absence de surcharge du réseau. La deuxième (section 3.3.3) démontre les avantages de MBCC par rapport aux mécanismes de contrôle de congestion traditionnels comme ceux de TCP. Au final, la partie 3.4 conclut cette section et introduit les évolutions à venir pour l'approche MBN.

Définition de l'architecture MBA s'appuyant sur des mesures

Conscient des différents aspects relatifs à la problématique de l'amélioration de la QdS dans l'Internet ainsi que les problèmes liés à la variabilité et à la dynamicité du trafic et des ressources disponibles dans le réseau, il est aisé de comprendre qu'une solution statique optimale pour l'ensemble des connexions [3] n'est pas possible à établir. Cette constatation nous a amené à proposer l'approche MBN qui permet de réagir en temps réel, globalement, localement et ponctuellement à différents évènements se produisant dans le réseau.

– *Prise en compte de l'hétérogénéité de la topologie Internet*
L'approche MBN propose de guetter des changements qui se produisent dans le réseau ou le trafic par l'intermédiaire de la mesure des paramètres de QdS et du trafic. La figure 3.6 décrit comment ces outils de mesure doivent être déployés dans le réseau. Elle détaille le cas plus spécifique d'une connexion MBCC régie par l'approche MBN entre une source et une destination, traversant deux SA Internet ainsi que les routeurs de bordure et de coeur. Ces routeurs intègrent le mécanisme MSP (permettant de signaler aux équipements réseaux concernés ces résultats de mesure). Il est à noter que les équipements de mesure sont de plus en plus déployés au sein des différents SA Internet à l'heure actuelle. Néanmoins, même si les nœuds du réseau ne seront sans doute jamais tous équipés d'outils de mesure, nous pensons qu'en collectant et en utilisant les résultats de mesure des sondes effectivement déployées dans l'Internet, nous pouvons améliorer considérablement la gestion du réseau et de son trafic. Ainsi, MBN est pensé selon l'idée suivante : les performances et la QdS peuvent être grandement améliorées et même devenir optimales en utilisant des informations de mesure sur le réseau, mais même si en certains points du réseau l'information de mesure n'est pas disponible, le

3. Par exemple en proposant de remplacer TCP par TFRC, comme les résultats obtenus dans le chapitre 1 pourraient le faire penser.

réseau doit continuer à fonctionner avec de bonnes performances et une bonne QdS.

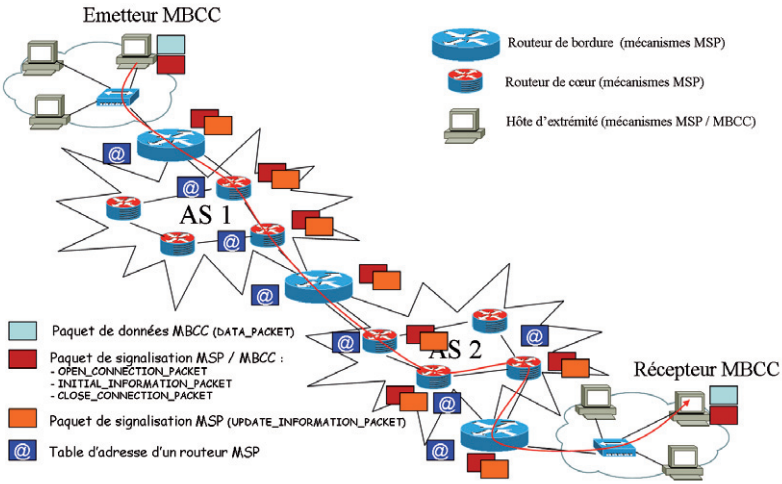


FIGURE 3.6 – Déploiement architectural de MBN : exemple du contrôle de congestion MBCC

- *Utilisation des mesures intra-domaine*
  La structure administrative de l'Internet nous amène à considérer différentes techniques de mesure. En effet, les mesures intra-domaines peuvent être réalisées par des équipements passifs (systèmes basés sur SNMP, NetFlow, DAG...) déployés par l'opérateur qui souhaite réaliser une gestion du réseau plus pertinente. Il pourra ainsi disposer d'informations sur le niveau de bande passante utilisée et disponible, le nombre de flux actifs dans son réseau, le taux de perte... A l'inverse la mesure du délai sera plus facile en employant des techniques actives.
- *Utilisation des mesures inter-domaines*
  Pour ce qui est de la mesure inter-domaine, le problème est différent. En effet, l'opérateur considéré ne disposera pas nécessairement d'informations fiables sur un SA concurrent. Dans ce cas, il est nécessaire d'utiliser des techniques de mesures actives [Lab05].
  Ainsi, l'ensemble de ces mesures réalisées en temps réel et signalées à l'ensemble des équipements réseaux concernés (les sources de trafic par exemple), donne une connaissance précise de l'état du réseau et du trafic et permet ainsi de parfaitement adapter leur débit d'émission (dans le cas de MBCC par exemple) aux ressources disponibles. Il est important de noter qu'un aspect primordial de MBN a trait à la définition d'un protocole permettant de signaler les informations de mesure dans le réseau à la fois en intra-domaine mais aussi éventuellement entre différents opérateurs ou FAI si ceux-ci décident de coopérer étroitement pour un objectif commun de fourniture de QdS. De plus, l'utilisation de mesures actives permettra de vérifier que les mesures annoncées par les différents opérateurs sont bien honorées par ces derniers.

– *Positionnement de MBA par rapport aux approches IntServ et DiffServ*
MBA permet une collecte de l'état des liens de différents domaines par l'intermédiaire du mécanisme MSP que nous allons détailler dans la suite de ce chapitre. Il apporte donc une réponse à l'absence de mécanisme de signalisation de l'approche DiffServ appliquée à une topologie multi-domaine et permet donc de contourner le problème des niveaux de QdS différents que l'on obtient lorsqu'un flux à QdS traverse plusieurs domaines DiffServ différents.
De plus, bien que la gestion de la QdS dans MBA se fasse par flux individuel (comme nous allons le voir dans la suite), le verrou d'IntServ concernant son passage à l'échelle de l'Internet est contourné car notre approche ne doit être appliquée que sur le service "éléphant". En effet, comme nous l'avons vu au chapitre 2, les éléphants représentent la très forte majorité du trafic en volume mais seulement quelques pourcents en nombre de flux. Dès lors, le nombre de flux qui est éligible pour ce service reste faible dans l'Internet actuel et ainsi le passage à l'échelle de notre solution est conservée.

3.3.2 Détails des mécanismes déployés dans MBN

Le mécanisme de gestion des informations de mesure (MSP)

MSP est un composant clé de l'architecture MBA mais il nécessite de trouver pour lui le compromis entre efficacité et capacité à fonctionner à large échelle. En effet, il est nécessaire de fournir des informations sur le trafic le plus rapidement possible pour permettre aux composants du réseau de réagir vite suite à la réception d'informations très récentes sur l'état du réseau, tout en évitant de saturer le réseau avec des informations de signalisation. Ce bon fonctionnement à grande échelle nécessite aussi que les composants MSP ne stockent pas une trop grande quantité d'informations : les tables de correspondance à rajouter pour MBN dans les routeurs doivent être aussi petites que possible (avec un nombre limité d'entrées) pour permettre de minimiser le temps de recherche d'une information.

La figure 3.6 présente le mode de fonctionnement de MSP permettant d'atteindre ces objectifs d'efficacité et de passage à l'échelle (plus de détails seront fournis dans la section qui suit relative à l'étude de cas de MBCC). Tout d'abord, MSP est orienté connexion, c'est à dire que le chemin signalé doit être le même pour tous les paquets d'une connexion choisie. Pour cela, nous avons utilisé le principe de RSVP [28] avec un premier paquet qui découvre le chemin de la source à la destination et ensuite un paquet de retour qui revient à la source en remontant le chemin aller. La différence entre RSVP et MSP est que le paquet de retour (qui est un paquet de réservation dans RSVP mais un paquet de signalisation dans MSP) transporte des informations de mesure. D'autre part, les paquets de signalisation sont envoyés à chaque fois que nécessaire, alors que dans RSVP ils sont juste envoyés lorsque la connexion est ouverte. MSP utilise simplement le principe de RSVP qui consiste à trouver un chemin et à revenir le long de ce chemin. Cette méthode permet à MSP d'identifier parfaitement quels sont les composants réseaux (les routeurs) rencontrés sur le chemin et de limiter le nombre de sources et de destinations pour les messages de mesure.

Pour permettre de prendre en compte le problème du facteur d'échelle rencontré par RSVP dans son déploiement Internet, nous avons aussi choisi :

– de ne considérer que les flux éléphants (selon la nouvelle définition présentée au début de ce chapitre). En fait, comme nous l'avons mis en évidence dans le chapitre précédent, les souris ne créent pas de réel problème dans le trafic et l'essentiel des dommages sont

générés par des éléphants volumineux [20]. Ainsi, les routeurs MSP conservent juste une information sur les flux éléphants les traversant. Cette technique permet de limiter le nombre d'entrées dans la table de connexion étant donné que les éléphants représentent une très petite proportion du nombre total de flux [Lar04b] ;

– d'envoyer les informations de mesure seulement lorsqu'une rupture est détectée dans le trafic. Cette technique permet de générer du trafic de signalisation seulement quand les conditions du réseau changent [4]. Ce principe va donc limiter la quantité de données de signalisation et permettre aux émetteurs et aux routeurs de disposer très rapidement d'informations importantes sur l'évolution du réseau et du trafic : rappelons que les mesures sont réalisées tout au long du chemin entre la source et la destination et potentiellement très proches de la source.

En procédant de la sorte, nous souhaitons résoudre le problème du facteur d'échelle qui a été précédemment rencontré dans l'Internet dans les tentatives d'amélioration et de garantie de la QdS (par exemple l'approche IntServ présentée au début de ce chapitre). Il est important de noter que la prise en compte individuelle des caractéristiques des différents flux (i.e. une solution proche de l'approche IntServ) ne posera pas le problème du facteur d'échelle rencontré par cette dernière. En effet, nous ne prenons en compte qu'une toute petite portion du nombre de flux total circulant dans le réseau (5% du total représenté par les flux éléphants) et ainsi le nombre d'informations à stocker dans les routeurs intermédiaires reste raisonnable. A l'inverse, en ne privilégiant pas une approche DiffServ, nous pouvons affiner le traitement appliqué à chacun des flux et ainsi fournir un service plus précis. Ces différents points seront détaillés dans la section qui suit. De plus, les performances de MSP seront précisément évaluées dans la section 3.3.3.

Le mécanisme de contrôle de congestion orienté mesure (MBCC)

Les objectifs de MBCC sont conjointement d'améliorer les caractéristiques du trafic et la performance du réseau en lissant le trafic (de façon à limiter les effets de la variabilité du trafic) et d'optimiser l'utilisation des ressources (la bande passante disponible) en utilisant l'infrastructure de mesure MBA / MSP. De plus, MBCC sera capable d'assurer une certaine équité à des flux concurrents et de continuer à fonctionner avec de bonnes performances même si certaines mesures manquent.

Dans les travaux [Lar03a] et [Owe04b] sur l'analyse des caractéristiques du trafic, la nature oscillatoire du trafic Internet a été mise en évidence. En particulier, il a été montré que ces oscillations persistantes dans le temps (sources de la LRD observée dans le trafic) étaient dûes à l'inadéquation de TCP pour la transmission des fichiers très volumineux sur des réseaux à haut débit. Ainsi, le problème le plus immédiat concerne la réduction des oscillations et plus précisément la régulation des oscillations persistantes qui ont un impact dramatique sur la QdS du trafic et les performances du réseau. C'est pour cela qu'un des objectifs de MBCC est d'offrir plus de stabilité aux flux éléphants. Pour supprimer les comportements oscillatoires observables à toutes les échelles de temps, le mécanisme de contrôle de congestion TFRC (TCP Friendly Rate Control) est capable d'apporter une contribution importante (cf. chapitre 1 et

4. Evidemment, un envoi périodique d'information de mesure est aussi intégré dans MSP pour détecter les variations très douces dans les fluctuations du trafic (c'est à dire un trafic sans rupture forte mais avec une composante de non-stationarité). Etant donné son principe de réaction basé sur la détection de rupture, la période pourra être très grande, induisant ainsi un faible trafic de signalisation.

[Owe04b]). Il essaie donc, d'éviter au maximum les variations brutales de débit qui apparaissent avec TCP dans le cas d'une reprise d'émission qui suit la détection d'une séquence de pertes. En associant un tel mécanisme au transfert des flux éléphants, représentant la majorité en volume du trafic, nous souhaitons contrôler les oscillations du trafic et augmenter la QdS et les performances globales du réseau.

Les bénéfices de l'utilisation de TFRC à la place de TCP ont été démontrés dans [Lar03a]. Cependant, si TFRC est capable de réduire les oscillations de TCP, il n'est pas capable de s'adapter aux ruptures brutales du trafic (pannes sur des liens impliquant un rééquilibrage du trafic, pics de trafic dus à un trafic légitime lié à la diffusion d'un évènement très populaire par exemple). L'approche MBN est proposée comme une solution pour faire face à ces ruptures. Ainsi, nous souhaitons que MBCC soit une solution optimale permettant d'améliorer TFRC, qui en moyenne est un petit peu moins efficace que TCP New Reno with SACK [5] [Lar03a]. Pour bénéficier des avantages de TFRC, nous avons défini MBCC comme une de ses extensions en le dotant d'une capacité à utiliser les résultats de mesure qui émanent des équipements de métrologie déployés dans le réseau. En faisant ce choix, nous sommes capables de produire de bons résultats (meilleurs que ceux de l'Internet actuel) même si les informations de mesure sont temporairement indisponibles.

1. *Principes de MBCC*

   Le principe de MBCC consiste à utiliser l'algorithme de TFRC pour calculer le taux d'émission nominal de chaque connexion et de corriger cette valeur grâce à la connaissance du niveau de bande passante disponible et consommée dans le réseau. Ainsi, si une fraction de la bande passante est disponible, les sources pourront générer plus de trafic qu'indiqué dans l'équation 1.8 (qui correspond au débit d'un flux TCP [9] et chapitre 1) sans pour autant créer des pertes et des congestions dans le réseau. Ainsi, le niveau de congestion du réseau devrait être significativement réduit en déployant des sources de trafic "pro-actives", capables d'adapter en temps réel leur débit d'émission en fonction des ressources disponibles. Un tel mécanisme devrait aussi aider à augmenter l'équité entre les flux, étant donné que la correction réalisée sur le débit d'émission ne devrait pas dépendre de la valeur du RTT mais de la réelle fraction de bande passante disponible équitablement partagée entre les flux concurrents. Comme dans [Lar03a], MBCC sera uniquement utilisé pour les flux éléphants qui sont les flux qui génèrent le plus de perturbations dans le réseau. A l'opposé, comme le trafic «souris» représente un bruit blanc gaussien [20], il n'induit pas de problème de transfert important et il n'est donc pas nécessaire de modifier leur protocole de transport.

2. *Détails de fonctionnement de MBCC*

   Ainsi, pour une période normale (quand les informations de mesure sont correctement reçues, qu'il n'y a pas de congestion et que de la bande passante est disponible dans le réseau), chaque flux éléphant peut utiliser une fraction supplémentaire des ressources qui sont disponibles. Cette fraction est calculée en divisant la bande passante totale disponible par le nombre de flux moyens éléphants dans le réseau à ce moment (ces informations étant fournies par les équipements de mesure rencontrés tout au long du chemin). Il est logique de diviser la bande passante disponible par le nombre moyen de flux actifs (N) traversant ce lien car il a été démontré que les arrivées de flux éléphants

5. Cette version de TCP a été choisie comme référence car elle est considérée comme la version la plus performante de ce protocole de transport.

sont proches d'un processus poissonien [20]. En effet, pour un processus de Poisson, comme la moyenne est égale à la variance, le nombre moyen est significatif car les valeurs du processus ne seront jamais très éloignées de cette valeur moyenne. A l'inverse, pour une période de congestion, les émetteurs MBCC devront réduire leur débit d'émission pour résorber la congestion et ceci en essayant d'être aussi équitable que possible. Dès lors, les sources MBCC envoient la valeur minimale entre le débit TFRC et le débit effectif obtenu par un flux à ce moment au niveau du goulot d'étranglement sur son chemin.

Ainsi, les équations de cet algorithme peuvent être résumées de la façon suivante :
– Pour une période sans congestion ($p = 0$) : $X_{MBCC} = X_{TFRC} + BPd_{flux}$;
– Pour une période de congestion ($p \neq 0$) : $X_{MBCC} = min(X_{TFRC}; BPc_{flux})$;
Avec :
– BPd_{flux} qui correspond à la bande passante disponible dans le(s) goulot(s) d'étranglement rencontré(s) sur le chemin. Il est calculé par l'intermédiaire du rapport $\frac{\text{bande passante totale disponible}}{N}$, cette information étant fournie par les routeurs MSP rencontrés sur le chemin ;
– BPc_{flux} qui correspond à la bande passante consommée par le flux au travers du goulot d'étranglement. Cette information est fournie par le récepteur MBCC avec les autres informations de bout en bout comme le RTT ou le taux de perte (cf. équation 1.8).

3. *Détails de MSP : illustration dans le cas du déploiement de MBCC*

Etant donné les principes de MBCC qui viennent d'être présentés, cette section va décrire comment les routeurs MSP se comportent pour transmettre les informations de signalisation aux sources MBCC que sont la bande passante disponible et le nombre moyen de flux éléphants mesurés sur le chemin emprunté par les flux MBCC. Le comportement de MSP est décrit sur la figure 3.6 et ses quatre étapes de fonctionnement détaillées ci-après :

(a) *Ouverture d'une connexion éléphant.* Un paquet de signalisation spécifique (OPEN_CONNECTION_PACKET) est émis par l'émetteur MBCC et indique à chaque routeur rencontré sur le chemin qu'une connexion éléphant va être initiée. Dans les différents routeurs traversés par le paquet de signalisation, une table d'adresse est mise à jour avec l'adresse de l'émetteur MBCC.

(b) L'agent récepteur MBCC envoie une information de mesure initiale à l'émetteur MBCC en utilisant un paquet de signalisation (INITIAL_INFORMATION_PACKET). Ce paquet est analysé par chaque routeur sur le chemin et mis à jour avec ces informations de mesure locale (voir (c) - pour les détails de la mise à jour). Ainsi, quand ce paquet arrive à l'émetteur MBCC, il peut ouvrir la connexion car il dispose ainsi d'informations de mesure sur l'état du réseau.

(c) Les paquets de données sont échangés entre l'émetteur MBCC et le récepteur (MBCC_DATA_PACKET). Dans le même temps, les routeurs envoient régulièrement des informations de mesure en utilisant des paquets de signalisation spécifiques (UPDATE_INFORMATION_PACKET). Cette information est envoyée quand une rupture est détectée par les routeurs dans l'ensemble des paramètres que l'on peut mesurer en temps réel (dans le cas des agents MBCC : nombre moyen de flux éléphants (N) et bande passante disponible). C'est un principe important

car il limite le nombre d'informations de mesure transitant dans le réseau. Les principes de l'algorithme sont les suivants :
- Si BP totale d_{Ri} (i.e. calculée par le routeur i) $< BP$ totale d_{sig} (i.e. incluse dans le paquet de signalisation) alors BP totale $d_{sig} = BP$ totale d_{Ri} ;
- Si $N_{Ri} > N_{sig}$ alors $N_{sig} = N_{Ri}$;

(d) *Fermeture de la connexion éléphant.* Les émetteurs MBCC envoient un paquet de signalisation spécifique (CLOSE_CONNECTION_PACKET) informant tous les routeurs sur le chemin de la fin de la connexion. Ces derniers suppriment donc de leur table d'adresse l'agent MBCC émetteur.

3.3.3 Validation expérimentale de l'approche MBN appliquée au contrôle de congestion

Dans cette section, nous présentons les résultats expérimentaux qui valident les mécanismes MSP et MBCC. En particulier, dans la section 3.3.3 nous quantifions les valeurs optimales pour les paramètres de MSP et MBCC de façon à trouver le meilleur compromis entre faible surcharge du réseau par les informations de mesure, temps de réponse faible et réactions précises des agents MSP et MBCC. En s'appuyant sur ces valeurs optimales, la section 3.3.3 présente les avantages de MBCC pour la stabilité du réseau et l'utilisation des ressources en le comparant aux mécanismes de contrôle de congestion traditionnels de TCP.

Principes des simulations

1. *Topologie de simulation*

   Les mécanismes MBCC et MSP ont été implémentés et évalués en utilisant NS-2. Il a été nécessaire de développer un ensemble d'outils pour mesurer la bande passante disponible et consommée dans le réseau simulé et pour échanger les résultats de mesure entre les routeurs et les sources de trafic.

   La topologie utilisée est décrite sur la figure 3.7. Dans ces simulations, nous avons créé une topologie multi-domaines avec plusieurs goulots d'étranglement. Les flux éléphants, utilisant soit MBCC, soit TCP SACK ainsi que le trafic de fond utilisant TCP New Reno sont échangés de façon à entrer en compétition dans ces goulots d'étranglement. L'objectif est donc de mesurer l'impact réciproque des flux MBCC, en théorie réguliers sur ceux TCP beaucoup plus variables. De plus, les liens de coeurs (ceux des SA 1 et 3) représentent les liens les plus «congestionnés» sur le chemin considéré. Ils induieront ainsi des périodes de congestion importantes où les capacités d'adaptabilité de MBCC seront estimées et son niveau de performance comparé avec les autres mécanismes de contrôle de congestion (ceux de TCP SACK). Chaque simulation s'appuie sur des traces de trafic collectées sur le réseau Renater. Elles sont rejouées dans le simulateur NS-2 avec une méthodologie spécifique détaillée dans [Owe04a] dont l'objectif est de produire des simulations réalistes [6] (cf. annexe C pour détails).

2. *Paramètres analysés*

   L'objectif principal de cette étude est de comparer les capacités d'adaptation de MBCC face à une augmentation (ou une diminution) de la charge du réseau et ce par rapport

6. Il s'agit de rejouer en simulation des échantillons de trafic Internet pour reproduire toutes les caractéristiques statistiques du trafic réel.

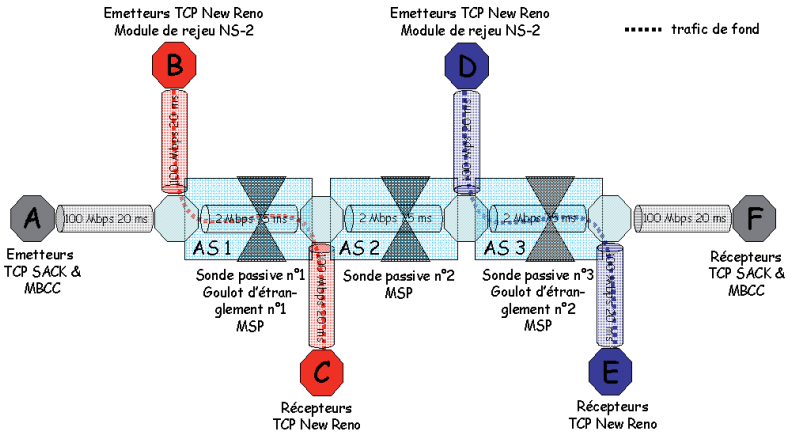


FIGURE 3.7 – Topologie du réseau utilisée pour les simulations NS-2

aux autres mécanismes de contrôle de congestion pour évaluer l'équité qu'il offre aux différents flux. Pour réaliser son évaluation, plusieurs paramètres ont été étudiés en simulation :

- l'impact du trafic de signalisation en calculant le pourcentage entre le débit moyen nécessaire pour la signalisation et le débit moyen global sur le réseau ;
- l'évolution du débit par type de trafic (TCP ou MBCC) en étudiant la variabilité du trafic. Pour cela, nous calculons le débit moyen (D), l'écart-type (σ) et un coefficient de stabilité défini de la façon suivante : $\mathrm{CS} = \frac{D}{\sigma}$;
- l'évolution du processus de perte permettant d'évaluer les capacités d'adaptation de MBCC et de les comparer à TCP ;
- la persistance des oscillations du trafic en calculant le facteur de Hurst par l'intermédiaire du diagramme LDEstimate.

Evaluation de la configuration optimale de MSP

Plusieurs simulations ont été menées, chacune divisée en deux scénarios différents : dans le premier, les flux éléphants (trafic entre les noeuds A et F) sont transmis en utilisant MBCC tandis que dans le second, ils utilisent TCP SACK. Le second scénario est utilisé comme référence expérimentale pour évaluer les avantages de MBCC (en termes de stabilité et de congestion dans le réseau). Dans ces deux scénarios, le trafic de fond (trafic entre les noeuds B à C et D à E), qui est constitué d'un trafic Internet normal, mélange de flux souris et éléphants, est émis en utilisant TCP version New Reno qui est la plus utilisée à l'heure actuelle dans l'Internet.

Chaque simulation dure 300 secondes. 100 éléphants et 2000 souris ont été rejoués. Un des

principaux objectifs de ces expérimentations a été d'étudier l'impact du trafic de signalisation généré par MSP sur la congestion du réseau et l'efficacité des réactions de MBCC. Pour cela, il a été nécessaire de trouver les valeurs optimales pour les différents paramètres de fonctionnement de MSP :

- Le premier paramètre est relatif à la granularité du système de mesure. En fait, la mesure doit être réalisée sur des intervalles courts et par exemple le débit instantané est ainsi calculé comme le débit moyen sur de courtes périodes de temps ($Periode$). Ce paramètre a un impact fort étant donné que plus la granularité est importante, plus le débit semble lisse. En conséquence, cette granularité agit sur le volume de trafic généré par MSP [7]. Plusieurs périodes de 0,2 à 5 secondes ont donc été testées.
- Le deuxième paramètre permet de fixer un seuil de détection pour les ruptures qui sont analysées dans le réseau. Il s'agit de déterminer la variation minimale ($Seuil$) entre deux mesures consécutives pour lesquelles nous pouvons considérer que les conditions du réseau ont changé et qu'il est nécessaire de le signaler aux sources de trafic pour leur permettre de s'adapter à ce changement. Ce seuil est exprimé en pourcentage de la capacité totale du lien.
- Pour finir, la valeur "Time Out" (TO) correspond au comportement périodique de MSP nécessaire dans le cas où aucune rupture ne se produit mais si l'évolution du réseau bien que très lente génère une tendance non-stationnaire. Cette valeur est définie par rapport au paramètre $Periode$, elle ne doit pas être beaucoup plus importante de façon à informer régulièrement les sources MBCC des variations même lentes de débit. En suivant ce principe, nous avons empiriquement sélectionné les couples ($Periode$, TO) : ($Periode = 0,2s$ et $TO = 2s$) ou ($Periode = 0,5s$ et $TO = 4s$) ou ($Periode = 1s$ et $TO = 5s$) ou ($Periode = 2s$ et $TO = 8s$) ou ($Periode = 5s$ et $TO = 10s$).

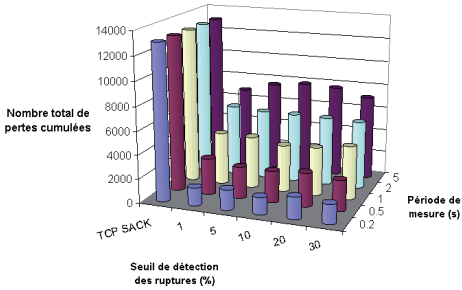
Nous avons fait plusieurs simulations en utilisant différentes traces pour trouver le couple optimal ($Periode_{optimale}$, $Seuil_{optimal}$). Les résultats sont présentés dans les figures 3.8 qui représentent le nombre cumulé de pertes dans le réseau, la surcharge de trafic induite par MSP (en pourcentage du trafic total) et le coefficient de stabilité mesuré pour le trafic échangé.

Inférence des paramètres optimaux

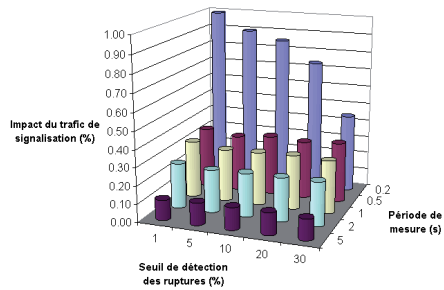
Tout d'abord, nous allons inférer la valeur optimale pour la $Periode$ de mesure. Un des objectifs principaux de MBCC est d'optimiser au mieux l'utilisation des ressources du réseau en générant le moins de pertes possibles. Dans la figure 3.8(a), seuls les résultats avec une $Periode \leq 1s$ sont acceptables [8] (quel que soit la valeur du seuil) : $pertes_{MBCC} \leq \frac{pertes_{TCPSACK}}{3}$. Un autre objectif de MBCC est de transférer les données avec un débit régulier et d'éviter les comportements oscillants qui induisent une mauvaise utilisation des ressources du réseau. Ainsi, dans la figure 3.8(c), seuls les résultats avec une $Periode \geq 1s$ sont acceptables (quelque soit la valeur du seuil) : CS(MBCC) $\geq CS$(TCP SACK). Lorsqu'on considère les résultats sur l'impact du trafic de signalisation, ils ne nécessitent pas de considération supplémentaire. Ainsi, en croisant les résultats des trois paramètres précédents (congestion, stabilité et trafic de signalisation), uniquement les résultats obtenus avec une $Periode = 1s$ respectent tous les critères de choix.

7. Plus la granularité choisie sera faible, plus la détection des variations se fera de façon précise et plus le volume de signalisation sera important.

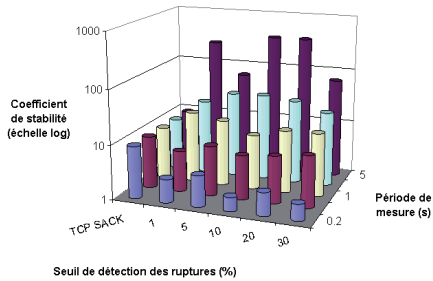
8. Pour une $Periode \geq 2s$, les niveaux de pertes entre MBCC et TCP SACK sont trop proches.



(a) Congestion du réseau



(b) Surcharge du trafic de signalisation par rapport au trafic total



(c) Stabilité du trafic

FIGURE 3.8 – Evolution des paramètres de performance en fonction des valeurs de fonctionnement de MSP

Dans un deuxième temps, nous allons inférer la valeur optimale pour le $Seuil$. Dans ce cas, seuls les résultats expérimentaux s'appuyant sur la stabilité du trafic peuvent nous apporter de l'information pour choisir le seuil optimal. Pour les deux paramètres restants (signalisation et perte), les résultats sont vraiment trop proches pour nous apporter une information utile. Ainsi, nous prenons en compte le seuil où le CS est maximum, il s'agit du cas où $Seuil = 1\%$. En conclusion, la paire de valeurs optimales est ($Periode_{optimale} = 1s$, $Seuil_{optimal} = 1\%$). Elles seront utilisées dans la section suivante pour quantifier précisément les avantages de MBCC par rapport à TCP SACK.

Contribution de MBCC à la régularité du trafic dans une configuration multi-domaines

Cette deuxième expérience permet de comparer précisément les impacts de MBCC et de TCP SACK sur la performance du réseau, la régularité du trafic et l'optimisation des ressources. Elle va illustrer les capacités de MBCC pour améliorer la régularité du trafic pour

les flux MBCC et devrait aussi montrer comment MBCC peut améliorer le profil du trafic en terme de stabilité pour les flux qui n'utilisent pas MBCC mais qui sont en concurrence avec les autres flux MBCC dans les goulots d'étranglement de l'Internet. Pour cela, la topologie utilisée est la même que dans l'expérience précédente (cf. figure 3.7). Les conditions des différents scénarios sont les mêmes à l'exception du nombre de flux qui a augmenté (d'un facteur 10) et les simulations durent 1800 s de façon à introduire des transferts d'éléphants plus longs comme c'est le cas dans l'Internet. Les mécanismes utilisés pour échanger les éléphants sur le réseau sont tour à tour dans le :

- Scénario 1 : le mécanisme MBCC ;
- Scénario 2 : le mécanisme TCP SACK ;
- Scénario 3 : le mécanisme TCP SACK, de plus nous avons procédé au déploiement du mécanisme ECN sur les routeurs des différents domaines ;
- Scénario 4 : le mécanisme TFRC.

L'objectif des scénarios 1 et 2 est de réaliser une stricte comparaison entre notre nouveau mécanisme de contrôle de congestion MBCC et la version la plus performante de TCP dans l'Internet actuel (TCP SACK). Le scénario 3 va permettre d'étudier l'effet du mécanisme ECN sur le trafic réseau et mesurer si la réactivité de TCP SACK aux phénomènes de congestion peut être accrue par rapport à celle de MBCC dans l'approche MBN. Enfin, le scénario 4 reproduit l'expérience présentée à la fin du chapitre 2 de ce manuscrit et va permettre de précisément quantifier l'apport et les avantages de MBCC par rapport au mécanisme TFRC.

Les différents scénarios de cette expérience permettent aussi d'évaluer les capacités à grandes échelles des mécanismes proposés, étant donné que le nombre d'éléphants est augmenté de façon conséquente. D'autre part, les routeurs MSP sont configurés avec le couple de valeurs optimales, inféré précédemment ($Periode = 1s$, $Seuil = 1\%$) et les paramètres étudiés dans les simulations sont les mêmes que dans la partie précédente.

Tout d'abord, le débit du trafic a été calculé. Le tableau 3.1 montre les résultats pour les scénarios 1 et 2. Cette expérience met donc en évidence que MBCC est plus performant que TCP SACK car le débit et l'utilisation des ressources sont plus élevés et le trafic est aussi plus régulier. D'autre part, une autre information intéressante concerne le trafic de fond des goulots d'étranglement 1 et 2 quand le trafic éléphant MBCC est présent dans le réseau (scénario 1). Nous pouvons voir que dans le cas où TCP SACK est utilisé pour transmettre les éléphants entre A et F (scénario 2), le débit moyen du trafic de fond est plus bas et présente plus de variabilité que quand MBCC est utilisé dans le réseau ($CS(\text{TCP New Reno}_{\text{Scénario 2}}) < CS(\text{TCP New Reno}_{\text{Scénario 1}})$).

TABLE 3.1 – Analyse de la variabilité du trafic (scénarios 1 et 2)

	Scénario 1			Scénario 2		
	MBCC	TCP New Reno Goulot d'étranglement n°1	TCP New Reno Goulot d'étranglement n°2	TCP SACK	TCP New Reno Goulot d'étranglement n°1	TCP New Reno Goulot d'étranglement n°2
Débit moyen (B / s)	109434.9	111822.5	111572.5	109420.2	101651.1	101357.3
Ecart-type du débit (σ) (B /s)	31840.4	57127.7	60299.4	44704.3	83943.6	84291.9
Coefficient de stabilité (SC)	3.437	1.957	1.850	2.448	1.211	1.202

Ce résultat est confirmé avec l'analyse du processus de perte. En effet, la figure 3.9 décrit un niveau de pertes plus important dans le réseau si TCP SACK est utilisé (cf. les courbes

du scénario 2) plutôt que quand MBCC (cf. les courbes du scénario 1). Ce résultat est en plus observable à la fois pour le trafic éléphant et pour le trafic de fond global. En effet, la variabilité du trafic dans le scénario 2 est plus importante, les congestions apparaissent donc plus facilement dans le réseau et le nombre de pertes est plus élevé.

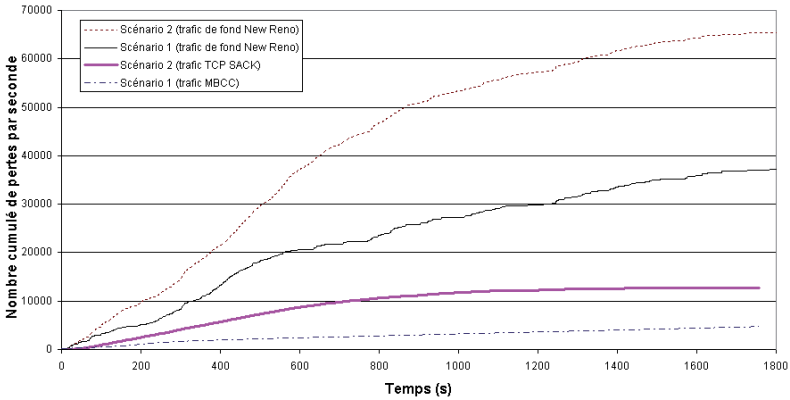


FIGURE 3.9 – Estimation du niveau de congestion (scénarios 1 et 2)

Lorsque l'on s'intéresse aux résultats obtenus dans le scénario 3 (cf. table 3.2) on peut s'apercevoir que le déploiement du mécanisme ECN dans le réseau couplé à l'utilisation de TCP SACK, diminue le caractère variable du trafic généré (CS(TCP SACK$_{\text{Scénario 2}}$) < CS(TCP SACK$_{\text{Scénario 3}}$)) mais les résultats en termes de stabilité et de débit moyen offerts aux flux éléphants restent moins intéressants que dans le cas du scénario 1 (utilisation de MBCC) ainsi CS(TCP SACK (ECN) $_{\text{Scénario 3}}$) < CS(MBCC$_{\text{Scénario 1}}$)). De plus, le niveau global de congestion généré dans le réseau reste plus important dans le cas du scénario 3 que dans le cas du scénario 1 (cf. figure 3.10).

TABLE 3.2 – Analyse de la variabilité du trafic (scénarios 3 et 4)

	Scénario 1	Scénario 3	Scénario 4
	MBCC	TCP SACK (ECN)	TFRC
Débit moyen (B / s)	109434,9	109430,0	96939,5
Ecart-type du débit (σ) (B /s)	31840,4	40587,1	36702,3
Coefficient de stabilité (SC)	3,437	2,696	2,641

En ce qui concerne le scénario 4 (utilisation de TFRC), on retrouve des résultats similaires à ceux présentés dans le chapitre 2. En effet, la régularité du trafic en utilisant TFRC est accrue par rapport à TCP mais les performances moyennes de ce mécanisme de contrôle de

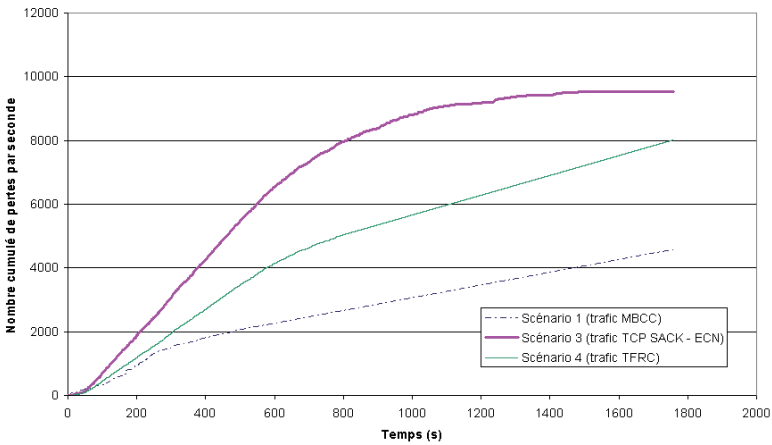


FIGURE 3.10 – Estimation du niveau de congestion (scénarios 3 et 4)

congestion entraine un débit moyen légèrement moins important que celui offert par TCP (cf. table 3.2). Ces résultats illustrent bien la valeur ajoutée de MBCC et de l'approche MBN qui permettent de pallier les limitations de TFRC en intégrant les informations de mesures apportées par le mécanisme MSP. Ainsi ces nouveaux mécanismes développés dans le cadre de l'approche MBN permettent de maximiser l'utilisation des ressources disponibles tout en générant un trafic le plus régulier et le moins variable possible ce qui a pour conséquence de limiter le nombre de pertes dans le réseau (cf. figure 3.10).

MBCC a aussi un impact très bénéfique sur la LRD du trafic (figure 3.11). Grâce à MBCC, la LRD est beaucoup plus réduite dans le trafic éléphant (voir le scénario 1 où $H = 0,51$) en comparaison du trafic TCP SACK éléphant de référence où la LRD est très élevée ($H = 0,92$ dans le scénario 2). En conséquence, il y a moins d'oscillations (cf. coefficient de stabilité du tableau 3.1). De plus, l'analyse de la LRD du trafic de fond (cf. le bas de la figure 3.11) fait apparaître que le trafic global, quand les éléphants sont transmis avec MBCC ($H = 0,74$ dans le scénario 1), est moins dépendant à long terme qu'avec TCP SACK ($H = 0,83$ dans le scénario 2), cette spécificité générant plus de stabilité dans le profil du trafic et donc moins de congestion dans le réseau.

3.4 Conclusion

Dans cette section, nous avons proposé une nouvelle approche qui utilise en temps réel les résultats de métrologie pour améliorer la QdS Internet avec l'objectif final de pouvoir proposer un service stable et de meilleure qualité. Cette approche a été appliquée pour la conception d'un mécanisme de contrôle de congestion (MBCC) dont l'objectif est de lisser le trafic (un besoin primordial pour pouvoir fournir des services stables et garantis), limiter le nombre de

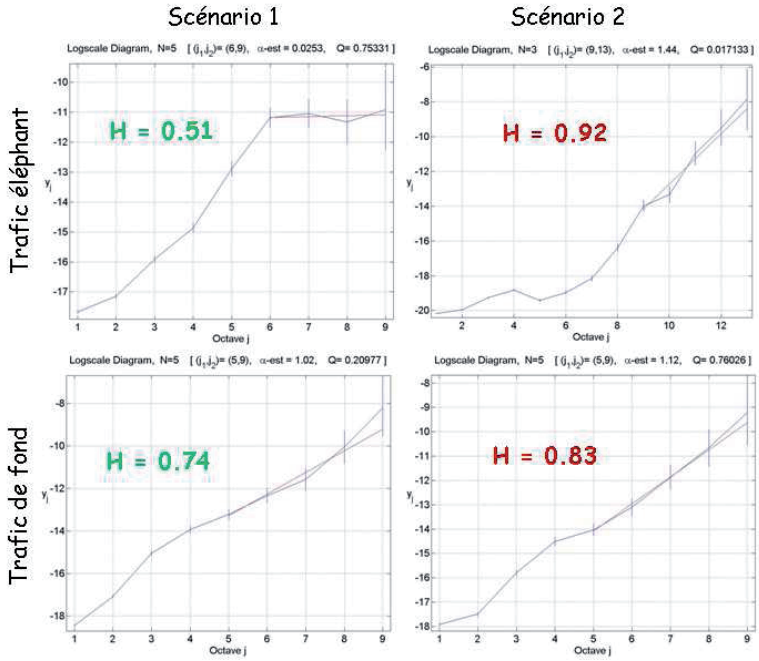


FIGURE 3.11 – Estimation de la LRD du trafic

pertes, optimiser l'utilisation des ressources et fournir de l'équité. Il faut noter que MBCC repose sur une approche originale qui consiste davantage à gérer de façon intelligente le trafic par rapport à ses caractéristiques complexes (oscillations et ruptures) plutôt que de réagir uniquement à des congestions. Les résultats expérimentaux prouvent que MBCC atteint ses objectifs : il fournit un débit optimal et régulier, il utilise toutes les ressources et fournit aussi plus d'équité entre les flux.

Au final, il est clair que les résultats de MBCC démontrent les bénéfices de notre approche MBN appliquée au contrôle de congestion. Nous croyons aussi que MBN a une vocation plus universelle pour gérer l'Internet et son trafic notamment son extrême dynamicité et sa forte variabilité. En effet, MBN peut être utilisé pour fournir une solution adaptée qui puisse faire face à différents types de réseaux, de trafics ou de conditions de fonctionnement. En particulier, MBN pourrait être utilisé dans d'autres domaines comme l'ingénierie du trafic, la tarification ou encore la sécurité réseau...

En particulier, lorsqu'on parle de trafic variable, un cas extrême concerne le trafic contenant des attaques de DdS par exemple. En effet, l'analyse de trace contenant des attaques (comme cela sera présenté dans le chapitre suivant) fait apparaître des phénomènes oscillatoires très

prononcés. C'est pour cela que dans le chapitre qui suit, nous introduisons une application de MBCC pour améliorer la robustesse du réseau et nous étudions ainsi précisément comment il capable de faire face à ce type de trafic.

Chapitre 4

Application du mécanisme MBCC pour l'amélioration de la robustesse du réseau Internet

Dans ce chapitre, nous allons mettre en évidence les apports du mécanisme MBCC présenté dans le chapitre 3 lorsqu'il est confronté à un trafic contenant des attaques de déni de service (DdS). Ce cas de figure représente une configuration extrême d'un trafic incluant des ruptures telles qu'elles ont été définies jusqu'à présent. En effet, comme nous le verrons dans ce chapitre, la génération d'attaques de DdS induit une augmentation du phénomène de variabilité du trafic par rapport au niveau de variabilité que l'on peut mesurer dans un trafic Internet standard (i.e. sans attaque). Dans la suite, nous présenterons dans un premier temps les notions de base qui sont relatives aux principes d'attaques dans l'Internet et dans un deuxième temps, nous illustrerons la capacité de MBN et MBCC à fournir un service stable (par rapport à ce que TCP peut proposer dans l'Internet actuel) même dans le cadre d'un trafic contenant des attaques de déni de service. Il est à noter que ce travail a été mené plus spécifiquement dans le cadre d'un projet intitulé METROSEC, labellisé par l'ACI Sécurité & Informatique. Nous allons donc le présenter dans ses grandes lignes au début de la section qui suit.

4.1 Analyse de trafics d'attaques

4.1.1 Le projet METROSEC

Nous avons vu dans le chapitre précédent que la QdS de l'Internet était sensible à un certain nombre de ruptures. Parmi celles-ci se trouvent notamment les moments où une attaque de déni de service, simple ou distribuée, est perpétrée, et pendant lesquels le réseau devient incapable de fournir les services demandés. Cette extrême fragilité de l'Internet souligne le lien étroit qui existe entre sécurité informatique et QdS. L'objectif de ce projet est donc d'augmenter la robustesse et l'insensibilité du réseau vis-à-vis des ruptures dans le trafic et la topologie, afin qu'il puisse continuer à fournir un service acceptable et de garantir la QdS demandée (réduisant ainsi à néant l'effet de possibles attaques). Le projet METROSEC se propose donc d'abord de développer et mettre en œuvre des outils de métrologie actifs et passifs et de supervision et de surveillance des caractéristiques du réseau et de son trafic.

L'analyse des traces et mesures doit permettre de mettre en évidence la nature et l'importance de l'impact de ces ruptures sur la QdS du réseau, ainsi que sur la propagation dans le temps et dans l'espace (à travers la topologie du réseau) d'éventuelles altérations de celle-ci. L'un des axes de recherche de ce projet se fonde sur les premiers résultats de caractérisation et de modélisation du trafic issus des travaux de métrologie menés dans METROPOLIS en général et en particulier ceux présentés dans les chapitres précédents. Ces travaux ont montré que les phénomènes d'invariance d'échelle constituaient l'une des caractéristiques majeures qui décoraient les statistiques du trafic Internet moderne. Ces travaux menés par certains partenaires de METROSEC ont, de plus, établi que les attaques perpétrées sur un réseau induisent des variations fortes dans les paramètres caractérisant les invariances d'échelle.

Ces approches ayant donné d'encourageants premiers résultats, l'objectif de cet axe de recherche est donc de construire des outils de traitement du signal permettant de détecter, mettre en évidence et caractériser des ruptures, des variations "anormales" des caractéristiques du trafic. Ces variations seront, dans un premier temps, recherchées, par analyses en ondelettes, décomposition modale empirique ainsi que par des méthodes de type filtre de Kalman multi-échelle. De façon complémentaire, le projet propose d'utiliser des outils de théorie des graphes pour détecter les ruptures dans le comportement du réseau. Il s'agit ici de surveiller les variations dans la topologie observée du réseau ou des échanges. Les outils statistiques d'analyse des graphes et de leur dynamique permettent d'espérer une description fine de ces topologies et de l'impact des ruptures de comportements du réseau sur leurs propriétés. Il s'agit donc de mesurer cet impact, de l'analyser et de développer des méthodes de détection et de réaction appropriées. A partir des analyses précédentes, METROSEC proposera des améliorations architecturales, protocolaires et topologiques pour le maintien du réseau à un niveau élevé de QdS, malgré l'occurrence de ruptures. La robustesse vis-à-vis des ruptures donnera aux outils de métrologie et d'analyse (traitement du signal et graphe) le temps de classer la rupture en temps réel et de mettre en place les réactions appropriées. En cas d'attaque, par exemple, des outils d'identification et d'élimination des paquets incriminés seront développés et mis en œuvre, ainsi que des mécanismes d'identification des attaquants.

A terme, METROSEC doit fournir un ensemble cohérent d'outils de métrologie et d'analyse de trafics et de topologies, qui permettront le développement de méthodes efficaces de surveillance, de supervision et de réaction aux anomalies. Ces méthodes combinées aux nouvelles solutions architecturales et protocolaires de communication augmenteront significativement la qualité des services réseaux, même face à une attaque. Assurer l'intégration et la complémentarité des quatre champs disciplinaires (réseau, traitement du signal, théorie des graphes et systèmes répartis), constitue un enjeu et un défi essentiel de ce projet. Cette synergie multidisciplinaire a déjà été ébauchée au travers du travail de l'Action Spécifique 88 du département STIC du CNRS, "métrologie des réseaux de l'Internet" [93], et éprouvée à travers ses conclusions.

4.1.2 Principes des attaques

Les attaques qui peuvent être perpétrées à partir d'un réseau sont très nombreuses, et leur nombre augmente régulièrement avec l'apparition de nouveaux logiciels ou de nouvelles techniques. Etant donné la thématique de notre travail de thèse, nous ne nous sommes focalisés que sur les attaques de DdS modifiant la QdS du réseau. Il s'agit des attaques qui n'affectent pas une machine ou un ensemble de machines particulier mais qui ont un effet sur le niveau

de service qui peut être rendu pour le réseau au moment de l'attaque. Tradionnellement, ces attaques se traduisent par une forte augmentation du niveau de variabilité et d'instabilité du trafic. Ainsi avant de présenter la contribution du mécanisme MBCC à la robustesse du réseau face à ce type de trafic d'attaques nous devons au préalable décrire le principe de ces attaques qui peuvent être liées à des bugs informatiques ou protocolaires (erreurs de programmation ou mauvaise utilisation) et qui génèrent des trafics qui peuvent par exemple s'apparenter à des attaques de déni de service. Nous donnerons par la suite quelques exemples des attaques considérées par la suite dans nos expérimentations : attaque d'inondation, attaque collaborative et attaque distribuée de type "smurf".

Notion d'inondation ("flooding")

Les attaques de "flooding" visent à surcharger les ressources d'un système afin que celui-ci ne puisse plus assurer le service qu'il a à rendre. Ce procédé consiste à envoyer un flux maximal de requêtes ou de données vers une cible définie. Un des problèmes de ce type d'attaque vient du fait que n'importe qui peut les provoquer. En effet, il suffit de disposer d'un logiciel permettant de générer le type de requète demandé (il est à noter que ces logiciels sont très facilement recupérable sur le réseau Internet à l'heure actuelle). Les conséquences d'une telle attaque peuvent aller du simple ralentissement système, en passant par un blocage voire même un effondrement du serveur qui est visé. Plus le nombre d'attaquants est important plus l'attaque peut être dangereuse pour la cible, on parle dans ce dernier cas d'attaque d'inondation distribuée (DDdS). Dans la suite, nous allons illustrer ce type d'attaque par l'exemple des attaques de "Syn-flooding" TCP, lesquelles ont été décrites dans [108], [83] et [104].

Exemple d'attaque d'inondation : "Syn-Flooding"

Le principe des attaques de Syn-flooding est de générer envers un serveur ou un groupe de serveurs un grand nombre de requêtes d'ouverture de connexion qui ne seront ensuite pas utilisées pour transporter des données, mais seulement allouées pour bloquer les ressources de serveurs FTP ou Web et ainsi dégrader la qualité de leur service, voire même saturer ces équipements qui ne pourraient plus alors accepter de nouvelle requête – d'où le nom de "déni de service". Ce type d'attaque exploite une faille du protocole TCP au niveau de l'ouverture de connexion ("three-way handshake"). En effet, de par son implémentation réelle, le récepteur comporte forcément un nombre limité d'entrées pour les requêtes de connexions. De plus, le protocole fixe un délai d'attente, habituellement de 75 s, pour la réception de la confirmation d'ouverture de connexion après réception de la demande d'ouverture. Ainsi, en générant suffisamment de requêtes d'ouverture de connexion, toutes les entrées du système récepteur peuvent être saturées, d'où le rejet de toute nouvelle demande de connexion. Les recommandations actuelles pour lutter contre le Syn-flooding consistent à augmenter la taille de la table des requêtes de connexion et à diminuer le délai acceptable entre demande et confirmation d'ouverture de connexion [83]. Toutefois, l'augmentation de taille de la table des connexions pendantes fait baisser les performances du système (recherche plus lente dans une table plus grande), et la diminution du délai risque de provoquer le refus d'ouverture de nouvelles connexions pourtant légitimes, venant de réseaux peu performants à longs délais. Les figures 4.1, 4.2 et 4.3 montrent comment une telle attaque se manifeste lorsque l'on observe le trafic sur un lien avec un outil de métrologie de type DAG. Le lien supervisé est un lien d'accès à 155 Mbits/s chargé au tiers de sa capacité en heures pleines. En observant la figure

4.1, rien ne permet de déceler que le réseau subit une attaque. En revanche, c'est en observant les figures 4.2 et 4.3, qui mesurent respectivement le nombre de paquets transportés par le réseau et le nombre de flux actifs, que l'on peut voir apparaître deux pics qui ne se traduisent pas par une augmentation en terme de trafic. Une analyse plus appuyée montre que les pics sont dûs à une augmentation dramatique du nombre de paquets de synchronisation de TCP (Syn). Vu le nombre de flux ouverts pendant les périodes d'attaque, le serveur visé est saturé ou au moins chargé plus que de coutûme et ne peut plus ouvrir de nouvelles connexions. De même, les connexions déjà ouvertes risquent de voir la qualité de service réseau qui leur était offerte se dégrader de façon importante, du fait du plus grand nombre de paquets que doivent traiter les routeurs. Ainsi, sur l'attaque analysée sur les figures 4.1, 4.2 et 4.3, on voit que pendant une heure, le service offert par l'opérateur a peut être été dégradé et le SLA signé avec ses clients violé. Ainsi, une telle attaque, même si la cible est un serveur chez un client, a aussi un fort impact sur le réseau et ses services. De telles attaques doivent donc être détectées au plus tôt au niveau du réseau. Ce qui précède a détaillé spécifiquement le cas des attaques de Syn-flooding. Le principe reste le même pour tous les types de protocole, par exemple ICMP, UDP, DNS... souvent en utilisant des options inadaptées. De plus, ces attaques peuvent cibler différents types de démons pour faire stopper un service.

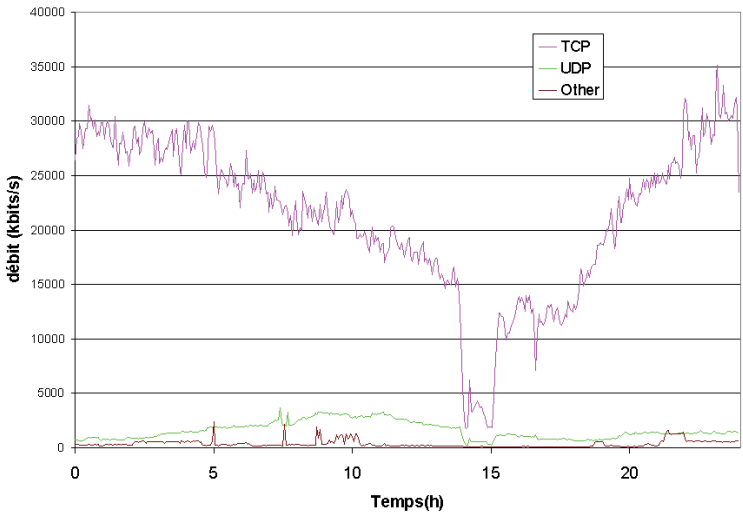


FIGURE 4.1 – Trafic instantané sur un lien d'accès à 155 Mbits/s

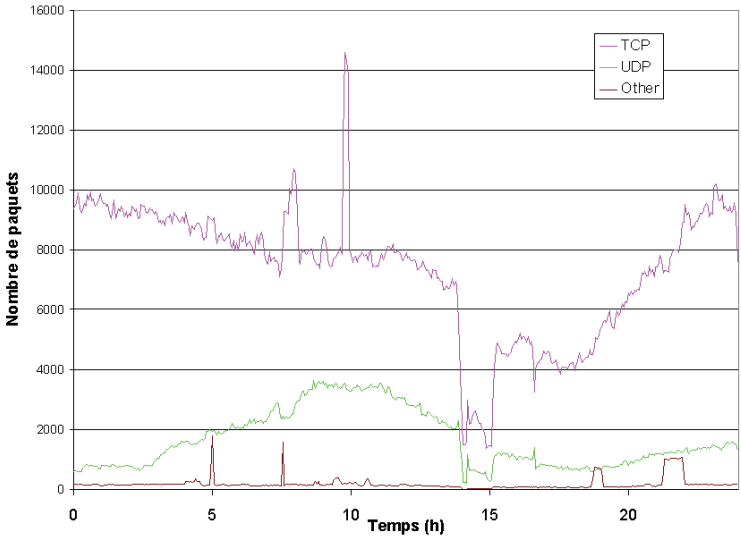


FIGURE 4.2 – Nombre de paquets transitant sur un lien d'accès à 155 Mbits/s

Notion d'attaque évoluée : exemple de l'attaque de "Smurf"

Comme nous l'avons mentionné dans l'introduction, pour prendre le dessus sur les équipements de sécurité de plus en plus répandus et de plus en plus performants, les pirates imaginent des attaques de plus en plus évoluées, comme les attaques distribuées, ayant des stratégies qui s'adaptent aux systèmes de défense en présence ou qui impliquent plusieurs machines ou plusieurs pirates de manière collaborative. Nous allons illustrer cette notion au travers d'un exemple d'attaque intitulée "smurf". La technique du "smurf" est basée sur l'utilisation de réflecteur broadcast pour paralyser un réseau. Un réflecteur est un serveur capable de dupliquer un message et de l'envoyer à toutes les machines présentes sur le même réseau que lui.

Le scénario d'une attaque est le suivant : la machine attaquante envoie un paquet ICMP (par l'intermédiaire d'une commande *ping* de préférence) à un (ou plusieurs) serveur(s) broadcast en falsifiant sa propre adresse IP (l'adresse à laquelle le serveur devrait théoriquement répondre par un pong) et en fournissant l'adresse IP de la machine cible. Lorsque le serveur broadcast va dispatcher le *ping* sur tout le réseau, toutes les machines du réseau vont répondre par un *pong*, que le réflecteur va rediriger vers la machine cible. Ainsi lorsque la machine attaquante adresse le *ping* à plusieurs réflecteurs situés sur des réseaux différents, l'ensemble des réponses de tous les ordinateurs des différents réseaux va être rerouté sur la machine cible.

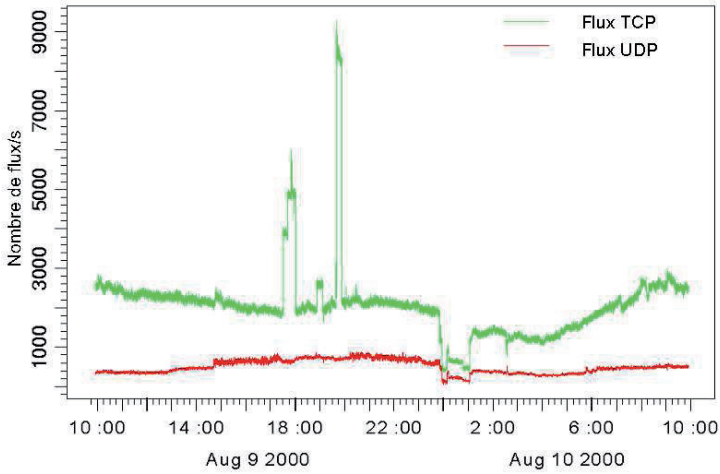


FIGURE 4.3 – Nombre de flux actifs sur un lien d'accès à 155 Mbits/s

Ce type d'attaque permet de perpétrer des attaques vers des serveurs connectés à très haut débit même depuis une machine connectée à bas débit. Ceci montre la puissance et le danger des attaques de DDdS car elles peuvent mettre à mal toute la robustesse du réseau par le trafic qu'elles génèrent. Nous allons maintenant nous intéresser dans la section qui suit aux avantages d'utiliser MBCC face à ce type de trafic.

4.2 Evaluation de l'impact de MBCC sur la robustesse d'un réseau confronté à des attaques de DdS

4.2.1 Principes des expérimentations

Dans cette section nous présentons des résultats expérimentaux qui illustrent les possibilités de MBCC pour améliorer la robustesse de la QdS réseau. En particulier, nous comparons TCP avec MBCC dans le cadre d'un trafic contenant des attaques de DdS ou de DDdS.

Ces nouveaux protocoles de contrôle de congestion (MBCC) et de signalisation (MSP), tous deux présentés et validés dans le chapitre précédent, vont être confrontés dans cette section à un trafic contenant un nombre de ruptures bien plus important et une amplitude bien plus forte que ceux mesurables dans l'Internet en temps normal. Il s'agit en effet de profils de trafic résultant d'une attaque de déni de service distribuée. La variabilité du trafic, comme

nous allons le voir par la suite, sera donc bien supérieure à celle d'un trafic classique d'un réseau de recherche par exemple car il possèdera des caractéristiques propres aux attaques. Il est à noter que toutes les expérimentations ont été réalisées à l'aide de NS-2.

Topologie de simulation

La topologie utilisée est décrite dans la figure 4.4. Dans ces simulations, nous avons créé un goulot d'étranglement pour augmenter la fragilité du lien face aux attaques que nous allons générer. L'approche suivie en simulation repose sur les mêmes principes expérimentaux que ceux exposés dans le chapitre précédent. De plus, les flux éléphants utilisent comme mécanisme de contrôle de congestion, tour à tour dans des scénarios différents, soit MBCC, soit TCP SACK. Le trafic de fond est échangé avec TCP New Reno. Tous ces flux sont transférés de façon à tous rentrer en concurrence dans le goulot d'étranglement et ainsi mieux tester la robustesse du réseau en période d'attaques [1]. L'objectif est donc d'étudier comment ces flux se comportent les uns par rapport aux autres et de comparer l'impact des attaques de DDdS sur la QdS des flux et du réseau lorsque MBCC ou TCP sont utilisés pour envoyer des éléphants.

Le lien de cœur représente le lien le plus congestionné sur le chemin considéré : il s'agit du lien qui va influencer le plus les débits d'émission MBCC ou TCP. Chaque simulation est basée sur des traces de trafic réel collectées sur le réseau Renater et rejouées dans le simulateur NS-2.

Nous retrouvons les mêmes conditions expérimentales que dans le chapitre précédent. La différence vient du profil du trafic de fond qui va être injecté dans les simulations : il s'agit d'un trafic comportant de nombreuses ruptures dues aux attaques de DDdS qu'il inclut. Ainsi, dans les simulations, les flux cours et longs sont toujours différenciés. Les premiers (les souris) n'induisent pas de problèmes de transfert dans le réseau (comme nous l'avons illustré dans le chapitre 2). Aussi, ils seront transmis en utilisant TCP New Reno qui est la version de TCP la plus utilisée dans l'Internet. A l'opposé, les flux éléphants créent dans le réseau des oscillations à long terme qui entrainent des congestions. C'est la raison pour laquelle MBCC a été défini pour transmettre plus efficacement ce type de flux. Ainsi, les simulations comparent le cas où les éléphants sont transmis en utilisant notre nouveau mécanisme de contrôle de congestion MBCC (scénario 2) et celui dans lequel ils sont transmis en utilisant TCP SACK (scénario 1).

Pour permettre d'évaluer la robustesse de MBCC vs. TCP, nous avons rejoué une trace de trafic incluant une attaque de DDdS. Cette trace a été capturée sur le réseau d'accès du LAAS vers l'Internet au moment où une attaque (volontaire) distribuée "d'UDP flooding" était perpétrée à destination de notre laboratoire. La trace que nous utilisons dans cette section dure 40 minutes. Cette capture a été réalisée avec des équipements DAG [35]. Ses caractéristiques de débit ont été représentées dans la figure 4.6. Les 16 premières minutes (cf. intervalle 1 de la figure 4.6) représentent un trafic Internet standard qui contient la plupart des applications classiques de l'Internet : web, mail, ftp... Le reste de la trace (cf. intervalle 2) contient en plus de ce même trafic initial, le trafic résultant de l'attaque de DDdS générée depuis plusieurs ordinateurs situés en dehors du réseau du LAAS et vers un ordinateur spécifique de notre réseau. Cette attaque de DDdS a été perpétrée depuis 4 sites distants vers une cible unique (cf. figure 4.5 pour détails). Les sources de l'attaque étaient l'université de Mont de Marsan, le LIP6, l'université de Coïmbra (Portugal) et un client situé sur une plaque ADSL parisienne.

1. Il est important de noter que le trafic de fond peut contenir en fonction du scénario du trafic résultant des attaques de DdS.

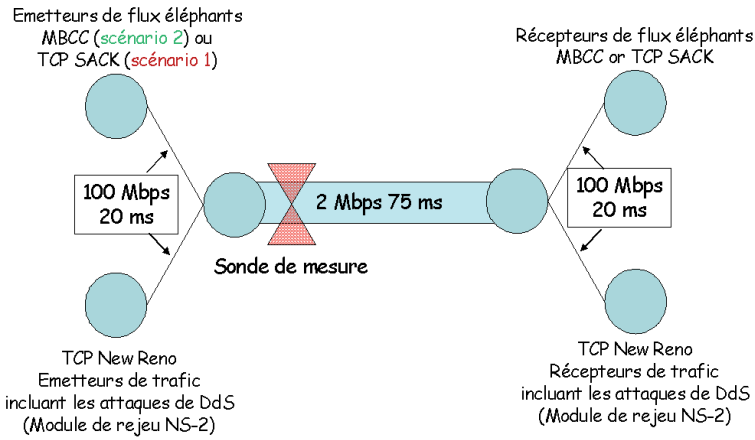


FIGURE 4.4 – Topologie du réseau utilisé dans les simulations NS-2

Toutes ces machines ont généré des paquets UDP à destination d'une machine unique située dans le réseau de recherche du LAAS-CNRS. Plus précisément l'attaque de DDdS est faite par inondation UDP. Chaque inondation est composée de 10.000 paquets dont les paramètres d'intensité et de fréquence changent au cours de l'attaque :

- l'intensité de l'attaque est modulée par la taille des paquets UDP émis (0, 20, 40, 100, 1000 ou 1500 octets) ;
- la fréquence de l'attaque est modulée par la période de temps entre deux paquets UDP consécutifs : 100 ns, 1.000 ns ou 10.000 ns).

Deux flux d'inondations consécutifs sont séparés par un intervalle de silence de 30 secondes.

	NIVEAU PAQUET			NIVEAU OCTET		
	INTERVALE 1	*INTERVALE 2*	*Augmentation (%) (intervale 1 comparé à intervale 2)*	*INTERVALE 1*	*INTERVALE 2*	*Augmentation (%) (intervale 1 comparé à intervale 2)*
Débit moyen	1811,9	2254,9	19,6	1050506,5	1476928,5	28,9
Ecart-type du débit	605,7	1013,5	40,2	534163,2	973696,6	45,1

TABLE 4.1 – Valeurs des débits concerntant les caractéristiques de l'attaque de DDdS

Paramètres considérés

Pour évaluer MBCC et sa contribution à la robustesse de la QdS, nous considérons les mêmes paramètres qui ont été évalués dans le chapitre précédent :

- le débit moyen (D), l'écart-type (σ) et un coefficient de stabilité défini de la façon suivante : $\text{CS} = \frac{D}{\sigma}$;
- l'évolution du processus de perte ;

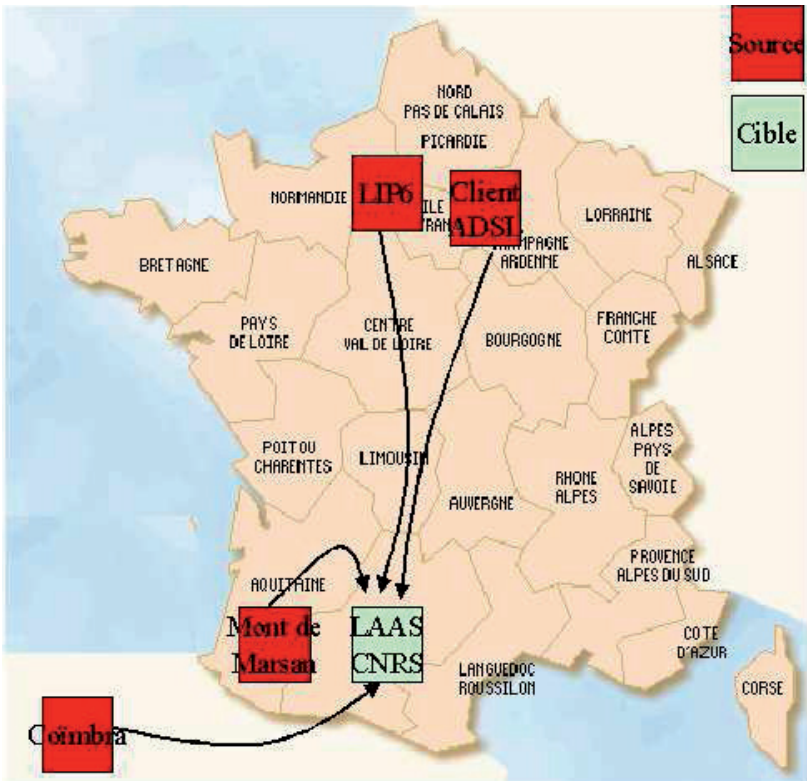


FIGURE 4.5 – Répartition des différentes machines attaquantes (DDdS à destination du LAAS-CNRS)

– le facteur de Hurst.

4.2.2 Résultats expérimentaux

Le premier scénario (éléphants échangés avec TCP SACK) est utilisé comme référence expérimentale. Dans ces deux scénarios, le trafic de fond (si nous excluons le trafic de DDdS) est un trafic Internet standard constitué à la fois de flux souris et éléphants. Ils sont tous envoyés avec TCP New Reno. Nous allons donc analyser la capacité des mécanismes MBCC et TCP SACK à garantir la robustesse du réseau en cas d'attaques. La robustesse aux attaques est définie par la capacité du mécanisme considéré à offrir le même niveau de QdS (en termes de débit, nombre de pertes, congestion ou variabilité...) lorsque le trafic est constitué d'ap-

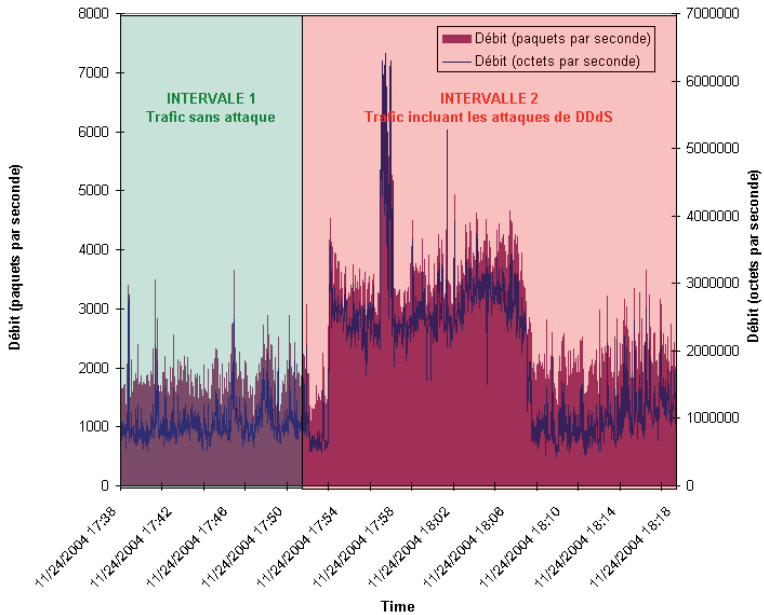


FIGURE 4.6 – Caractéristiques de l'attaque de DDdS

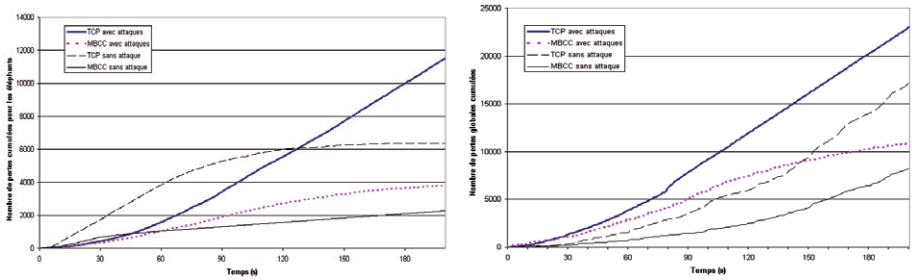
plications standards comme nous l'avons analysé dans le chapitre précédent et lorsque l'on considère le cas, plus extrême, d'un trafic contenant un grand nombre de flux de DdS.

Ainsi, chaque simulation dure 200 secondes. 100 éléphants et à peu près 4000 flux (à la fois souris et éléphants appartenant au trafic de fond) ont été rejoués. Le trafic de fond inclut les paquets constituant l'attaque de DDdS. Les résultats de simulation sont les suivants : tout d'abord, le débit du trafic a été calculé pour la période 2 (trafic avec l'attaque de DDdS). Le tableau 4.2 montre les valeurs résultats pour ces deux scénarios. Cette expérience a montré que MBCC est plus performant que TCP SACK étant donné que le débit et l'utilisation des ressources sont plus élevés et que le profil du trafic est plus régulier. D'autre part, un autre résultat intéressant à trait au trafic de fond dans le lien étroit lorsque du trafic éléphant est présent dans le réseau (cf. scénario 2). Nous pouvons voir que dans le cas où TCP SACK est utilisé pour transmettre les éléphants (cf. scénario 1), la moyenne du trafic de fond est plus basse et met en évidence plus de variabilité que lorsque MBCC est utilisé dans le réseau (cf. par exemple $SC(\text{TCP New Reno}_{\text{Scénario 1}}) < SC(\text{TCP New Reno}_{\text{Scénario 2}})$). Ainsi, ceci démontre que MBCC est capable de favoriser l'état de QdS dans le réseau en le rendant plus robuste et ainsi conserver un trafic plus lisse que ce que peut générer TCP SACK ; en particulier dans le cadre d'un trafic d'attaque de DDdS.

TABLE 4.2 – Analyse de la variabilité du trafic

	INTERVALLE 2 : trafic incluant les attaques de DDdS					
	Scénario 1 (TCP)			Scénario 2 (MBCC)		
	Trafic global	*Trafic de fond*	*Trafic Eléphant (TCP SACK)*	*Trafic global*	*Trafic de fond*	*Trafic Eléphant (MBCC)*
Débit moyen (octets / s)	243.872,13	221.827,32	22.044,81	248.004,64	227.691,13	22.313,51
Ecart-type du débit (octets / s)	31.553,07	83.943,63	44.704,32	22.690,51	57.127,70	31.840,41
Coefficient de stabilité pour le débit (SC)	7,73	2,64	0,49	10,93	3,99	0,70

Ce résultat est confirmé avec l'analyse du processus de pertes. En effet, la figure 4.7 présente un niveau de pertes plus important dans le réseau en utilisant TCP SACK qu'en utilisant MBCC. Ce résultat a été analysé à la fois sur le trafic éléphant seul (cf. figure 4.7(a)) mais aussi sur l'ensemble du trafic échangé (cf. figure 4.7(b)). En effet, la variabilité du trafic avec TCP est plus importante, ce qui rend les congestions plus nombreuses dans le réseau et augmente le nombre de pertes.



(a) Trafic éléphant (b) Trafic global

FIGURE 4.7 – Analyse du niveau de congestion dans le réseau

Finalement, comme la figure 4.8 l'illustre, MBCC joue un rôle très positif sur le niveau de LRD du trafic. En fait, grâce à MBCC, la LRD est plus réduite dans le scénario 2 (où $H = 0.69$) en comparaison du scénario de référence avec TCP SACK où la LRD est très élevée ($H = 0.88$). Par conséquent, il y a moins d'oscillations (cf. les cœfficients de stabilité associés aux deux scénarios dans le tableau 4.2), cette spécificité induit plus de stabilité sur le profil du trafic et donc moins de congestion dans le réseau. Ce phénomène se traduit par une diminution des variations de débits induites par l'attaque dans le réseau. En effet, avec l'utilisation du mécanisme TCP les ruptures sont très marquées dans le réseau et le "front" des attaques est très visible, à l'inverse avec MBCC sa capacité à absorber les ruptures dans le profil du trafic le rend beaucoup moins sensible aux conséquences de l'attaque de DDoS.

En conclusion, ces résultats prouvent que MBCC est capable de rentre la QdS plus robuste que celle obtenue en utlisant TCP. En effet, l'impact d'une attaque de DDdS est beaucoup plus limité avec MBCC qu'avec TCP : le débit global est plus élevé, les pertes et le niveau de congestion dans le réseau est plus bas ainsi que la LRD qui est beaucoup plus basse. Tout ceci induit un trafic beaucoup plus lisse et une meilleure QdS dans le réseau.

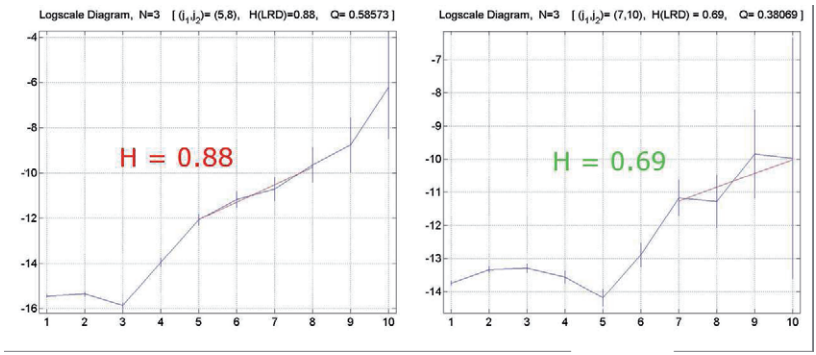


FIGURE 4.8 – Analyse de la LRD du trafic avec TCP (figure de gauche) et MBCC (figure de droite)

4.3 Conclusion

Cette section a illustré la contribution de notre approche orientée mesures (MBN) pour améliorer la robustesse du réseau, et permettre ainsi de continuer à fournir une haute qualité de service même quand ils sont attaqués.

Etant donné, d'une part, le mauvais impact de la variabilité du trafic sur la QdS réseau et ses performances, et d'autre part, la source de cette large variabilité du trafic (les mécanismes TCP en majeure partie cf. chapitre 1 pour détails), nous avons proposé un nouveau mécanisme de contrôle de congestion MBCC qui s'inclut dans une nouvelle approche orientée mesure, MBN. Dans le chapitre précédent, nous avons démontré que MBCC atteignait bien ses objectifs et augmente de façon significative la QdS du réseau en fournissant un trafic très régulier (et réduisant par la même la LRD du trafic qui est responsable de la plupart de ces problèmes de performance et de QdS). Mais dans cette section, nous avons voulu allez plus loin et montrer que MBCC est capable d'augmenter la robustesse de la QdS du réseau, en particulier lorsqu'il est confronté à du trafic contenant des attaques (i.e. un trafic extrêmement variable par nature). En effet, dans ce cas de figure extrême (qui génère des conditions particulièrement handicapantes pour le réseau et ses utilisateurs), MBCC continue à fournir aux utilisateurs le même niveau de QdS.

Bien que ces résultats soient très encourageants, de nombreuses études doivent encore être menées. Premièrement, il est nécessaire d'étendre l'étude de l'efficacité de MBCC sur le trafic d'attaque en considérant d'autres types d'attaques et en particulier la nouvelle famille d'attaques "légères". Ces attaques sont qualifiées par ce terme car soit elles ne produisent pas des ruptures visibles à l'oeil nu (par l'intermédiaire d'une courbe de débit par exemple), soit elles génèrent des variations qui restent faibles pour s'apparenter à du trafic légitime, soit il s'agit d'une combinaison de plusieurs trafics très faibles qui en s'agrégeant vers une cible donnée représente un danger pour cette dernière. Dans tous les cas, ces nouvelles attaques nécessitent de nouvelles techniques de détection car elles ne génèrent plus de ruptures franches dans le profil du trafic.

Ainsi, étant donné les premiers résultats d'analyse des attaques de DdS, nous allons continuer à étudier les nouvelles solutions reposant sur MBN pour augmenter la QdS du réseau et la rendre encore plus robuste. En particulier nous allons approfondir le travail sur l'analyse métrologique des attaques pour faire ressortir avec précision les caractéristiques spécifiques de chacunes d'elles.

Conclusion

L'objectif d'optimisation et de garantie de QdS dans l'Internet est un problème essentiel aujourd'hui. Nous venons de présenter, le résultat de notre contribution de thèse dans le domaine des réseaux informatiques qui cible cet objectif ambitieux. Ainsi, la thématique de notre travail a été la métrologie de l'Internet et ses applications pour permettre une amélioration de la QdS dans le réseau. Nous avons en particulier, fait la proposition d'une nouvelle architecture orientée mesures qui permet l'optimisation des services sur Internet et par la même améliore la robustesse de ce dernier.

Il est maintenant clair que la métrologie du trafic Internet, et notamment son analyse montre que les mécanismes de transport actuels (TCP) introduisent des propriétés de LRD qui se traduisent par une grande variabilité du trafic et obligent à sur-dimensionner les ressources de communication. L'objectif de cette thèse a donc été de trouver des protocoles de transport reposant sur une architecture de mesures qui réduisent cette auto-similarité afin d'optimiser l'utilisation des ressources de communication. Ainsi, dans un premier temps, nous avons vu que les résultats de caractérisation du trafic font ressortir une évolution des usages de l'Internet en terme d'applications (cf. chapitres 1 et 2). Ils mettent en évidence la proportion de trafic P2P qui ne cesse d'augmenter depuis l'an 2000. Ce résultat attendu, s'accompagne d'une augmentation de la quantité d'éléphants transmise sur le réseau. Il est en effet bien connu que les applications P2P sont la plupart du temps utilisées pour échanger des fichiers musicaux ou vidéo qui sont traditionnellement beaucoup plus gros que des pages web. Par contre, l'impact de la transmission de ces flux éléphants n'était pas attendu (cf. chapitre 2). En effet, la nouvelle nature à queue lourde des distributions des tailles de flux introduit, par l'intermédiaire des mécanismes de contrôle de congestion de TCP, de la LRD qui induit dans le trafic des oscillations très importantes. Etant donné que la LRD et les oscillations sont très néfastes pour les performances du réseau et la stabilité des services proposés, il est très difficile, dans de telles conditions, de mettre en place dans l'Internet, des services garantis.

La contribution majeure de ce travail de caractérisation concerne les nouvelles directions de recherche qui sont ouvertes par les premiers résultats d'analyse. En effet, les caractéristiques du trafic (et leurs causes) ne vont pas dans le bon sens pour l'Internet. C'est la raison pour laquelle nous les avons prises en compte pour orienter notre travail de recherche. Tout d'abord, nous avons défini de nouveaux mécanismes pour permettre de diminuer la LRD. Le lien avec les mécanismes de contrôle de congestion de TCP ayant été démontré dans le chapitre 1, nous avons travaillé sur la définition de nouveaux mécanismes plus réguliers permettant une diminution de la LRD. Pour cela, nous avons réalisé une étude permettant de mesurer l'impact de TFRC sur les caractéristiques du trafic : en particulier la LRD et les oscillations. Les résultats de ce travail, ont montré qu'une réduction de la LRD, en modifiant les mécanismes de contrôle de congestion, permet au réseau de fournir des services plus stables.

D'autre part, l'augmentation des oscillations à la fois en amplitude et en degré de persistance font voler en éclats toutes les solutions uniquement basées sur le dimensionnement et la gestion de la taille des tampons dans les routeurs. Avec les résultats présentés dans ce manuscrit, il est évident que les versions actuelles de TCP génèrent un trafic tellement instable et irrégulier qu'il est impossible de dimensionner les files d'attente dans le but d'obtenir des routeurs à la fois fiables et rapides. En effet, la réduction des tampons va diminuer la fiabilité alors que l'augmentation développera le phénomène de LRD. Il apparaît clairement que ce problème peut trouver une solution cohérente au niveau transport.

Ainsi, dans un deuxième temps, nous avons proposé une nouvelle approche qui utilise en temps réel les résultats de métrologie pour améliorer la QdS Internet avec l'objectif final de pouvoir proposer un service stable tout en pouvant réagir en temps réel aux modifications se produisant dans le réseau comme par exemple des variations inattendues dans le profil du trafic (cf. chapitre 3). Cette approche a été appliquée pour la conception d'un mécanisme de contrôle de congestion (MBCC) dont l'objectif est de lisser le trafic (un besoin primordial pour pouvoir fournir des services stables et garantis), limiter le nombre de pertes, optimiser l'utilisation des ressources et fournir de l'équité. Il faut noter que MBCC repose sur une approche originale qui consiste davantage à gérer de façon intelligente le trafic par rapport à ses caractéristiques complexes (oscillations et ruptures) plutôt que de réagir uniquement à des congestions. Les résultats expérimentaux prouvent que MBCC atteint ses objectifs : il fournit un débit optimal et régulier, il utilise toutes les ressources et il fournit aussi plus d'équité entre les flux.

Au final, il est clair que les résultats de MBCC démontrent les bénéfices de notre approche MBN appliquée au contrôle de congestion. Nous croyons aussi que MBN a une vocation plus universelle pour gérer l'Internet et son trafic. En effet, MBN peut être défini de façon à fournir une solution adaptée qui puisse faire face à différents types de réseaux, de trafic ou de conditions de fonctionnement. Ce fut l'objet de la dernière partie de notre travail de thèse (cf. chapitre 4) où nous avons appliqué le mécanisme MBCC pour remplir l'objectif final d'amélioration de la robustesse du réseau (i.e. fournir de façon continue des services de haute qualité même dans le cas où le réseau est attaqué). Le trafic d'attaque correspond au cas pire des variations qui peuvent se produire sur un réseau. Il était donc intéressant d'évaluer la performance de MBCC dans cette configuration extrême (trafic très fortement variable). En conclusion, il est clair que les résultats de MBCC démontrent les bénéfices de notre approche MBN pour améliorer la robustesse de la QdS, en particulier dans le cas d'attaques, MBCC continuant à fournir aux utilisateurs le niveau de QdS proche du niveau qu'il fournit face à un trafic sans attaque et, dans tous les cas, bien supérieur au niveau offert par TCP SACK.

Cependant, comme nous l'avons mentionné à la fin du chapitre précédent, ces travaux devront être poursuivis. Il faudra notamment prendre en compte l'impact d'autres types d'attaques et rechercher en parallèle de nouvelles optimisations pour rendre le mécanisme MBCC encore plus performant. En effet, dans un premier temps, ce travail s'est focalisé uniquement sur les attaques d'inondation (et sur la famille de l'UDP flooding). Ainsi, le travail futur incluera l'analyse de l'impact des autres types d'attaques de DdS et en particulier la nouvelle famille des attaques "légères". De plus, en se basant sur les résultats d'analyse des attaques de DdS, nous allons continuer à étudier de nouvelles solutions basées sur MBN pour améliorer la QdS du réseau et le rendre plus robuste. Ainsi, nous souhaitons étendre notre approche, dans un deuxième temps, à l'étude des mécanismes qui régissent les routeurs ainsi que les protocoles de routage pour qu'ils puissent tirer parti des informations de mesures fournies par l'architecture MBA de façon à pouvoir agir directement dans le coeur du réseau pour mieux

combattre et ainsi diminuer l'impact des attaques de DdS. Au final, nous souhaitons pouvoir améliorer le transfert des flux dans l'Internet en suivant une approche complète : pour transférer au mieux les informations, il est en effet nécessaire de pouvoir agir à la fois au niveau transport mais aussi dans le cœur de réseau en considérant les problèmes qui se produisent dans les nœuds intermédiaires et lors des décisions de routage.

En parallèle de cette perspective de travail sur l'amélioration de la robustesse du réseau, nous souhaitons valider la solution MBN et l'ensemble des mécanismes MSP et MBCC en considérant une configuration réelle en ajoutant un degré de réalisme supérieur à celui obtenu en simulation. Pour cela, nous sommes en train de déployer l'ensemble de l'architecture MBA sur un émulateur de grande taille par l'intermédiaire du projet Grid Explorer. Ce projet vise à mettre en œuvre un grand instrument d'émulation pour les communautés des systèmes distribués à grande échelle, des réseaux et des utilisateurs des Grids ou des systèmes P2P. Ce grand émulateur est constitué d'un grand cluster et d'outils de conduite et d'analyse d'expériences. Il s'agit d'une plateforme réelle de 300 noeuds que nous allons utiliser de façon à démontrer la faisabilité de notre approche MBN lors d'un passage à l'échelle de l'Internet.

Enfin, à plus long terme, nous pouvons imaginer que MBN aura des applications dans d'autres domaines comme l'ingénierie du trafic, la tarification ou encore la sécurité réseau étant donné qu'il se retrouve au cœur de l'évolution de l'Internet actuel par son caractère par essence réactif et sa capacité à la fois à collecter des informations sur l'état du réseau et à s'adapter à cette évolution permanente par l'intermédiaire des mécanismes existant (MBCC) et ceux restant à définir (par exemple un routage réactif orienté mesures).

Annexe A

Mise en place d'une plate-forme de mesures passives

Cette annexe se décompose en deux parties : une première qui a trait à la mise en place matérielle de la plateforme de mesure passive – nous exposerons en particulier comment les sondes DAG et QoSMOS qui ont été déployées – et une deuxième qui a trait à la mise en forme des traces. En effet, le format DAG (ERF) est difficile à exploiter, en particulier car il est variable. Il a donc fallu développer les scripts nécessaires pour permettre aux sondes DAG de capturer des traces avec les bons formats, c'est à dire incluant les entêtes IP et TCP (mais pas les entêtes applicatifs pour rester dans les limites imposées par la CNIL) et une estampille précise et exploitable. En particulier, il a fallu développer des macros pour manipuler les estampilles dans des grandeurs humainement compréhensibles (secondes, ms, μs...) et pas en nombre de tics d'horloges comme cela est fait par la carte DAG.

Un des objectfis du projet METROPOLIS consiste à préparer des logiciels (sous-projet numéro 3 intitulé "Analyse") pour exploiter les traces de trafic produites et collectées par les équipements de métrologie (sous-projet numéro 7 intitulé "Instrumentation"). Toutefois, dans cette annexe nous ne traiterons que de l'analyse des traces passives obtenues avec les sondes DAG et QoSMOS [Owe04d]. Les logiciels développés dans ce sous-projet vont des outils de base pour la manipulation de traces brutes (et ce afin de les convertir à un format facile à exploiter) jusqu'à des outils de plus haut niveau pour la caractérisation du trafic et dont les résultats sont des bases notamment pour les travaux de modélisation.

Avant de rentrer dans les détails des logiciels qu'il a été nécessaire de développer pour manipuler les traces passives collectées sur la plate-forme METROPOLIS nous allons présenter dans la section qui suit les grands principes qui ont guidé notre choix pour le déploiement de cette plateforme de mesures passives.

A.1 Mise en œuvre de la plate-forme de mesures passives

A.1.1 Contraintes et besoins

La principale contrainte qui se pose pour l'installation de sondes de mesure est due au fait que le réseau dont nous souhaitons analyser le trafic est un réseau opérationnel, et que malgré la présence de la sonde, ce réseau doit continuer à fonctionner sans aucune dégradation du service qu'il offre. Le premier besoin pour le système de mesure à mettre en place est donc une

transparence totale pour le réseau et son trafic. Cela signifie que pour être non intrusif, cet équipement ne devra pas provoquer de pannes, d'erreurs de transmission et ne pas introduire de délais pour ne pas modifier le profil du trafic.

Le second besoin lors du choix des sondes de mesure passive concerne sa précision et la validité des traces qu'elle produira. Ainsi, il est essentiel de ne pas "manquer" de paquets transitant sur le réseau, et d'avoir des informations précises sur le passage de ces paquets, notamment au niveau temporel, ce qui représente, aujourd'hui, une des difficultés majeures avec les systèmes actuels. Le système devra donc être bien dimensionné et offrir une horloge précise qui ne dérive pas.

Enfin, le troisième besoin qui apparaît concerne la possibilité de corréler des événements de plusieurs traces, par exemple de suivre un paquet en plusieurs points du réseau, ou d'analyser de façon croisée le passage des paquets et de leurs acquittements, etc. Pour pouvoir analyser finement de tels événements se produisant en des points géographiquement distants et à des instants distincts mais faiblement éloignés temporellement, il est nécessaire de disposer d'une base temporelle commune et universelle pour toutes les sondes.

A.1.2 La solution DAG

Pour répondre à ces besoins (transparence, précision temporelle, temps universel), la solution existante la mieux adaptée est indéniablement une solution basée sur les cartes DAG conçues et développées à l'université de Waikato en Nouvelle-Zélande et commercialisées, maintenues et améliorées par la société ENDACE aujourd'hui. Le premier avantage de cette carte est de pouvoir travailler en dérivation du lien à analyser. Ainsi, dans le cadre de réseaux sur fibres optiques, le principe de branchement de la sonde consiste à insérer un "splitter" optique qui laisse passer 80 % de la puissance optique sur la fibre originelle (chemin normal), et récupère 20 % de cette puissance à destination de la sonde DAG. Ainsi, le trafic n'est absolument pas perturbé, aucun délai n'est introduit au niveau du "splitter" et le trafic conserve donc les mêmes caractéristiques et profils. Le système de mesure est ainsi totalement transparent. A noter que dans le cas où le support physique est de la paire torsadée, il n'y a pas de "splitter", inutile avec les propriétés naturelles de propagation de l'électricité, et un système de "bypass" le remplace pour garantir un bon fonctionnement du réseau même si la sonde s'arrête.

De son côté, la carte DAG est une carte dédiée qui réalise, en temps-réel, l'extraction des entêtes de tous les paquets qui passent sur le lien. La taille de l'entête est précisée au moment de la configuration de la carte pour la capture. Dans notre cas, nous souhaitons pouvoir capturer les entêtes IP et TCP. Enfin, pour chaque paquet capturé, la carte ajoute une estampille codée sur 64 bits à l'entête capturé. Le tout est ensuite stocké sur disque. Il est à noter que les traces ainsi constituées deviennent rapidement très volumineuses, surtout sur les réseaux à hauts débits, et nécessitent donc d'utiliser des disques de grandes capacités et en nombres suffisants. Toutes les machines sont donc équipées de 3 disques durs de 73 Go.

Pour la même raison, le trafic qui transite entre la carte DAG et le disque dur de la station hôte est très élevé, et pour les réseaux aux capacités les plus fortes, les bus PCI classiques des ordinateurs habituels ne suffisent pas. Il est nécessaire dans ce cas là d'utiliser des bus PCI-X, à savoir les bus 64 bits à 66 MHz qui offrent des bandes passantes bien supérieures aux bus PCI traditionnels de 32 bits et à 33 MHz. Tous ces éléments (carte dédiée temps-réel, bus haute capacité, mémoire importante et disques durs de grandes capacités) sont les éléments

indispensables pour garantir un système bien dimensionné capable de capturer une trace de tous les paquets ayant transité sur le lien mesuré.

En ce qui concerne l'estampille de passage de chaque paquet, stockée avec l'entête du paquet, une référence GPS est utilisée. La carte est en effet directement reliée à une antenne GPS. Ainsi, l'horloge de la station qui héberge la carte DAG est resynchronisée chaque seconde sur un signal GPS qui transporte le temps universel venant des horloges atomiques de référence. Ainsi, la dérive de l'horloge est quasiment inexistante, garantissant une grande précision des mesures temporelles, ainsi que le temps universel, car toutes les sondes seront effectivement synchronisées sur le temps de référence universel.

Enfin, pour analyser les traces capturées par les sondes, une plate-forme de stockage des traces et d'analyse a été conçue et mise en place. Cette plate-forme était hébergée au LAAS à Toulouse et était ouverte aux partenaires de METROPOLIS. Les besoins de cette plate-forme sont donc essentiellement une grande capacité de stockage, et une grande capacité de traitement (processeurs et mémoire essentiellement). Cette plate-forme est donc composée de 2 PowerEdge 4600 dont les caractéristiques sont :

- Poweredge 4600
- Rack 6U
- Bi-Pentium Xeon 2.4 Ghz
- RAM : 8 GB
- HD : 8 x 73 GB (RAID0)
- Archivage : DLT1

D'autre part, il a fallu penser à un système pour rapatrier les fichiers de traces des sondes de mesures vers la plate-forme d'analyse. La solution idéale aurait été de pouvoir utiliser les réseaux académiques, mais face à la charge supplémentaire qu'aurait représentés ces transferts, certains liens auraient eu du mal à tenir. Il a donc été décidé d'équiper toutes ces machines avec des lecteurs de bandes au format DLT1, et nous faisons ces transferts en envoyant les bandes par les services postaux [134]. Cette solution a aussi l'avantage de pouvoir utiliser les bandes comme système d'archivage des traces que nous n'utilisons plus pendant quelques temps, ce qui a permis de réduire la capacité des disques de la plate-forme d'analyse, et de réduire sensiblement son prix.

A.1.3 La solution QoSMOS

Dans le chapitre 2, nous avons présenté des résultats qui mettent en évidence la répartition du trafic par application. Nous avons fait appel pour cela à une technique de classification du trafic en fonction de l'application qui utilise une méthode de détection qui reconnait la famille applicative de chaque flux en se basant sur les premiers paquets échangés par l'application considérée. A l'aide de cette méthode (et contrairement aux méthodes "classiques" d'analyse des numéros de port) nous sommes donc capable de reconnaitre un trafic échangé sur un numéro de port qui n'est pas traditionnellement dédié à cette application. Cette méthode est implémentée dans des sondes développées par la société QoSMOS [1] et intitulées Traffic Designer.

Il s'agit d'un boitier permettant l'anayse du trafic, s'appuyant sur une architecture PC et équipé avec une carte Ethernet "bypass". Il permet l'analyse en temps réel du trafic entrant et sortant sur un lien donné et fournit des statistiques macroscopiques sur ce trafic. Cet outil

1. QoSMOS est une jeune pousse issue du laboratoire LIP6 à Paris.

dispose d'un certain nombre d'autres fonctionnalités mais qui dépasse le cadre de notre travail de thèse. Nous ne nous intéressons donc qu'à ses fonctionnalités de reconnaissance applicative des protocoles par "pattern matching".

Cette sonde fournit, en effet, un grand nombre d'informations macroscopiques sur le trafic. Elle se branche en rupture sur un lien d'accès d'un laboratoire ou d'une entreprise à l'Internet, et en fonction des mesures et des analyses sur le trafic qu'elle réalise, adapte sa gestion du trafic de façon à optimiser la QoS perçue par les utilisateurs. Seule les fonctionalités de mesure et supervision nous ont intéressé pour le projet METROPOLIS, et notamment les possibilités offertes par l'interface graphique de cet outil, et surtout son mécanisme de classification applicative basé sur une reconnaissance des premiers paquets transmis sur une connexion. Toutefois, ces équipements se sont avérés trop peu robustes pour fonctionner sur nos liens d'accès, et sont souvent tombés en panne. Nous n'avons donc pas pu les maintenir en place pour ne pas pénaliser nos collègues au LAAS. Nous avons donc demandé à la société QoSMOS de nous développer un logiciel permettant de rejouer les traces DAG dans les boitiers TrafficDesigner. Ainsi, nous utilisons, grâce à ce logiciel TDplayer les sondes QoSMOS en temps différé, et nous pouvons donc utiliser leurs fonctionalités importantes de classification applicative et de représentation graphique.

Nous réalisons cette opération en "rejouant" les traces [2] (par l'intermédiaire du logiciel Traffic-Designer Player de la société QoSMOS) dans le boitier Traffic Designer développé par la société QoSMOS [120].

A.2 Collecte et mise en forme des traces passives

Après avoir présenté, dans la section précédente, les différentes sondes passives que nous avons utilisées dans le projet METROPOLIS, cette section est dédiée à l'ensemble des réglages et des calibrages qui ont été nécessaires pour exploiter les captures de trafic réalisées sur la plate-forme de mesures passives déployée dans le cadre du projet METROPOLIS. En particulier, nous traiterons des problèmes de la taille des enregistrements ainsi que de l'estampillage temporel des traces, problèmes que nous avons rencontrés lors des phases de déploiements matériel et logiciel du projet METROPOLIS.

A.2.1 Taille des enregistrements

La taille des enregistrements pour les captures microscopiques passives est un problème crucial dans le projet METROPOLIS. En effet, cette taille conditionne le degré de complexité des analyses qui pourront être menées a posteriori sur une trace. En effet, une capture n'enregistrant que l'en-tête TCP/IP des paquets circulant sur le réseau ou une capture permettant de stocker l'ensemble des données contenues dans un paquet TCP/IP n'offriront pas les mêmes possibilités en terme de caractérisation. Par exemple, prenons le cas d'un ingénieur réseau qui souhaite analyser le trafic qui passe sur son réseau. Une capture au niveau TCP/IP lui apportera par l'intermédiaire des numéros de port des informations quantitatives sur les différentes applications qui transitent sur son réseau. Mais cette information sera de plus en plus incomplète car de plus en plus d'applications, notamment pair à pair (P2P), n'utilisent pas un numéro de port fixé à l'avance pour échanger leurs informations. Ainsi, leur détection nécessite de faire une reconnaissance syntaxique de ces flux. Cette tâche nécessite de pouvoir avoir

2. Ces traces sont collectées par l'intermédiaire des boitiers DAG déployés sur notre réseau.

accès au contenu applicatif de chacun des paquets capturés. Malheureusement, le stockage de la totalité de l'en-tête applicatif de chaque paquet ne nous est pas autorisé par la CNIL. Pour ces raisons, nous utiliserons le logiciel de classification applicatif QoSMOS [118] pour réaliser ce type de caractérisation car il permet d'analyser à la volée le contenu de chaque paquet et de ne stocker qu'une partie agrégée de ces informations (par exemple la répartition en volume des données circulant sur le réseau classées par type d'application que nous avons présenté dans le chapitre 2).

Pour garantir un degré d'analyse suffisant, nous avons fixé la valeur minimale d'un paquet capturé à 70 octets (16 octets pour l'en-tête DAG + 14 octets pour l'en-tête Ethernet (sans les 4 octets du CRC) + 20 octets pour l'en-tête IP + 20 octets pour l'en-tête TCP (hors option)).

Les cartes DAG utilisées pour réaliser les captures passives proposent plusieurs formats de stockage pour les paquets capturés. La mise à disposition de ces différents formats par la société Endace [49], provient de la nécessité de garder une compatibilité ascendante avec les différents outils développés pour chacune des générations de cartes DAG. Etant en possession de la dernière génération, nous avions le choix entre deux variantes [3] du format le plus récent appelé ERF (format préconisé par la société Endace). Après plusieurs expérimentations, nous avons choisi d'utiliser la version du format ERF avec une taille d'en-tête fixe. En effet, c'est ce format qui nous a permis de simplifier au maximum le développement des outils de traduction de format qui ont été nécessaires pour permettre à chacun des partenaires de disposer de traces dans un format dont il avait l'habitude (par exemple le format PCAP [117] produit par le logiciel TCPDUMP). Le détail des différents champs contenus dans un paquet capturé dans le format ERF avec un en-tête fixe est disponible sur la figure A.1.

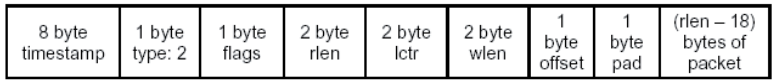


FIGURE A.1 – Détail du format ERF des paquets capturés par une sonde DAG (pour une architecture TCP/IP basée sur Ethernet 10/100 Mbps)

Le détail des différents champs est donné ci-après :
- *timestamp* : date d'arrivée du paquet,
- *type* : type de trame de niveau liaison (Eth, ATM, PoS)
- *flags* : informations diverses sur l'état de la capture (numéro de l'interface, enregistrement tronqué, etc),
- *rlen* (record length) : longueur totale de l'enregistrement transféré entre la carte de capture et le périphérique de stockage,
- *lctr* (loss conter) : nombre de paquets perdus entre la carte DAG et le périphérique de stockage (dans le cas d'une surcharge du bus PCI).
- *wlen* (wire length) : longueur du paquet capturé (rlen moins les informations rajoutés par la carte DAG),

3. La différence entre les deux variantes porte sur la taille des en-têtes qui dans un cas peut être fixe et dans l'autre variable. Cette spécificité permet une optimisation de l'espace utilisé pour le stockage de la trace capturée.

– $offset$ / pad : nombre d'octets qui n'ont pas été capturés au début de la trame (pour le moment cette fonctionnalité n'est pas implémentée sur les cartes DAG).

La taille maximale de capture d'un paquet est fixé par le protocole de niveau liaison sur lequel la carte réalise la capture. Dans le cas qui nous intéresse pour le présent document, les captures sont réalisées sur un réseau local de type Fast-Ethernet. Etant donné qu'une trame Ethernet ne peut transporter plus de 1500 octets (valeur de la MTU de données, la taille maximale d'une trame capturée est limitée à 1532 octets : 1500 octets pour les données utiles du paquet Ethernet + 14 octets pour l'en-tête Ethernet (sans les 4 octets du CRC) + 18 octets pour l'en-tête DAG).

L'ensemble des tailles possibles pour chaque enregistrement dont nous disposons au moment de la capture est donc relativement important en théorie. En pratique, les captures sont soit réalisées sur les en-têtes TCP/IP seuls, soit sur l'ensemble du paquet TCP/IP. En effet, la capture partielle d'un paquet de données serait un inconvénient pour son analyse a posteriori. L'information capturée ne serait pas complète ce qui pourrait nuire à l'analyse des informations. Considérons pour illustrer cette notion l'exemple de l'analyse d'un trafic généré par une application de VoIP. Il arrive de plus en plus souvent que ce type de trafic soit encapsulé dans du trafic HTTP. Si la capture de l'en-tête applicatif du protocole HTTP est incomplète, il sera impossible de faire la différence entre un paquet HTTP contenant les données d'une page Web et un paquet HTTP servant à véhiculer du trafic VoIP.

A titre de remarque, signalons que les cartes DAG capturent l'ensemble du trafic qui circule sur le réseau. Ainsi, tous les paquets générés par des protocoles de niveaux réseau et transport sont aussi capturés (ICMP, ARP, UDP...). Nous n'illustrerons pas quantitativement l'impact de ce type de paquet sur la taille des enregistrements réalisés par les cartes DAG étant donné que le trafic TCP/IP est le trafic majoritaire sur les réseaux analysés dans METROPOLIS [4]. Nous tenons juste à signaler que ces différents paquets entraineront pour la plupart une modification de la taille de l'en-tête de niveau transport de la trame capturée sur le réseau.

A.2.2 Estampillage temporel

L'ensemble de la plate-forme de mesure METROPOLIS est synchronisée temporellement par des horloges GPS (à l'aide de cartes GPS installées dans les machines de capture). Ainsi, tout paquet qui est capturé sur le réseau est aussitôt estampillé par l'ajout d'un champ de 64 bits (cf. le champ "timestamp" de la figure A.1). Dès lors, cette datation universelle permet de réaliser des calculs extrèmement précis (cette précision étant requise dans certains cas, par exemple, dans le cadre d'un calcul du délai de propagation de routeur en routeur pour des paquets IP dans un réseau de type MAN ou WAN).

Précisons que l'information temporelle délivrée par les cartes DAG est exprimée en "tics" d'horloge. Il a donc été nécessaire de développer des macro-commandes pour permettre de manipuler ces valeurs dans des unités plus courantes (microseconde, milliseconde ou seconde).

4. En effet, dans l'Internet plus de 80 % du trafic, quelle que soit la granularité d'étude, est du trafic TCP/IP.

Annexe B

Le logiciel Zoo, un outil de caractérisation zoologique

Tous les outils d'analyse de traces qui ont permis de présenter les résultats de caractérisation de trafic au cours des chapitres 1 et 2 ont été intégrés dans une boite à outils d'analyse des traces. Cette boite à outils a été baptisée ZOO. Ce nom peut surprendre de prime abord, mais nous espérons dans cette annexe convaincre le lecteur de la pertinence de ce nom. En effet, nous avons vu au cours du chapitre 2 que souvent la caractérisation du trafic ne peut se faire de façon globale, et qu'il faut passer par des sous-classes du trafic. Deux des plus célèbres classes de flux portent les noms de "souris" et "éléphants", suivant que ces flux comportent respectivement un petit ou un grand nombre de paquets. En s'inscrivant dans cette même démarche, nous avons donc défini de nouvelles classes de flux qui présentent des caractéristiques communes et que nous avons choisi de désigner par des noms d'animaux. Ainsi, cette annexe est pleine de tortues, libellules, buffalos, etc., et ZOO est donc un outil qui permet d'analyser très simplement chacun de ces types de flux et leur trafic associé.

B.1 Motivations

Modéliser le trafic n'est pas une tâche aisée. Ce problème est soulevé depuis plusieurs années par une grande partie de la communauté réseau mais à l'heure actuelle aucun modèle proposé n'a été accepté par cette dernière. En effet, les premiers résultats (cf. chapitre 1) montrent que les caractéristiques du trafic ne sont pas simples et difficilement intégrables dans un modèle. Plus précisément, les premiers résultats ont montrés que le trafic Internet est dépendant à long terme (LRD) [1], auto-similaire [98] ou encore multi-fractal [44]. Dans tous les cas, même si des modèles pour le trafic Internet peuvent être proposés, ils ne sont pas du tout simples et ne décrivent pas tous les aspects du trafic. Cette complexité des modèles entraîne que, même de nos jours, chercheurs et ingénieurs réseaux continuent d'utiliser des sources de trafic Markoviennes (de type Poisson la plupart du temps), étant donné qu'il reste difficile de développer des générateurs de trafic intégrant les spécificités des modèles auto-similaires ou fractaux. Il en va de même pour les modèles utilisés pour adresser les problèmes d'évaluation de performance.

De plus, les modèles de trafic doivent être aussi simples que possible ; par exemple, Markoviens ou encore mieux Poissonniens. De nombreux outils implémentant ces modèles existent et

permettent aux utilisateurs de facilement analyser et valider leurs propositions. Malheureusement, le trafic Internet n'est pas Poissonnien dans sa globalité (comme nous l'avons présenté dans le chapitre 2) ! Ainsi, le travail détaillé dans cette annexe porte donc sur la détermination d'une méthode pour classifier les flux d'un trafic donné en différentes classes dans le but d'associer à chaque composante du trafic identifiée une classe particulière qui suivrait un modèle simple et bien connu (Poisson, Markov, Gaussien, etc.). Cette idée, qui va de pair avec le besoin d'appliquer des actions différentes en fonction des différentes classes de trafic, provient d'une première analyse réalisée sur le trafic Internet en s'intéressant à la taille des flux (i.e. savoir si le flux considéré est une "souris" ou un "éléphant"). Les résultats de cette étude, détaillés précédemment dans ce manuscript (cf. chapitre 2), ont montré que le trafic des souris (du moins au niveau paquet) peut être assimilé à un processus Poissonnien. D'autre part, les arrivées de flux éléphants peuvent aussi suivre un processus Poissonnien. Ainsi, nous avons souhaité allez plus loin dans ce schéma de classification et définir d'autres classes de flux : les tortues, les libellules, les buffles, etc. Pour réaliser cet objectif, nous avons développé un outil générique pour analyser les propriétés de base du trafic pour ces différentes classes, et ainsi pouvoir valider si elles suivent ou non un modèle de processus simple. Cet outil s'appelle "ZOO". Son nom s'est imposé à nous étant donné qu'il permet de mettre en évidence, comme nous allons le voir dans la suite, toute la "ménagerie" des flux qui sont présents dans l'Internet.

Ainsi, cette annexe décrit le travail de classification réalisé dans l'outil ZOO et présente les différentes familles d'animaux qui ont été créées pour désigner les différentes sortes de flux. Le but final de ce travail sera de trouver une classe pour chaque flux, et un trafic pour chaque classe qui suit un modèle basé sur une loi simple. Ainsi, la modélisation du trafic Internet apparaitra aussi simple qu'une agrégation de processus stochastiques de base. Dans tous les cas, même si toutes les classes de flux ne peuvent être créées pour suivre des processus simples, ce travail sera d'une grande importance si un grand nombre de classes de flux suivent un modèle simple, ceci entrainant une forte simplification de la tâche de modélisation du trafic, ainsi que des autres tâches associées : dimensionnement du réseau, gestion de la QdS, ingénierie du trafic, planification du réseau, etc.

Une autre besoin important à l'origine du développement du logiciel ZOO, concerne la caractérisation simple d'un grand nombre de traces. En effet, dans le cadre du projet METROPOLIS, nous avons collecté un nombre très important de traces. Il était nécessaire de pouvoir mettre en évidence les différences existant entre elles (charge du lien analysé, nombre de flux actifs, taux de pertes...). Un certains nombre de fonctionnalités relatives à la caractérisation simple d'une trace de trafic ont donc été intégré dans ZOO. Nous allons maintenant les détailler dans la section qui suit.

B.2 Description du logiciel ZOO

B.2.1 Fonctionnalités du logiciel

Les fonctionnalités du logiciel ZOO sont décomposables en deux grandes classes. Une première qui permet d'extraire d'une trace de trafic toutes les caractéristiques basiques que nous allons lister dans la suite et une seconde, qui est en relation avec le besoin, expliqué dans le paragraphe précédent, de décomposer le trafic en différentes classes.

Caractérisation simple du trafic

La plateforme de capture METROPOLIS nous permet de réaliser et de stocker un nombre important d'échantillons de trafic. Ces derniers, de part leur durée ou l'état du réseau au moment de leur réalisation, peuvent posséder des caractéristiques et illustrer des comportements très différents. Il est donc nécessaire de disposer d'un ensemble de paramètres de base permettant de les comparer. Pour cela, nous avons sélectionné les caractéristiques suivantes :

- Les premières sont statiques :
  - nombre de paquets et d'octets dans la trace (tous protocoles confondus, uniquement TCP ou UDP) ;
  - nombre de flux (tous protocoles confondus, uniquement TCP ou UDP) ;
  - répartition du nombre de paquets, d'octets et de flux par type d'application ;
  - nombre d'acquittements TCP ;
  - nombre de pertes TCP ;
- D'autres dynamiques :
  - évolution du débit global au cours du temps ;
  - évolution du débit TCP ou UDP ou par application au cours du temps ;
  - évolution du RTT des flux TCP au cours du temps ;
- Ainsi que certaines statistiques :
  - distribution des inter-arrivées des flux TCP ;
  - distribution de la taille des flux TCP ;
  - distribution de la durée des flux TCP ;
  - distribution des inter-arrivées des paquets TCP ;
  - distribution des tailles des paquets TCP.

Nous sommes conscients que cette liste de paramètres n'est pas exhaustive. En effet, un certain nombre d'autres caractéristiques pourraient être ajoutées et ainsi améliorer la précision de la caractérisation. Néanmoins, avec l'ensemble des paramètres sélectionnés nous disposons des différents niveaux d'analyse (octet, paquet ou flux) nécessaires pour pouvoir porter différents regards sur l'état du réseau, illustrer ses comportements et ainsi isoler certaines de ses propriétés. Ainsi, les paramètres statistiques permettent de réaliser une évaluation des performances du réseau considéré en estimant, par exemple, le niveau moyen d'utilisation de ses différents liens ou encore une estimation des comportements utilisateurs en permettant de déterminer le profil général des flux (TCP ou UDP) qui y sont échangés (par exemple en terme de taille, de durée ou de loi d'arrivée).

Cependant, nous avons déjà montré que lorsqu'on souhaite dépasser le cadre de la supervision réseau et s'orienter vers des phases de modélisation ou d'optimisation du réseau, ces informations ne sont plus suffisantes. Il apparaît alors nécessaire de se focaliser sur des paramètres plus complexes (par exemple des grandeurs statistiques d'ordre 2 comme nous l'avons vu dans la section 1.3.4) ou encore de proposer des méthodes d'analyse du trafic qui vont au delà de la simple décomposition en octet, paquet ou flux. C'est l'objet de la section qui suit.

Caractérisation zoologique du trafic

Ainsi, le logiciel ZOO, à partir d'une trace de trafic, permet une analyse plus avancée aux niveaux paquets et flux et offre les décompositions suivantes :

- *souris / éléphant* : il s'agit de distinguer les flux selon leur volume en nombre de paquets

(moins de 10 = flux souris, plus de 100 = flux éléphants) [1].
- *libellules / tortues*[2] : il s'agit de distinguer les flux selon leur durée de vie dans la trace (moins de 2 secondes = flux libellule, plus de 15 minutes = flux tortue).

L'un des objectifs principaux de cet outil étant de trouver un processus simple à chaque classe de flux identifiée, ZOO fournit donc à l'utilisateur un certain nombre de paramètres lui permettant d'infirmer ou de confirmer si les hypothèses stochastiques faites pour une classe particulière du trafic sont correctes, ces paramètres (valables pour chaque classe analysée) sont :
- la distribution des arrivées de flux ;
- la distribution des arrivées de paquets ;
- le niveau de corrélation des arrivées de flux ;
- le niveau de corrélation des arrivées de paquets ;
- le niveau de LRD des arrivées de flux ;
- le niveau de LRD des arrivées de paquets.

Ainsi, l'utilisateur à la possibilité de déterminer si une caractéristique stochastique précise (par exemple le caractère Poissonnien des lois d'arrivées des paquets pour une classe de flux donnée) est recevable. Pour cela, l'utilisateur peut appliquer, pour chaque classe considérée, une démarche systématique qui se décompose selon les différentes étapes ci-dessous :

1. Déterminer les distributions des inter-arrivées ;
2. Chercher si cette distribution suit une loi exponentielle par l'intermédiaire de la décroissance de la distribution et de son niveau d'auto-corrélation ;
3. Evaluer le niveau de dépendance de la série de valeurs en se référant :
   - au diagramme LDestimate représentant l'évolution de la variance en fonction de la largeur d'analyse temporelle (cf. analyse statistique au second ordre par la méthode de la transformée en ondelettes [1]) ;
   - à l'évaluation quantitative du niveau de dépendance à long terme donné par le facteur de Hurst.

B.2.2 Fonctionnement du logiciel

Pour simplifier l'explication du fonctionnnement du logiciel ZOO, nous allons d'abord présenter l'algorithme de traitement et de classement des flux en classe dans sa globalité avant de présenter plus en détails les traitements spécifiques appliqués aux différentes classes.

Description globale de l'analyse

Pour analyser l'ensemble de la trace, l'algorithme extrait les paquets un par un puis analyse l'en-tête de ces paquets. Les informations de cet en-tête permettent de déduire l'origine, la destination et le protocole de chaque paquet.

L'analyse se poursuit ensuite en fonction du protocole, sachant que l'analyse d'un paquet TCP est plus complexe que celle d'un paquet UDP (car les paquets TCP sont classés en flux et

1. Ces bornes ont été choisies en s'appuyant sur la définition des flux souris et éléphants présentée dans [95] mais elles restent modifiables par l'utilisateur du logiciel.
2. Cette décomposition a été présentée dans [30], nous nous sommes inspirés de ce travail pour déterminer les bornes temporelles qui permettent de différencier un flux libellule d'un flux tortue.

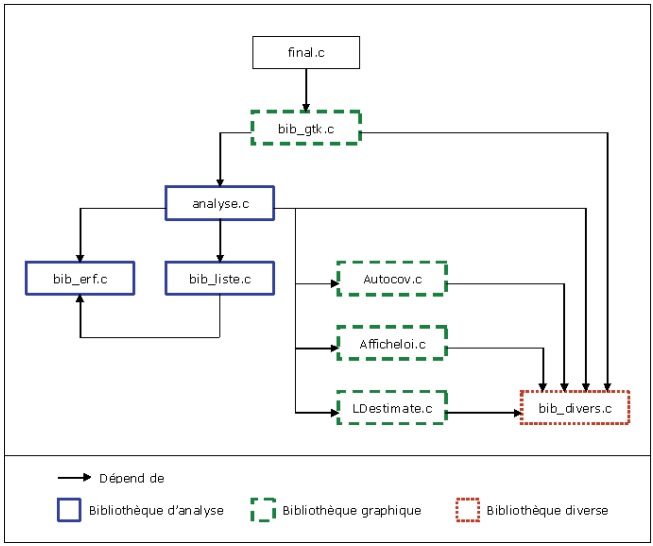


FIGURE B.1 – Structure du logiciel ZOO

on ne fait pas cette opération pour les paquets UDP). Les statistiques pour chaque protocole sont ensuite mises à jour à chaque nouvelle occurence d'un paquet de ce protocole.

Pour pouvoir caractériser le trafic selon deux niveaux (paquet et flux), il est nécessaire d'associer chaque paquet à un flux. Ainsi, chaque paquet est attribué à un ancien flux et si le flux n'existe pas (paquet SYN rencontré) ce dernier est créé. Il est ensuite mémorisé puis actualisé à chaque paquet jusqu'à ce que le flux s'achève (paquet FIN rencontré). Une fois le flux terminé, il est alors possible de savoir s'il s'agit d'un flux Eléphant, Souris, Tortue ou Libellule (cette répartition ne se faisant que dans le cas des flux TCP) pour pouvoir ensuite caractériser séparément chacun de ces types de flux.

Pour informer l'utilisateur de l'avancement de l'analyse, la fenêtre de progression est actualisée tous les 5 000 paquets. Cette analyse précise le nombre de flux, paquets et octets analysés ainsi que la durée de l'analyse.

Enfin, dès que l'analyse paquet par paquet est achevée, le fichier trace analysé et les fichiers résultats (débit, inter-paquets, inter-flux...) sont fermés. Les étapes suivantes sont l'enregistrement des derniers résultats (calcul du niveau de corrélation ou de LRD) puis la génération des graphes.

Le détail de la structure logique retenue pour le squelette du logiciel ZOO ainsi que les différentes bibliothèques développées sont présentés dans la figure B.1.

Nous allons maintenant détailler le traitement opéré en fonction des différents protocoles rencontrés (TCP ou UDP) et les différentes classes de flux rencontrées (souris, éléphants, tortues ou libellules).

Caractérisation des flux TCP

Cette caractérisation se fait lorsque le numéro de protocole trouvé dans l'en-tête du paquet (champ protocol) est égal à 6, c'est à dire lorsqu'il s'agit d'un paquet TCP encapsulé dans un paquet IP. Ensuite, l'analyse se déroule en plusieurs étapes détaillées ci-après :

1. *Enregistrement de la taille du paquet* pour le calcul et l'affichage futur de la distribution des tailles de paquet.
2. *Sauvegarde de la durée inter-paquets* (il s'agit de la durée séparant l'arrivée de deux paquets successifs du même type) dans la distribution des durées inter-paquets, puis mémorisation de la valeur des inter-paquets dans un fichier annexe (données nécessaire pour le calcul de l'auto-corrélation ou de la LRD) ainsi que de la date des arrivées des paquets.
3. *Recherche du flux :* pour cela nous vérifions si le paquet appartient à un ancien flux déjà enregistré ou s'il s'agit du premier paquet d'un flux.

   S'il s'agit d'un nouveau flux, nous mettons à jour le compteur de flux, nous sauvegardons la valeur temporelle de l'inter-flux de la même façon que cela a été fait avec l'inter-paquet puis nous sauvegardons les informations sur ce nouveau flux. Cela fait nous ajoutons le nouveau flux à la liste des flux en ayant au préalable ajouté le paquet à la liste des paquets de ce flux. Il est nécessaire de sauvegarder tous les paquets des flux TCP car l'analyse des flux particuliers (Eléphants, Souris, Libellules, Tortues...) n'est possible qu'une fois le flux terminé.

   S'il ne s'agit pas d'un nouveau flux, nous mettons à jour les caractéristiques du flux puis nous ajoutons le paquet à la liste des paquets de ce flux.
4. *Evaluation du paquet de fin,* pour cela si le paquet analysé est le dernier paquet du flux (bit FIN à 1), on exécute les étapes 5, 6 et 7. Sinon, on passe directement à l'étape 8.
5. *Enregistrement de la durée et de la taille du flux* dans les distributions correspondantes.
6. *Traitement* suivant que le flux appartient à la classe des flux Eléphant, Souris, Tortue ou Libellule (cf. point ci-après de la section B.2.2).
7. *Suppression du flux :* pour cela nous supprimons le flux de la liste chaînée des flux puis nous libérons l'espace mémoire (cette étape est très importante pour garantir une vitesse d'exécution du programme raisonable).
8. *Mise à jour du compteur* du nombre de paquets.
9. *Enregistrement du débit,* pour cela nous enregistrons le débit dans le fichier résultat à chaque fois que la durée atteint le niveau de granularité souhaitée par l'utilisateur.

Caractérisation des flux UDP

L'analyse des paquets UDP est à peu de chose près similaire à l'analyse des paquets TCP. Cette caractérisation se fait lorsque le numéro de protocole trouvé dans l'en-tête du paquet (champ protocol) est égal à 17, c'est à dire lorsqu'il s'agit d'un paquet UDP encapsulé dans un paquet IP.

Le processus de caractérisation du paragraphe précédent reste valable pour les étapes 3 à 9 en évitant les étapes 5, 6 et 7 décrites ci-dessus. Néanmoins, certaines particularités différencient l'analyse TCP de l'analyse UDP. Tout d'abord, il n'est pas possible, avec le

protocole UDP, de connaître la fin d'un flux en consultant un champ de l'en-tête. C'est pour cela que nous avons recours à la fonction qui vérifie dans la liste complète des flux UDP s'il existe un flux inactif depuis au moins 900 secondes (soit 15 minutes).

Ensuite, seuls le nombre de paquets, le nombre de flux et la quantité d'octets nous intéressent dans la caractérisation des flux UDP. De ce fait, aucune distribution, aucun calcul de débit et aucune valeur d'inter-paquets ou d'inter-flux ne sont sauvegardées. Ces informations bien qu'intéressantes à analyser ne sont pas à l'heure actuelle extraites des traces. Ces fonctionnalités seront ajoutées dans la version suivante du logiciel si nécessaire.

Enfin, il est inutile pour le protocole UDP de conserver les paquets dans le flux puisque nous n'analysons aucun flux spéciaux de ce protocole comme nous le faisons avec TCP pour les flux Eléphants, Souris... De la même façon que pour le calcul des distribution la décomposition en classe de flux pour le trafic UDP n'est pas implémentée dans cette version du logiciel. Ces fonctionnalités feront partie de la prochaine version de ZOO.

Caractérisation des flux Eléphants, Souris, Tortues et Libellules

Dans la version actuelle de ZOO, il est possible d'analyser 4 types de flux différents selon leur taille ou leur durée. On définit ainsi qu'un flux Eléphant comporte au moins 100 paquets et un flux Souris comporte au plus 10 paquets. On définit également qu'un flux tortue dure au moins 900 secondes tandis qu'un flux Libellule dure au plus 2 secondes. Toutes ces données sont les valeurs par défaut du logiciel, elles sont issues de l'article [30] qui définit la méthodologie pour distinguer les différentes classes de flux au sein du trafic Internet. Néanmoins, l'utilisateur a la possibilité de les modifier s'il le souhaite.

La distinction Eléphant/Souris se fait donc en fonction du nombre de paquets. La distinction Tortue/Libellule, quand à elle, se fait en fonction de la durée du flux.

L'analyse de ces flux se fait pour le protocole TCP uniquement et lorsqu'un flux est achevé. En effet, il n'est évidemment pas possible de connaître la durée ou la taille d'un flux tant que ce dernier n'est pas terminé.

Pour tous les types cités ci-dessus la procédure d'analyse est la suivante :

1. *Mise à jour des compteurs* de paquets et de flux,
2. *Ajout des paquets du nouveau flux* dans la liste des paquets existants par ordre chronologique.
3. *Mise à jour des distributions* des tailles de paquets, d'inter-paquets, des tailles de flux et des durées de flux et enregistrements des valeurs dans le fichier résultat,
4. *Enregistrement ordonné* de la date de fin du flux,
5. *Enregistrement des débits* dans le fichier résultat.

Caractérisation des autres flux

Cette caractérisation se fait lorsque le protocole trouvé dans l'en-tête du paquet n'est ni TCP ni UDP. Dans ce cas, l'analyse est très simple puisque nous ne nous intéressons ici qu'au nombre de paquets et au volume d'octets échangés.

Le nombre de flux n'est dans ce cas pas comptabilisé car il est très difficile de définir la fin d'un flux lorsqu'on s'intéresse à plusieurs protocoles simultanément. En effet, il faudrait dans ce cas, s'intéresser à la sémantique de chaque flux pour déterminer quel type de protocole

FIGURE B.2 – Fenêtre d'introduction

de niveau transport ou application l'a généré. Cette méthode n'est à l'heure actuelle pas implémentée dans le logiciel ZOO.

B.2.3 Utilisation du logiciel

Nous allons présenter dans cette section les différentes fonctionnalités de l'interface graphique du logiciel ZOO (générée par l'intermédiaire de la bibliothèque Gtk 2.0 [63]).

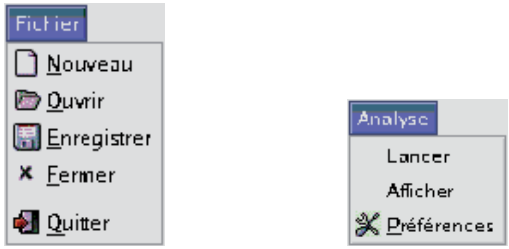
Fenêtre d'introduction

Au démarrage du logiciel ZOO, une fenêtre d'introduction (cf. figure B.2) s'ouvre en avant de la fenêtre principale. Cette fenêtre d'introduction présente le logiciel, les auteurs de ce projet et propose de continuer ou de quitter. Si le bouton *Non* est sélectionné, le programme est fermé tandis qu'un clic sur le bouton *Oui* ferme la fenêtre d'introduction et rend active la fenêtre principale.

Fenêtre principale

La fenêtre principale (cf. figure B.3) est composée de :
- *Une barre de menu :* cette barre permet d'accéder à toutes les fonctionnalités du logiciel. Elle inclut les menus Fichier, Analyse et Aide détaillés dans les sections qui suivent.
- *Une barre d'outils :* cette barre contient les fonctionnalités les plus utiles du logiciel à savoir Nouveau, Ouvrir, Enregistrer, Analyser, Afficher, Options et Quitter.
- *Un "notebook" :* cette zone permet d'afficher les graphes et les résultats statistiques obtenus après analyse.
- *Une barre d'état :* cette barre permet d'afficher des messages pour l'utilisateur (aide, message d'erreur, etc.).

FIGURE B.3 – Fenêtre principale



(a) Menu Fichier (b) Menu Analyse

FIGURE B.4 – Détails de la barre de menu principale

Le Menu Fichier

Le menu fichier (cf. figure B.4(a)) permet de gérer les projets et de quitter le programme. Il est composé des sous menus Nouveau, Ouvrir, Enregistrer, Fermer et Quitter.

– *Nouveau*
Le sous menu Nouveau permet de créer un nouveau projet [3] en spécifiant le répertoire et les noms de fichier dans lesquels seront enregistrés les résultats. En outre, la création d'un nouveau projet réinitialise toutes les options.
En cliquant sur Nouveau, et si aucun autre projet n'est déjà ouvert, la fenêtre de la figure B.5 s'ouvre alors.
En haut de cette fenêtre, un champ permet d'entrer le nom du répertoire (dans lequel les résultats seront sauvegardés) soit directement, soit à l'aide du bouton Parcourir

3. Un projet représente l'ensemble des paramètres d'une analyse, une trace sur laquelle réalisée cette analyse et l'ensemble des fichiers résultats obtenus après analyse.

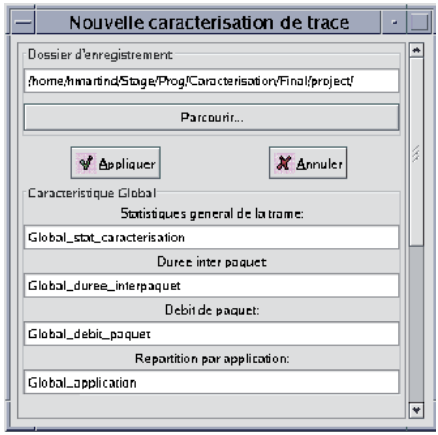


FIGURE B.5 – Fenêtre nouveau

qui permet de sélectionner un répertoire dans l'arborescence. Ensuite, l'utilisateur peut entrer les noms des fichiers dans lesquels seront enregistrés les résultats. Par commodité, toutes les entrées sont déjà remplies mais l'utilisateur a la possibilité de changer les noms de fichiers si les noms par défaut ne lui conviennent pas.
Enfin, l'utilisateur a la possibilité de cliquer sur Appliquer pour fermer la fenêtre et mémoriser les fichiers ou sur Annuler pour revenir à la fenêtre principale sans sauvegarder les noms de fichier.

– *Ouvrir*
Le sous menu Ouvrir permet d'ouvrir un projet déjà sauvegardé. Les données sauvegardées sont le nom du répertoire de sortie, les noms des fichiers et toutes les options modifiables par l'utilisateur. Une fois le projet ouvert, toutes les valeurs sont initialisées avec celles contenues dans le fichier, il est alors possible de lancer une nouvelle analyse ou d'afficher des anciens résultats.
En sélectionnant le sous menu Ouvrir, et si aucun autre projet n'est ouvert, une fenêtre s'ouvre. L'utilisateur a alors la possibilité de choisir le nom du projet.
– *Enregistrer*
Le sous menu Enregistrer permet d'enregistrer un projet ouvert. Les données sauvegardées sont le nom du répertoire de sortie, les noms des fichiers résultats et toutes les valeurs des options modifiables par l'utilisateur.
– *Fermer*
Le sous menu Fermer permet de fermer un projet ouvert et de réinitialiser l'affichage de la fenêtre principale. La fermeture d'un fichier permet à l'utilisateur de pouvoir ouvrir un autre projet car l'ouverture simultanée de deux projets est impossible.
– *Quitter*
Le sous menu Quitter permet de quitter le programme. Un message de confirmation apparaît alors pour s'assurer du choix de l'utilisateur.

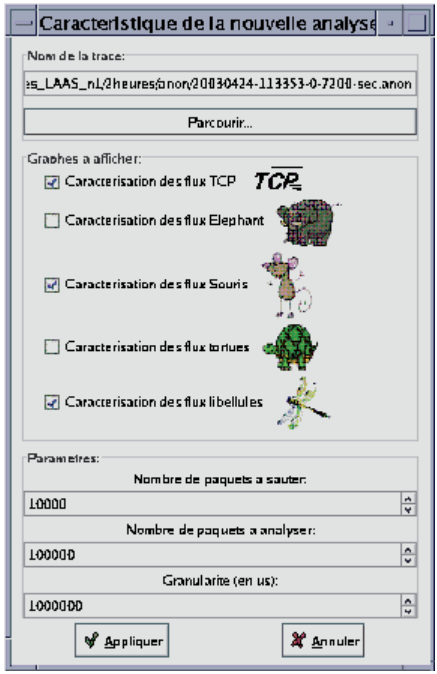


FIGURE B.6 – La fenêtre de lancement d'une analyse

Le Menu Analyse

Le menu Analyse (cf. figure B.4(b)) permet de gérer l'analyse d'un projet. Il est composé des sous menus Lancer, Afficher et Préférences.

- *Lancer*
  Le sous menu Lancer (cf. figure B.6) permet de lancer une nouvelle analyse à condition qu'un projet soit ouvert au préalable par l'utilisateur. Un clic sur ce menu provoque l'affichage d'une fenêtre permettant de choisir les paramètres principaux de l'analyse qui sont :
  - Le nom de la trace à analyser : il peut être choisi directement en entrant le nom dans le champ de saisie ou bien en passant par un sélectionneur de fichier pour choisir une trace dans l'arborescence.
  - Les graphes à afficher : en cliquant sur le "checkbox" associé, l'utilisateur peut désactiver l'analyse d'un type de flux. La désactivation permet de sauter l'analyse, la caractérisation et l'affichage d'un type de flux (cette possibilité est utile pour accélérer l'analyse globale par exemple).
  - Les paramètres principaux de l'analyse.

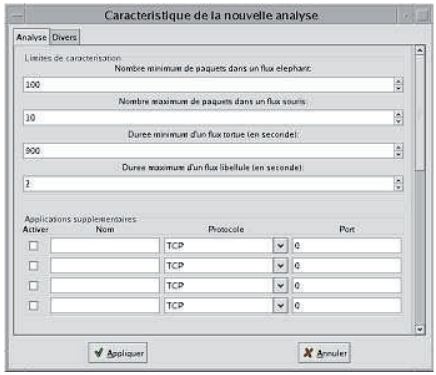


FIGURE B.7 – Les options d'analyse

Une fois les paramètres d'analyse sélectionnés, l'utilisateur peut choisir de lancer l'analyse en cliquant sur Appliquer ou de fermer simplement la fenêtre avec le bouton Annuler.

– *Afficher*
  Le sous menu Afficher permet d'afficher les graphes et les statistiques d'une ancienne analyse à condition qu'un projet soit ouvert (l'ouverture d'un projet permet en effet de connaître les noms des fichiers et les paramètres de l'analyse) et que tous les graphes au format *.PNG [4] existent dans le répertoire courant.
– *Préférences*
  Le sous menu Préférences permet d'afficher et de modifier les paramètres secondaires d'analyse. Les options sont divisées en deux catégories :
  – Les options d'analyse (cf. figure B.7) : l'utilisateur peut définir les limites caractéristiques des flux qu'il souhaite analyser, les applications supplémentaires qu'il souhaite caractériser et la largeur d'analyse pour les calculs d'auto-corrélation.
  – Les options diverses (cf. figure B.8) : l'utilisateur peut définir le coefficient de grossissement appliqué dans l'affichage des graphes et les noms des fichiers à conserver lors de la sortie du logiciel.

Le menu Aide (' ?')

Ce menu permet d'accéder à l'aide utilisateur ainsi qu'aux informations diverses du logiciel.

Après l'analyse

Une fois l'analyse terminée, le notebook est mis à jour et les graphes sélectionnés dans les paramètres principaux d'analyse s'affichent. Les figures B.9 et B.10 ci-dessous montrent un exemple typique d'affichage après analyse.

4. Le format PNG a été choisi pour sa gratuité et son caractère ouvert.

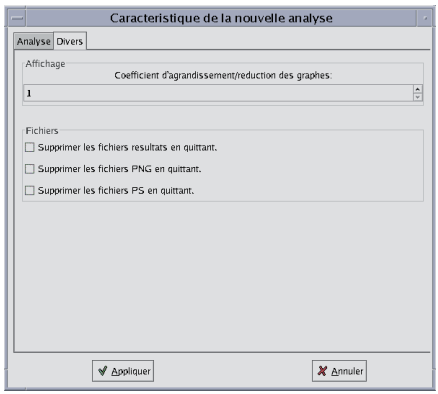


FIGURE B.8 – Les options diverses

Les premières pages du notebook présentent les statistiques globales de l'analyse puis présentent les résultats obtenus par type d'application. Les applications utilisateurs sont ajoutées à la suite des applications de base. Les pages suivantes affichent les graphiques générés après analyse en les regroupant par type (TCP, Eléphant, Souris, Tortue et Libellule) et par niveau (flux et paquet). Si l'analyse révèle un nombre insuffisant de paquets ou de flux d'un type particulier, les graphes de ce type ne peuvent pas être générés. Dans ce cas, un message d'erreur avertit l'utilisateur et les pages du type en question ne sont pas ajoutées au notebook.

FIGURE B.9 – Affichage des statistiques de l'analyse

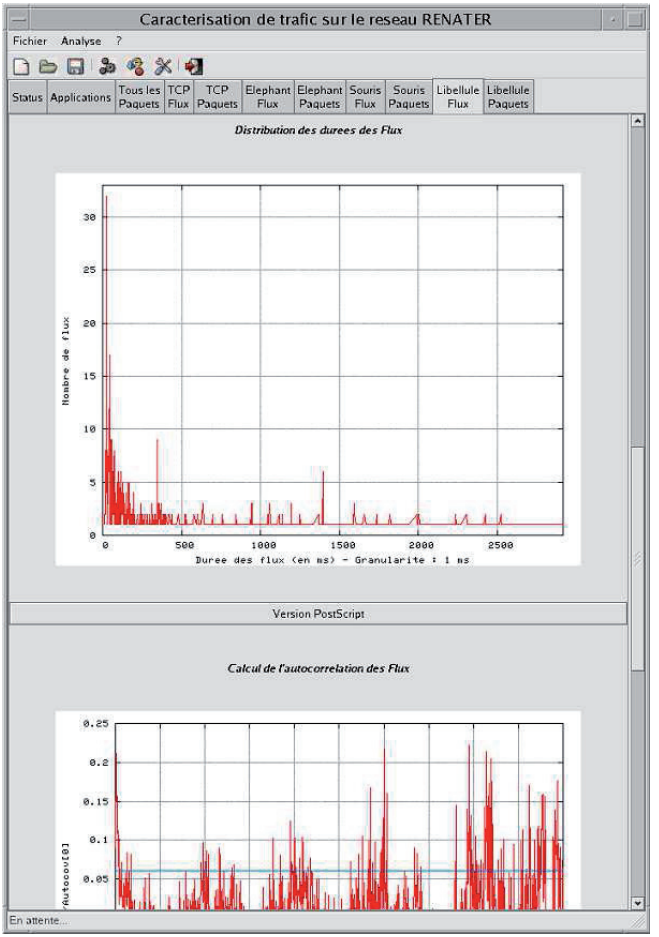


FIGURE B.10 – Affichage des graphiques générés après analyse

Annexe C

Une nouvelle méthode pour améliorer le réalisme des simulations

C.1 Problématique de la simulation des réseaux de l'Internet

C.1.1 Pourquoi est-il si difficile de simuler l'Internet ?

Les caractéristiques des trafics observés sur les différents liens que nous avons étudiés au cours des chapitres 1 et 2 montrent bien que le problème de la modélisation des trafics de l'Internet reste pour le moment une tâche ardue. En effet, de nombreuses approches reposant sur des modèles connus comme les modèles Poissonniens, Markoviens, les modèles ON/OFF [75], les modèles de files d'attente, M/G/1/N, Mk/Gk/1/N [62], les mouvements fractionnels Browniens caractéristiques des comportements auto-similaires, ou même les modèles à base de fractales [53] ne parviennent pas à représenter toutes les caractéristiques du trafic Internet sur un lien. Cette difficulté dans la modélisation des trafics engendre également des difficultés pour réaliser des simulations réalistes de l'Internet. C'est ce qui est mis en évidence dans [57] qui montre qu'il est extrêmement complexe de simuler de tels trafics, en particulier à cause des caractéristiques d'auto-similarité, LRD ou multi-fractalité qui ont pu être mises en évidence lors des premières analyses de trafic des liens de l'Internet. Les résultats du chapitre 2 ont bien montré que cette difficulté est une réalité, déjà sur un seul lien. Or, il apparaît également que les caractéristiques, et donc le modèle pour les représenter, sont différents d'un lien à un autre, sans qu'il soit aujourd'hui possible de connaître les règles de dépendance d'un lien à l'autre. L'ingénierie des réseaux de l'Internet à partir de modèles formels du trafic est donc une activité à développer, et qui prendra certainement de nombreuses années. Pour pouvoir continuer à développer et améliorer l'Internet actuel, il est toutefois essentiel de mettre en place des techniques de simulations réalistes. Cette notion de réalisme est un problème majeur des moyens de simulations qui existent aujourd'hui. Par exemple, dans les simulateurs actuels, les sources de trafic sont généralement des sources régulières comme des générateurs constants ou respectant des processus d'émission markoviens, soit, dans tous les cas, des sources de trafic plus régulières que le trafic de l'Internet. Cette régularité est assez dommageable pour le réalisme des simulations actuelles car les protocoles à étudier ne sont pas confrontés aux contraintes réelles du trafic, mais à des contraintes moins dures. Souvent, les protocoles

ou nouvelles architectures issues de ces simulations, et qui donnaient satisfaction lors des simulations ne fournissent pas les mêmes résultats lors des déploiements en environnement réel. Nous nous sommes donc attachés à utiliser la métrologie pour améliorer le réalisme des environnements dans lesquels sont faits les simulations Internet.

C.1.2 Les deux approches de simulation

De façon évidente, la métrologie modifie le processus d'ingénierie réseau en ajoutant en amont une phase de caractérisation et d'analyse du trafic, comme cela a été développé dans ce manuscrit. D'autre part, vu les difficultés à modéliser le trafic, les approches entièrement formelles qui pouvaient être utilisées jusqu'à présent dans l'ingénierie des réseaux et des protocoles (et représentées sur la figure C.1 comme l'approche 1), sont aujourd'hui difficilement applicables et le resteront tant qu'un modèle formel du trafic ne sera pas trouvé. De fait, nous sommes forcés de nous rabattre sur une approche de simulation informelle, avec un simulateur comme NS-2 par exemple, qui est le simulateur officiel de l'IETF. Avec ce type de simulateur, comme avec des simulateurs utilisant des approches formelles, il est difficile d'obtenir des résultats réalistes pour les mêmes raisons que précédemment : les sources de trafic ne prennent pas en compte toutes les caractéristiques d'irrégularité du trafic Internet, ni qualitativement, ni quantitativement. Pour cela, l'approche que nous proposons (et représentée comme l'approche 2 sur la Figure C.1) consiste à utiliser la métrologie en rejouant les traces capturées par les équipements de métrologie dans le simulateur, de façon à avoir des sources de trafic réalistes et reproduisant les comportements des utilisateurs et de leurs applications. La suite va donc présenter comment fonctionne la méthode de rejeu de traces de métrologie que nous avons développée.

C.2 Présentation de la méthode de rejeu

C.2.1 Les outils de simulation

L'objet de ce paragraphe est de présenter le module que nous avons développé pour le simulateur NS-2 qui permet de jouer des traces de trafic réel afin de simuler dans un environnement réaliste de nouveaux protocoles ou de nouvelles architectures Internet. Ce module permettra une vérification des comportements des nouveaux protocoles qui seront imaginés par la suite par les chercheurs. Le module de simulation de traces réelles développé dans le cadre de ce travail de thèse est destiné à s'intégrer à la distribution la plus récente du simulateur NS-2 disponible au début de cette thèse : la version 2.1 beta 8 a.

Présentation de l'outil de simulation utilisé : le simulateur NS-2 de l'IETF

Le projet VINT [131] est une collaboration entre USC / ISI, Xerox, LBNL et UCB. Il a eu pour objectif principal de construire un simulateur multi-protocoles pour faciliter l'étude de l'interaction entre les protocoles et le comportement d'un réseau à différentes échelles. Le projet contient des bibliothèques pour la génération de topologies réseaux, des trafics ainsi que des outils de visualisation tel que l'animateur réseau NAM. L'étude des comportements à des échelles différentes d'un réseau n'est pas obtenue par la simulation parallèle (qui peut être utile pour accélérer la simulation) mais par l'utilisation de techniques d'abstraction appliquées à différents éléments de la simulation. VINT est un projet qui a donné naissance au

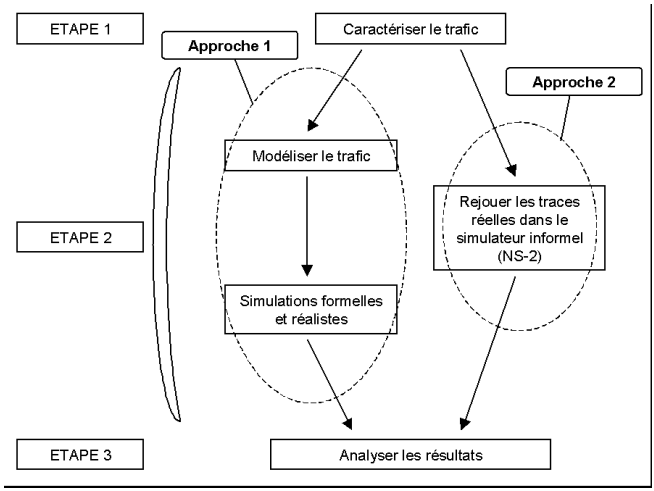


FIGURE C.1 – Processus de recherche en réseau

simulateur NS-2. Le simulateur NS-2 actuel est particulièrement bien adapté aux réseaux à commutation de paquets et à la réalisation de simulations de petites tailles. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage point à point ou multipoint, des protocoles de transport, de session, de réservation, des services intégrés, des protocoles d'application comme HTTP, etc. De plus, le simulateur possède déjà une palette de systèmes de transmission (couche 1 de l'architecture TCP / IP), d'ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion. La liste des principaux composants actuellement disponibles dans NS-2 par catégorie est :

- *Application* : Web, FTP, Telnet, générateur de trafic (CBR, VBR, etc.) ;
- *Transport* : TCP, UDP, RTP, SRM, TFRC ;
- *Routage* : statique, dynamique (à base de vecteur distance) et routage multipoint (DVMRP, PIM) ;
- *Gestion de file d'attente* : RED, DropTail, Token bucket ;
- *Discipline de service :* CBQ, SFQ, DRR, Fair queueing ;
- *Système de transmission* : CSMA / CD, CSMA / CA, lien point à point.

Prises ensembles, ces capacités ouvrent le champ à l'étude de nouveaux mécanismes au niveau des différentes couches de l'architecture réseau. NS-2 est devenu l'outil de référence pour les chercheurs du domaine. Ils peuvent ainsi partager leurs efforts et échanger leurs résultats de simulations voire leurs modules protocolaires. Cette façon de faire se concrétise aujourd'hui par l'envoi dans certaines listes de diffusion électronique de scripts de simulations NS-2 pour illustrer les points de vue. NS-2 est un outil de simulation de réseaux de données. Il est bâti autour d'un langage de programmation appelé TCL dont il est une extension. Du point de vue de l'utilisateur, la mise en œuvre de ce simulateur se fait via une étape de programmation

qui décrit la topologie du réseau et le comportement de ses composants, puis vient l'étape de simulation proprement dite et enfin l'interprétation des résultats. Cette dernière étape peut être prise en charge par un outil annexe, appelé NAM qui permet une visualisation et une analyse des éléments simulés. Le déploiement de nouvelles fonctionnalités dans l'Internet est une tâche difficile et lourde de conséquences. En effet, on ne peut pas se permettre de mettre en place, par exemple un nouveau protocole, si celui-ci n'a pas fait l'objet de test répétés qui garantissent son comportement en environnement réel. Les différents utilisateurs de l'Internet, demandent une fiabilité toujours croissante étant donné le rôle économique des échanges réalisés sur ce réseau. Ainsi, pour permettre un déploiement optimal des nouveaux services de l'Internet, une phase de simulation réaliste doit avoir lieu en amont pour permettre la vérification des nouveaux protocoles. Dans ce but, la métrologie couplée avec l'utilisation du module NS-2 permettant le rejeu de trafic réel doit permettre la réalisation de cette tâche. Nous allons maintenant présenter le module que nous avons développé pour le simulateur NS-2.

C.2.2 Utilisation de NS-2 dans un contexte métrologique

NS-2, simulateur officiel de l'IETF, permet de rejouer des traces. Il faut cependant appliquer un certain nombre de traitements aux traces métrologiques brutes pour permettre ce rejeu. Certaines modifications du code de NS-2 sont également nécessaires. Les paragraphes suivants visent à présenter les différents outils et modifications que nous avons réalisés pour permettre de rejouer des traces réelles dans NS-2.

De la trace brute à NS-2 : les traitements à mettre en place

NS-2 ne fournit qu'une seule classe pour rejouer une trace réelle : la classe "trafficTrace" qui ne permet de générer qu'un unique flux à partir d'un fichier représentant la liste des tailles des paquets de ce flux. Il nous faut donc créer et fournir au script de simulation un fichier par flux à rejouer. D'autre part, comme les dates de début des flux sont à préciser directement dans le script de simulation, il nous faut également lui fournir un fichier contenant les dates de début de flux et le nom des fichiers traces associés. En conséquence, nos outils de traitement doivent :

- Parcourir la trace, classer les paquets TCP par flux en enregistrant pour chaque flux la date de début et la taille de chaque paquet composant le flux.
- Créer un fichier de trace par flux. Le format de ce fichier, imposé par le "trafficTrace" du simulateur, doit correspondre à une succession d'enregistrements de la forme :
  typedef struct_trec {
  unsigned int trec_time ; //durée inter-paquet
  unsigned int trec_len ; //taille du paquet
  } trec ;
  Comme TCP gère lui-même les instants d'envois des paquets par flux, on donne au paramètre de durée inter-paquet la valeur 0 lorsqu'on rencontre un paquet TCP dans la trace.
- Créer un fichier contenant les dates de début de chaque flux et le nom des fichiers traces associés. Le détail pratique de ces diverses tâches est exposé dans la section qui suit.

Rejeu de la trace dans le simulateur NS-2

Une fois les traitements présentés ci-dessus effectués, il faut lancer un script NS utilisant les fichiers traces et le fichier info. Vous trouverez dans la section qui suit un script NS de base qui met en place les éléments pour rejouer les fichiers traces et le corps général du script. Pour finir, il faut préciser que pour respecter exactement nos choix, en particulier le respect de la taille des paquets, il nous a fallu apporter des modifications au code C++ de NS-2. En effet, les agents NS-2, éléments de base pour les protocoles de niveau transport, ont un fonctionnement basé sur une taille de paquet fixe : ainsi, les agents UDP ou TCP, pour un flux donné, émettent des paquets de taille constante. Nous avons donc été obligés de modifier ces composants pour que les paquets que nous voulions rejouer soient émis à la taille voulue et non pas à une taille fixe. Ainsi, nous avons apporté des modifications aux classes C++ suivantes : agent.cc, tcp.cc, udp.cc et traffictrace.cc.

C.2.3 Détails de fonctionnement du module NS-2 permettant le rejeu de trafic réseau

Cette section décrit les différentes étapes nécessaires au rejeu sous NS-2 de trafic à partir de traces réseau réelles.

Extraction des flux de la trace brute

Cette étape permet d'extraire tous les flux TCP actifs et terminés d'une trace de les enregistrer dans un fichier sous le format suivant :

[0] //1er flux à se terminer : syntaxe pour séparer les infos de chaque flux
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 ...
[1] //2ème flux à se terminer
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 ...
...
[N] //(N+1)iéme et dernier flux à se terminer
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 ...

Cette tâche est réalisée par le programme en langage C “traceExtracteur.c”. Il parcourt tous les paquets de la trace et classe les différents paquets TCP par flux. Lorsque la terminaison d'un flux TCP est détectée (lecture d'un paquet TCP FIN), le flux est stocké dans le fichier résultat selon le format présenté ci-dessus. Tous les flux qui ne sont pas terminés à la fin de l'analyse de la trace ne sont pas enregistrés.

La ligne de commande permettant de réaliser cette étape est la suivante :

$$./traceExtracteur\ nom\_fichier\_trace\ nom\ nombre\_de\_paquets\_1\ nombre\_de\_paquets\_2 \tag{C.1}$$

Où :

– *nom_fichier_trace* : nom du fichier trace à analyser.
– *nom* : nom du fichier résultat dans lequel sont stockés les flux. C'est également le nom de base à partir duquel travaillent les programmes suivants. A noter que le programme crée également le fichier "nom_info" contenant le nombre de flux analysés et la durée de l'analyse. Ces informations seront utiles pour les étapes suivantes.
– *nombre_de_paquets_1* : il s'agit du nombre de paquets à sauter avant de commencer l'analyse.
– *nombre_de_paquets_2* : il s'agit du nombre de paquets sur lesquels va porter l'analyse.

Ordonnancement des flux

Cette étape permet de trier chronologiquement les flux du fichier généré par le programme précédent. Elle n'est pas obligatoire mais permet une meilleure vision du trafic global. Elle est réalisée par le programme en langage C "trieur.c" qui retourne un fichier formaté comme suit :

[0] //1er flux à débuter : syntaxe pour séparer les infos de chaque flux
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 ...
[1] //2ème flux à débuter
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 ...
...
[N] //(N+1)iéme et dernier flux à débuter
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 ...

La ligne de commande permettant de réaliser cette étape est la suivante :

$$./trieur\ nom \tag{C.2}$$

Où :
– *nom* : c'est le nom de base passé en paramètre au programme traceExtracteur.
Le fichier résultat se nomme "nom_trie".

Création des fichiers traces des différents flux

Le programme en langage C "preparationNS.c" travaille sur le fichier nom_trie généré lors de l'étape 2 :
– il crée les différents fichier traces correspondant chacun à un flux au format imposé par NS-2 ;
– il actualise le fichier "nom_info" avec les informations suivantes : début de chaque flux et nom des fichiers flux associés générés dans cette étape.

La ligne de commande permettant de réaliser cette étape est la suivante :

$$./preparationNS\ nom \tag{C.3}$$

Où :

– *nom* : c'est le nom de base passé en paramètre au programme traceExtracteur.

Simulation de la trace

La ligne de commande pour lancer le script est :

$$ns\ script.tcl\ nom\_info \tag{C.4}$$

Où :
– *script.tcl* : script de simulation basé sur le script topologie_base.tcl,
– nom_info : c'est le nom du fichier contenant les informations nécessaires à la simulation (le nom des fichiers flux, la date de début de ces flux, etc.).

Le script de base permettant le rejeu des flux TCP contenus dans une trace réelle doit comporter les étapes suivantes :

Récupération du nom du fichier info contenant le nombre de flux à rejouer, ...
Ouverture du fichier
Lecture des paramètres contenus dans le fichier
Paramètres de la simulation
Description de la simulation
Création un objet simulateur
Création d'un fichier nam
Création d'un fichier de trace permettant de récolter les résultats de la simulation
Création des generateurs de trafic
#————————————————#
Description de la topologie :
création des noeuds et des liens
#————————————————#
Création des noeuds
Création des liens
#————————————————#
Création des différents agents et sources
(avec connection des sources aux agents et
des agents aux noeuds)
Connection émetteurs/récepteurs
#————————————————#
Création des agents et sources émetteurs
Création des agents récepteurs
Connection émetteurs/récepteurs
#————————————-#
Mise en place du scénario
————————————#
Description du scénario
Choix des résultats de la simulation à écrire dans le fichier de trace
Procédure de fin
Ferme les fichiers résultats
Execute nam sur le fichier resultat
Lancement de la simulation

Définition de la topologie de simulation

Etant donné que l'environnement de simulation a la lourde tâche de mettre en forme le profil d'émission il est très important de le créer de façon à générer du trafic ayant les mêmes caractéristiques que le trafic réel. Nous avons vu, dans les paragraphes précédents, le rôle des agents d'émission et de réception NS qui vont injecter dans le réseau les flux aux moments opportuns et selon les tailles des paquets lus dans la trace réelle. En particulier, pour rejouer intégralement une trace les éléments caractéristiques du trafic réel qui doivent intervenir sont :

- les dates relatives des débuts de flux qui représente le comportement réel et aléatoire des utilisateurs ;
- les tailles des paquets à l'intérieur de chaque flux ; il a été en effet montré que c'est un des éléments qui peut être à l'origine de l'auto-similarité du trafic, propriété que l'on souhaite étudier dans les résultats de simulation.

Nous allons maintenant nous intéresser à la définition de la topologie de simulation nécessaire. Les travaux précédents sur les caractéristiques du trafic Internet ont montré qu'il possédait des propriétés de LRD, auto-similarité qui étaient dûes à TCP et ses mécanismes de contrôle de congestion [96] [130]. Comme les mécanismes sont basés sur une réponse prédéfinie aux pertes, il apparait que les principales caractéristiques du trafic réel à améliorer dans les simulations sont associées au processus de pertes. De plus, il apparait que le RTT [1] est un paramètre important pour les mécanismes de contrôle de congestion. En effet, il joue un rôle primordial pour les performances de TCP et le profil du trafic. Ainsi, en respectant ces paramètres pour les flux simulés nous devrions améliorer de façon importante le réalisme des simulations. Il faut noter que l'approche défendue dans cette annexe ne se focalise pas sur des topologies réseaux très complexes. En fait, la topologie générique que nous proposons est la plus simple qui soit capable de reproduire aussi précisément que possible le taux de perte [2] et le RTT moyen observés dans les traces de trafic réelles. Pour cela, nous analysons chaque flux de la trace originale et nous mesurons le taux de perte. L'objectif est donc de reproduire grâce à la topologie de simulation le même taux de perte originalement mesuré pour chacun des flux. De façon à construire une topologie réseau adaptée, il est aussi nécessaire d'extraire des traces originales les autres paramètres des flux :

- le taux de perte expérimenté par chacun des flux pendant leur transfert sur le réseau ;
- le RTT expérimenté par chacun des flux ;
- le débit moyen obtenu par chacun des flux ;
- la durée de chaque flux.

Pour limiter la complexité de la topologie de simulation, et en se basant sur les analyses des taux de pertes, nous avons décidé de définir seulement six classes de taux de pertes différentes (cf. tableau C.1 pour détails).

Avec les informations extraites des traces originales, nous sommes capables de déduire à la fois la bande passante et la taille de la file d'attente de chacun des liens de la topologie de simulation où le flux devra être transmis, en fonction de la classe de flux auquel le flux appartient. La bande passante du lien pour la classe i (Bw_{Ci}) est calculée selon l'équation :

$$Bw_{Ci} = \frac{\sum_{n=1}^{N_{flux}} d_i * Th_i}{d_{trace}} \tag{C.5}$$

1. Il s'agit du temps aller retour nécessaire pour échanger une information entre la source et la destination.
2. Nous espérons ainsi reproduire le taux de perte réel.

TABLE C.1 – Classes de flux utilisées pour rejouer les traces de métrologie

Classe	Taux de perte de la classe (%)
C0	0
C10	0-10
C20	10-20
C30	20-30
C50	30-50
C100	50-100

Où :
– N_{flux} est le nombre de flux de la classe i,
– d_i est la durée du flux i,
– Th_i est la moyenne du débit du flux i,
– d_{trace} est la durée de la trace.
Ainsi la taille de la file d'attente de la classe i (QL_{Ci}) est déduit selon l'équation :

$$QL_{Ci} = Bw_{Ci} * (100 - taux_{perte}) \tag{C.6}$$

Où :
– $taux_{perte}$ est la moyenne du taux de perte (en %) obtenu par chaque flux de la classe.
Au final, la topologie expérimentale qui sera utilisée pour rejouer la trace considérée dans l'exemple est décrite dans la figure C.2 (RTT_{Ci} est la moyenne des RTT de l'ensemble des flux appartenant à la classe i).

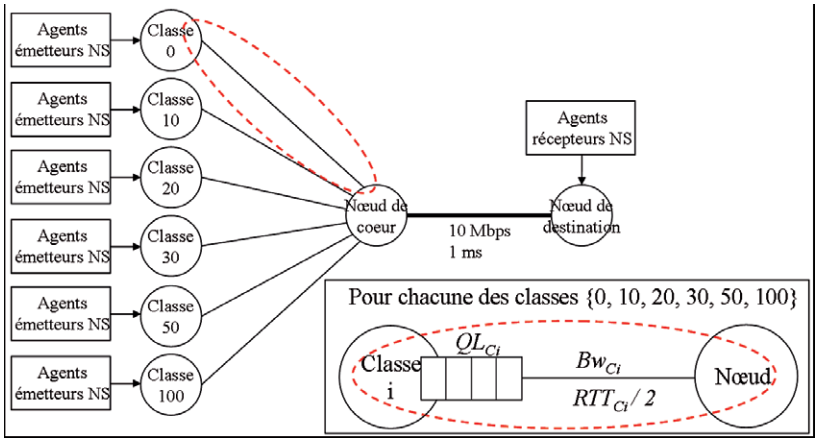


FIGURE C.2 – Topologie expérimentale

Des traces ont été capturées sur un lien Ethernet à 10 Mbps (réseau du LAAS-CNRS, cf. annexe E.3 pour détails). C'est pourquoi dans la topologie expérimentale, le lien de coeur a une capacité de 10 Mbps. La valeur du délai pour ce lien est de 1 ms pour éviter toute influence sur l'émission des flux des différentes classes [3]. En effet, la moyenne des RTT pour chaque classe de perte a été calculée et positionnée sur les différents liens d'accès pour les six classes considérées.

C.3 Evaluation de la méthode de rejeu

C.3.1 Objectifs de l'évaluation

Les travaux de [100] et [57] recommandent de considérer pour les traces de trafic des invariants que nous avons observés dans nos traces de trafic. Ces invariants que nous avons mis en évidence au préalable (cf. chapitres 1 et 2) sont :

- une corrélation à long terme existant entre les paquets ;
- une distribution de la taille de flux en loi de puissance (log-normale le plus souvent) ;
- une distribution à queue lourde pour ce qui est de l'activité du réseau.

C.3.2 Principes de l'évaluation

Ainsi dans la suite, l'évaluation de la méthode de rejeu sera faite en comparant le trafic réel qui a été capturé et le "même" trafic rejoué dans le simulateur. Les paramètres qui sont comparés sont bien évidemment les paramètres traditionnels du trafic (débit, nombre de paquets, ...), mais aussi toute les paramètres qui sont en relation avec la dynamique du trafic, en particulier les moments statistiques d'ordre deux comme l'autocorrélation du trafic ou la LRD.

Dans la prochaine section, nous présentons plusieurs résultats expérimentaux afin de valider notre approche de rejeu. Nous avons testé notre méthode de rejeu sur un grand nombre de traces, et nous avons obtenus les mêmes résultats avec chacune d'elles, en les comparant au trafic rejoué. Nous rappellons que pour chaque analyse nous nous focalisons principalement sur l'ensemble de la dynamique du trafic : les paramètres les plus difficiles à reproduire et à contrôler en simulation et qui sont responsables de la plupart des problèmes de performances de l'Internet.

C.3.3 Resultats d'analyse

Tout d'abord, la moyenne du taux de perte que nous avons obtenu en simulation est la même que le taux réel pour chacune des classes considérées. Mais le problème principal relatif à cette partie du trafic est de pouvoir obtenir des simulations où la mise en forme des paquets est similaire au cas réel. Pour cela, les temps d'inter-arrivées des paquets sont analysés de deux façons : en simulation et dans le réseau réel. La figure C.3 présente la représentation Q-Q-Plot des deux séries d'inter-arrivées. Il apparait que la correspondance entre les séries simulée et réelle est très bonne pour l'ensemble des valeurs à l'exception des très grands quantiles. En effet, les deux analyses montrent que la seule différence vient de la proportion de paquets très proches temporellement qui est plus importante dans la trace réelle que dans

3. Il devrait être de 0 mais cette valeur est impossible à positionner dans notre version de NS.

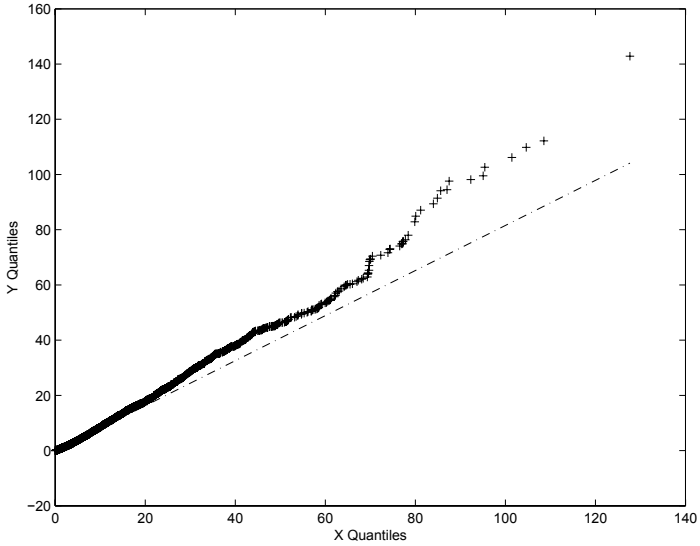


FIGURE C.3 – Q-Q Plot des instants d'inter-arrivées des paquets (ms) – données de la trace (axe des Y) vs. données de la simulation (axe des X)

celle simulée. Dans le cas réel, les paquets séparés par des durées très courtes sont celles des flux qui expérimentent un RTT très bas. En simulation, étant donné que nous avons défini pour tous les flux d'une classe le même RTT, les flux avec des RTT courts ne sont pas très bien rejoués.

Notre technique de simulation orientée rejeu de trace a prouvé qu'elle donnait de bons résultats pour les statistiques de premier ordre (cf. fonction de distribution analysée grâce à l'étude de la représentation en Q-Q Plot). Mais pour vérifier que les deux processus qui génèrent les deux traces (réelle et simulée) sont identiques, il est aussi nécessaire de montrer qu'ils ont le même moment à tous les ordres statistiques. D'un point de vue pratique, le troisième ordre et au-delà ont très peu d'influence. Dans de telles évaluations expérimentales, il est généralement admis qu'il est suffisant de faire la vérification pour les deux premiers ordres, ceci validant que les deux processus étudiés sont similaires. C'est pourquoi la figure C.4 représente la fonction d'auto-corrélation pour les deux cas. Il apparait clairement sur la figure C.4 que notre simulation orientée trace donne de très bons résultats pour les statistiques de second ordre, ce qui est un des problèmes principaux quand on souhaite rejouer du trafic.

Pour compléter notre analyse, il est nécessaire de calculer la LRD du trafic. En fait la LRD donne une évaluation de la dépendance induite dans le trafic à toutes les échelles. L'objectif est donc de regarder pour chaque échelle temporelle, si la dépendance dans le trafic est la même pour la trace simulée et réelle. La LRD a été calculée en utilisant l'outil LDEstimate [1] dont l'utilisation est détaillée en annexe D. Les résultats dans les deux cas sont présentés

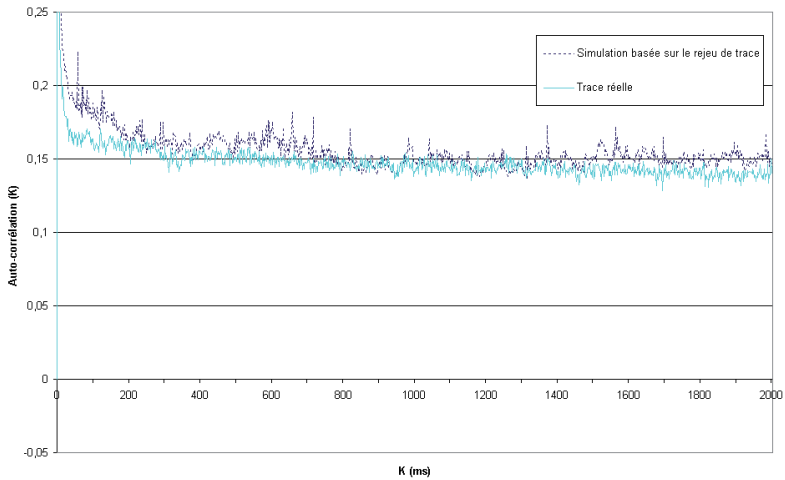
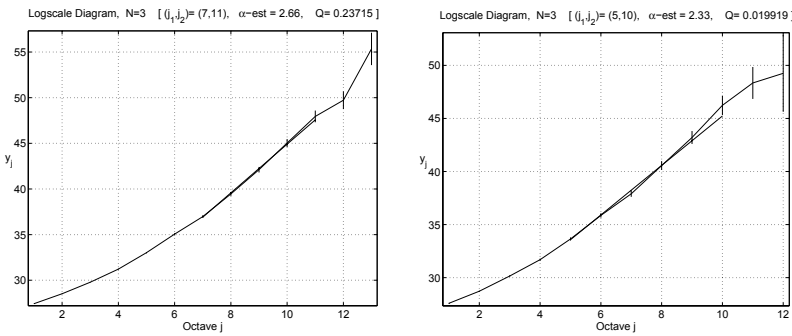


FIGURE C.4 – Fonction d'autocorrélation des instants d'inter-arrivées des paquets



(a) Trace originale

(b) Simulation à base de trace rejouée

FIGURE C.5 – Diagrammes LDEstimate des instants d'inter-arrivées des paquets (ms)

sur la figure C.5. Il apparait que la LRD pour notre trace rejouée et pour la trace réelle sont très proches. Ceci signifie que le trafic réel est hautement dépendant à long terme et que notre approche est capable de parfaitement reproduire ce niveau de complexité sur les larges échelles de temps. Ce résultat était attendu étant donné que la LRD est due à la distribution des tailles de flux à queue lourde ou log-normale [96] et notre approche de simulation rejoue les flux avec leur taille réelle.

C.4 Conclusion

Nous avons décrit dans cette annexe une nouvelle approche orientée trace pour la simulation Internet, et plus particulièrement basée sur le rejeu de traces de trafic capturées par un système de métrologie passif. Cette approche à l'avantage important de construire des sources de trafic simulées possédant des caractéristiques et spécificités réalistes en terme de processus d'arrivée (de flux, de sessions) en accord avec ce qui est observé dans l'Internet. Elle semble être un bon moyen pour arriver à obtenir des simulations réalistes, et ceci tant qu'un modèle complet du trafic Internet ne sera pas disponible. Nous avons aussi proposé une technique pour construire les topologie de simulation qui est basée sur la connaissance du taux de perte de chaque flux. L'objectif est ici de pouvoir reproduire aussi précisément que possible le processus de perte, étant donné que c'est un paramètre essentiel dans la génération du profil des paquets Internet (tout du moins pour le trafic TCP).

Les résultats obtenus avec notre approche de simulation basée sur le rejeu de trace ont montré que les trafics simulés reproduisent la complexité des trafics actuels et particulièrement leur dynamique. Nos résultats de simulation sont vraiment encourageants et montrent que la simulation profite grandement de l'analyse métrologique du réseau. Cependant, nous avons vu dans la section C.3.3 quelques petits défauts notamment en ce qui concerne les paquets séparés par des durées très courtes. Ce point devra être abordé lors de travaux futurs. Nous essaierons pour cela de définir une méthode permettant de construire une topologie de simulation qui se base à la fois sur des taux de pertes variables mais aussi des RTTs différents à l'intérieur d'une même classe de perte.

Annexe D

Détails de fonctionnement de l'outil LDEstimate

Cet outil permet une quantification du phénomène de dépendance mesurable dans une série de données. Les séries temporelles, telles que le nombre de paquets transmis par unité de temps par exemple, y sont analysées en termes d'un spectre d'énergie établi en fonction d'un facteur d'échelle temporel. Nous avons récupéré le code source librement disponible pour disposer d'une méthode initiale de calcul du facteur de Hurst. Nous souhaitons préciser que nous avons eu l'aide d'un des deux auteurs de cet outil, Patrice ABRY, pour utiliser son outil sur nos traces de trafic Internet.

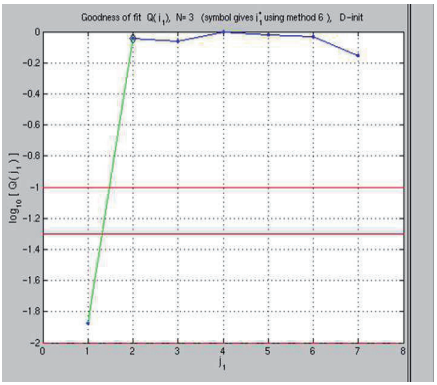
La marche à suivre pour utiliser cette méthode est la suivante :

- Il faut disposer d'un processus à analyser sous la forme d'une série d'estampilles temporelles (durées inter-paquets ou série de débits par unité de temps par exemple).
- Il faut disposer de la version 6 release 12 de Matlab.
- La méthode de calcul se lance par l'intermédiaire de la fonction Matlab suivante :

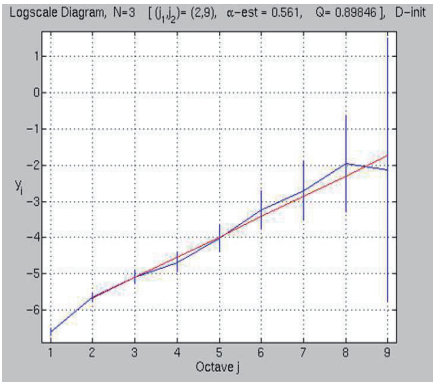
$$LDestimate(data, regu, j_1, j_2, discrete\_init, calcj_1, printout) \tag{D.1}$$

Les différents paramètres d'entrée signifient :

- *data* : la donnée d'entrée est un vecteur ligne (série d'estampilles temporelles) ;
- *regu* (regularity) : le nombre de "vanishing" moments (2 permet une précision suffisante pour une série de 10.000 points) ;
- j_1 : la borne inférieure pour l'échelle choisie (1 < j1 < valeur_max-1) ;
- j_2 : la borne supérieure pour l'échelle choisie (2 < j2 < valeur_max) ;
- *discrete_init* (1 par défaut) :
  1 : applique une initialisation MRA pour obtenir une série de valeurs discrètes ;
  autre : implique que les données d'entrée sont déjà initialisées (i.e. il s'agit déjà de valeurs approximées) ;
- $calcj_1$ (1 par défaut) :
  1 : lance la fonction $newchoosej_1$ qui va afficher un graphique de $Q(j_1)$ en fonction de j_1, ainsi que retourner la valeur optimale de j_1 ;
  else : ne lance pas la fonction $newchoosej_1$;
- *printout* (1 par défaut) :
  1 : un graphique log-log est affiché avec l'interpolation linéaire des principaux points, les différents résultats sont affichés (dont le paramètre H) ;



(a) $calcj_1 = 1$



(b) $printout = 1$

FIGURE D.1 – Résultats graphiques obtenus par la méthode par ondelettes

autre : rien n'est affiché.

Les résultats que l'on peut obtenir sont représentés par la figure D.1 (le graphique de gauche correspond à $calcj_1 = 1$ et celui de droite à $printout = 1$).

En plus des résultats graphiques, la méthode implémentée par Patrice Abry et Darryl Veitch donne à chaque pas de calcul la valeur du paramètre de Hurst pour la précision courante et permet de raffiner le calcul en précisant un nouvel intervalle de valeurs $[j_1; j_2]$ et ainsi relancer l'algorithme pour une nouvelle itération.

L'interprétation de ces résultats a déjà exposée dans le premier chapitre de ce manuscrit, nous ne reviendrons donc pas dessus dans cette annexe.

Annexe E

Listes des différentes traces analysées

E.1 Trace FT : cœur de réseau

Il s'agit d'un fichier qui contient 271 millions de durées inter-paquets (IP) consécutives. Ces données ont été collectées sur le réseau commercial de France Telecom et plus particulièrement dans leur cœur de réseau Parisien, sur un lien OC48 (2,5 Gbps). Cette trace est intéressante car elle représente un degré d'agrégation bien supérieur aux autres traces dont nous disposons. Ces paramètres principaux sont :

- Nombre des durées inter-paquets : 271 millions ;
- Moyenne des durées : 26,0114 ms.

E.2 Trace FT : lien montant d'une plaque ADSL

Il s'agit d'une trace collectée sur le lien montant (technologie Gigabit Ethernet) d'une plaque ADSL parisienne. Les caractéristiques principales de cette trace sont :

- Date de capture : mercredi 15 octobre 2004 ;
- Emplacement de la sonde de capture : dérivation d'un lien à haut débit connectant différentes plaques ADSL de la région parisienne de Fontenay aux Roses (seul le trafic du lien descendant est analysé) ;
- Heure de départ : 19H01 ;
- Durée de la capture : 1300 secondes ;
- Nombre total de paquets : 134.434.541 ;
- Nombre total de flux : 9.636.105.

E.3 Trace Renater de réseau d'accès – LAAS-CNRS

Il s'agit d'un trafic collecté sur le lien d'accès du LAAS-CNRS à RENATER qui est un lien Fast-Ethernet. Les caractéristiques principales de cette trace sont :

- Date de capture : lundi 2 juillet 2003 ;
- Emplacement de la sonde de capture : lien de sortie du LAAS-CNRS ;
- Heure de départ : 10H50 ;

– Durée de la capture : 3600 secondes ;

E.4 Trace Renater de réseau de bordure – Jussieu

Il s'agit d'un trafic collecté sur le lien d'accès du réseau de Jussieu à RAP puis Renater qui est un lien Giga-Ethernet. Les caractéristiques principales de cette trace sont :

– Date de capture : lundi 1er octobre 2004 ;
– Emplacement de la sonde de capture : réseau du campus de Jussieu ;
– Heure de départ : 14H50 ;
– Durée de la capture : 3600 secondes ;
– Nombre total de paquets : 80.437.378 ;
– Nombre total de flux : 2.322.931.

E.5 Trace NLANR

Il s'agit de traces collectées par le groupe de travail WAND du NLANR (université de Waikato, Nouvelle Zélande) sur le point de présence d'Auckland. Elles ont été capturées par l'intermédiaire de sonde DAG sur un réseau local Ethernet à 10 Mbps : pour détails voir http ://pma.nlanr.net/Traces/long/auck4.html.

E.6 Trace SPRINT

Il s'agit de traces collectées sur le réseau de l'opérateur américain SPRINT en 2000 dans le cadre du projet IPMON (cf. chapitre 1). Les liens analysés sont des liens 0C12 et il s'agit de différents P0P situés un peu partout sur le territoire américain : San Diego, Boston ou encore Houston. Pour plus de détails, consulter le site : http ://ipmon.sprint.com.

Table des figures

Liste des tableaux

Bibliographie

Publications de l'auteur

Revues internationales

[Owe03a] P. OWEZARSKI, N. LARRIEU, *Coherent charging of differentiated services in the Internet depending on congestion control aggressiveness*, Computer Communications Journal, Issue 13, Vol.26, pp.1445-1456, August 2003.

[Lar04a] N. LARRIEU, P.OWEZARSKI, *Towards a measurement based networking approach for Internet QoS improvement*, Rapport LAAS N. 04193, 29p., Décembre 2004, accepté pour publication dans la revue Computer Communications.

[Sal05] K. SALAMATIAN, S. D'ANTONIO, J. DOMINGO-PASCUAL, M. ESPOSITO, M. JANIC, N. LARRIEU, I. MARSH, P. OWEZARSKI, T. ZSEBY, *Internet measurements : state and some challenges*, Rapport LAAS N. 05008, Janvier 2005, 21p., accepté pour publication dans la revue Computer Communications.

Revue nationale

[Lar04b] N. LARRIEU, P. OWEZARSKI, *De la métrologie pour l'ingénierie des réseaux de l'Internet*, revue Technique et Sciences Informatiques, numéro thématique "Réseaux et Protocoles", vol. 23, n. 5-6/2004, septembre 2004, 33 p.

Conférences internationales

[Lab05] Y. LABIT, P. OWEZARSKI and N. LARRIEU, *Evaluation of active measurement tools for bandwidth estimation in real environment*, Rapport LAAS N. 05093, Mars 2005, 10p, IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON) 15th May 2005, Nice-Acropolis, Nice, France.

[Lar03a] N. LARRIEU, P. OWEZARSKI, *TFRC contribution to Internet QoS improvement*, 4th COST 263 International Workshop on Quality of Future Internet Services (QoFIS'2003), Stockholm (Sweden), 1-2 October 2003, Lecture Notes in Computer Science 2811, Quality for all, Eds. G. Karlsson, MI. Smirnov, 2003, Springer, pp.73-82.

[Lar03b] N. LARRIEU, P. OWEZARSKI, *Une extension du modèle de tarification "smart market" pour l'Internet basé sur le contrôle de congestion*, Rapport LAAS N. 03220, 10ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'2003), Paris (France), 7-10 octobre 2003, pp.377-392.

[Lar04c] N. LARRIEU, *A measurement based networking approach for improving Internet congestion control*, IFIP World Computer Congress (WCC'04), Student Forum, Toulouse (France), 22-27 August 2004.

[Lar05a] N. LARRIEU, P.OWEZARSKI, *Contrôle de congestion et gestion du trafic à partir de mesures*, Rapport LAAS N. 04609, Mars 2005, 17p., CFIP 2005.

[Lar05b] N. LARRIEU, *Monitoring based approach for congestion control aiming at improving Internet QoS*, Rapport LAAS, 3 p., Mars 2005, IEEE INFOCOM 2005, Student Workshop.

[Lar05c] N. LARRIEU, P. OWEZARSKI, *Measurement based networking approach applied to congestion control in the multi-domain Internet*, Rapport LAAS N. 04256, Mai 2005, 9p., IEEE/IFIP IM 2005.

[Owe04a] P. OWEZARSKI, N. LARRIEU, *A trace based method for realistic simulations*, IEEE International Conference on Communications (ICC'2004), 20-24 June, Paris (France).

[Owe04b] P. OWEZARSKI, N. LARRIEU, *Internet traffic characterization - An analysis of traffic oscillations*, 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'04), Toulouse (France), June 30 - July 2, 2004.

Papier invité

[Owe04c] P. OWEZARSKI, N. LARRIEU, *Measurement tools and techniques for traffic and QoS management*, International Conference on Integrated Modeling & Analysis in Applied Control & Automation (IMAACA'04), Genoa (Italy), 28-31 October 2004.

Conférences nationales

[Lar03c] N. LARRIEU, P. OWEZARSKI, *Un modèle de tarification du trafic Internet basé sur le contrôle de congestion*, Colloque de l'Ecole Doctorale Informatique et Télécommunications (EDIT'03), Toulouse (France), 14-15 avril 2003, pp. 98-102.

[Lar04d] N. LARRIEU, *Analyse des problèmes liés à l'évolution du trafic Internet*, Colloque de l'Ecole Doctorale Informatique et Télécommunications (EDIT'04), Toulouse (France), 29-30 mars 2004.

[Lar04e] N. LARRIEU, P. OWEZARSKI, *Contrôle de congestion orienté mesures pour l'Internet*, Rapport LAAS N. 04266, 6èmes Journées Doctorales Informatique et Réseau (JDIR'04), Lannion (France), 2-4 Novembre 2004, pp.167-176.

[Owe05a] P. OWEZARSKI, N. LARRIEU, *Un mécanisme de contrôle de congestion orienté mesures pour une QdS robuste dans l'Internet*, Rapport LAAS N. 05010, Juin 2005, 10p., SAR 2005.

Rapports de contrats :

[Fri04] T. FRIEDMAN, K. SALAMATIAN, P. OWEZARSKI, N. LARRIEU, G. YONNET, E. DA COSTA, F. X. ANDREU, *Rapport intermédiaire du sous-projet 6 : Tarification et SLA*, contrat RNRT METROPOLIS, janvier 2004, 51 p.

[Lar04f] N. LARRIEU, P. OWEZARSKI, K. SALAMATIAN, A. SOULE, *Rapport intermédiaire du sous-projet 3 : Analyse du réseau*, contrat RNRT METROPOLIS, janvier 2004, 56 p.

[Owe04d] P. OWEZARSKI, N. LARRIEU, *Rapport intermédiaire du sous-projet 7 : Conception et mise en place de la plate-forme de mesures passives*, contrat RNRT METROPOLIS, janvier 2004, 10 p.

[Owe04e] P. OWEZARSKI, N. LARRIEU, L. BERNAILLE, W. SADDI, F. GUILLEMIN, A. SOULE, K. SALAMATIAN, *Distribution of traffic among applications as measured in the French METROPOLIS project*, Rapport LAAS N. 04628, Contrat RNRT METROPOLIS, Octobre 2004, 13p.

[Owe05b] P. OWEZARSKI, N. LARRIEU, *Definition of monitoring equipment and software and location points*, EuQoS project, Work Package n. 2, Delivrable 2.1.1, 86 p., Mars 05.

[Owe05c] P. OWEZARSKI, F. RACARU, G. AURIOL, N. LARRIEU, *Technical requirements for the trial, tasks and scheduling*, EuQoS project, Work Package n. 5, Delivrable 5.1.1, 161 p., Mars 05.

[Owe05d] P. OWEZARSKI, F. RACARU, G. AURIOL, N. LARRIEU, *Connectivity and performance tests report for local and pan-European (across GEANT) testbed design for the Trial*, EuQoS project, Work Package n. 5, Delivrable 5.1.2, 96 p., Mars 05.

Rapports techniques :

[Lar02] N. Larrieu, *Métrologie des réseaux IP : développement de nouveaux outils pour caractériser, analyser et rejouer le trafic réseau*, rapport de diplôme ingénieur INSA, 83 pages, juin 2002.

[Owe03b] P. OWEZARSKI, N.LARRIEU, *Congestion control based pricing model and charging mechanisms for Internet traffic*, Rapport LAAS N. 03183, avril 2003, 10p.

[Owe03c] P. OWEZARSKI, N. LARRIEU, *An analysis of Internet traffic characteristics and related issues*, Rapport LAAS N. 03496, novembre 2003, 12p.

[Owe04f] P. OWEZARSKI, N.LARRIEU, *Measurement based approach of congestion control for enforcing a robust QoS in the Internet*, Rapport LAAS N. 04722, Décembre 2004, 20p.

Papiers soumis en cours d'évaluation :

[Lar05d] N. LARRIEU, P. OWEZARSKI, Y. ZHANG, *Characterization and analysis of main Internet application traffic*, Rapport LAAS N. 04549, Février 2005, 7 p., soumis à la conférence ITC 19.

[Lar05e] N. LARRIEU, Y. ZHANG, P. OWEZARSKI, *Caractérisation et analyse du trafic Internet en fonction du type d'application*, Rapport LAAS, Février 2005, 4 p., soumis à la conférence GRETSI 2005.

[Lar05f] P. BORGNAT, N. LARRIEU, P. ABRY, P. OWEZARSKI, *Détection d'attaques de "Déni de Services" : ruptures dans les statistiques du trafic*, Rapport LAAS, Février 2005, 4 p., soumis à la conférence GRETSI 2005.

Bibliographie générale

[1] P. Abry, D. Veitch, *Wavelet Analysis of Long Range Dependent Traffic*, Trans. Info. Theory, Vol.44, No.1 pp.2-15, Jan 1998.

[2] P. Abry, D. Veitch and P. Flandrin, *"Long-Range Dependence : Revisiting Aggregation with Wavelets"*, Journal of Time Series Anal., Vol.19, No.3 pp.253- 266 May 1998.

[3] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi, D.Veitch, *The Multiscale Nature of Network Traffic : Discovery. Analysis, and Modelling*, IEEE Signal Processing Magazine vol 19 , no 3, pp 28- 46, Mai 2002.

[4] A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S. B. Moon, V. Paxson and D. Towsley, *The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior*, IEEE Communications, 38(5), May 2000.

[5] C. Adjih, P. Jacquet and N. Vvedenskaya, *Performance evaluation of a single queue under multi-user TCP/IP connections*, Tech. Report RR-4141, INRIA, March 2001.

[6] E. Altman, K. Avrachenkov, C. Barakat and R. Nunez Queija, *State dependent M/G/1 type queueing analysis for congestion control in data networks*, Proceedings of IEEE INFOCOM (Anchorage, Alaska), April 2001.

[7] M. Allman and V. Paxson, *On estimating end-to-end network path properties*, SIGCOMM, 1999, pp. 263-274.

[8] E. Altman, K. Avrachenko, C. Barakat and R. Nuñez-Queija, *TCP modeling in presence of nonlinear window growth*, ITC'17 (Salvador da Bahia), 2001.

[9] E. Altman, K. Avrachenkov and C. Barakat, *A Stochastic Model of TCP/IP with Stationary Random Losses*, in Proceedings of ACM SIGCOMM, Stockholm, Sweden, 2000.

[10] G. Almes, S. Kalidindi and M. Zekauskas, *A One-way Delay Metric for IPPM*, RFC 2679, September 1999.

[11] G. Almes, S. Kalidindi and M. Zekauskas, *A One-way Packet Loss Metric for IPPM*, RFC 2680, September 1999.

[12] G. Almes, S. Kalidindi and M. Zekauskas, *A Round-trip Delay Metric for IPPM*, RFC 2681, September 1999.

[13] AMP web site, http ://www.watt.nlanr.net.

[14] J. Apsidorf, *OC3MON : Flexible, affordable, high performance statistics collection*, Proceedings of INET, June 1997.

[15] Site web du projet Européen AQUILA, http ://www-st.inf.tu-dresden.de/aquila/.

[16] S. Aubert, *Les dénis de service réseau*, Journées Réseaux (JRES'2001), Lyon, France, 10-14 Déc. 2001.

[17] F. Baccelli and D. Hong, *TCP is max-plus linear*, ACM-SIGCOMM'00 (Stockholm), no. 4, 2000, pp. 219-230.

[18] F. Baccelli and D. Hong, *AIMD, fairness and fractal scaling of TCP traffic*, Tech. Report 4155, INRIA, Domaine de Voluceau, Rocquencourt B.P.105, 78153 Le Chesnay Cedex, April 2001.

[19] C. Barakat, P. Thiran, G. Lannaccone, C. Diot and P. Owezarski, *A flow-based model for Internet backbone traffic*, IMW 2002 (Internet Measurement Workshop),Janvier 2002.

[20] N. Ben Azzouna and F. Guillemin, *Analysis of ADSL traffic on an IP backbone link*, In Proc. Globecom 2003, San Francisco, December, 2003.

[21] N. Ben Azzouna, C. Fricker and F. Guillemin, *Modeling ADSL traffic on IP backbone link*, France Télécom and INRIA Research Report, 2003.

[22] J. Beran, *Statistics for Long-Memory Processes*, Monongraphs on Statistics and Applied Probability, Chapman and Hall, New York, NY, 1994.

[23] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, *Long-range dependence in Variable-Bit-Rate video traffic*, IEEE Trans. Comm., Vol. 43, No 2/3/4, pp. 1566-1579, 1995.

[24] U. Black, *TCP/IP and related protocols*, McGraw-Hill, 1992.

[25] S. Blake, D. Black and M. Carlson, *An Architecture for Differentiated Services*, RFC 2475, 1998.

[26] M. Boldi, *Une approche de l'analyse mathématiques des données circulant en réseaux informatiques*, projet de semestre Département de mathématiques de l'EPFL, été 1999.

[27] J. Bolot, *End-to-end packet delay and loss behavior in the Internet*, ACM Sigcomm '93 (San Fransisco, CA) (ACM, ed.), September 1993, pp. 289-298.

[28] R. Braden and L. Zhang, *Resource ReSerVation Protocol (RSVP) – Version 1 message processing rules*, RFC 2209, September 1997.

[29] P. Brown, *Resource sharing of tcp connections with different round trip times*, Proc. IEEE Infocom (Tel-Aviv, Israel), March 2000.

[30] N. Brownlee and K. Claffy, *Understanding Internet Traffic Streams : Dragonflies and tortoises*, IEEE Communications, 2002.

[31] R. Caceres, N. Duffield, D. Towsley and J. Horowitz, *Multicast-based Inference of Network-internal loss characteristics*, IEEE Transactions on Information Theory, vol. 45, no. 7, November 1999.

[32] CAIDA web site, http ://www.caida.org.

[33] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, *Internet traffic tends to Poisson and independent as the load increases*, Tech. Rep., Bell Labs, 2001.

[34] R. Casellas et al., *Rapport du sous-projet 4 : méthodologie pour la mesure et l'échantillonage*, rapport du projet RNRT METROPOLIS, janvier 2004.

[35] J. Cleary, S. Donnely, I. Graham, A. McGregor and M. Pearson, *Design principles for accurate passive measurement*, PAM (Passive and Active Measurements) Workshop, Hamilton, New Zealand, April 2000.

[36] CoralReef website, http ://www.caida.org/tools/-measurement/coralreef.

[37] D. R. Cox , *long-Range Dependance : A Review*, The Iowa State University Press, 1984.

[38] M. Crovella and A. Bestavros, *Explaining World Wide Web traffic self-similarity*, Boston, MA, Computer Science Department, Boston University, 1995.

[39] M. Crovella and A. Bestavros, *Self-similarity in world wide web traffic evidence and possible causes*, IEEE ACM Transactions on Networking vol. 5, no 6, 9 pages, 1996.

[40] D. Dacunha-Castelle and M. Duflo, *Probabilités et statistiques Tome 1 Problèmes à temps fixe*, Editions Masson, Collection mathématiques appliquées pour la maîtrise, 2ième édition, pages 47-48, 220 pages, 1994.

[41] J. Daemenand and V. Rijmen, *AES proposal : Rijndael*, Technical Report, Computer Security Resource Center, National Institute of Standards and Technology.

[42] *Dag 4 SONET network interface* http ://dag.cs.waikato.ac.nz/dag/dag4-arch.html.

[43] *Dag synchronization and timestamping*, http ://dag.cs.waikato.ac.nz/dag/docs/-dagduck_v2.1.pdf.

[44] T. D. Dang, S. Molnár and I. Maricza, *Capturing the Complete Multifractal Characteristics of Network Traffic*, GLOBECOM 2002, Taipei, Taiwan, November 17-21, 2002.

[45] M. Diaz and P. Senac, *Time Stream Petri Nets : a Model for Multimedia Stream Synchronization*, First International Conference on Multimedia Modeling (MMM'93), Singapour, novembre 1993.

[46] A. B. Downey, *Evidence for long tailed distributions in the Internet*, ACM SIGCOMM Internet Measurement Workshop, November 2001.

[47] V. Dumas, F. Guillemin, and P. Robert, *A Markovian analysis of Additive- Increase Multiplicative-Decrease (AIMD) algorithms*, Advances in Applied Probability 34 (2002), no. 1.

[48] A. Elwalid, D. Mitra, and R.H. Wentworth, *A new approach for allocating buffers and bandwidth to heterogeneous regulated traffic in an ATM node*, IEEE Journal on Selected Areas in Communications 13 (1995), no. 6, 1115-1127.

[49] *Endace Web Site*, http ://www.endace.com.

[50] A. Erramilli, O. Narayan and W. Willinger, *Experimental queuing analysis with long range dependent packet traffic*, IEEE/ACM Transactions on Networking, Vol. 4, No. 2, pp 209–223, 1996.

[51] A. Erramilli, O. Narayan and W. Willinger, *Self-similarity in high-speed network traffic measurements : fact or artifact ?*, 12 pages, In Proc. of the 12th Nordic Teletraffic Seminar, NTS12, Espoo, Finland, 22-24 August 1995.

[52] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford and F. True, *Deriving traffic demands for operational IP networks : Methodology and Experience*, In Proc. ACM SIGCOMM Conference, Stockholm, 2000.

[53] A. Feldmann, A. Gilbert and W. Willinger, *Data networks as cascades : Investigating the multifractal nature of Internet WAN traffic*, Proc. of ACM SIGCOMM'98, Vancouver, Canada, 1998.

[54] A. Feldmann, A. C. Gilbert and W. Willinger, *Data networks as cascades : Explaining the multifractal nature of Internet WAN traffic*, Proc. of ACM SIGCOMM'98, Aug. 1998.

[55] A. Feldmann, A. C. Gilbert, W. Willinger and T. G. Kurtz, *The changing nature of network traffic : Scaling phenomena*, Computer Communication Review, Vol. 28, No 2, April 1998.

[56] S. Floyd, M. Handley, J. Padhye and J. Widmer *Equation-Based Congestion Control for Unicast Applications*, SIGCOMM 2000, August 2000.

[57] S. Floyd and V. Paxson, *Difficulties in simulating the Internet*, IEEE/ACM Trans. on Networking, Vol. 9, No 4, pp. 392-403, Aug. 2001.

[58] S. Floyd, *Connections with multiple congested gateways in packet-switched networks part 1 : One way traffic*, Computer Communication Review 21 (1991), no. 5, 30-47.

[59] S. Floyd and K. Fall, *Promoting the use of end-to-end congestion control in the Internet*, In Proc. IEEE ACM Transactions on Networking, 14 pages, February 1998.

[60] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owerzarski, D. Papagiannaki and F. Tobagi, *Design and deployment of a passive monitoring infrastructure*, In Passive and active measurements workshop, Amsterdam, April 2001.

[61] D. R. Figueiredo, B. Liu, V. Misra and D. Towsley, *On the Autocorrelation Structure of TCP Traffic*, UMass CMPSCI Technical Report TR 00-55 1998.

[62] J.M. Garcia, D. Gauchard, O. Brun, P. Bacquet, J. Sexton and E. Lawless, *Modélisation différentielle du trafic et simulation hybride distribuée*, Calculateurs parallèles, Vol. 18, No. 3, 2001.

[63] *GTK Web Site*, http ://www.gtk.org.

[64] F. Guillemin, P. Robert and B. Zwart, *Performance of TCP in the presence of correlated packet loss*, 15th ITC Specialist Seminar on Internet Traffic Engineering and Traffic Management (Wurzburg), July 2002.

[65] F. Guillemin, P. Robert and B. Zwart, *AIMD algorithms and exponential functionals*, Annals of Applied Probability (2003).

[66] L. Guo, M. Crovella and I. Matta, *How does TCP generate pseudo-selfsimilarity ?*, 9 pages, Proc. the International Workshop on Modeling, Analysis and Simulation ofComputer and Telecommunications Systems (MASCOTS'01), Cincinnati, OH, Août 2001.

[67] L. Guo , M. Crovella and I. Matta, *TCP Congestion Control and Heavy Tails*, 2000.

[68] M. Handley, S. Floyd, J. Pahdye and J. Widmer, *TCP Friendly Rate Control (TFRC) : Protocol Specification*, RFC 3448, Proposed Standard, January 2003.

[69] D. Heyman, *Some issues in performance modeling of data teletraffic*, Performance Evaluation, Vol. 34, pp. 227-247, 1998.

[70] N. Hohn, D. Veitch and P. Abry, *Cluster Process, a Natural Language for Network Trafic*, IEEE Transactions on Signal Processing, Special Issue on Signal Processing in Networking, vol. 51, no. 8, pp. 2229-2244, 2003.

[71] S. Kalidindi and M.J. Zekauskas, *Surveyor : an infrastructure for Internet performance measurements*, Proceedings of INET'99, June 1999.

[72] F.P. Kelly, *Effective bandwidths at multi-type queues, Queueing Systems*, Theory and Applications 10 (1991), no. 1-2, 5-15.

[73] E. Kohler, M. Handley, S. Floyd and J. Padhye, *Datagram Congestion Control Protocol*, IETF draft, draft-ietf-dccp-spec-03.txt, May 2003.

[74] F. Lau, S. H. Rubin, M. H. Smith and L. Trajkovic, *Distributed denial of service attacks*, IEEE Int. Conference on Systems, Man and Cybernetics, Nashville, TN, USA, Oct. 2000.

[75] W. Leland, M. Taqqu, W. Willinger and D. Wilson, *On the self-similar nature of Ethernet traffic*, ACM SIGCOM, September 1993.

[76] J. Madhavi and S. Floyd, *TCP-friendly unicast rate-based flow control*, End2end-interest mailing list, January 1997.

[77] B.B. Mandelbrot, *Long-Run Linearity, Locally Gaussian Processes, H-Spectra and Infinite Variances*, Intern. Econom. Rev. 10, 82-113, 1969

[78] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, *TCP Selective Acknowledgment Options*, Request for Comments 2018, October 1996.

[79] *Site web du projet METROPOLIS*, http ://www.laas.fr/ owe/METROPOLIS/-METROPOLIS.html.

[80] I. Miloucheva, P.A. Gutierrez, D. Hetzer, A. Nassri, M. Beoni, *Intermon architecture for complex QoS analysis in inter-domain environment based on discovery of topology and traffic impact*, Inter-domain Performance and Simulation Workshop, Budapest, March 2004

[81] G. Minshall, *TCPdpriv Command Manual*, 1996.

[82] Projet IST MOME : http ://www.ist-mome.org.

[83] P. Mutaf, *Defending against a Denial-of-Service attack on TCP*, 2nd Int. Workshop on Recent Advances in Intrusion Detection (RAID'99),West Lafayette, Indiana, USA, Sept. 1999.

[84] *Netflow Services Solutions Guide*, http ://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/-netflsol/.

[85] *Netsizer web site*, http ://www.netsizer.com.

[86] I. Norros, *A storage model with self similar input, Queueing Systems*, Theory and Applications 16 (1994), 387-396, number 23.

[87] I. Norros, *On the use of fractional brownian motion in the theory of connectionless networks*, IEEE Journal on Selected Areas in Communications 13 (1995), 953-962.

[88] B. Norton, *Evolution of the U.S. Peering Ecosystem*, in Proceedings of the North American Network Operators' Group Workshop, May 2004, San Francisco, CA.

[89] C. J. Nuzman, I. Saniee, W. Sweldens and A. Weiss, *A compound model for TCP connection arrivals*, Proc. of ITC Spec. Seminar on IP Traffic Modeling, Measurement and Management, Sept. 2000.

[90] P. Olivier and N. Benameur, *Flow level IP traffic characterization* , Proc. of ITC'17 Moreira de Souza, Fonseca and de Souza e Silva (eds.), December 2001.

[91] T. J. Ott, J.H.B. Kemperman and M. Mathis, *The stationary behavior of ideal TCP congestion avoidance*, Unpublished manuscript, August 1996.

[92] P. Owezarski , D. Andreu , C. Fricker, K. Salamatian, C. Chekroun, N. Benameur, P. Olivier, J. Roberts, P. Robert and F. Guillemin, *Projet METROPOLIS. Sous-projet 1 : Rapport état de l'art*, rapport, Janvier 2003, Contrat RNRT METROPOLIS.

[93] P. Owezarski, P. Abry, K. Salamatian, D. Kokman, A. Aussem, F. Guillemin, P. Robert, *Métrologie des réseaux de l'Internet*, Rapport LAAS N 03548, Action Spécifique 88, Décembre 2003, 12p.

[94] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, *Modeling TCP throughput : A simple model and its empirical validation*, IEEE/ACM Transactions on Networking 8 (2000), 133-145.

[95] N. Papagiannaki, K. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian and C. Diot, *A pragmatic definition of Elephant in Internet backbone traffic*, 2001.

[96] K. Park, G. Kim and M. Crovella, *On the relationship between file sizes, transport protocols, and self-similar network traffic*, IEEE ICNP, 1996.

[97] K. Park, G. Kim and M. Crovella, *On the Effect of Traffic Self-similarity on Network Performance*, SPIE International Conference on Performance and Control of Network Systems, November, 1997.

[98] K. Park and W. Willinger, *Self-similar network traffic : an overview*, In Self-similar network traffic and performance evaluation, edited by K. Park and W.Willinger, J.Wiley & Sons, 2000.

[99] V. Paxson, G. Almes, J. Mahdavi and M. Mathis, *Framework for IP Performance Metrics*, RFC 2330, May 1998.

[100] V. Paxson and S. Floyd, *Wide area traffic : The failure of Poisson modeling*, IEEE/ACM Trans. on Networking, Vol. 3, pp. 226-244, 1995.

[101] V. Paxson and S. Floyd, *Why we don't know how to simulate the Internet*, Winter Simulation Conference, 1997, pp. 1037-1044.

[102] V. Paxson, *End-to-end Internet packet dynamics*, IEEE/ACM Transactions on Networking 7 (1999), no. 3, 277-292.

[103] V. Paxson, A. Adams and M. Mathis, *Experiences with NIMI*, Proceedings of Passive and Active Measurement, 11 pages, 2000.

[104] L. Ricciulli, P. Lincoln and P. Kakkar, *TCP SYN flooding defense*, Communication Networks and Distributed Modeling and Simulation, 1998.

[105] *RIPE NCC web site*, http ://www.ripe.net.

[106] J. Roberts, *Engineering for Quality of Service*, In Self-similar network traffic and performance evaluation, edited by K. Park and W. Willinger, J. Wiley & Sons, 2000.

[107] P. Salvador, A. Nogueira and R. Valadas, *Modeling Local Area Network Traffic with Markovian Traffic Models*, 2001.

[108] C. L. Schuba, I. V. Krsul and M. G. Kuhn, *Analysis of a denial of service attack on TCP*, IEEE Symp. on Security and Privacy, Oackland, CA, USA, 1997.

[109] P. Senac, *Contribution à la modélisation des systèmes multimédias et hypermédias*, Doctorat, Institut National des Sciences Appliquées, Toulouse, juin 1996.

[110] B. Sikdar and K. Vastola, *On the contribution of TCP to the selfsimilarity of network traffic*, Proceedings of the 2001 Tyrrhenian International Workshop on Digital Communications : Evolutionary Trends of the Internet, Springerv 2001.

[111] B. Sikdar, K. Vastola and S. Kalyanaraman, *On reducing the degree of self-similarity in network traffic*, In Proc. IEEE International Conference on Network Protocols, pp. 171–180, October 1998.

[112] A. Simon, *netMET -Network's METrology :Une solution de métrologie générale pour les réseaux régionaux, métropolitains et de campus*, Journées réseaux (JRES' 2001), Lyon, France, 10-14 décembre 2001.

[113] O. Spatscheck, *Defending against Denial of Service Attacks in Scout*, 3rd OSDI Symposium, Feb. 1999.

[114] A. Soule, A. Nucci, R. Cruz, E. Leonardi and N. Taft, *How to identify and estimate the largest traffic matrix elements in a dynamic environment*, in Proceedings of the joint international conference on Measurement and modeling of computer systems, 2004, pages 73–84, New York, NY, USA.

[115] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Scwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang and V. Paxson, *Stream Control Transmission Protocol*, RFC 2960, octobre 2000.

[116] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen and P. Conrad, *SCTP Partial Reliability Extension*, Draft Internet, juin 2003.

[117] *Site web du logiciel TCPDUMP*, http ://www.tcpdump.org.

[118] *Site web QoSMOS*, http ://www.qosmos.net.

[119] M. Taqqu, V. Teverovsky and W. Willinger, *Estimators for long-range dependence : an empirical study*, Boston University, Fractals, vol. 3, no. 4, pp. 785-788, 18 pages, 1995.

[120] *Traffic Designer Web site*, http ://www.qosmos.fr/EN/home.htm.

[121] K. Thompson, G. Miller and M. Wilder, *Wide-area Internet traffic patterns and characteristics*, IEEE Network, Vol. 11, No 6, Nov./Dec. 1997.

[122] Ph. Tran-Gia and N. Vicari (editors), *Impacts of new services on the architecture and performance of broadband networks (Final report of COST 257)*, Chapter on "Traffic measurement and data analysis", 2000.

[123] S. Uhlig and O. Bonaventure, *Understanding the long-term self-similarity of Internet traffic*, Infonet group, université de Namur, QoFIS 2001.

[124] *Site web pour l'implémentation de la méthode des ondelettes*, http ://www.emulab.-ee.mu.oz.au/ darryl/secondorder_code.html.

[125] *Site web pour l'implémentation de la méthode des ondelettes*, http ://www.cubinlab.ee.mu.oz.au/ darryl/MS_code.html, Verlag Lecture Notes in Computer Science Taormina, pp. 596-613, Italy, September 17-20, 2001.

[126] M. Venkatachalam, J. Kaur and H. Vin, *End-to-End Model for a Flow in the Internet*, Technical Report, Department of Computer Sciences, University of Texas at Austin, TAX 2.124, Austin, TX 787 12-1188, USA, August 2000.

[127] D. Veitch and P. Abry, *A wavelet based joint estimator for the parameters of LRD, Special issue on Multiscale Statistical Signal Analysis and its Applications"* IEEE Trans. Info. Th. April 1999, Volume 45, No.3, 1999.

[128] A. Veres and M. Boda, *The chaotic nature of TCP congestion control*, in Proceedings of IEEE INFOCOM'2000, 9 pages, March 2000.

[129] A. Veres and M. Boda, *On the Impact of Short Files and Random Losses on Chaotic TCP Systems*, in Proc. IFIP ATM & IP 2000 Workshop, Ilkley, UK, July 2000.

[130] A. Veres , Zs. Kenesi S.Molnar and G. Vattay, *On the Propagation of Long-Range Dependence in the Internet*, 2000.

[131] *Site web du projet VINT*, http ://netweb.usc.edu/vint.

[132] M. Vojnovi and J.-Y. Le Boudec, *Some observations on equation-based rate control*, Proc. ITC'17 (Salvador de Bahia, Brazil), 2001.

[133] M. Vojnovi, J.-Y. Le Boudec, and C. Boutremans, *Global fairness of additive-increase and mutliplicative-decrese with heteregeneous round trip times*, IEEE Infocom (Tel Aviv, Israel), March 2000.

[134] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy, *Turning the Postal System into a Generic Digital Communication Mechanism*, Proc. of ACM SIGCOMM 2004, August 2004.

[135] J. Cao, W. S. Cleveland, D. Lina and D. X. Sun, *Internet Traffic Tends Toward Poisson and Independent as the Load Increases*, Bell-labs, technical report, http ://cm.bell-labs.com/cm/ms/departments/sia/doc/lrd2poisson.pdf, 18 pages, 2001.

[136] W. Willinger, *Traffic Modeling of High-Speed Networks : Theory and Practic*, Stochastic Networks, Kelly and Williams eds, Springer-Verlag 1994.

[137] W. Willinger, V. Paxson and M. Taqqu, *Self-Similarity and Heavy Tails : Structural Modeling of Network traffic*, In A Practical Guide To Heavy Tails : Statistical Techniques and Applications, ISBN 0-8176-3951-9, 1998.

[138] W. Willinger, M. Taqqu, R. Sherman and D. Wilson, *Self-similarity through high variability : statistical analysis of Ethernet LAN traffic at the source level*, In ACM Sigcomm'95, pages 100–113, 1995.

[139] J. Xu, J. Fan, M. Ammar and S. Moon, *On the Design and Performance of Prefix-Preserving IP Traffic Trace Anonymization*, Proc. of 10th IEEE International Conference on Network Protocols (ICNP 2002).

[140] M. Yajnik, J. Kurose, and D. Towsley, *Packet loss correlation in the MBone multicast network experimental measurements and Markov chain models*, Tech. Report UM-CS-1995-115, University of Massachusetts, Amherst, 1995.

[141] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, *On the Constancy of Internet Path Properties*, Proc. ACM SIGCOMM Internet Measurement Workshop (IMW'2001), San Francisco, California, USA, November 2001.