Métrologie dans les réseaux Peer-To-Peer

Oualid Saddi

La métrologie dans les réseaux p2p

THÈSE

présentée et soutenue publiquement le 12 octobre 2007

pour obtenir le titre de

Docteur de l'Université Pierre et Marie Curie - Paris VI

(spécialité informatique et télécommunications)

par

Oualid Saddi

Composition du jury

Rapporteurs :	Catherine Rosenberg (University of Waterloo, Canada). Philippe Owezarski (LAAS-CNRS, France).
Examinateurs :	Kavé Salamatian (LIP6, France). Patrice Abry (ENS Lyon, France)
Directeurs :	Serge Fdida (LIP6, France). Fabrice Guillemin (France Telecom R&D).

Mis en page avec la classe thloria.

Remerciements

Je tiens à remercier, en tout premier lieu, Monsieur Serge Fdida, professeur à l'université Pierre et Marie Curie, et Monsieur Fabrice Guillemin, ingénieur recherche et développement à France Telecom, qui ont dirigé cette thèse. Merci à Monsieur Fabrice Guillemin qui a su orienter mes recherches aux bons moments grâce à ses compétences scientifiques. Je voudrais aussi le remercier pour sa rigueur et sa patience qui ont énormément faciliter ma tâche et m'ont permis d'aboutir à la production de cette thèse. Merci à lui, ainsi qu'à Monsieur Serge Fdida, pour leurs précieux conseils.

Je remercie tous particulièrement Madame Catherine Rosenberg, professeur à l'université de Waterloo, ainsi que Monsieur Philippe Owezarski, chargé de recherches au LAAS-CNRS à Toulouse, qui ont accepté de juger ce travail et d'en être les rapporteurs.

Je tiens également à remercier Monsieur Patrice Abry, directeur de recherche au CNRS, et Monsieur Kavé Salamatian, maitre de conférence à l'université de Pierre et Marie Curie, d'avoir accepté de participer au jury de cette thèse.

Je remercie Mr Christian Guillemot, responsable du laboratoire "Core Packet Networks for NGN & IMS" à France Telecom R&D, pour m'avoir accueilli au sein de cette institution.

Mes plus sincères remerciements vont également à Monsieur Jean Philippe Le Brenn, responsable de l'unite de recherche et développement "Traffic and Networks Security" à France Télécom R&D, qui m'a chaleureusement accueilli dans son équipe. Ses conseils et ses commentaires ont été fort utiles.

Je remercie tous ceux sans qui cette thèse ne serait pas ce qu'elle est : aussi bien par les discussions que j'ai eu la chance d'avoir avec eux, leurs suggestions ou contributions. Je pense ici en particulier à Monsieur Fabrice Clérot et Monsieur Marc Boullé membres de l'unité de recherche et développement "Statistique et Traitement de l'Information" à France Telecom R&D, à Monsieur Philippe Robert, membre de l'unité de recherche RAP à l'INRIA Rocquencourt.

J'aimerais également exprimer le plaisir que j'ai eu à travailler avec ma collègue Madame Stéphanie Moteau, je tiens aussi à la remercier pour le soutien moral qu'elles m'a fourni tout au long de la réalisation de ces travaux et pour les moments agréables que j'ai passés avec elle (merci pour sa relecture des mes productions écrites, sans elle, mon manuscrit aurait contenu le double de fautes d'Aurthografe).

Je remercie aussi profondément Monsieur Joel François et Monsieur Thierry Houdoin, qui m'ont fourni les données expérimentales sur lesquelles est basée une grande partie de ma thèse. Je n'oublierai pas les aides permanentes reçues de Jean-Louis Baron, Catherine Blanquart, André Castelli, Gérard Augoyat, Thierry Thalagrand pour résoudre les problèmes d'ordre informatique. Mes plus vifs remerciements à Mme Françoise Douerin et Mme Sylvie Gillot, secrétaire du laboratoire CORE/CPN à France Telecom R&D, pour leur sympathie et leur disponibilité.

Mes remerciements vont encore à tous mes collègues de l'unité de recherche TNS à France Telecom R&D, qui m'ont permis de passer des années très agréables et enrichissantes. Je leur exprime ma profonde sympathie et leur souhaite beaucoup de bien. Mes remerciement vont aussi à mes collègues de l'URD FAME avec lesquels, j'ai eu des discussions très fructueuses et partagé les plaisirs des pauses café très instructives.

Je souhaite aussi remercier mes amis, en particulier Moez, Imène, SidMo, Yassine, Ramzi, Narjess, Marie, Estelle, Riadh Z, Nadhem, Asma, Wassym, Liv, Zineb, Romak, Julie, Alexandra, Daly, Mohammed, Riadh K, Walid, Sophie, Sana, Nassima, Jean Louis, Mounir (désolé pour ceux qui ne figurent pas dans la liste, le nombre de pages dans une thèse est malheureusement limité à 3000 pages) donc merci à tous mes amis témoins de mes joies, de mes fatigues, de mes enthousiasmes et de mes hauts et bas.

Pour finir ("*last but not least*"), ma gratitude est adressée à ma mère Hayet, à mon frère Ziad, à ma sœur Dhouha et à mon oncle Chedly dont les encouragements et la générosité sont inestimables. Mes plus chaleureux remerciements vont à Achtart, ma grande source d'inspiration, qui a accepté avec plaisir ma vie de nomade du savoir.

Je dédie ce travail aux personnes, qui grâce à leur savoir, m'ont fait découvrir l'étendue de mon ignorance.
à la mémoire de mon père Mustapha et de ma grand mère Beya

Résumé

Cette thèse s'inscrit dans le domaine de la métrologie du trafic Internet. Nous portons dans notre étude un intérêt particulier au trafic peer-to-peer (p2p) et plus précisément le trafic eDonkey, principale composante du trafic dans les réseaux de France Telecom. Nous avons analysé les différentes méthodes d'identification du trafic p2p afin d'extraire cette importante composante du trafic. Après avoir obtenu une proportion de trafic jugée représentative de l'échantillon de la population eDonkey étudiée, nous avons étudié les caractéristiques en termes de trafic de cette composante p2p en utilisant entre autres la dichotomie souris/éléphant. Outre les caractéristiques du trafic, nous avons également analysé la topologie du réseau eDonkey dans le réseau de France Telecom à l'échelle nationale et internationale. Un modèle mathématique décrivant la manière avec laquelle le réseau eDonkey se construit est également présenté.

Abstract

This thesis is a contribution to the domain of Internet metrology. We study in this document the characteristics of p2p traffic and especially eDonkey traffic. eDonkey protocol gives rise to the prevalent part of traffic in some European countries (France, Germany, etc.). We have analyzed many real traffic traces from high speed links of France Telecom networks carrying ADSL traffic. In order to analyze this type of traffic, several p2p identification methods have been developed. By extracting a representative part of eDonkey traffic, we have analyzed its characteristics using the mouse/elephant dichotomy. Furthermore, the topology of the eDonkey community in the France Telecom network has been investigated. Finally, to explain some observed phenomena, a simulation based model has been proposed.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

En 1972, Bob Khan et Vinton Cerf, deux ingénieurs américains, présentaient officiellement et pour la première fois les résultats d'un projet lancé depuis la fin des années soixante : ARPANET. Le lancement du projet était intimement lié à l'ambiance de la guerre froide qui régnait à l'époque. En effet, les concepteurs de l'ARPANET cherchaient à créer une plateforme de communications invulnérable face à une attaque nucléaire touchant l'un des constituants de cette plateforme. Ce premier réseau plat a évolué depuis cette première démonstration et a donné naissance au réseau des réseaux : Internet. Pendant cette évolution, Internet est devenu au fur et à mesure très hiérarchique et caractérisé par une intense centralisation. Ce chemin inverse au concept de la genèse d'Internet que le monde de la recherche et de l'industrie a emprunté se traduit par la prédominance d'une architecture extrêmement centralisée qu'est l'architecture Client/Serveur. Le web, l'illustration la plus pédagogique de cette architecture, a été l'élément clé dans la démocratisation d'Internet dans le monde au début des années 90. Le web est l'application qui a marqué le monde des réseaux tout au long de la dernière décennie du XX siècle. Malgré les tailles réduites des pages HTML plus ou moins enrichies avec des images, le web représentait la composante majoritaire du trafic Internet. Le trafic web a fait l'objet de diverses études de métrologie et de modélisation mathématique. Cette hégémonie ne va pas tarder à disparaître pour céder la place à une nouvelle forme d'application plus proche dans leur philosophie du concept originel du réseau plat. En 1999, le réseau Napster, considéré depuis comme le premier réseau Peer-to-Peer (p2p), permettait à ses utilisateurs de télécharger pas moins de 2 millions de titres MP3. Au paroxysme de sa réussite, le serveur de Napster a été fermé à cause des poursuites judiciaires lancées à son encontre. Le serveur Napster a bousculé l'industrie audiovisuelle en proposant un accès gratuit aux contenus numériques soumis aux droits d'auteurs. Cette affaire Napster, n'a pas empêché l'emergence d'une panoplie d'applications p2p de partage de fichiers. Ces applications p2p sont devenues très rapidement la source de la majeure partie du trafic dans les réseaux IP des opérateurs des télécommunications dans le monde entier.

En effet, aujourd'hui, l'observation des liens transportant le trafic ADSL montrent que la part du trafic des services p2p est relativement grande et dépasse parfois la barre des 70% du trafic global. Cette observation est commune pour tous les réseaux publics et en particulier pour le réseau de France Télécom. La compréhension du fonctionnement des applications p2p et la prédiction des éventuels impacts sur le réseau des opérateurs sont devenues alors indispensables pour un opérateur comme France Télécom. L'émergence de ce phénomène est due à plusieurs facteurs très variés. Cependant, la démocratisation des accès haut débit reste le facteur le plus déterminant dans l'évolution vertigineuse de ces applications p2p. En effet, les temps de téléchargement des contenus audiovisuels (musique, vidéo) sont énormément réduits avec les bandes passantes offertes par la technologie ADSL. Aujourd'hui, la majeure partie des données véhiculées dans les réseaux p2p sont des données audiovisuelles. En général, ces conte-

nus échangés ne sont pas libres de droits. Cet aspect légal a été marqué par l'affaire Napster, le pionnier des systèmes p2p d'échanges de contenu. Les poursuites judiciaires lancées à l'encontre des concepteurs de Napster ont conduit à la fermeture du serveur. Mais contrairement à ce qu'on attendait, cette décision n'a fait qu'amorcer le développement d'une multitude de nouveaux systèmes d'échanges de fichiers plus robustes et moins vulnérables.

Cependant, ces poursuites judiciaires ont énormément compliqué les études de ce type de réseau. En effet, les utilisateurs préfèrent camoufler leur trafic p2p via les différents mécanismes mis à leur disposition. La caractérisation de la population eDonkey s'est révélée en conséquence relativement difficile. En effet, celle-ci passe par une identification plus au moins précise du trafic engendré par l'application eDonkey. L'approche classique pour observer le trafic d'une certaine application consiste à analyser les ports source et destination. Cette approche que nous avons adoptée au début n'est pas très efficace dans le cas des applications p2p. La contribution d'eDonkey est bizarrement en dessous de nos prédictions. Cette observation nous laisse présager qu'une bonne partie du trafic passe par les ports non usuels. L'étude des réseaux p2p passe tout d'abord par une étape d'identification. Cette étape préliminaire est essentielle pour la caractérisation de ces réseaux en termes de trafic et de topologie. Le premier chapitre de la thèse traite de cette problématique d'identification du p2p et essentiellement de l'eDonkey, en présentant une panoplie de méthodes qui existent dans la littérature ainsi que d'autres méthodes développées au cours de la thèse. Après avoir obtenu une proportion de trafic jugée représentative de l'échantillon de la population eDonkey étudiée, nous avons étudié les caractéristiques en termes de trafic de cette composante p2p en utilisant entre autres la dichotomie souris/éléphant

Outre les caractéristiques du trafic, nous avons analysé, dans le deuxième chapitre, la topologie du réseau eDonkey dans le réseau de France Telecom à l'échelle nationale et internationale. La première étape de la caractérisation d'une topologie d'un système donné revient à caractériser une image figée dans le temps et formalisée sous la forme d'un graphe dont les sommets représentent les pairs et où les arcs représentent une information binaire (est ce qu'il y a eu une connexion ou pas entre ces deux pairs ?). Afin d'enrichir cette image, trois actions possibles peuvent être menées. La première consiste à introduire des poids sur les liens. Ces poids nous renseignent sur les volumes écoulés entre deux sommets ou encore le débit moyen qu'on a pu observer entre eux. La deuxième action consiste à obtenir des poids instantanés au lieu des poids moyennés sur la période de la mesure. Ceci permet d'intégrer le paramètre du temps pour cerner la dynamique du système. La connaissance du comportement d'un seul client p2p aidera à comprendre et à formaliser cette dynamique. Une troisième action agit plutôt sur les sommets du graphe. Ces sommets représentent les adresses IP des clients p2p. Afin de connaître la capacité des systèmes p2p à utiliser d'une façon optimale, la topologie physique sous-jacente, d'autres niveaux de granularité doivent être introduits. Une agrégation des adresses IP selon des entités plus grandes, intimement liées aux mécanismes de transport chez l'opérateur, comme les AS, les pays ou les routeurs de raccordement au réseau doit être réalisée. En effet, un client eDonkey ne possède aucune conscience géographique et il est possible parfois de rapatrier des fichiers à partir de sources situées un peu partout dans le monde même si le contenu recherché est disponible chez un voisin géographique. Dans le système eDonkey, un voisin logique n'est pas forcément un voisin géographique. L'équivalence entre ces deux types de voisins est bénéfique non seulement aux FAI (coût de peering, trafic inter FAI) mais aussi pour les utilisateurs qui auront des temps de téléchargement et de réponse plus petits. Nous avons tendance à considérer que le système eDonkey n'est régi par aucune loi et s'étend sur le monde entier d'une façon aléatoire. Afin de comprendre un peu plus les

aspects géographiques du trafic eDonkey, nous avons choisi de caractériser la matrice de trafic eDonkey dans le réseau national de France Telecom. Les outils de métrologie de trafic nous permettent d'avoir une certaine connaissance sur le trafic p2p. Des travaux antérieurs sur la métrologie, menés au sein de France Télécom dans le cadre d'une thèse précédente, ont permis d'inférer quelques caractéristiques propres au trafic p2p. Ce trafic est marqué par une forte activité de signalisation, qui se manifeste par des rafales de petits messages. Parallèlement à la signalisation, on observe également des phases de transfert des données caractérisées par des connexions de longues durées et un nombre élevé de paquets. L'apparition de ces deux composantes de trafic p2p est indissociable du comportement des clients p2p et surtout du mode de fonctionnement des protocoles qui régissent la communication dans les réseaux logiques p2p. Ces deux composantes de trafic se prêtent bien à une modélisation mathématique dès qu'on adopte un découpage approprié du trafic observé. Le trafic de signalisation peut être décrit par un modèle basé sur des processus stochastiques liés à des files d'attente du type $M/G/\infty$. Ce trafic, malgré sa prédominance en termes de nombre de flots, contribue à moins de 5% du volume total. Ce déséquilibre entre nombre de flots et contribution en volume est la différence fondamentale avec la deuxième composante du trafic p2p, constituée d'un petit nombre de flots mais engendrant la majeure partie du volume observé. En se basant sur une agrégation adéquate, le trafic des données peut être lui aussi décrit par un modèle mathématique simple. En plus de la modélisation du trafic p2p, il est important de compléter ces travaux par une analyse du comportement des systèmes p2p responsable de ce trafic. L'analyse de ce comportement permet de comprendre plusieurs phénomènes que nous pouvons observer dans la réalité. En effet, l'observation des systèmes réels montre des systèmes d'une extrême complexité. Cette complexité est attendue étant donné que les systèmes en question se présentent comme une gigantesque nébuleuse constituée d'un très grand nombre de pairs très hétérogènes et sollicitée par des humains aux comportements très divers. Afin de prédire le comportement de ces systèmes complexes et d'évaluer les performances qu'ils offrent aux utilisateurs, leur modélisation est une tâche nécessaire. Dans la littérature, plusieurs modèles des réseaux p2p sont proposés. Ces modèles utilisent une myriade de formalismes mathématiques (réseau fermé ou ouvert de files d'attentes, modèles fluides,...etc.). Dans le troisième chapitre nous présentons notre propre modèle mathématique qui décrit la manière avec laquelle le réseau eDonkey se construit. Un simple modèle préliminaire basé sur la simulation a été proposé. Ce modèle nous fournit une explication plausible de la forte activité de signalisation que nous observons dans le système eDonkey réel.

Chapitre 1

Les premières observations du trafic

Sommaire

1.1 La caractérisation des réseaux pair à pair

1.1.1 Les paramètres de caractérisation

Pour caractériser un système p2p, un ensemble de paramètres doit être mesuré. La littérature identifie quatre classes de paramètres de caractérisation :

La topologie : Les paramètres de cette classe visent à caractériser la topologie du système. La connaissance de la distribution des pairs géographique ou logique (identifiés selon plusieurs niveaux d'agrégation par leurs adresses IP ou par les préfixes AS,...)permet de connaître la taille du système. La connaissance des degrés des paires du système est un premier pas vers la détermination de la topologie. La caractérisation de la topologie consiste dans un premier lieu à établir un graphe dont les paires (selon un certain niveau d'agrégation) représentent les sommets. Les arcs qui lient les sommets représentent à ce point de l'analyse une information binaire (une connexion est en cours entre ces deux paires) sans aucune précision supplémentaire (pas de poids sur les liens).

Le Trafic : Les paramètres de trafic permettent de donner un poids aux liens qui lient les sommets du graphe. Les poids peuvent représenter les volumes écoulés entre deux sommets ou le débit observé entre eux pendant la mesure. Ces paramètres préparent le terrain pour une éventuelle comparaison entre l'écoulement de trafic d'une application p2p et d'autres applications. Malgré la présence du paramètre temps dans le débit, le graphe établi jusqu'à maintenant reste une image figée du réseau réel au moment de la mesure. Cette image n'exprime pas le caractère évolutif du système étudié.

La dynamique : Ces paramètres caractérisent l'évolution du système au cours du temps. Les systèmes p2p évoluent très rapidement. Les paires arrivent et quittent le système d'une façon aléatoire et la durée de vie des liens est très variable. Cette aspect évolutif du système étudié doit être pris en compte en déterminant d'autres paramètres tel que la durée d'une connexion, la durée d'activité d'un paire ou les périodes de son inactivité.

Les paramètres sociaux : L'étude de ces paramètres revient à déterminer et à caractériser d'éventuelles communautés sémantiques. Dans cette classe on pourra intégrer des connaissances sur le degré de coopération des paires et leur contribution dans le système étudié.

1.1.2 Les mesures

Pour caractériser la topologie des réseaux pair à pair réels, la campagne de mesures est une étape préliminaire indispensable. Dans la littérature, il existe une multitude d'approches pour faire des mesures mais que nous pouvons classer tout de même dans deux grandes familles : les mesures actives et les mesures passives.

Mesures Actives

Les mesures actives consistent à intégrer un crawler actif dans le système. Le crawler est un client d'une application donnée que l'on modifie dans le but de faire de mesures sur le réseau associé à cette application. En faisant partie intégrante de ce dernier, le crawler actif procédera à une collecte d'informations et de caractéristiques des différents éléments du système.

En général, un crawler actif est un client p2p modifié qui permet d'avoir des caractéristiques diverses du système. Le client p2p modifié peut récolter des données variées comme la bande

passante des clients, les délais de propagation, les fréquences de connexion/déconnexion, les fichiers partagés ou le degré de coopération,...etc.

Le degré de sophistication du crawler actif varie énormément. Il peut partir d'une approche très simple en utilisant des outils basés sur le protocole ICMP (ping/pong basique). Cependant, les résultats de cette approche restent limités. Au contraire, des méthodes plus évoluées nécessitent un logiciel spécialisé [58]. Entre les deux, des approches intermédiaires sont possibles et offrent un bon compromis entre l'efficacité et la simplicité. Pour étudier le réseau Gnutella, les auteurs de [73] ont utilisé un outil appelé LF. Cet outil se base sur la plateforme de mesure Sting [76].

Dans une première étape, les auteurs [73] recensent les clients connectés au réseau Gnutella. La deuxième étape consiste à contacter à nouveau tous les clients recensés afin de déterminer un certain nombre de leurs caractéristiques. L'ensemble des pairs Gnutella présentent une hétérogénéité remarquable en termes de temps de latence, de bande passante et de disponibilité. Les auteurs [73] ont noté également la tendance des paires à fournir de fausses informations. En plus des temps de latence ou de la bande passante disponible des pairs du système Gnutella, le crawler actif permet également de donner la topologie qui maille l'ensemble de ses éléments.

Les auteurs de [70] ont utilisé un crawler actif dans le but d'établir une topologie du réseau Gnutella. Pour ce faire, le crawler est préconfiguré avec une liste de pairs Gnutella. Cette liste est parcouru séquentiellement et chaque pair est contacté. Le crawler récupère la liste des voisins du pair correspondant présente dans les messages *Pong* du protocole Gnutella. Cette approche consomme énormément de ressources (ressources réseau et CPU). En effet, pour construire une topologie avec seulement 4000 nœuds, la collecte a duré plus de 50 heures. Pour contourner ce problème, les auteurs de [70] ont eu l'idée de paralléliser la tâche entre plusieurs clients. Chaque client s'occupe d'une partie de la liste publique. Cette parallélisation accélère l'opération mais au fond le problème est loin d'être résolu.

La nature de l'objet de mesure exige énormément de ressources CPU et beaucoup de bande passante. En effet, la topologie d'un réseau p2p évolue avec une rapidité extrême. En conséquence, le crawler doit être pourvu d'énormes ressources pour obtenir une copie fidèle du réseau. Sans ressource, le crawler serait très rapidement incapable de dresser la topologie d'un réseau de taille conséquente. En plus des ressources, l'approche du crawler actif exige aussi une connaissance très précise du protocole étudié. Cette contrainte n'est pas toujours facile à satisfaire. En effet, les spécifications des clients pour quelques protocoles sont tout simplement tout absents.

Pour finir avec ce type d'outil de mesure, notons que le crawler actif peut endosser le rôle de n'importe quel élément du système peer to peer. Prenons l'exemple du système eDonkey. L'indexation des contenus se fait d'une façon centralisée au niveau d'un certain nombre de serveurs géographiquement dispersés. Le serveur peut être installé sur n'importe quelle machine désirant jouer un rôle dans l'indexation. En conséquence, un serveur peut jouer, en plus de son rôle d'indexation, un rôle d'espion sur le système en analysant les requêtes envoyées par les clients et par les autres serveurs d'indexation. L'analyse de ces deux entités (requêtes et réponses) aide à déterminer, par exemple, la popularité d'un certain contenu (nombre de requêtes reçues) ou la liste des pairs susceptibles d'être contactés par le client qui a fait la requête.

L'observation du système eDonkey à partir d'un serveur d'indexation offre une vue plus panoramique que l'approche client. Mais malgré son efficacité pour l'analyse de composants sémantiques du système p2p, cette approche est incapable de fournir des informations sur

l'écoulement du trafic entre les pairs. En effet, la liste des sources de contenu envoyée par le serveur vers le client n'implique pas forcément un échange de données entre ce client et les sources. En plus, il est possible d'initier un transfert sans être amener à interroger le serveur.

Par exemple, les auteurs du client emule supposent que deux pairs qui s'échangent un fichier ont une forte probabilité d'avoir des centres d'intérêts en commun. En conséquence, le logiciel permet aux clients en communication de s'échanger leurs listes de fichiers partagés sans passer par un serveur d'indexation.

Mesures passives

Pour les mesures passives, l'outil d'observation ne fait plus partie du système étudié. Ce caractère non intrusif de la méthode limite les éventuelles interférences entre l'outil d'observation et l'objet observé. Les mesures passives peuvent être exhaustives ou échantillonnées.

Une mesure exhaustive consiste à capturer tous les paquets d'un lien donné. La première étape consiste à choisir le lieu de l'observation de trafic. Pour étudier le trafic Kazaa, les auteurs de [33] ont fait leur capture sur un lien de sortie d'un réseau d'un campus universitaire à Washington. L'auteur de [83] a également utilisé une capture d'un lien de sortie dans un réseau universitaire pour étudier le trafic eDonkey. Quant aux auteurs de [44] et de [43], le trafic étudié est un trafic commercial. Le lien sur lequel le trafic a été capturé est un lien OC48 (2.5Gbps) d'un fournisseur d'accès Internet américain. Le but de cette de capture est de caractériser le trafic p2p.

1.1.3 Constat sur les paramètres des réseaux p2p réels

L'analyse des caractéristiques des réseaux p2p révèlent un fort degré d'hétérogénéité entre les pairs de ces systèmes.

Le premier niveau de disparité existe entre les systèmes p2p. Ceci est relativement prévisible étant donné que les protocoles et les principes qui régissent deux systèmes p2p distincts sont généralement très différents.

Il existe aussi un deuxième degré d'hétérogénéité qui est au contraire inattendu. Cette hétérogénéité est observable au sein d'un même système p2p. En effet, les différents clients d'un même réseau peuvent évoluer les uns indépendamment des autres. Cette évolution continue et peut induire des comportements différents.

Le troisième et dernier degré d'hétérogénéité est relatif aux différences d'usage entre les utilisateurs des réseaux pair à pair. La contribution de chaque nœud en termes de trafic n'est pas du tout homogène. Cette disparité se traduit par l'apparition de la notion des clients-serveurs (heavy users) et des clients-consommateurs (free riders). Mais en général, ces disparités ont tendance à disparaître dès que nous atteignons un certain niveau d'agrégation. En effet, dans une étude d'AT&T [78], l'analyse des traces Netflow issues des routeurs de bord montre que les volumes entre les systèmes autonomes restent relativement stables au cours du temps.

1.2 Les premières observations sur un lien GigabitEthernet

Dans cette section, nous présentons quelques résultats sur le trafic observé sur un lien Gigabit Ethernet (GE) reliant le Réseau Backbone de la Collecte IP (RBCI) à plusieurs plaques ADSL. La capture a été faite dans le sens descendant (sens du RBCI vers les plaques ADSL) le 04 Novembre 2004 entre 18h00 et 18h30. La charge du lien était de 40.27%. Sur

les 137,725,328 paquets observés sur le lien, 81% sont des paquets TCP. Ces paquets TCP représentent environ 88% du volume cumulé sur les 30 minutes.

Les premières analyses se sont focalisées sur le trafic engendré explicitement par les clients p2p du réseau eDonkey. Le trafic p2p passe théoriquement sur les ports standard 4665 et 4672 en UDP et 4661 et le 4662 en TCP. Dans ce qui suit, le trafic eDonkey qui passe par les ports standard sera appelé trafic eDonkey clair ; il représente 27% du volume total cumulé sur les 30 minutes (TCP et UDP).

En prenant en compte le trafic eDonkey clair, on observe que ce protocole engendre 42% des paquets TCP. Cette contribution est plus faible (17%) pour les paquets UDP. Lorsqu'on s'intéresse aux volumes cumulés, on remarque que la contribution d'eDonkey dans le trafic TCP est de 30% et qu'elle n'est que de 4% pour UDP.

Dans le tableau 1.1, on peut trouver la composition du trafic ADSL (TCP) capturé sur le lien GigabitEthernet, en termes de volume.

	Application	Pourcentage
p2p	eDonkey	30.36%
	Gnutella	2.44%
	Bittorrent	0.35%
	Kazaa	0.35%
	Napster	0.95%
non p2p	http	27.77%
	NNTP	1.52%
	RTSP	1.01%
	FTP	0.77%
	POP3	0.88%
	Autres	31.46%

TAB. 1.1 – Composition du volume par application (TCP) .

L'analyse du trafic UDP montre que la contribution d'eDonkey est plus significative en nombre de paquets qu'en volume, ce dernier restant relativement négligeable. Cela confirme le fait que dans le réseau eDonkey, UDP ne sert pas à réaliser les transferts de fichiers ; son rôle se limite à la recherche et l'indexation de contenus.

1.2.1 Une analyse des numéros de port

Analyse des numéros de port

L'analyse du trafic sur le lien GE confirme l'importance de la contribution du p2p. Même en se limitant à une analyse de la partie claire du trafic p2p, celui-ci apparaît dans la liste des applications les plus présentes dans le trafic global. Parmi les dix premiers ports de destination en volume, on trouve huit ports de destination connus comme ports standard du p2p (eDonkey, Gnutella, Bittorrent, Napster,...). En ce qui concerne les ports de source, 3 ports standard du p2p figurent dans le liste des 5 premiers ports qui contribuent le plus en volume. Le protocole eDonkey est de loin le plus productif parmi les applications p2p. L'analyse des ports montre que les port 4662 et 4661 représentent plus de 20% (resp. 10%) de tous les ports de destination (resp. source).

Nous notons également, l'importance du trafic Web. Celui-ci est le plus volumineux lorsqu'on analyse les ports de source avec un pourcentage avoisinant les 25%. En regardant les

ports de destination, le trafic web est presque inexistant (moins de 1%). Notons ici que le port 80 dédié habituellement au protocole http peut être utilisé comme port d'écoute pour le p2p afin de déguiser celui-ci en trafic web et contourner ainsi d'éventuels filtrages au niveau des mécanismes de pare-feu. Par ailleurs, quelques réseaux p2p utilisent explicitement le protocole http pour la signalisation. Citons par exemple Bittorrent où l'on trouve un serveur http tournant sur le Tracker (coordinateur de téléchargement d'un fichier).

En dehors du web, on remarque également une évolution du trafic due aux news. Le protocole NNTP se classe tout de suite après les deux plus importantes applications p2p (eDonkey et Gnutella). Ceci peut être expliqué par l'utilisation des news pour les échanges de fichiers multimédia (films, MP3,...). Cette migration est très probablement la conséquence de la médiatisation des poursuites judiciaires contre les utilisateurs du p2p. Une autre conséquence de cette médiatisation est l'utilisation des ports non standard. Les ports 5662, 14662 ou 40662 sont sans doute des ports d'écoute pour les applications du réseau eDonkey. Par exemple, le port 5662 (en tant que port source) arrive devant le port du protocole ftp en ce qui concerne le volume. Les mêmes constatations sur la contribution des applications p2p en volume restent vraies pour la contribution en nombre de paquets pour ce numéro de port. En tant que port de destination, celui-ci arrive en seconde position avant les applications p2p Bittorrent, Gnutella ou Napster.

L'analyse des paquets UDP montre que eDonkey et Gnutella sont pratiquement les seules applications qui utilisent conjointement TCP et UDP. Avec WinMX (presque inexistant dans le trafic TCP), eDonkey et Gnutella sont les seules applications p2p dans la liste des 5 premiers ports les plus productifs en volume. Dans cette liste, on trouve en première place, le protocole d'encapsulation L2TP qui contribue à plus que 70% du volume en UDP. Le reste est partagé entre les applications p2p, les jeux en réseau (HalfLife, Quake) et aussi le port 5672 qui confirme la constatation faite sur le port 5662 pour TCP. La prédominance de L2TP en ce qui concerne le trafic UDP est due au fait que le trafic des ISP tiers est transporté dans le réseau de France Télécom dans des tunnels L2TP.

Conclusion partielle sur l'identification du trafic p2p. Pour conclure cette section, on peut remarquer que le trafic eDonkey, en dehors des ports standard 4661 et 4662, est essentiellement transmis en utilisant des ports "voisins" tels que 5662, 14662 ou 40662. En fait, en prenant en compte ces ports et les ports standard, on arrive à identifier une bonne partie du trafic eDonkey, à savoir 90% dans le cas particulier considéré dans cette section, qui n'est peut être pas représentatif à cause de sa durée limitée (30 minutes).

Analyse des couples de ports source/destination

En analysant le trafic TCP, on dénombre 1,278,301 couples de ports (source, destination). En moyenne un couple de ports engendre 107 paquets et un volume de 63 Ko ; la moitié de ces couples de ports se sont échangés moins de neuf paquets. Les ports relatifs au protocole eDonkey sont les ports qui communiquent le plus avec d'autres ports distincts.

Alors que la moyenne est de 20 ports de destination pour un seul port source, le port 4662, en tant que port de destination, a communiqué avec 58,999 ports de source (sur seulement 30 minutes). Sachant qu'on observe la même moyenne lorsqu'on analyse les ports de source, on constate que le port 4662 a communiqué avec 52,707 ports de destination distincts.

A titre de comparaison, le port 80 relatif au protocole HTTP est le port d'écoute pour 2,717 ports de source distincts (4,179 ports de destination si le port 80 est le port source).

Les ports 4662 et 4661 communiquent rarement entre eux ; dans le réseau eDonkey, ils sont plutôt des ports d'écoute. Un client qui veut initier une connexion avec un autre client, joint ce dernier sur son port d'écoute (4662 ou 4661 par défaut) et utilise un numéro port source aléatoire entre 1025 et 65535 (1025 est le numéro de port qui a le plus communiqué avec le port 4662).

Conclusion partielle. En observant les ports de destination, on constate, en supposant que chaque utilisateur a un numéro de port de source différent, qu'un nombre colossal d'utilisateurs (plus de 50,000) a essayé de communiquer en eDonkey clair avec les terminaux connectés aux plaques ADSL desservies par le lien GE observé et ceci seulement sur 30 minutes. Cette observation laisse présager de la taille gigantesque du réseau eDonkey.

1.2.2 Analyse des adresses

Sur les 30 minutes de capture, on dénombre 30,425 adresses de destination qui ont communiqué en TCP comme indiqué dans le tableau 1.2. Seulement 33.1% de ces adresses de destination ont au moins communiqué une fois en p2p clair (i.e. reçu au moins un paquet avec 4662 ou 4661 comme port de source ou de destination). Ce pourcentage atteint environ 72% pour les adresses des sources : sur les 1,191,465 adresses de source (TCP) observées, 856,761 adresses ont communiqué en p2p clair. En ce qui concerne les couples (source, destination), le pourcentage des couples qui ont communiqué en p2p clair est intermédiaire et avoisine les 62% (2,388,909 couples sur les 3,821,608 couples).

	TCP	eDonkey
Adresses Source.	1,191,465	856,761 (72%)
Adresses Dest.	30,425	10,072 (33.1%)
Couples	3,821,608	2,388,909 (62%)

TAB. 1.2 – Les adresses de source et de destination TCP et eDonkey.

Un couple d'adresses eDonkey clair s'échange (statistique sur un seul sens) en moyenne 24 paquets (36 paquets pour TCP). Le volume de données échangé par couple est en moyenne égal à 10 Koctet (cumulé sur 30 mn), il est le double pour TCP (le volume eDonkey est contenu dans le volume TCP total). En analysant par adresse, on remarque qu'en moyenne, une adresse de source de type eDonkey envoie environ 28 Koctets (toujours sur 30 minutes). Une adresse de destination reçoit en moyenne environ 2.5 Mo. Ces données reçues proviennent en moyenne de 238 adresses de source et vont vers 3 adresses de destination. Le faible volume, en moyenne, transmis par les sources s'explique par le fait que les utilisateurs qui rapatrient des données à partir d'un terminal connecté à l'une des plaques ADSL desservies par le lien observé n'envoie que des messages d'acquittement ou alors que les clients ont communiqué avec un super nœud. Par contre, les clients ADSL qui rapatrient des données reçoivent un volume conséquent (sur 30 mn).

En analysant les adresses de source de manière plus précise, on s'aperçoit que le serveur belge Razorback est la source qui a le plus communiqué avec des adresses de destination (875 adresses). Dans le trafic TCP, il n'est cependant classé que 43ème. De plus sa contribution en volume est négligeable. Sur les adresses de source eDonkey, il n'est que le 1341ème client en volume. Ceci s'explique par le fait qu'un super nœud eDonkey sert seulement à la tâche d'indexation ; il ne fournit ni ne télécharge de fichiers. Les plus volumineux sont essentiellement

des clients Internet. Les trois premiers sont des abonnés de FAIs francophones (le Suisse Cablecom, Free, le Québécois Sympatico). Les clients français semblent avoir une préférence linguistique (films en version française par exemple).

Sur toutes les adresses de source observées sur les 30 minutes, la plus importante en volume est celle du serveur de news news-europe.giganews.com ; ce serveur a servi 4 adresses de destination. Ceci confirme une fois de plus l'augmentation des news dans le trafic observé. Ce serveur de news a le plus contribué dans le volume total TCP. Il est en l'occurrence responsable de la majeure partie du trafic des news. Par ailleurs, plus de 8.500 adresses de source génèrent 80% du trafic eDonkey. Cette constatation reste vraie pour les adresses de destination mais le volume est mieux réparti sur les clients. En effet, 80% du volume sont à destination d'environ 8% des clients soit 750 adresses de destination. La loi de Pareto (une faible proportion des adresses engendre la majorité du volume) n'est pas spécifique au trafic p2p. La figure 1.1 montre en outre que le trafic eDonkey est plus réparti sur les adresses destination que le trafic TCP global. Le phénomène est encore présent lorsque on s'intéresse aux couples d'adresses et il est même plus intense. Plus de 90% du trafic eDonkey est engendré par seulement 1% des couples d'adresses.

Ces gros fournisseurs (ou consommateurs) du trafic eDonkey clair sont également parmi les gros consommateurs de la bande passante du trafic TCP. Le trafic eDonkey des fournisseurs représente 94% de leur trafic total. Pour les consommateurs (adresse de destination) ce pourcentage est de l'ordre de 70%.

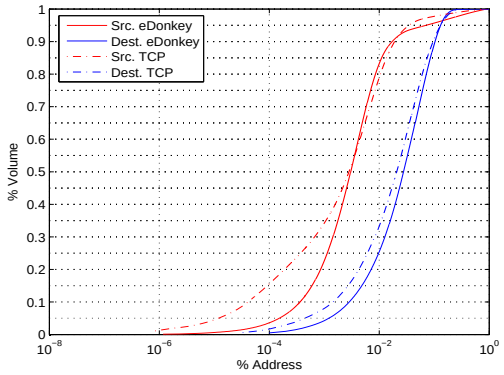


FIG. 1.1 – Répartition du volume suivant le pourcentage des adresses de source et de destination

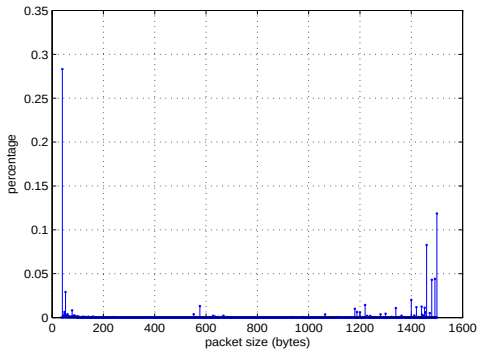
Conclusions partielles. L'analyse des adresses donne des renseignements très importants sur la structure du réseau eDonkey :

1. environ 30% des clients ont reçu du trafic sur les ports standard ; si ceux-ci sont utilisés comme port d'écoute, on peut s'attendre à ce que 30% des clients jouent le rôle de serveurs p2p ; ils correspondent à peu près aux "heavy users" en termes de trafic qui sont régulièrement observés lors de l'analyse du trafic ADSL ;
2. une faible proportion de clients rapatrie des données (8% sur la demi-heure observée) ;
3. les clients p2p semblent avoir une préférence linguistique.

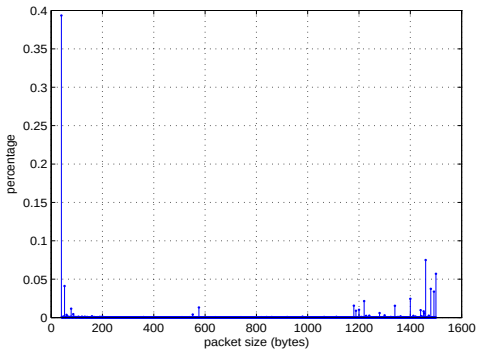
1.2.3 La dichotomie Souris/Eléphants

L'application de la dichotomie Souris/Eléphants sur le trafic eDonkey clair et le trafic TCP montre que le nombre des éléphants (flots avec plus 20 paquets) est largement inférieur à celui des souris (flots avec moins 20 paquets). En effet, les flots souris représentent plus de 96% du nombre total des flots (98% pour le trafic eDonkey). Malgré l'infériorité numérique en termes de flots, les éléphants représentent 93% du volume total pour le trafic TCP et également pour le trafic eDonkey clair. Lorsqu'on s'intéresse au nombre de paquets, le contraste entre ces deux types de flots est moins intense ; les éléphants engendrent environ 62% des paquets et les souris sont essentiellement formées de paquets de petite taille.

L'analyse de la répartition de la taille des paquets des éléphants montre l'existence de deux classes (voir 1.2(a) et 1.2(b)). La première classe est concentrée autour de la taille de 40 octets ; ce sont les paquets des éléphants ACK (acquittements). La deuxième classe se situe dans l'intervalle 1200 et 1500 octets. Ce sont les éléphants réguliers, qui sont associés aux transferts de données. Un pic est présent au niveau de la taille de 576 octets, relative à la valeur de la MTU du protocole IP standard.



(a) Paquets TCP



(b) Paquets eDonkey clair

FIG. 1.2 – La répartition de la taille des paquets

Les paquets souris sont essentiellement de petite taille. Cette dernière ne dépasse pas les 140 octets. Le pic principal est situé au niveau de la valeur de 40 octets (40% des paquets ont cette taille).

En se basant sur la taille moyenne des paquets d'un flot, on sépare les flots éléphants en deux classes : les éléphants d'acquittement (éléphants ACK) et les éléphants réguliers qui correspondent au transfert de données. Les flots réguliers ne représentent que 58% de la totalité du nombre des éléphants, mais en revanche, il sont responsables de la majorité du volume des éléphants cumulé sur les 30 minutes (97% du volume).

L'analyse des deux classes montrent une certaine similitude lorsqu'on s'intéresse au nombre de paquets par flot. Par contre, des disparités apparaissent dès qu'on s'intéresse au critère volume. Un flot éléphant régulier transporte en moyenne 0.8 Mo. Cette moyenne est 25 fois plus petite pour les flots ACK. Ces moyennes sont calculées sur la durée de la capture de 30 minutes. Il est possible que des flots soient tronqués ce qui explique l'absence d'une valeur caractéristique (taille d'un chunk par exemple).

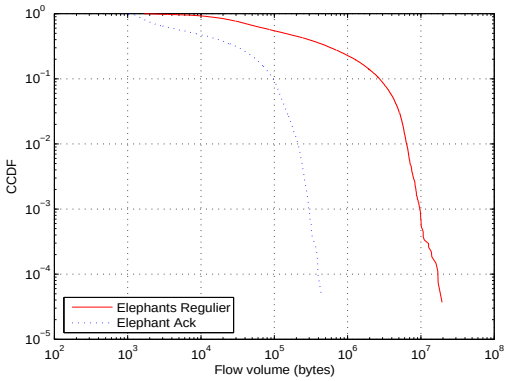


FIG. 1.3 – Fonction de répartition cumulative complémentaire de la taille des éléphants

1.3 Les premières observations sur un lien OC3

Dans cette section, on analyse une trace de 3 heures réalisée dans les deux sens d'un lien OC3 le 7 Mars 2005 entre 18h et 21h. On se limite dans ce qui suit à une analyse des paquets TCP. On appelle sens descendant le sens qui véhicule le trafic vers les utilisateurs. Le sens qui remonte des utilisateurs vers le RBCI est appelé sens montant. Les adresses de destination du lien descendant et les adresses de source du lien montant seront appelées adresses internes (i.e. les adresses des clients ADSL connectés à la plaque ADSL observée) par opposition aux adresses externes.

1.3.1 Généralités

Le nombre de paquets TCP observés pendant les 3 heures dans le sens descendant est de 82,540,876. Ces paquets engendrent un volume égal à 49,900,262,357 octets. Sur le sens montant, on capture 2% de plus de paquets TCP (84,140,885). En revanche, le volume cumulé sur les 3 heures dans le sens montant ne représente environ que 67% du volume cumulé dans le sens descendant. Cette dissymétrie naturelle de l'accès ADSL n'est pas surprenante mais elle est malgré tout assez modeste, ce qui indique une utilisation de plus en plus symétrique du réseau avec la montée en puissance des protocoles p2p ; une symétrisation parfaite est encore contrariée par des débits d'accès différents d'un sens à l'autre.

Pour illustrer le phénomène de symétrisation des flux, prenons par exemple le réseau eDonkey. Le volume engendré par ce réseau dans le sens montant (52.55% du volume) est deux fois plus important que le volume observé dans le sens descendant. L'évolution du trafic montant des clients, négligeable avant la montée en puissance des applications p2p, va probablement pousser les FAIs à repenser leur modèle économique. En effet, ce modèle basé sur une hypothèse d'asymétrie des deux sens, n'est plus valable quand le p2p est plus important que le web dans le palmarès des applications les plus populaires. Ce dernier est marqué par un fort contraste entre les deux sens. L'analyse de la trace montre que le trafic Web représente 34.57% du volume descendant et seulement 6.22% du volume cumulé pendant 3 heures sur le sens montant. La répartition du volume pour le reste des applications est donnée dans le tableau 1.3.

	Application	Pourcentage Up	Pourcentage Down
p2p	eDonkey	53%	17%
	Gnutella	5%	5%
	Bittorrent	$< 1\%$	$< 1\%$
	Napster	1%	1%
non p2p	HTTP	6%	35%
	NNTP	$< 1\%$	1%
	Autres	34%	41%

TAB. 1.3 – Répartition du volume par application (sens montant et descendant).

Notons ici que l'on entend par trafic eDonkey le trafic TCP qui passe non seulement par les ports standard de l'application eDonkey mais aussi celui qui passe par les ports voisins tels que le 5662, 40662 et le 14662. Dans le tableau 1.4, on donne la contribution de chacun de ces ports dans le trafic eDonkey obtenu.

	Port	Pourcentage Up	Pourcentage Down
Ports standards	4662	90.52%	66.24%
	4661	2.88%	5.63%
Ports voisins	5662	5.17%	20.08%
	14662	$< 1\%$	$< 1\%$
	40662	1.12%	7.65%

TAB. 1.4 – Pourcentage en volume des ports standard et ports voisins.

1.3.2 Analyse des adresses

Sur les 3 heures d'observation, on dénombre 2,848 adresses de destination TCP dans le sens descendant. Près de la moitié de ces adresses ont communiqué en eDonkey (soit 1,423 adresses). Le pourcentage des adresses de source qui ont communiqué en p2p est plus important. Sur les 684,941 adresses de source TCP, 436,658 ont communiqué en p2p (soit 63.75%). Les couples d'adresses p2p représentent 58% du nombre de couples TCP (679,096 sur 1,170,084).

Comme indiqué dans le tableau 1.5, l'analyse du sens montant montre que le nombre d'adresses internes TCP est légèrement différent du nombre d'adresse interne vues dans le sens descendant (2,639 adresses contre 2,848) : soit ces adresses ne répondent pas aux sollicitations des clients extérieurs, soit ces adresses ne sont plus actives (terminaux éteints ou en veille, ou adresses plus allouées). Cette constatation n'est pas valable pour les adresses externes : 684,941 adresses externes dans le sens descendant ont été observées et dans le sens montant le double a été observé (1,366,782).

	TCP Down	TCP Up	eDonkey Down	eDonkey Up	HTTP Down	HTTP Up
Ext.	684,941	1,366,782	436,658 (63.7%)	467,851 (34.2%)	23,786 (3.4%)	75,181 (5.5%)
Int.	2,848	2,639	1,423 (49.9%)	732 (27%)	2,638 (92.62 %)	2,450 (92.8%)
Couples	1,170,084	1,861,887	679,096 (58.0%)	726,941 (39.04%)	100,292 (8.5%)	150,450 (8%)

TAB. 1.5 – Répartition des adresses internes et externes en fonction des applications.

Ce phénomène est inversé quand on analyse le trafic eDonkey clair. Pour les adresses externes, les nombres dans les sens montant et descendant sont sensiblement les mêmes (467,851

contre 436,658) alors que pour les adresses internes, on observe 732 adresses dans le sens montant et 1,423 dans le sens descendant.

Les adresses eDonkey externes ont communiqué majoritairement en eDonkey. En moyenne, 95% du volume reçu ou envoyé par ces adresses est du trafic eDonkey (plus de 83% de ces adresses ont communiqué uniquement en eDonkey). Ce comportement n'est pas observé au niveau des adresses internes. Pour le sens montant, le trafic eDonkey de ces adresses représente 38% de leur trafic total. Ce pourcentage descend à 16% lorsqu'on analyse le trafic descendant. Ceci confirmera les constatations que nous avons pu faire lorsqu'on a appliqué la méthode méthode d'identification basée sur les annuaires (voir la section 1.4). Les adresses eDonkey externes ne sont contactées que parce qu'elles appartiennent au réseau eDonkey. Le trafic reçu ou envoyé par ces adresses externes est un trafic purement eDonkey. En revanche, le trafic envoyé ou reçu par les adresses internes est le résultat d'une agrégation d'une multitude d'applications (p2p, http, mail,...). Cela ne veut pas dire pour autant que les adresses externes p2p n'utilisent que les applications p2p. En fait, le réseau, associé à un point d'observation particulier, devient une fonction de filtrage intrinsèque. Cette fonction permet d'éliminer le trafic des applications non p2p des adresses p2p. En contrepartie, une partie de leur trafic p2p est probablement ignorée.

1.3.3 Le trafic d'un client eDonkey

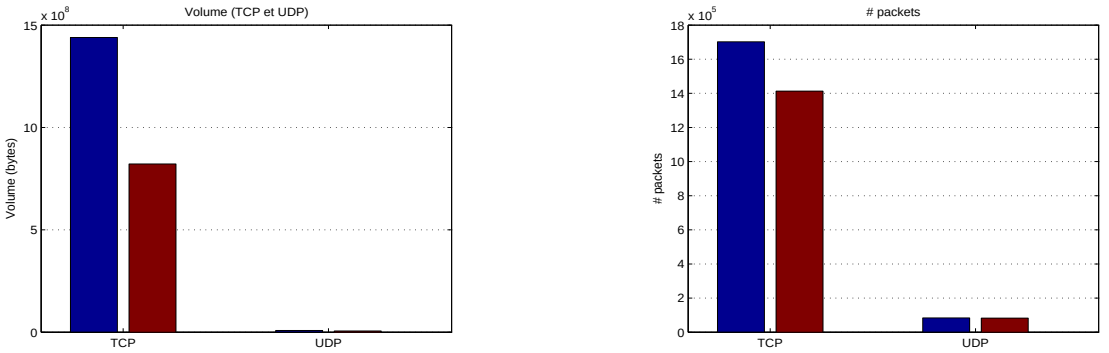
Cadre expérimental

Nous analysons dans ce rapport, le trafic d'un client eDonkey sur une durée de 15 heures. Afin de se mettre dans un contexte réel et pour se comporter comme un client type du réseau eDonkey, nous mettons en partage un fichier f_1 en partage. Le fichier f_1 a une taille de 732,476,072 octets. Nous lançons en parallèle le téléchargement d'un fichier f_2 d'une taille de 733,878,272 octets. Nous configurons le client avec les numéros de port d'écoute 4662 pour les connexions TCP et le 4672 pour les communications UDP. Nous limitons le nombre de sources pour un fichier à 500 sources et nous limitons le nombre de connexions simultanées à 600 connexions. Une troisième et dernière limite a été imposée par rapport aux débits de réception et d'émission (920 kbps en réception et 200 kbps en émission).

En analysant la répartition du trafic entre TCP et UDP, nous remarquons d'emblée que le trafic UDP est négligeable en termes de volume et de nombre de paquets par rapport au trafic TCP. Cette observation explique la tendance des études de modélisation de trafic qui se concentrent sur l'étude du trafic TCP. Une deuxième observation aussi importante que la première concerne la symétrie entre le sens montant et descendant du trafic. Le trafic pair à pair renverse le sens de l'asymétrie up/down du trafic web. Le volume cumulé envoyé par le client durant la capture est supérieure au volume cumulé qu'il reçoit. Le constat est le même lorsque nous nous intéressons au nombre cumulé de paquets (voir les figures 1.4(a) et 1.4(b)).

Le trafic TCP

Nous nous intéressons dans un premier lieu au trafic TCP. Dans le système eDonkey, le protocole TCP joue un rôle prépondérant dans la signalisation et dans le transfert des données. Afin de mettre en évidence ce double rôle assuré par le protocole TCP, nous appliquons la dichotomie souris/éléphant sur les flots TCP observés durant la capture. En plus de cette dichotomie, nous avons scindé les éléphants en deux classes supplémentaires. La première classe contient les éléphants dits réguliers. Les éléphants réguliers sont les éléphants dont la



(a) volume

(b) nombre de paquets

FIG. 1.4 – Statistiques TCP et UDP

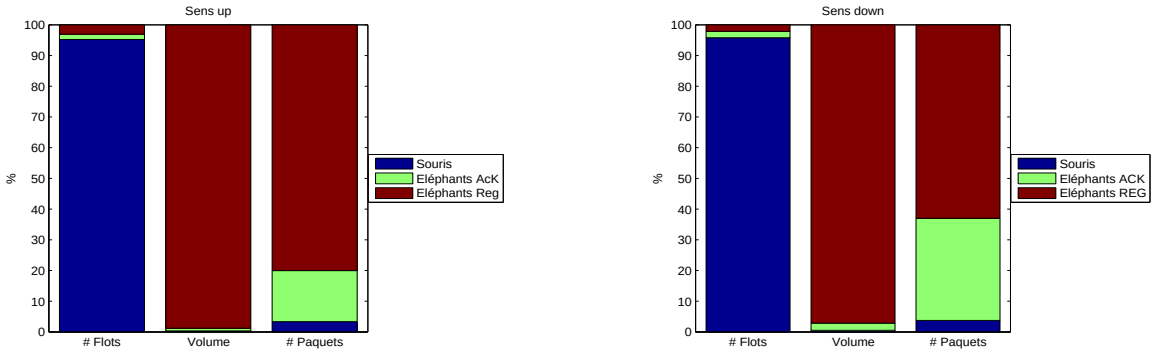
taille moyenne des paquets est supérieure à 80 octets. Les éléphants dont la taille moyenne des paquets est inférieure à ce seuil sont dits éléphants d'acquittement. La deuxième classe d'éléphants et comme son nom l'indique représente les flots qui acquittent le rapatriement d'un fichier dans le sens opposé. La première classe d'éléphants représente le transfert de fichier. L'analyse de cette classe montre que les éléphants réguliers sont entachés de périodes d'inactivité. Le transfert des données se fait avec des rafales de transmission séparées de périodes d'inactivité. Nous prenons la valeur de 20 secondes comme durée d'inactivité [6]. Un éléphant régulier est alors une succession de groupements de paquets séparés par des périodes d'inactivité. Si le groupement contient moins de 20 paquets, il est dit souris d'éléphant. Par ailleurs, si le groupement contient plus de 20 paquets il est dit mini-éléphants. En conséquence, un éléphant régulier peut être alors l'agrégation de mini-éléphant et de souris d'éléphant. Le tableau 1.6 indique la contribution de chaque entité dans le trafic global en termes de volume, nombre de flots et en termes de nombre de paquets.

		Souris	Éléphants	E ACK	E REG	Mini-E	SE
Up	Nb flots	6,715	337	124	213	237	754
	Volume	4,876,902	1,434,290,896	12,834,599	1,421,456,297	1,421,048,759	407,538
	Nb paquets	57,094	1,644,887	282,829	1,362,058	1,359,659	2,399
Down	Nb flots	6,752	295	146	149	223	162
	Volume	4,418,271	817,474,470	19,312,380	798,162,090	797,841,242	320,848
	Nb paquets	52,735	1,360,499	470,198	890,301	889,789	512

TAB. 1.6 – Dichotomie elephant souris

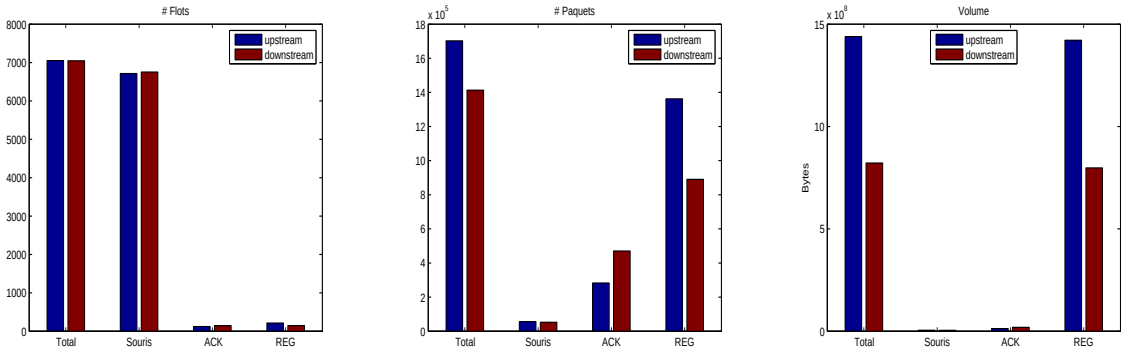
Confirmant les études antérieures sur la dichotomie souris/éléphants, cette analyse permet de mettre en évidence la prédominance des souris en termes de nombre de flots. En termes de volumes, les éléphants sont largement prépondérants. Les statistiques sur les flots sont résumées dans les figures 1.5(a), 1.5(b), 1.6(a), 1.6(b) et 1.6(c).

Par ailleurs, la caractérisation des mini-éléphants dans la trace du client montre que les transferts se font généralement d'une façon continue. En effet, dans la majorité des cas, les éléphants réguliers sont constitués d'un seul mini-éléphant (65% des éléphants réguliers sont formés d'un seul mini-éléphant, 25% sont formés de deux mini-éléphants). L'intégrité des



(a) sens montant (b) sens descendant

FIG. 1.5 – Dichotomie souris éléphants : statistiques I



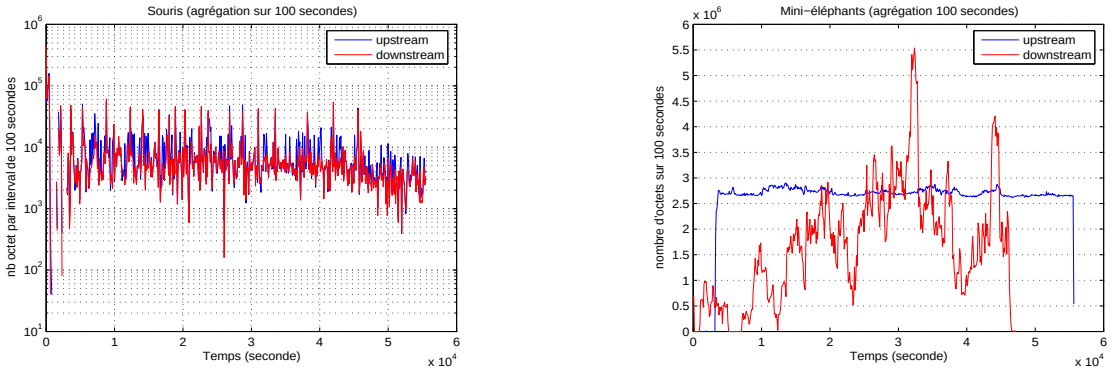
(a) nombre de flots (b) nombre de paquets (c) volume

FIG. 1.6 – Dichotomie souris éléphants : statistiques II

éléphants réguliers garantit une certaine stabilité dans leurs débits mais ne garantit en aucun cas la stabilité du débit global à cause de l'absence de synchronisme entre les sources. Ceci est du, premièrement au caractère très dynamique du réseau pair à pair (connexion, déconnexion, etc..) et deuxièmement au mécanisme de péremption des files d'attente au niveau des clients emule.

Sur les figures 1.7(a) et 1.7(b) nous représentons les débits (avec un intervalle d'intégration égal à 100 secondes). La figure 1.7(b) montre que le débit descendant est instable au cours du temps. En revanche, nous remarquons que le débit montant est plutôt stable au cours du temps. Cette constatation montre l'impact de la disponibilité du client sur la variabilité du débit. Notre client demeure connecté et donc disponible le long de la durée de la capture (pas de connexion/déconnexion). En plus, le nombre de mini-éléphants simultanés reste relativement constant dans le sens montant contrairement au sens descendant (voir les figures 1.8(a) et 1.8(b)).

Notons aussi que des limites sur les débits d'émission et de réception ont été instaurées. Nous remarquons que le débit d'émission se stabilise au niveau de la limite instaurée(200 kbps). Cette stabilisation au niveau de la limite ne contredit pas l'argumentation que nous avons présentée plus haut sur la stabilité du débit d'émission. En effet, malgré l'intervalle d'intégration relativement grand, le débit descendant est extrêmement variable contrairement au débit montant qui montre une variabilité négligeable autour de la limite de 200 kbps. Le débit d'émission ne chute pas au cours du temps en dessous de cette limite.

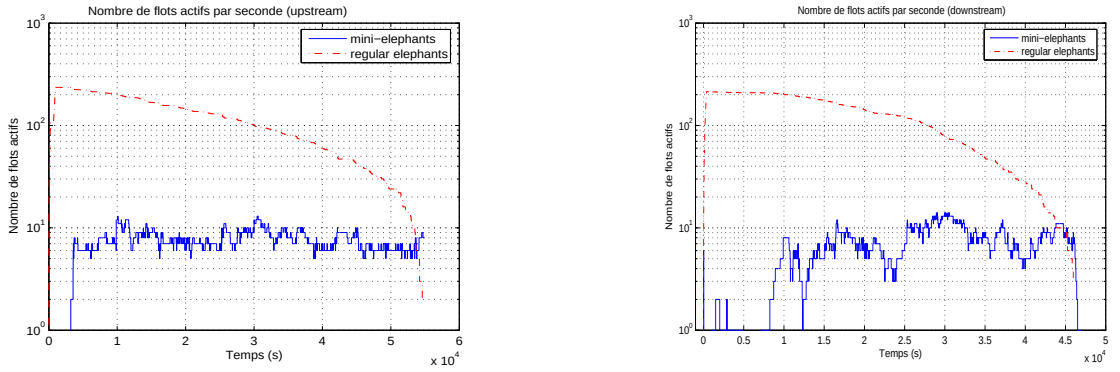


(a) Souris (b) Mini-Éléphants

FIG. 1.7 – Débits des différentes classes du trafic

Avant de s'intéresser à l'analyse du comportement des adresses externes, nous avons étudié les caractéristiques des grands transferts de fichiers représentés par les éléphants réguliers. L'étude des caractéristiques des éléphants réguliers nous permet de déceler les spécificités des transferts eDonkey selon quelques critères donnés. Pour ce faire, nous avons choisi quatre critères.

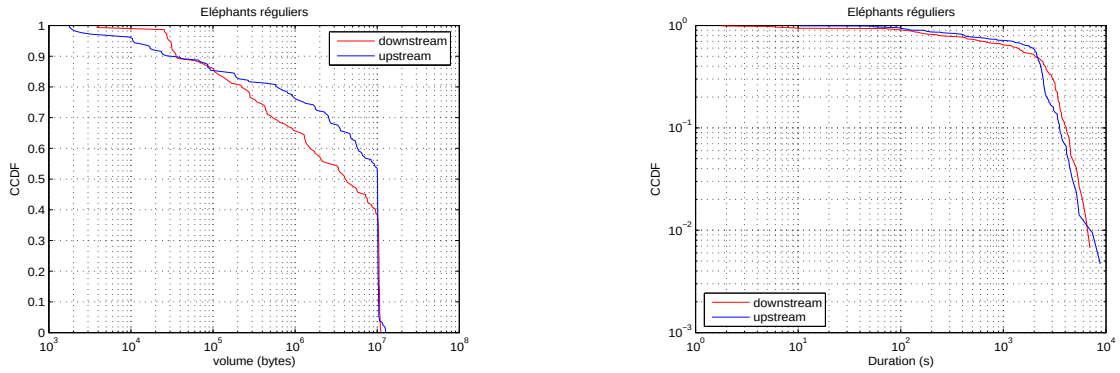
Les deux premiers critères sont le volume et la durée. Sur les figures 1.9(a) et 1.9(b), nous avons représenté les fonctions de répartition complémentaires selon ces deux critères pour les éléphants réguliers. Sur les figures 1.10(a) et 1.10(b), nous avons représenté les fonctions de répartition complémentaires associées aux deux autres critères. Ces deux critères sont le débit et la taille moyenne des paquets. Indépendamment du sens observé, le volume transporté par un éléphant régulier reste inférieur à 10 Mo. En plus, nous remarquons qu'un éléphant typique



(a) sens montant (b) sens descendant

FIG. 1.8 – Nombre d'éléphants actifs par seconde

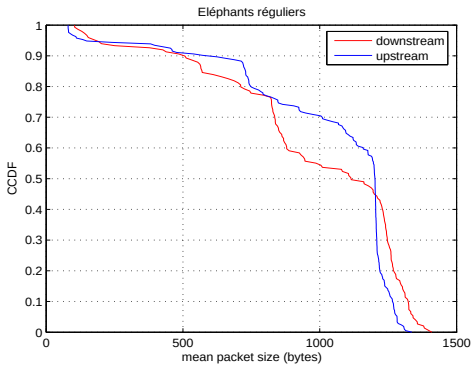
dure environ 30 minutes. Concernant les deux autres critères, nous pouvons constater que les débits des éléphants restent inférieurs à la valeur de 100 kbps. Cette observation a été déjà faite lors de l'analyse d'une trace Gigabits Ethernet en 2003 [6]. Concernant la taille moyenne des paquets, aucune valeur typique n'a été observée sauf autour de la valeur de la MTU de l'IP standard (512 octets). La taille des paquets reste inférieure à la valeur de 1,300 octets. Le tableau 1.7 donne les statistiques du premier et de deuxième ordre selon ces quatre critères pour les éléphants réguliers.



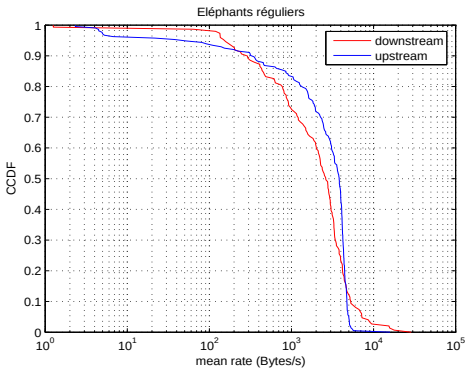
(a) Volume (b) Durée

FIG. 1.9 – ccdf : Les éléphants réguliers

Nous nous intéressons maintenant à l'analyse des adresses externes. Le nombre d'adresses externes s'élève à 1,603 adresses. Environ 99% de ces adresses ont échangé au moins une souris avec le client observé. En revanche, moins de 6% des adresses externes ont échangé au moins un éléphant régulier avec le client. Nous notons alors une forte activité de signalisation mais qui se transforme rarement en transfert de fichier. Par ailleurs, le client envoie des éléphants réguliers à des adresses externes sans aucune distinction. En effet, 56.9% des adresses externes ayant reçu un éléphant régulier de la part du client n'ont envoyé aucun éléphant régulier. Par contre, seulement 36.8% des adresses externes qui ont envoyé un éléphant régulier à notre client n'ont reçu aucun de sa part. Les adresses externes semblent adhérer à un principe de



(a) Taille moyenne des paquets



(b) Le débit moyen

FIG. 1.10 – ccdf : les éléphants réguliers

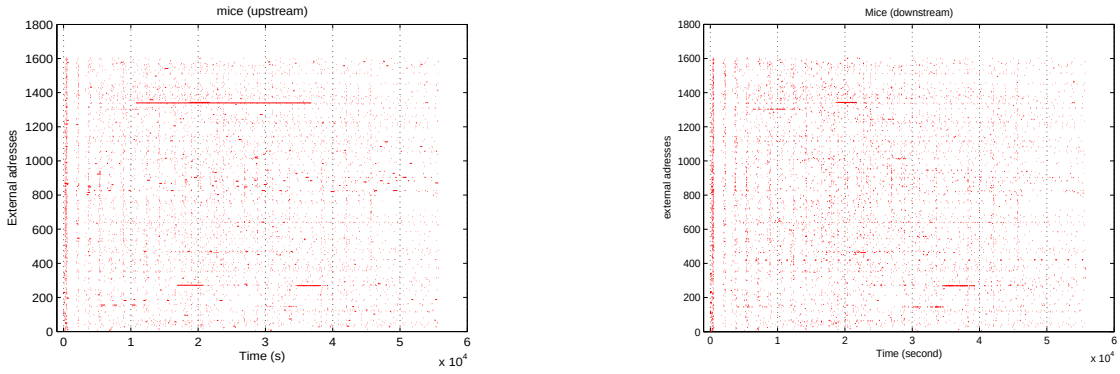
	Duration		Volume		Mean packet size		Mean rate	
	up	down	up	down	up	down	up	down
min	10.42	1.18	1,781	3,879	80.18	104.6	2.33	1.255
max	17,7530	7,413	12,7M	10,27M	1,337	1,410	16K	28.8K
mean	2,033	2,064	6.67M	5.35M	1,030	988.4	3,161	3,091
median	2,139	2,042	10.11M	4.18M	1,202	1,115	3,821	2,631
Std	1,740	1,658	4.47M	4.70M	314.3	344.5	1,841	3,486

TAB. 1.7 – Les statistiques du premier et de deuxième ordre

réciprocité avec le client. L'analyse de cette réciprocité en termes de volume révèle une forte asymétrie entre le sens montant et le sens descendant. En effet, parmi les 65 adresses externes qui ont envoyé et reçu un éléphant régulier, 52 adresses ont envoyé un volume supérieur au volume qu'elles ont reçu. En moyenne, le rapport du volume envoyé par le volume reçu est égal à 8 pour ces 65 adresses externes.

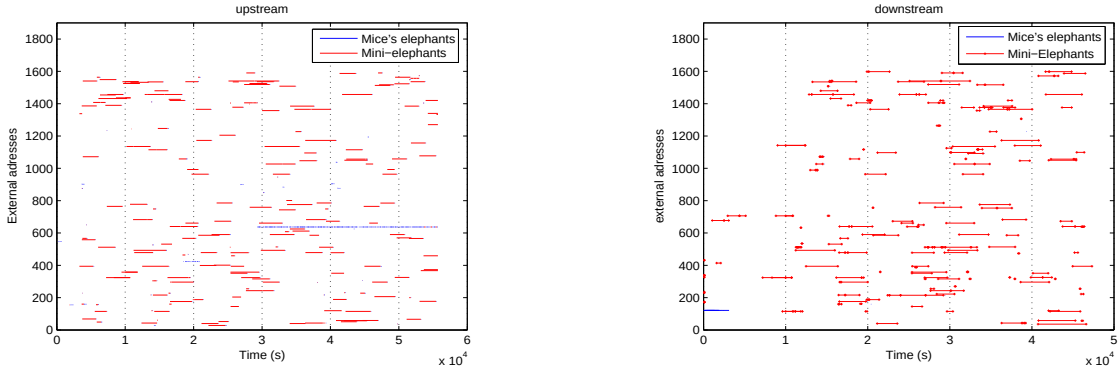
Dans les figures 1.11(a) et 1.11(b), nous représentons l'activité des adresses externes en termes de souris dans les deux sens du lien. Sur les deux figures, nous indexons les adresses de 1 à 1603. Prenons à titre d'exemple le sens montant, la longueur des segments horizontaux représente la durée du flot et la position par rapport à l'axe des ordonnées indique l'index de l'adresse externe ayant reçu ce flot. Comme nous pouvons le remarquer, les souris sont relativement de courtes durées et se répartissent bien sur le temps et sur les adresses externes. En revanche, la trace commence par une forte activité de souris marquée par la traînée verticale sur les deux figures 1.11(a) et 1.11(b) immédiatement après le déclenchement de la capture (et du client également).

La répartition relativement homogène des souris sur le temps et sur les adresses disparaît lorsque nous nous intéressons aux transferts de fichiers. Ces transferts mettent du temps avant de commencer d'une façon relativement homogène pour le sens montant (voir 1.12(a) et 1.12(b)). Dans le sens descendant, les transferts sont localisés sur des plages d'indexes et se font d'une façon complètement aléatoire dans le temps. Notons aussi que le rapatriement des données (*download*) commence tardivement par rapport au transfert de données (*upload*) dans le sens montant. Ce retard est sans doute dû aux mécanisme de file d'attente instauré dans le système eDonkey.



(a) sens montant (b) sens descendant

FIG. 1.11 – Activité des adresses externes en termes de souris

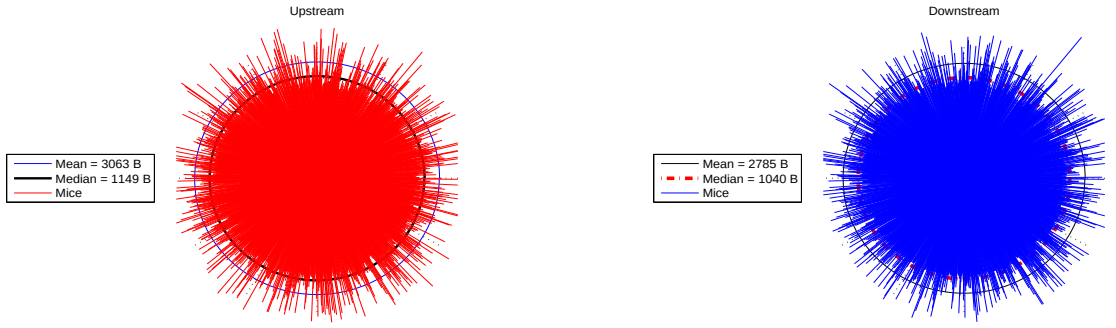


(a) sens montant (b) sens descendant

FIG. 1.12 – Activité des adresses externes en termes de mini-éléphants

Afin d'évaluer le degré d'homogénéisation des flots sur les adresses externes, nous représentons les flots dans un repère polaire (voir 1.13(a), 1.13(b), 1.14(a), 1.14(b), 1.15(a) et 1.15(b)). L'origine du repère polaire représente le client observé. En considérant les coordonnées polaires, une adresse externe est définie avec un angle donné et le rayon indique le volume envoyé (ou en l'occurrence reçu) par l'adresse externe considérée. Nous remarquons que les souris sont bien réparties sur les adresses externes. En effet, chaque adresse externe reçoit au moins une souris. Dans les figures 1.16(a) et 1.16(b) , nous représentons le nombre de souris (envoyées ou reçues) pour chaque adresse externe afin d'évaluer l'activité de ces adresses en termes de nombre de flots. Afin de déterminer la répartition de l'activité sur les adresses externes, nous représentons sur la figure 1.3.3 la fonction de répartition complémentaire du nombre de souris par adresse externe. Nous remarquons qu'une grosse partie des adresses (75% des adresses externes) envoie au plus trois souris. En conséquence, nous pouvons conclure que les adresses ne présentent pas une grande disparité en termes de nombre de souris par adresse. La forte activité de signalisation est suivie par une activité moins forte de transfert de données. En effet, nous remarquons dans les figures 1.14(a) et 1.14(b) que seulement une partie d'adresses découvertes par les souris reçoit ou envoie un éléphant régulier. Le sens montant montre une répartition un peu plus équitable sur les adresses externes que le sens descendant. Le client sert plus d'adresses externes en éléphants réguliers.

Pour finir avec l'étude des adresses externes, nous donnons dans les figures 1.17(a), 1.17(b), 1.18(a) et 1.18(b) le nombre d'éléphants réguliers et d'acquittement pour chaque adresse externe en gardant la même indexation. En moyenne, les adresses s'échangent un seul éléphant régulier et un seul éléphant d'acquittement.

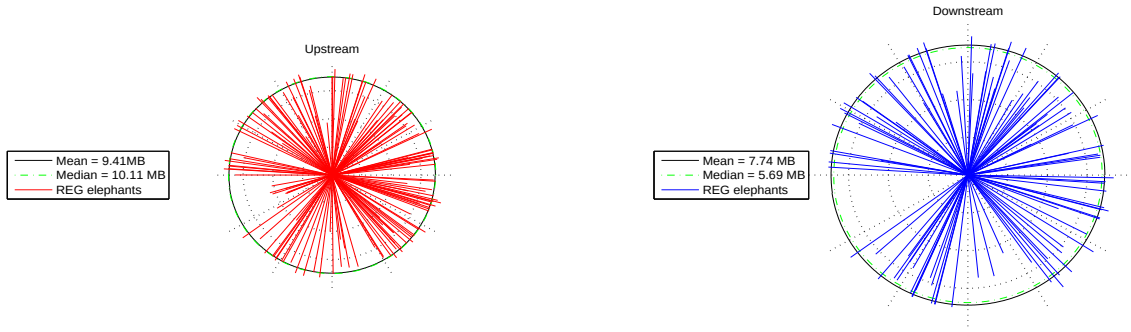


(a) sens montant (b) sens descendant

FIG. 1.13 – Souris (volume en échelle logarithmique)

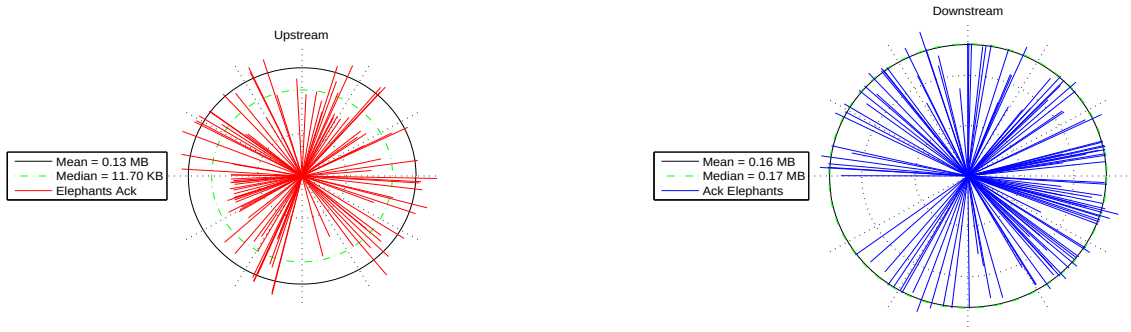
Le trafic UDP

Afin de compléter l'étude du comportement du client observé en terme de trafic, nous avons procédé à une étude du trafic UDP. L'étude du trafic UDP part du fait qu'un client eDonkey utilise conjointement les protocoles TCP et UDP. Cependant, le rôle du protocole UDP se restreint à la signalisation (voir spécification). En effet, dans le réseau eDonkey, les transferts de données se font exclusivement en TCP.



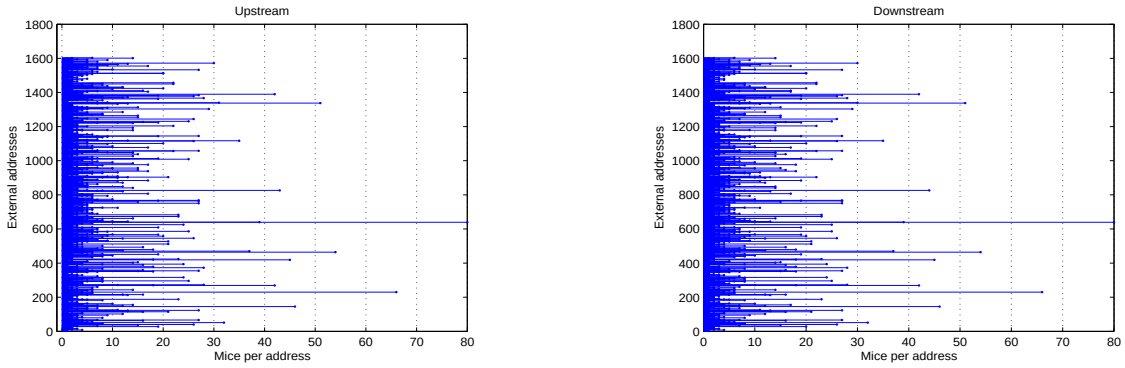
(a) sens montant (b) sens descendant

FIG. 1.14 – Éléphants réguliers (volume en échelle logarithmique)



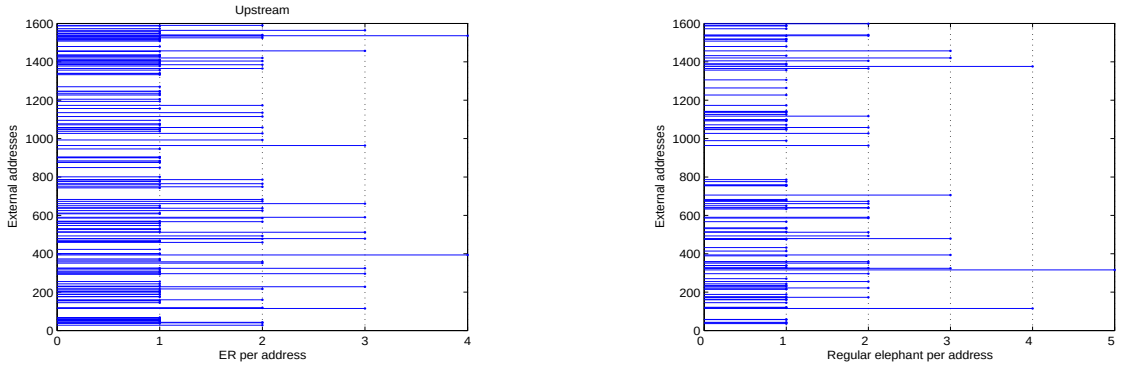
(a) sens montant (b) sens descendant

FIG. 1.15 – Éléphants ACK (volume en échelle logarithmique)



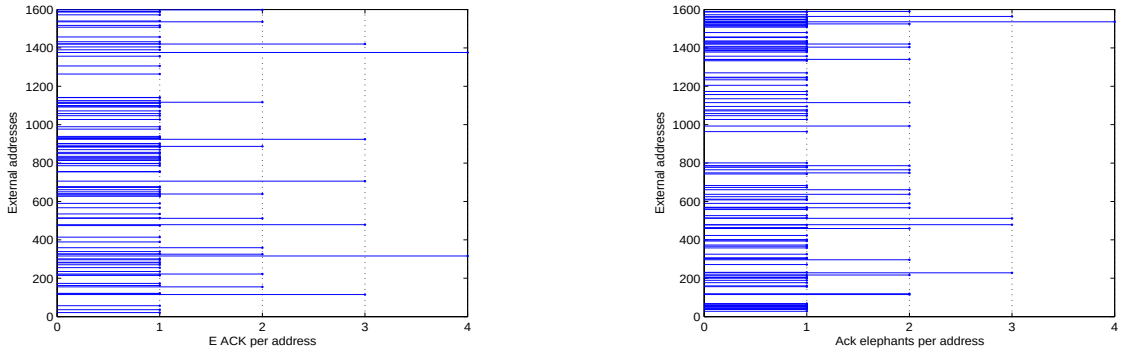
(a) sens montant (b) sens descendant

FIG. 1.16 – Nombre de souris (reçues ou envoyées) par adresse externe



(a) sens montant (b) sens descendant

FIG. 1.17 – Nombre d'éléphants réguliers (reçus ou envoyés) par adresse externe



(a) sens montant (b) sens descendant

FIG. 1.18 – Nombre d'éléphants réguliers (reçus ou envoyés) par adresse externe

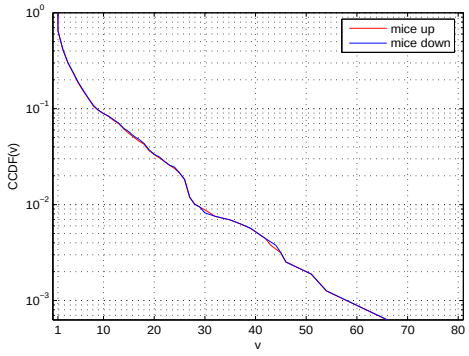


FIG. 1.19 – CCDF du nombre de souris par adresse externe

Pour étudier le trafic UDP, nous avons gardé la même granularité adoptée dans l'étude du trafic TCP. En effet, nous nous plaçons au niveau flot dans le but d'appréhender le comportement du client observé. Dans le tableau 1.8, nous donnons la répartition des flots UDP en souris et en éléphants. Notons ici qu'un autre critère a été pris en compte. En effet, nous avons tenu compte de l'existence d'une connexion TCP parallèlement à ce flot UDP. Nous appelons alors flots TCP tous les flots dont l'adresse externe a échangé au moins une fois un flot TCP avec le client observé. Rappelons ici que les adresses externes sont les adresses de destination (resp. de source) dans le sens montant (resp. descendant).

En termes de nombre de flots, les souris sont majoritaires. Cette constatation est en cohérence avec toutes les observations antérieures faites sur la dichotomie souris/éléphant. Cependant, en termes de volume, nous observons la limitation de cette approche lorsqu'il s'agit d'un trafic exclusivement réservé à la signalisation. En effet, le nombre d'éléphants UDP non TCP est négligeable et leur contribution en termes de volume est très réduite. Ainsi dans le cas, des éléphants UDP non TCP, il est plus judicieux de parler de grande souris plutôt que d'éléphant. En effet, dans la figure 1.3.3, nous observons que le nombre de paquets dans les éléphants UDP est réduit (plus de 98% des éléphants non TCP sont formés de moins de 60 paquets). Les flots UDP de type non TCP assurent une mission de découverte de réseau avec des messages de battement de cœur. Par ailleurs, les flots UDP de type TCP accompagnent les connexions TCP afin de garantir quelques services supplémentaires (gestion de la file d'attente). La figure 1.21(a) montrent que les flots de type TCP durent en général plus longtemps que les flots UDP de type non TCP et garantissent un rôle de signalisation comme peuvent l'indiquer les volumes relativement réduits de ces entités. En effet, la figure 1.21(b) confirme cette constatation pour les deux types d'éléphants UDP.

		Souris TCP	Souris non TCP	Éléphant TCP	Éléphant non TCP
Up	Nb flots	973	31,523	240	49
	Volume	454,130	7,322,335	528,432	221,378
	Nb paquets	7,013	67,437	7,406	1,864
Down	Nb flots	910	31,134	223	47
	Volume	283,102	5,498,057	308,122	125,630
	Nb paquets	6,588	67,939	6,823	1,543

TAB. 1.8 – Le trafic UDP : dichotomie élephant souris

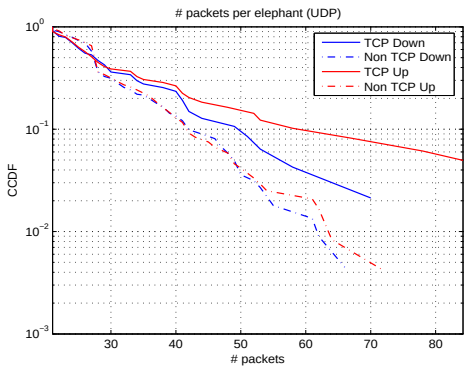
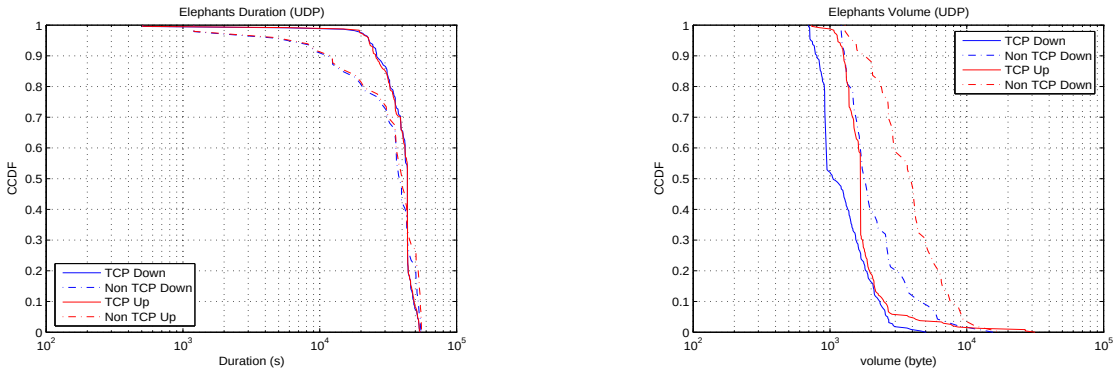


FIG. 1.20 – Nombre de paquets par éléphant



(a) Durée (b) Volume

FIG. 1.21 – Les durées des flots UDP

Pour conclure cette étude du trafic UDP, nous nous intéressons à l'analyse des adresses. Durant les 15 heures de capture, le client communique avec 31,900 adresses externes par l'intermédiaire du protocole UDP. Dans la majorité des cas, l'échange se fait dans les deux sens (du client vers l'adresse externe et vice versa). En général, les adresses externes reçoivent ou envoient un seul flot UDP (98% des adresses). Cependant, le client prend l'initiative dans un peu plus de 90% des cas. Dans le cas du protocole TCP, ce taux reste relativement élevé mais descend tout de même à un peu moins de 70% des échanges. Ainsi, nous pouvons appréhender à travers cette observation la gigantesque taille du réseau pair à pair découvert par un client pendant une durée relativement courte (en tenant compte des durées de téléchargement d'un fichier de grande taille dans un réseau comme eDonkey).

1.4 Heuristiques pour l'identification du trafic eDonkey

L'analyse du trafic TCP fondée sur la dichotomie souris/éléphants a mis en évidence plusieurs caractéristiques du trafic p2p. Grâce aux mesures, nous savons que maintenant que l'importance du trafic peer-to-peer dans le réseau le RBCI. Ceci découle du pourcentage conséquent que représente le trafic p2p dans les traces que nous avons analysé (ex. le trafic eDonkey représente 53% du volume total. Une seconde propriété est relative à la forte activité de signalisation du trafic eDonkey. Cette propriété est commune à toutes les applications (TCP). Cependant, deux caractéristiques relatives à l'application eDonkey. La première concerne la signalisation sur les ports standards (nous verrons comment cette caractéristique pourrait nous servir dans l'identification du trafic eDonkey). La deuxième caractéristique est relative à la connectivité de la communauté d'utilisateurs étudiée avec le reste du réseau eDonkey. Le nombre d'adresses externes contactées ou contactant notre communauté est extrêmement élevé. Finalement, une caractéristique concernant le transfert du fichier dévoile le mécanisme de segmentation des fichiers utilisé par eDonkey (trace du client eDonkey). Dans ce chapitre, nous nous intéressons à l'identification du trafic p2p et plus particulièrement au trafic eDonkey qui engendre actuellement la majeure partie du trafic dans le RBCI.

1.4.1 Principes théoriques et réalité de l'identification

Pour interagir avec d'autres pairs, un client p2p ouvre en principe un port d'écoute, une application possédant en général un tel port défini par défaut. Ce port d'écoute permet en retour de caractériser l'application utilisée par un internaute quand du trafic est observé sur celui-ci. L'identification du trafic engendré par une application revient ainsi à analyser les ports de source et/ou de destination dans l'entête des segments TCP ou des datagrammes UDP. Pour identifier le trafic de l'application eDonkey lors d'une capture sur un lien du réseau, il suffit, en théorie, d'analyser les ports de source et de destination des paquets collectés. En effet, il est connu à partir des spécifications que le protocole eDonkey utilise les ports 4662 et 4661 comme ports standard pour communiquer en TCP et 4665 et 4672 en UDP.

Dans la réalité, cette affirmation est loin d'être vérifiée non seulement dans le réseau eDonkey mais aussi dans les autres réseaux p2p. Ceci est dû au fait que les applications p2p tiennent leur réussite des types de contenus (films, musique,...) qu'elles permettent de partager. Or, comme les fichiers multimédia sont généralement protégés par les droits d'auteurs et de la propriété intellectuelle, les concepteurs des réseaux p2p ont introduit la possibilité pour les pairs de changer dynamiquement les numéros de ports pour réaliser des transactions p2p, afin de rendre leur identification plus difficile. Dans [80], en analysant les données remontées par un client, on estime à 45% la proportion de clients qui changent intentionnellement le numéro de leur port d'écoute sur le réseau eDonkey. Par exemple, le trafic p2p peut désormais passer par le port 80 réservé en principe au protocole HTTP afin de court-circuiter les mécanismes de pare-feu et les proxy. Cependant, cette possibilité semble peu utilisée dans la pratique ; les auteurs de [80] affirment que le pourcentage des clients utilisant le port 80 pour échanger du trafic eDonkey ne dépasse pas les 3%.

La possibilité de changer intentionnellement les ports pour les transactions p2p explique en partie la baisse du pourcentage du trafic p2p apparent, i.e. quand celui-ci n'est identifié que par l'analyse des numéros de ports [13, 39]. En contrepartie, le trafic jugé comme non identifié a vu son pourcentage augmenter, ce qui est cohérent avec une utilisation de ports non standard par les utilisateurs des réseaux p2p. Ceci doit être néanmoins nuancé par le fait que certaines observations, et comme on le verra par la suite pour le trafic dans le RBCI, montrent que eDonkey et Bittorrent utilisent plus souvent les ports standard que les autres applications p2p (cf. par exemple [77]).

Pour une étude du trafic p2p qui se veut rigoureuse et précise, l'identification de ce trafic déguisé devient une étape préliminaire indispensable. Notons que plusieurs études sur le trafic p2p se sont limitées au trafic p2p passant par les ports standard [73, 83]. Les méthodes d'identification existantes rejoignent les efforts de quelques industriels dans la création d'outils de filtrage du trafic p2p [2, 3, 52, 61]. Dans les sections suivantes, nous présentons quelques méthodes pour identifier ce dernier.

1.4.2 État de l'art des méthodes d'identification du trafic pair à pair

Analyse des contenus

La première approche, qui est aussi la plus naturelle pour caractériser une application, consiste à identifier un champ spécifique dans les paquets qu'elle engendre. Ce champ sera alors une empreinte applicative du trafic. Les méthodes basées sur l'identification d'une signature sont généralement déployées dans les systèmes de détection d'intrusion [1]. Afin d'identifier un champ particulier, la consultation des spécifications est indispensable. Or, cette tâche se

révèle rapidement difficile [77] car une documentation complète et riche fait défaut dans le monde des réseaux p2p. Les protocoles employés ne sont pas normalisés et sont en constante évolution, sauf peut-être pour Kazaa qui est un protocole relativement stable mais dont la spécification n'est pas publiquement disponible. Nous donnons dans ce qui suit deux exemples pour l'approche par analyse de signatures applicatives. Le premier exemple concerne le réseau eDonkey, le deuxième traite le réseau BitTorrent.

Le protocole eDonkey. Chaque message de ce protocole possède un entête de 6 octets. En plus des champs indiquant le type de message (1 octet) et la taille du message (4 octets), on trouve un champ fixe d'un octet contenant la version du client (par exemple 0xE3 pour le client eMule pour les communications en TCP). L'identification d'un paquet p2p revient alors à analyser l'octet qui suit les 40 octets (sans compter les options) des entêtes IP et TCP. L'analyse des 5 derniers octets de l'entête permettrait de connaître de manière plus précise le protocole en question.

Le réseau Bittorrent. L'identification d'un message Bittorrent est une tâche plus compliquée en termes de coût calculatoire. Les messages échangés entre deux clients Bittorent sont caractérisés par un préfixe constant de 20 octets (un octet dont la valeur est 0x13 et les 19 octets qui code la chaîne de caractère "BitTorrent protocol"). Il est également possible de déceler la clé ".torrent" dans les premiers paquets de la connexion.

Théoriquement, l'analyse des signatures paraît être une approche relativement simple et efficace. Dans la pratique et pour des implémentations en ligne, l'opération devient cependant de plus en plus difficile dès que la taille de la signature dépasse un certain seuil. Ces méthodes sont utilisées par exemple par ALLOT [3] pour identifier le trafic p2p dans le but de le filtrer. Mais en général, cette approche est confrontée à deux problèmes.

Le premier problème est lié à la nature évolutive des protocoles étudiés. La signature peut changer d'une version du "client" à une autre et ainsi rendre obsolète la routine d'identification. Pour résoudre ce problème, comme une bonne partie de ces protocoles ont un code source disponible, il est possible de réviser et mettre à jour continuellement ces routines. Il existe également une approche automatisée que les auteurs de [25] ont proposé afin de pallier à ce problème. Dans [25], la méthode proposée consiste à transformer la signature applicative (donc les n premiers octets de la charge utile du paquet) et à passer le vecteur associé à des méthodes d'apprentissage afin de déceler une certaine régularité par application. En conséquence, l'apprentissage manuel devient inutile.

Le deuxième problème est un peu plus difficile à résoudre. L'utilisation des protocoles p2p dont les messages sont cryptés complique la tache d'identification basée sur la reconnaissance d'un champ particulier dans les messages échangés entre pairs. Parmi les protocoles p2p qui sont passés au mode crypté, citons l'exemple de Winny [86] (très répandu au Japon) ou encore de Filetopia[27]. L'émergence attendue de l'IPv6 compliquera certainement l'identification des applications utilisées avec toutes les fonctions d'authentification et de sécurité que propose ce protocole.

Il existe également d'autres méthodes qui se basent sur une analyse de contenu. Celles-ci sont destinées à filtrer le trafic p2p et non à l'étudier. On pourra citer les outils proposés par Advestigo [2] ou par Audible Magic [52] qui calculent une empreinte du contenu appelée théraogramme. Cette signature peut être retrouvée même après une manipulation du fichier (compression, découpage,...). Vu la complexité des outils qu'elle utilise, cette méthode est inappropriée pour une implémentation en ligne dans un réseau de cœur.

Analyse protocolaire

Les méthodes d'identification du trafic p2p décrites ci-dessus se montrent relativement inefficaces vis à vis de l'évolution des protocoles et des débits dans les réseaux. D'autres méthodes fondées sur une analyse protocolaire semblent plus prometteuses pour une identification plus précise du trafic p2p. Dans [43], les auteurs proposent deux heuristiques pour l'identification du trafic p2p. Un client p2p est identifié à partir du couple (adresse IP, port d'écoute).

La première méthode considère un couple (adresse IP, numéro de port) comme couple p2p dans le cas où il a été contacté par N adresses sources distinctes en utilisant N ports source distincts. Cette approche part de l'hypothèse suivant laquelle il n'existe pas plusieurs connexions entre deux pairs, contrairement au protocole HTTP qui permet d'initier plusieurs connexions entre le serveur et le client afin d'accélérer le rapatriement de documents. Cette méthode risque de donner de fausses alertes car plusieurs protocoles peuvent ouvrir plusieurs connexions à partir d'un même terminal, comme par exemple les protocoles de messagerie (SMTP, POP) et les application de jeux en réseau.

Une deuxième méthode repose sur le fait que les adresses qui communiquent simultanément en TCP et en UDP ont une forte probabilité de faire partie d'un réseau p2p. Cette propriété n'est cependant pas réservée aux protocoles p2p. C'est par exemple le cas de la résolution des noms de domaines par DNS, ce protocole fonctionnant sur TCP et UDP. Une différence, malgré tout avec les protocoles p2p, est que tout échange de données est effectué en utilisant le port standard 53. Par ailleurs, les deux types de trafic (trafic DNS et trafic p2p) ont d'autres caractéristiques différentes comme par exemple la taille des paquets transmis. Une analyse basée sur la distribution de la taille des paquets pourrait être utilisée en complément pour distinguer les deux types de trafic, comme cela est fait dans les travaux relatifs au trafic des applications temps réel [60].

Une troisième approche pour identifier le trafic p2p peut exploiter l'existence d'un comportement spécifique, induit par la nature des protocoles et par la structure des réseaux p2p. Prenons l'exemple d'un client appartenant à un réseau p2p pur. Ce client jouera, s'il respecte la règle du jeu, à la fois le rôle de serveur et le rôle de client. Ce double rôle confère aux échanges p2p pur une spécificité par rapport aux communications que l'on peut observer dans un réseau régi par le modèle client/serveur classique. Dans [59], les auteurs essaient de partir de cette constatation pour identifier les pairs du réseau Winny et donc identifier le trafic p2p qu'ils engendrent.

Même si cette approche peut éventuellement être appliquée aux réseaux p2p purs, son efficacité peut être critiquée. La détection des pairs dans [59] est faite en intégrant un client leurre dans le réseau p2p et en analysant ensuite son historique. De plus, l'identification du trafic est réalisée sur un lien de sortie d'un réseau de taille réduite, formé d'un petit nombre de machines. Cette méthode n'a pas une visibilité totale du réseau p2p. Dans [31], l'auteur se place également au niveau d'un réseau local et propose ainsi une méthode pour l'identification des utilisateurs p2p dans celui-ci. Pour ce faire, l'auteur propose d'utiliser les connexions UDP. En effet, l'analyse des fichiers logs issus des proxies montre que les applications p2p ont un comportement spécifique. Ce comportement est associé au protocole UDP. Dans le cas particulier de l'eDonkey, des messages de battement sont régulièrement envoyés par les pairs vers des nouveaux serveurs d'indexation. Ces messages servent à rafraîchir les listes de serveurs d'indexation. Les pairs intègrent dans leurs listes seulement les serveurs actifs. En effet, en guise de réponse au message UDP, un serveur d'indexation actif répond par un message UDP envoyé sur le port d'écoute UDP du client. En conséquence, un couple (adresse IP, port

UDP) donné est jugé comme un utilisateur du p2p s'il est contacté par plusieurs adresses IP avec des ports différents. Cette approche d'identification se base alors sur un comportement spécifique des protocoles peer to peer. Ce comportement spécifique a été représenté sous une forme de signatures graphiques appelées *graphlets* dans [45]. Une *graphlet* est une description graphique des échanges protocolaires d'une application donnée. L'identification consiste alors à trouver la *graphlet* qui correspond le mieux à une transaction vue dans une trace réelle. Cette technique d'identification permet de classer les transactions dans des classes d'application. En revanche, cette classification n'atteint pas un niveau élevé de précision. En effet, elle est incapable de classifier les applications peer to peer dans des classes distinctes. Afin de surpasser cette limitation, il est nécessaire d'enrichir les *graphlet* pour qu'elles réussissent à identifier séparément les applications peer to peer. L'enrichissement de ces *graphlets* nécessite une étude approfondie de la spécification des protocoles en question. L'étude de la spécification d'un protocole permet de déduire une signature transactionnelle qui lui est associée.

Parallèlement à cette signature transactionnelle, la détermination d'une signature statistique d'une application donnée est un autre moyen pour identifier son trafic que l'on peut trouver dans la littérature. En effet, des études plus récentes (voir [72], [56], [25] et [21]) ont utilisé les techniques de la classification et de la reconnaissance des formes pour résoudre ce problème d'identification. Pour ce faire, une étape préliminaire et indispensable consiste à choisir les paramètres discriminants. Les paramètres utilisés dans la littérature sont généralement les mêmes. Parmi ces paramètres, on peut citer la taille moyenne des paquets, la distribution des temps d'inter-paquets ou la durée du flot, etc. Certains de ces paramètres dépendent énormément des conditions du réseau dans lequel les paquets sont acheminés. En plus, ces méthodes souffrent du même problème que les méthodes de classification basées sur les signatures transactionnelles. En effet, cette méthode permet de faire une classification relativement grossière du trafic (application de transfert de données, applications transactionnelles,...etc). Cependant, ce niveau de précision reste suffisant pour décider de la politique de qualité de service dans une entreprise mais se révèle aussitôt très pénalisant pour une étude visant à comprendre et à modéliser une application donnée. Outre le fait que ces méthodes sont incapables de servir des études sur une application précise, il est important de noter leur incapacité à être mise en œuvre dans un contexte temps réel. En effet, lors de l'apprentissage, les méthodes citées plus haut dégagent les caractéristiques du trafic à partir des flots complets. Lors de la phase de classification proprement dite, il faut attendre la fin du flot pour avoir une décision sur l'application qui le génère. Néanmoins, un bon choix du paramètre discriminant permettrait éventuellement d'aboutir à une détection rapide du type de l'application observée. Par exemple,dans [8, 9, 7], les auteurs utilisent la taille des premiers paquets de données d'une connexion TCP afin d'identifier l'application qui lui est associée.

Les auteurs de [8, 9, 7] prennent en considération les premiers paquets de la connexion en négligeant les paquets de la mise en place de la connexion TCP (SYN, ACK). Ces premiers paquets de données représentent un paramètre ayant un pouvoir discriminant très intéressant. Ce paramètre discriminant permet de fournir des résultats satisfaisants. En effet, une application donnée génère une série de paquets qui lui est spécifique. Ce comportement est identifié grâce à une étude préliminaire sur des traces de trafic réel issues des réseaux universitaires. Grâce à l'outil commercial Qosmos, les auteurs identifient l'ensemble des connexions TCP qui ont aboutit. L'outil permet aussi d'identifier l'application qui correspond à cette connexion TCP grâce à une analyse de la charge utile des paquets de données. Au final, on dispose d'une batterie de N connexions TCP où chaque connexion correspond à une application bien identifiée. Notons ici que le terme connexion TCP est utilisé au lieu de flot. En effet, le terme

connexion TCP sous-entend la prise en compte des deux directions du trafic. En conséquence, chaque connexion TCP est caractérisée par la taille des p premiers paquets de données vus sur les deux directions. Pour distinguer les deux directions de l'observation, un signe négatif est attribué au sens "serveur vers client". Le client désigne l'adresse IP qui a initié la connexion TCP, en d'autres termes c'est l'adresse IP qui a envoyé le premier paquet SYN. Selon cette convention, la *ième* connexion TCP est caractérisée par le vecteur ε_i défini dans Z^p qui s'écrit sous la forme :

$$\varepsilon_i = (x_i^1, x_i^2, \ldots, x_i^{p-1}, x_i^p)$$

où $|x_i^j|$ représente la taille du $j ème$ paquet de la connexion i. Le signe x_i^j est déterminé par le sens du $j ème$ paquet (le signe est négatif si le paquet est acheminé du serveur vers le client et le signe est positif dans le cas contraire). En conséquence, chaque connexion est représentée dans un espace à p dimensions. Grâce à une analyse de la couche applicative, chaque connexion est labellisée avec une application donnée. Il s'agit alors d'une classification supervisée. En effet, la matrice N est une séquence d'apprentissage qui sert à mettre en place une règle de classification des connexions TCP non labellisées. Cette méthode de classification résiste à l'encryptage des données étant donné qu'elle traite des paramètres qui ne peuvent pas être encryptés. Ces données sont la taille du paquet et son sens d'acheminement. Les études faites à l'université John Hopkins [87] rejoignent cette piste en utilisant les mêmes paramètres mais en utilisant une autre caractérisation des connexions TCP. Dans [87], le vecteur ε_i d'une connexion TCP comptabilise le nombre des petits paquets et le nombre des gros paquets selon un seuil de taille donné. Le jeu de vecteurs d'observation détermine alors les paramètres d'une chaîne de Markov cachée associée à une application donnée. La génération de cette chaîne de Markov se base sur l'algorithme de Baum-Welch. L'opération de classification consiste à trouver dans la bibliothèque des modèles markoviens construite lors de l'apprentissage, le modèle de chaînes de Markov cachées (HMM) qui semble être le plus proche de l'observation courante (le modèle le plus vraisemblable sachant l'observation). Les auteurs de [87] utilisent l'algorithme de Viterbi. Les paramètres adoptés par [8, 9, 7] pour représenter une connexion TCP semblent être une représentation discriminante. Afin de vérifier cette assertion, nous l'avons testé sur des traces issues du réseau de France Telecom (voir section 1.4.5).

Pour conclure cette partie d'identification, il est important de noter que parmi toutes les méthodes présentées ci-dessus, aucune n'est idéale. En effet, chaque méthode a ses inconvénients et ses avantages et aucune d'elle ne permet une identification avec un risque d'erreur nul. Ces méthodes sont plutôt complémentaires et elles sont souvent implémentées en parallèle. Cependant, la méthode basée sur les signatures applicatives reste théoriquement une méthode fiable. Elle est généralement utilisée pour estimer les taux de non détection des autres méthodes. Elle se trouve néanmoins confrontée à des problèmes d'ordre calculatoire dès qu'il s'agit d'analyser des champs de très grande taille. Il existe également un problème relié à l'encryptage qui commence à devenir un sujet d'actualité dans la communauté des utilisateurs du p2p.

1.4.3 Le trafic non identifié et potentiellement eDonkey sur le lien GE

Bien qu'elle représente un peu moins du tiers du volume, la contribution du trafic eDonkey reste en dessous des estimations annoncées par la littérature (plus de 50%). Ceci est dû au fait que les utilisateurs du réseau eDonkey ont tendance à camoufler leur trafic p2p. Afin de comptabiliser ce trafic p2p caché nous nous proposons de développer deux approches qui se

basent sur une analyse des adresses et non pas sur une analyse des signatures du protocole eDonkey.

La première approche consiste à créer un annuaire contenant les couples qui ont communiqué au moins une fois avec un numéro de port source ou destination standard de l'application eDonkey. Si les adresses de l'un de ces couples échangent un éléphant avec des numéros de port non standard, l'échange sera considéré malgré tout comme du trafic p2p. Ainsi, le trafic d'un couple d'adresses qui négocient en clair et utilisent ensuite des numéros de port non standard pour le transport des données sera comptabilisé comme un trafic p2p.

Les piles protocolaires "clients" de eDonkey ne permettent cependant pas ce genre de comportement. L'utilisateur spécifie dans les paramètres de son "client" p2p un seul numéro de port qu'il utilisera pour recevoir des connexions initiées par d'autres clients p2p. Ce port servira en même temps à la signalisation et au transfert des données. Cela ne veut pas dire pour autant que la séparation entre ces deux tâches est impossible. Elle existe déjà dans le monde des réseaux avec les clients FTP qui maintiennent une connexion permanente sur le port 21. Cette connexion véhicule toutes les commandes et les réponses entre le client et le serveur. Une deuxième connexion sur le port 20 sert ensuite au transfert de données. Une fois celui-ci achevé, cette connexion est rompue. Sur les 191,862 flots éléphants non p2p (environ 80% des éléphants TCP), seulement 3,156 flots (environ 1.6% des flots non p2p) émanent d'un couple d'adresses qui ont déjà communiqué en p2p clair. Ces flots représentent seulement 1.7% du volume total (1.1% du nombre de paquets TCP total). Ce faible pourcentage confirme que les principes protocolaires de eDonkey sont globalement respectés.

La deuxième approche consiste à raisonner par rapport aux adresses et non par rapport aux couples. Pour cela deux annuaires sont créés. Le premier annuaire contient l'ensemble des adresses de source ayant déjà communiqué en p2p clair. Le deuxième contient les adresses de destination obéissant au même critère. Le premier annuaire est appelé annuaire des sources, le deuxième est appelé annuaire des destinations.

Cette approche part du principe qu'un client utilisant un port non standard est malgré tout contraint à communiquer avec des clients utilisant un port standard pour bénéficier pleinement du réseau eDonkey. En effet, les clients eDonkey utilisant des ports standard ne sont pas si rares et forment une partie non négligeable du réseau p2p. Un client qui restreindrait ces communications à des clients utilisant uniquement des ports d'écoute non standard se priverait d'une grande partie du réseau p2p.

L'approche consiste alors à considérer tout flot éléphant dont l'adresse de destination ou de source figure dans l'un des deux annuaires comme flot eDonkey. Un flot éléphant dont l'adresse de source (resp. de destination) figure dans l'annuaire des sources (resp. destination) est dit "flot éléphant estimé eDonkey" par rapport à la source (resp.destination). Les flots éléphants concernés par cette analyse sont les flots qui a priori *ne sont pas* des flots eDonkey. Ils représentent un peu moins de 70% du volume total cumulé sur les 30 mn de capture.

Le nombre d'éléphants estimés eDonkey en utilisant l'adresse de source est égal à 15,767 soit 8% du nombre de flots a priori non eDonkey (10% du volume TCP total). Le même pourcentage mais cette fois-ci sur la base de l'adresse de destination est égal à 35.28% des éléphants non eDonkey, ce qui représente 28% du volume TCP total. Le premier lot de flots éléphant est pratiquement inclus dans le deuxième lot.

Afin d'évaluer la performance de cette deuxième approche, une identification basée sur les numéros de port est appliquée sur les éléphants estimés eDonkey par rapport à la destination, ceci afin d'évaluer le volume de faux positifs. Le volume des éléphants Bittorent, Gnutella ou Napster identifiés comme étant des éléphants eDonkey donne en effet un ordre de grandeur

du taux de fausse alerte. Pour trouver un ordre de grandeur du taux de la non détection, on évalue la capacité de la méthode à détecter le trafic qui passe par les ports 5662, 14662 et 40662. En termes de volume, cette méthode d'identification permet de détecter 98.95% du trafic qui passe par ces ports voisins lorsqu'on utilise l'annuaire des adresses de destination. Par ailleurs, une identification basée sur l'annuaire des adresses de source, permet de détecter plus de 60% de ce trafic. Mais malgré cela, la méthode souffre d'un taux de fausses alertes assez élevé. Comme indiqué dans le tableau 1.9, près du tiers du trafic Bittorrent et Gnutella est confondu avec du trafic eDonkey.

	Application	% eDonkey Source.	% eDonkey Dest.
p2p	Bittorrent	5.12%	30.77%
	Gnutella	11.26%	33.18%
	Kazaa	1.92%	6.54%
	Napster	9.00%	12.20%
non p2p	FTP	7.22%	45.90%
	HTTP	0.90%	20.95%
	P5662	62.58%	98.95%
	Inconnu	22.56%	45.54%
TCP Total		10.00%	28.00%

TAB. 1.9 – Pourcentage en termes de volume du trafic estimé eDonkey par rapport à la source et à la destination par application.

Cela montre que les utilisateurs adhèrent à plusieurs réseaux p2p afin de profiter des avantages des différents réseaux d'échange de fichiers. Cette constatation, bien qu'elle détermine l'un des aspects du profil typique des utilisateurs du p2p, montre les limitations de cette méthode de détection. L'utilisation de l'annuaire des adresses de destination entraîne une surestimation du volume du trafic eDonkey. En revanche, l'utilisation de l'annuaire des adresses de source permet de donner un taux de fausse alerte raisonnable. En contrepartie, l'utilisation du deuxième annuaire est plus coûteuse en termes de temps d'exécution, ce dernier étant bien plus grand que que dans le cas de l'annuaire des adresses de destination.

L'annuaire des adresses de source donne ainsi une image assez fidèle de la communauté eDonkey contactée par les clients de la plaque ADSL. Cependant, cette image est plus ou moins biaisée quand on s'intéresse aux réels échanges de volume. En fait, lors de la construction des annuaires, on n'a pas pris en compte la nature du trafic eDonkey clair. En conséquence, l'annuaire contient essentiellement les adresses qui ont assisté, durant la capture, à une phase de sollicitation de contenu. Cet annuaire ne fait alors aucune distinction entre les adresses qui participent dans les échanges actifs au moment de la capture avec celles qui figurent dans les tables d'indexation des serveurs eDonkey mais qui ne sont pas actives pendant la durée d'observation.

Afin de prendre en considération cette constatation, on procède à un tri supplémentaire sur les premiers annuaires pour garder les adresses actives dans le réseau eDonkey pendant la capture. Les adresses actives sont précisément les adresses qui sont impliquées dans des phases de transfert de fichiers. En d'autres termes, la nouvelle version de l'annuaire des adresses contient exclusivement les adresses de source ou de destination des flots éléphants réguliers dont les ports sont standard. Cela ne néglige pas pour autant le rôle des phases de recherche de contenu dans la caractérisation des échanges de fichiers dans le réseau eDonkey. Ce rôle se manifeste dans des cas particuliers. Par exemple, un régime de téléchargement établi, qui

se passe sur les ports non standard et dont les acteurs ne sont plus bavards (ils l'étaient certainement avant le début de la capture) passe inaperçu avec la méthode des annuaires. En effet, ce cas de figure est la conséquence directe de la durée de la capture ; les effets de bord peuvent réduire la richesse des annuaires. Afin d'éviter ce type de problème, nous traitons dans la suite une capture de trafic de plus longue durée.

1.4.4 Identification du trafic eDonkey potentiel sur le lien OC3

Méthode des annuaires

Nous nous intéressons à présent à l'identification du trafic eDonkey potentiel. La durée de cette capture est plus longue que la capture issue du lien GE. Une durée plus longue limitera sans doute les effets de bords et servira à enrichir les informations contenues dans les annuaires.

Dans un premier temps, nous appliquons la méthode en utilisant les annuaires de base. Comme expliqué dans la section réservée au lien GE, ces annuaires contiennent les adresses qui ont envoyé ou reçu au moins un paquet eDonkey clair. Dans un deuxième temps, nous utilisons une version triée des annuaires. Ces annuaires contiennent alors les adresses de destination ou de source d'éléphants réguliers de type eDonkey clair. Rappelons que ces adresses du deuxième annuaire, sont supposées être des adresses actives dans le réseau eDonkey. En d'autres termes, ces adresses contribuent plus au moins activement (en termes de volume) au trafic du réseau eDonkey.

Les résultats de l'utilisation de la première version des annuaires sont présentés dans le tableau 1.10. Notons ici que l'on entend par "p2p" les applications p2p observées sur le lien pendant la durée de capture à l'exception d'eDonkey. Nous remarquons que les taux observés par rapport à la source restent relativement cohérents avec ceux obtenus sur le trafic du lien GigabitEthernet de la section précédente. Cependant, les taux obtenus en utilisant les annuaires des adresses internes confirment les limitations de cette méthode et son incapacité à identifier avec précision le trafic eDonkey caché. En effet, le taux de fausse alerte est très élevé (plus des deux tiers du trafic p2p est pris pour du trafic eDonkey).

	Application	eDonkey Source	eDonkey Destination
Sens Descendant	p2p	6%	67%
	http	5%	58%
	inconnue	20.50%	81%
Sens Montant	p2p	64.10%	6%
	http	27.80%	5%
	inconnue	75.70%	24%

TAB. 1.10 – Pourcentage du trafic estimé être de l'eDonkey pour les principales composantes du trafic.

Afin de réduire ce taux de faux positifs, nous appliquons alors la méthode des annuaires dans sa version élaguée. Les résultats de cette deuxième analyse sont présentés dans le tableau 1.11. Malgré une diminution très importante par rapport aux valeurs de la première version des annuaires, le taux de faux positifs reste relativement élevé. Cependant, l'annuaire des adresses externes donne un taux de faux positifs très satisfaisant. En contrepartie, nous remarquons d'emblée une diminution brutale des taux associés aux applications inconnues.

Le pourcentage du volume http que l'on peut considérer comme eDonkey est a priori situé entre les deux valeurs obtenues par l'application des annuaires source et destination.

	Application	eDonkey Source	eDonkey Destination
Sens Descendant	p2p	<0.1%	20%
	http	1.34%	21%
	inconnue	1.74%	53%
Sens Montant	p2p	23%	<0.1%
	http	15.20%	1.1%
	inconnue	59.40%	1.0%

TAB. 1.11 – Pourcentage du volume considéré comme de l'eDonkey pour les principales composantes du trafic en considérant les annuaires "élagués".

Comme nous pouvons le remarquer, les annuaires que nous avons utilisés jusqu'à présent sont relatifs à un sens particulier du trafic. En effet, nous ne prenons pas en compte, lors de la construction des annuaires, le fait que l'on dispose des deux sens du lien. L'association des deux directions permet de donner une nouvelle définition des adresses eDonkey actives. On appellera "adresse eDonkey active" toutes les adresses qui ont reçu et envoyé au moins un paquet eDonkey clair. Cette définition permet de garder dans l'annuaire, toutes les adresses eDonkey qui répondent à des requêtes provenant du réseau p2p associé. Nous formons ainsi deux annuaires : un annuaire des adresses internes et un deuxième annuaire pour les adresses externes. Le premier contient toutes les adresses internes, déclarées comme "adresse eDonkey active". Le deuxième contient l'ensemble des adresses externes déclarées comme "adresse eDonkey active". Un éléphant a priori non eDonkey est déclaré comme "éléphant eDonkey" si son adresse de source et l'adresse de destination appartiennent respectivement aux deux annuaires. Ainsi, un éléphant a priori non eDonkey, vu dans le sens descendant, est déclaré comme "éléphant eDonkey" si son adresse de source et son adresse de destination appartiennent respectivement à l'annuaire des adresses externes et à l'annuaire des adresses internes. Ces deux annuaires représentent a priori les deux communautés (interne et externe) actives du réseau eDonkey que nous avons recensées au niveau de ce point d'observation. Les éléphants déclarés "eDonkey" par cette nouvelle version d'annuaire sont alors tous les éléphants que s'échangent les deux communautés eDonkey (interne et externe).

Nous appliquons cette approche sur les deux sens. Le tableau 1.12 donne les pourcentages en volume du trafic associés aux trois principales composantes du trafic (web, p2p et inconnu) et déclarés "eDonkey" par notre méthode. Nous remarquons que le taux de fausse alerte est réduit à moins de 5%. Cependant, le pourcentage de trafic a priori inconnu et déclaré "eDonkey" par la méthode est resté raisonnablement élevé contrairement à ce que nous avons remarqué par rapport à l'application des autres versions de l'annuaire. En effet, avec les premières versions de l'annuaire, la diminution du taux de fausse alerte s'accompagne d'une diminution brutale du pourcentage de trafic inconnu mais déclaré eDonkey.

Dans la section suivante, nous procédons à une analyse du trafic http dans le cadre d'une étude comparative avec le trafic eDonkey pour vérifier les résultats obtenus par rapport au trafic http.

Une comparaison avec le trafic web

Dans ce qui suit, nous serons amenés à faire des comparaisons entre eDonkey et le web étant donné que ce sont les deux applications dominantes dans le trafic étudié. L'analyse des différences entre ces deux applications mettra en évidence leurs spécificités. Par ailleurs,

	Application	eDonkey
Down	p2p	4.21%
	http	2.7%
	inconnu	17.22%
Up	p2p	4.6%
	http	3.2%
	inconnu	21.63%

TAB. 1.12 – Pourcentage du trafic déclaré "eDonkey" pour chaque application.

l'analyse des ressemblances permettrait probablement de déceler les utilisateurs qui utilisent le web à des fins de p2p (utilisation du port 80 pour l'application eDonkey).

Analyse des adresses L'analyse du trafic web permet de dénombrer les adresses qui ont communiqué avec le port standard du protocole http. Plus de 92% des adresses internes ont envoyé ou reçu des paquets http. Le web met en jeu plus d'adresses que eDonkey. En effet, les adresses internes qui communiquent en eDonkey ne représentent dans le meilleur des cas que 49,9% de l'ensemble d'adresses TCP (uniquement 27,7% dans le sens montant).

Ce constat est inversé lorsqu'on considère les adresses externes. Rappelons que les adresses eDonkey représentent environ 49% de l'ensemble des adresses TCP (63.7% pour le sens descendant et 34.2% pour le sens montant). Ces pourcentages relativement importants chutent pour atteindre 3.4% pour le sens descendant et 5.5% pour le sens montant lorsqu'on dénombre les adresses web. Ce contraste est également observé lorsqu'on analyse les couples d'adresses (58% contre 8.5% pour le sens descendant et 38.04% contre 8% pour le sens montant).

Ces observations peuvent s'expliquer par les natures différentes des deux applications. Le web est basée sur une architecture client/serveur par opposition à une architecture plus ou moins décentralisée pour l'application eDonkey. Pour cette dernière, le rapatriement d'un fichier se fera à partir de plusieurs utilisateurs extérieurs. Pour le web, un client verra sa requête satisfaite en communiquant avec un nombre limité de serveurs. Le nombre de serveurs de contenus extérieurs dans le p2p est plus grand que le nombre de serveurs web. Les nombres de clients (adresses internes) des deux applications sont comparables avec une petite avance pour le web. Le web a eu plus de temps pour faire partie des mœurs des utilisateurs d'Internet.

Dans la figure 1.22 , on représente la répartition du volume envoyé ou reçu par les adresses internes en fonction du pourcentage de cette population. Les deux applications étudiées et comparées sont eDonkey et le web (http). On peut remarquer que la loi de Pareto est toujours vérifiée pour les deux applications et ceci dans les deux sens. 10% de la population engendrent plus de 60% du volume.

La comparaison entre le sens montant et le sens descendant montre que le volume reçu par les adresses internes est moins réparti sur les clients que le volume envoyé par ces derniers. Cette disparité entre les deux sens de l'étude est plus grande pour l'application eDonkey que pour le web.

Une minorité des clients qui ont communiqué une fois avec du http (qui font probablement du http et en même temps de l'échange de fichiers) contribuent majoritairement dans le volume total . Le point à partir duquel la tendance est inversée indique que les nouveaux clients cumulé dans le pourcentage sont des clients qui font uniquement du web.

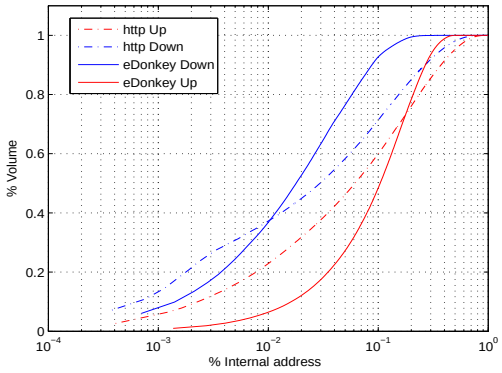


FIG. 1.22 – Comparaison entre les deux sens de capture : adresses internes

La suprématie du web sur la première partie des courbes représentées dans la figure 1.22 s'explique par le fait que l'échange de fichiers engendre un très grand volume par rapport à du web pur. Un échange de fichiers, déguisé en transaction http, peut être observé dans la courbe de la répartition des volumes sur les adresses. Le trafic purement http de ces adresses est nettement moindre par rapport à leur trafic http qui sert pour le p2p. Cependant, cette affirmation doit être nuancée par le fait que le transfert d'un fichier volumineux peut se produire dans un contexte purement http. Par exemple, le téléchargement d'une page web riche en images peut entraîner les mêmes observations sur les proportions en volume de ces adresses.

Étant donné que le sens montant est le sens contraire de l'acheminement des contenus multimédias (sens client vers serveurs http), on peut s'attendre à ce que le trafic http soit assez faible dans le sens montant et relativement diffus sur tous les utilisateurs. Or, comme pour le trafic eDonkey up, on peut remarquer que le trafic http up est relativement concentré sur les adresses des clients ADSL, même si le phénomène est moins net que pour le trafic p2p. Cette observation laisse présager que certains utilisateurs échangent des fichiers via le web. Ceci sera confirmé dans la suite quand on étudiera les volumes de données sur les applications dans le sens montant.

Dichotomie Souris/Eléphants La réalisation de la dichotomie souris/éléphants sur la trace considérée ne révèle pas de surprises. Le nombre de flots éléphants est toujours négligeable devant celui des souris comme indiqué dans le tableau 1.13.

		Nombre de flots		Volume	
	Application	% Eléphants	% Souris	% Eléphants	% Souris
Sens Up.	eDonkey	2%	98%	95%	5%
	http	9%	91%	68%	32%
	Tcp Total	4%	96%	93%	7%
Sens Down.	eDonkey	2%	98%	89%	11%
	http	12%	88%	88%	12%
	Tcp Total	5%	95%	7%	93%

TAB. 1.13 – Répartition en nombre de flots et en volume par type d'application.

Nous pouvons aussi remarquer la tendance habituelle des répartitions en termes de volume. Nous remarquons tout de même une exception au niveau du protocole http dans le sens montant. La contribution des souris en termes de volume est relativement importante, comme c'est prévisible puisque le sens montant sert, dans le cas du web, à transporter les requêtes des clients ADSL vers les serveurs, les clients ADSL étant en principe rarement amenés à envoyer des éléphants http volumineux lors de transactions http, bien que des éléphants de taille réduite peuvent être envoyés par ces clients à titre d'acquittements pour des transferts de fichiers dans le sens descendant. Mais malgré tout, le volume des éléphants http dans le sens montant reste prépondérant.

Afin de vérifier ces assertions, nous regardons la contribution, en termes de volume et en termes de nombre de flots, des éléphants http dans l'ensemble des éléphants TCP vus dans les deux sens. L'ensemble des résultats sont présentés dans le tableau 1.14. Les éléphants http contribuent faiblement dans le volume des éléphants du sens montant (5%) bien qu'ils soient plus nombreux que les éléphants eDonkey. Ces éléphants sont des éléphants d'acquittement. Toutefois, des éléphants http qui transportent de grands volumes existent. Ceci laisse penser que dans la masse des échanges de fichiers, ceux réalisés via des transactions http sont marginaux.

		Nombre de flots		Volume	
	Application	% Up	% Down	% Up	% Down
Eléphants	eDonkey	24%	19%	53%	16%
	http	41%	55%	5%	33%
	Autres	35%	26%	42%	51%
Souris	eDonkey	40%	47%	42%	26%
	http	17%	20%	30%	57%
	Autres	43%	33%	28%	17%

TAB. 1.14 – Répartition en nombre de flots et en volume par type d'application.

Dans la figure 1.23, on représente la répartition du volume sur les flots éléphants. Nous observons le même phénomène que pour l'analyse des adresses internes. En plus de la confirmation de la loi de Pareto, l'entrelacement des deux courbes relatives aux deux applications eDonkey et web existe et se produit au niveau du même point pour les deux sens. La loi de Pareto montre qu'un petit nombre des éléphants http contribue à la majorité du volume des éléphants. Les fonctions de répartition complémentaires mettent en évidence les caractéristiques de ces gros éléphants (voir les figures (1.24(a) et 1.24(b)).

La différence entre les éléphants http et les éléphants p2p est remarquable lorsqu'on s'intéresse aux durées. Un éléphant p2p dure en moyenne 703 secondes dans le sens descendant (587.35 secondes dans le sens montant). Ces moyennes ne dépassent pas les 100 secondes pour un éléphant http (72,4 secondes pour le sens descendant et 100 secondes pour le sens montant). La médiane pour la durée des éléphants p2p dépasse les 240 secondes. Elle est égale à 10 pour éléphants http.

Conclusion sur l'étude comparative entre le Web et e-Donkey Pour conclure cette étude comparative, nous avons appliqué la méthode des annuaires (dans sa deuxième version) et avons pu déduire que seulement 10 adresses internes (resp. 19) dans le sens montant (resp. sens descendant) communiquent via http pour échanger du trafic eDonkey. Ces valeurs donnent un ordre de grandeur de la taille de cette communauté et nous permet d'avoir une idée sur

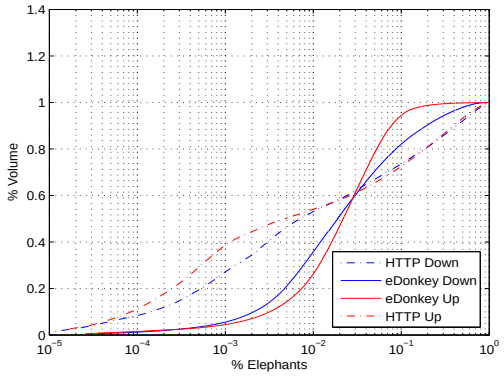
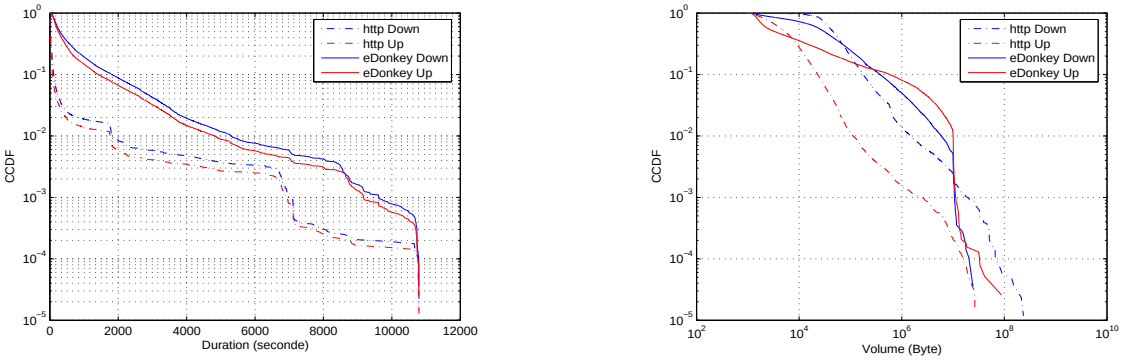


FIG. 1.23 – La répartition du volume sur les flots éléphants



(a) Durée

(b) Volume

FIG. 1.24 – Comparaison entre le sens montant et le sens descendant pour le p2p et le Web.

l'intensité d'utilisation du port '80' pour l'application eDonkey au moins pendant la durée de capture. Les proportions du trafic http sont bien réelles et certainement liées aux profils des clients ADSL desservies par la plaque observée. Pour le trafic inconnu, il est très difficile à ce niveau de l'étude de se prononcer sur sa nature. La méthode des annuaires donnent un ordre de grandeur des volumes du trafic potentiellement eDonkey surtout par rapport aux annuaires basés sur les adresses externes. Par ailleurs, l'association des deux sens de l'observation (trace bidirectionnelle) a permis d'affiner la méthode des annuaires. En effet, la création d'un annuaire interne et un externe permet de diminuer le taux de fausse alerte sans faire chuter pour autant le taux de trafic inconnu et qui est déclaré comme trafic "eDonkey". Ce trafic comporte une grande partie du trafic eDonkey caché. Cependant, il comporte également une partie du trafic p2p caché (autre qu'eDonkey). Une caractérisation plus fine de la composition du trafic inconnu et déclaré "eDonkey" nécessite une analyse de niveau applicatif.

Une amélioration dans la construction de ces annuaires est envisageable pour augmenter l'acuité de la méthode surtout par rapport aux annuaires des adresses internes. Il serait judicieux de respecter le principe de la causalité lors de la construction de l'annuaire. En conséquence, l'annuaire est construit au fur et à mesure. Cela permettra de diminuer le taux des fausses alertes.

1.4.5 Les premiers paquets au service de l'identification

Les paramètres adoptés par [7, 8, 9] pour représenter une connexion TCP semble être une représentation discriminante. Afin de vérifier cette assertion, nous avons utilisé cette représentation et nous avons évalué son pouvoir discriminant sur des traces issues du réseau de France Telecom. Pour ce faire, nous avons choisi d'utiliser deux traces issues de deux points différents du réseau de France Telecom grâce à la sonde OTARIE. La première trace a été capturé, le 11 septembre 2006 à Strasbourg. La deuxième trace est issue d'un lien situé à Rennes et a été capturée le 15 janvier 2007. Les deux liens sont techniquement équivalents, c'est à dire pourvus de la même capacité et situé au même niveau de la hiérarchie du réseau de cœur de France Telecom. Les deux traces fournissent les données complètes des deux sens du trafic. Nous avons pu en conséquence reconstruire les connexions TCP. L'analyse de la charge utile des paquets nous permet également de reconnaître les applications associées à ces connexions TCP. Nous disposons ainsi d'un jeu de connexions TCP (représentées par la taille des dix premiers paquets modulo le signe associé au sens du paquet) dont nous connaissons l'application associée (environ 70000 connexions TCP pour chaque trace). Nous créons ensuite, une batterie d'apprentissage qui représente 2/3 de la batterie totale.

Pour procéder à l'apprentissage, nous avons utilisé l'outil KHIOPS. Cet outil de classification a été développé au sein de France Telecom. L'outil KHIOPS se base sur une version amélioré du modèle Bayesien naïf standard. La méthode de modélisation par sélection [17] (d'où le nom de Selective Naive Bayes) améliore systématiquement et significativement les résultats du Bayesien Naïf "standard". Un atout de la méthode réside également dans sa robustesse et son pouvoir d'adaptation à l'échelle. En effet, le modèle de classification offre les mêmes performances pendant l'apprentissage et pendant la phase opérationnelle. En d'autres termes, le modèle de classification n'est pas le résultat d'un apprentissage par cœur.

Une autre spécificité de l'outil KHIOPS réside dans la méthode de traitement des attributs. En effet, cet outil repose sur une méthode optimale de prétraitement des attributs (les éléments du vecteur connexion TCP dans notre cas). Le prétraitement revient à une opération de

discrétisation pour les variables numériques (univariées [14] et bivariées [16]) ou à une opération de groupement de valeurs pour les variables catégorielles [15].

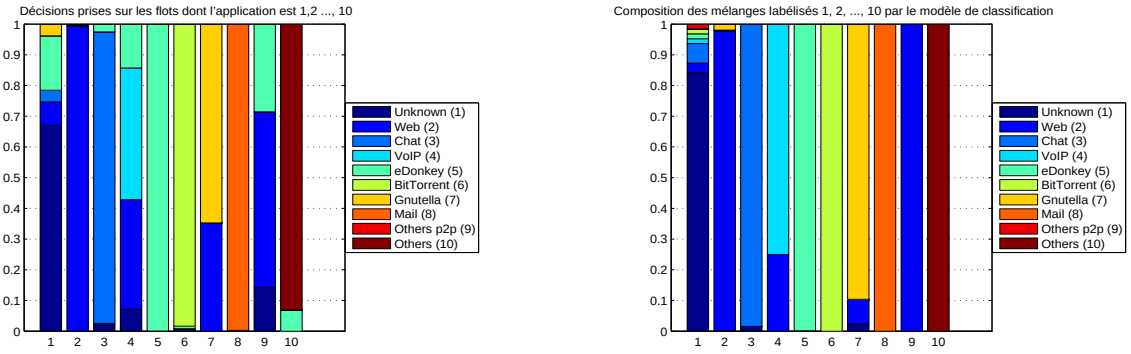
Pour tester cette approche, nous avons adopté le processus expérimental décrit par la suite. Parmi l'ensemble des flots TCP, nous gardons seulement les connexions TCP dont le handshake est réussi (une suite SYN, SYN/ACK, ACK). Ces Connexions TCP sont préalablement associées à une application donnée grâce à une analyse du niveau applicatif (OTARIE). Chaque connexion va correspondre alors à une application donnée. Dans un premier lieu, ce label applicatif va servir à créer le modèle de classification. Ce modèle est alors crée avec une proportion de la batterie de connexions TCP (environ 2/3 de la batterie). Le dernier tiers de la batterie sert à évaluer le modèle de la classification. L'évaluation du modèle revient à comparer les décisions du modèle avec les vrais labels applicatifs. Une décision revient à attribuer une connexion TCP à une application donnée. En conséquence, pour une connexion TCP qui fait partie de la batterie de test, deux labels applicatifs existent. Le premier label est attribué par l'analyse du niveau applicatif. Le deuxième label constitue la décision prise par le modèle de classification. Ce label est désigné dans la suite par le terme label décisionnel.

Afin d'évaluer les performances de la méthode de classification, ainsi que le pouvoir discriminant du paramètre choisi, nous avons repris deux critères classiques d'évaluation.

Pour une connexion donnée, nous associons deux labels : le premier label est issu de l'analyse payload, ce label est dit réel (l'analyse du payload est prise comme référence). Le deuxième label est attribué par la méthode de classification à cette connexion. Pour évaluer la méthode de classification, nous avons deux critères.

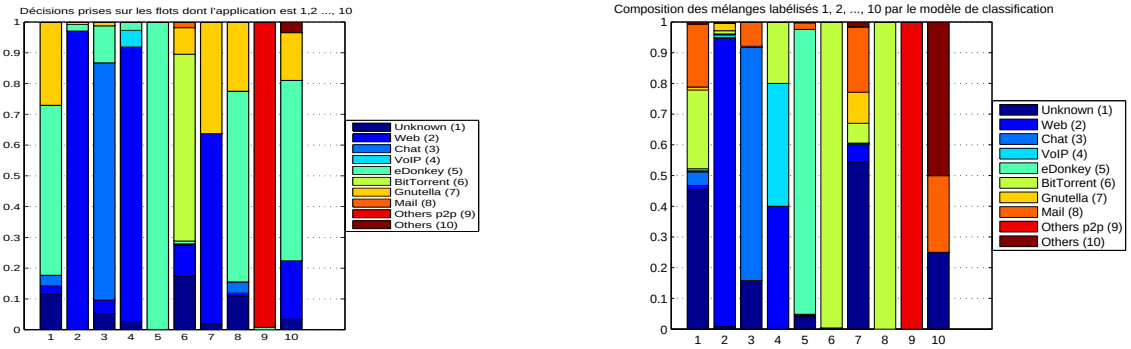
Afin de présenter le premier critère d'évaluation, prenons par exemple, l'ensemble de connexions ayant comme label réel '2'. Donc c'est l'ensemble des connexions web déclarées 'connexion web' par l'analyse de la signature applicative. Nous analysons ces connexions notre méthode de classification. Idéalement, la méthode devrait annoncer dans 100% des cas que c'est des connexions web. Mais dans la pratique, le classifieur peut se tromper et labelliser une partie de ces connexions web par d'autres labels différents (un pourcentage des connexions web est déclaré FTP, un autre est déclaré p2p, etc.). Donc le premier critère d'évaluation est de voir la capacité du classifieur à trouver toutes les connexions web et de minimiser le nombre de connexion web que l'on leur affecte un label différent du celui du web (p2p, ftp, etc.). Cependant ce premier critère n'est pas suffisant, pour la raison suivante : prenons un classifieur très simple, qui décide d'une façon systématique et affirme que n'importe quelle connexion TCP qui se présente est une connexion web. Dans ces conditions, ce classifieur est idéal par rapport au premier critère car il réussit à identifier la totalité des connexions web en affirmant que c'est du web. En revanche, un tel classifieur prendra l'ensemble des connexions (p2p, ftp, voip, etc.) pour du web. Ainsi "le sac" web (toutes les connexions labéllisées web par le classifieur), contient toutes les connexions web mais contient également beaucoup d'impuretés (toutes les connexions non web mais labélisées web par le classifieur). D'où l'utilité d'un deuxième critère, qui évalue la pureté du "sac labélisé web par le classifieur".

L'ensemble des figures (1.25(a), 1.25(b), 1.26(a), 1.26(b),1.27(a), 1.27(b) et 1.28(a), 1.28(b)) représentent les résultats des différents jeux possible d'apprentissage et de test selon les deux critères de performance. Les deux figures 1.25(a) et 1.25(b) représentent les résultats relatifs à une expérience où le modèle d'apprentissage est construit à partir des deux tiers de la trace de Strasbourg. Le troisième tiers de la trace de Strasbourg sert à tester ce modèle de classification. La première figure 1.25(a) représente les performances par rapport au premier critère cité plus haut. Les performances par rapport au deuxième critère sont représentées dans la deuxième figure 1.25(b).



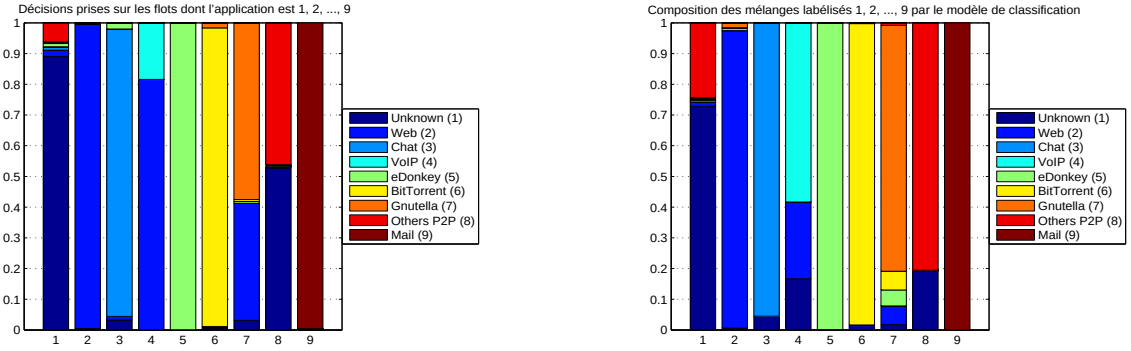
(a) Critère 1 (b) Critère 2

FIG. 1.25 – Apprentissage : Strasbourg 2006 et Test : Strasbourg 2006



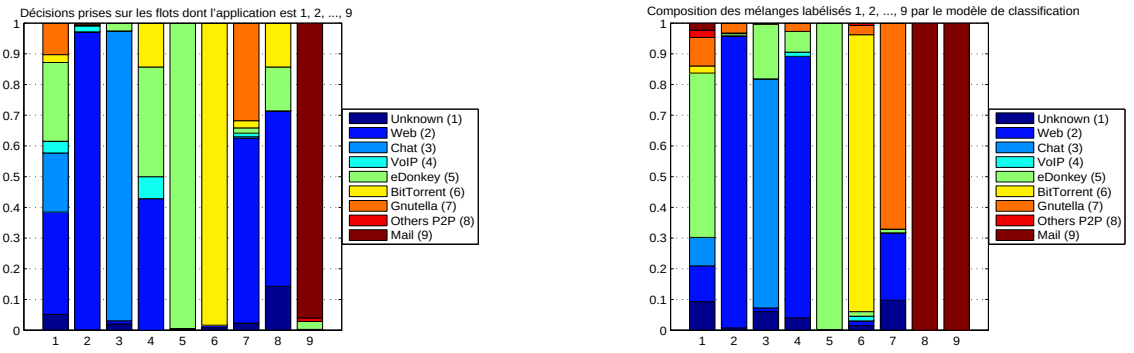
(a) Critère 1 (b) Critère 2

FIG. 1.26 – Apprentissage : Strasbourg 2006 et Test : Rennes 2007



(a) Critère 1 (b) Critère 2

FIG. 1.27 – Apprentissage : Rennes 2007 et Test : Rennes 2007



(a) Critère 1 (b) Critère 2

FIG. 1.28 – Apprentissage : Rennes 2007 et Test : Strasbourg 2006

Les résultats de l'application de la méthode sont assez satisfaisants. Les résultats sont meilleurs lorsque l'on utilisent la même trace pour l'apprentissage et pour le test (il faut bien noter que bien qu'elles soient issues de la même trace, les données servant à l'apprentissage sont différentes de celles que nous utilisons pour le test). Dans ce cas, nous atteignons des taux globaux de précision qui avoisinent l'unité. Quand on applique la méthode sur des jeux de test et d'apprentissage croisés (Rennes Vs Strasbourg), ces taux chutent mais restent relativement élevés (entre 0.887 et 0.9641). En effet, Les performances se dégradent avec un croisement (apprentissage Strasbourg/test Rennes, voir les figures 1.28(a) et 1.28(b)). La quasi-absence de la classe "others p2p" dans la 1ère trace (Strasbourg) explique cette dégradation. En fait, la méthode n'a pas assez d'enregistrements pour apprendre le comportement de cette classe. Rappelons d'ailleurs que la classe "others p2p" est en fait une nébuleuse d'applications p2p, du coup l'ensemble d'apprentissage de cette classe se retrouve une fois de plus scindé en une myriade de comportements représentés par un nombre restreint de flots. L'alimentation de la base d'apprentissage est indispensable dans la construction d'un modèle robuste et assez performant. Dans le cas d'une implémentation en temps réel, l'alimentation de cette base d'apprentissage doit être périodique afin de s'adapter aux changement du comportement de l'application étudiée. La périodicité de cette opération d'apprentissage est un paramètre à définir en fonction de notre connaissance sur l'application étudiée. La supervision de l'évolution du comportement d'une application permet de remonter des alarmes dés que l'on commence à dériver énormément de notre comportement de base appris, au moment de la construction du modèle.

Dans une implémentation temps réel, il serait judicieux également de traiter les deux sens séparément. Cependant, la séparation des deux directions du trafic réduit sans aucun doute l'information contenue dans le modèle de classification. Il s'agit donc d'un compromis entre, les difficultés que posent d'un coté un traitement conjoint des deux sens sur un lien de grande capacité et de l'autre coté une perte en connaissance et en pouvoir de séparation du modèle.

Une dernière amélioration du modèle consiste à classifier systématiquement les flots de test avec une logique "la classe la plus probable" tout en ayant une connaissance a priori des variables. Il serait alors judicieux de créer une espèce d'urne des inconnus (rempli moyennant un seuil sur la probabilité a posteriori). Sur les flots qui remplissent cette urne, nous pouvons appliquer une méthode de clustering dès que sa taille devient conséquente. L'application d'une

méthode de clustering sert à identifier des applications naissantes dans le réseau. Il est également possible de caractériser le comportement d'une application donnée en l'étudiant d'une façon isolée ç'est à dire au sein d'un réseau local par exemple. Cette démarche semble être une démarche incontournable dès qu'il s'agit d'une application cryptée donc très difficile à identifier avec les méthodes classiques.

1.5 Conclusion

En partant du principe qu'une identification au niveau applicatif était coûteuse en ressources et qu'une telle méthode pouvait être mise en défaut par l'introduction de procédés de cryptage, nous avons mis au point dans ce document une technique d'identification du trafic eDonkey qui repose sur des annuaires plus ou moins affinés. Cette technique donne des résultats assez peu fiables pour l'identification de eDonkey sur le trafic engendré par les clients rattachés à une plaque ADSL à cause d'un volume de faux positifs trop important. Cette limitation est due au fait que les clients utilisent plusieurs protocoles p2p et qu'un volume non négligeable de trafic que l'on sait pertinemment être du BitTorrent, Gnutella, etc. est considéré par la méthode des annuaires comme étant de l'eDonkey. Par contre, cette méthode donne des résultats tout à fait corrects lorsque nous associons les informations issues des deux sens du trafic (un taux de faux positifs inférieur à 5 %). L'application de la méthode des annuaires permet d'obtenir une borne supérieure pour le trafic eDonkey. Ceci permet en particulier d'observer que le Web est marginalement utilisé pour effectuer des transactions d'échange de fichiers dans le réseau eDonkey. Toutefois, le protocole http est utilisé d'une façon naturelle dans d'autres protocoles p2p.

Enfin, une identification basée sur les numéros de ports standards et voisins permet d'avoir un échantillon représentatif de la communauté eDonkey. Nous remarquons, grâce à la dichotomie souris/éléphants, que cette communauté est de grande taille en dépit de la courte durée des traces. Cependant, la communauté qui forme le réseau des vrais échanges de fichiers est de taille beaucoup plus modeste. En effet, la distinction entre la signalisation et les vrais échanges est possible grâce à la distinction entre souris et éléphants. Le nombre d'adresses impliquées dans des échanges de souris (signalisation) est extrêmement important. En revanche, le nombre d'adresses qui particpent dans des vraies échanges de données (éléphants réguliers) est plus modeste.

Chapitre 2

La topologie des réseaux pair à pair : du pair jusqu'au système autonome

Sommaire

2.1 Introduction

Nous nous intéressons dans ce chapitre à l'analyse des caractéristiques géographiques du trafic ADSL. Notre analyse est scindée en deux parties. Nous étudions, dans la première partie, le trafic ADSL de plusieurs populations d'utilisateurs ADSL français. Nous nous concentrons ainsi sur la répartition géographique de leur trafic sur les différents pays. La deuxième partie est centrée sur l'analyse de la répartition de ce même trafic ADSL sur une échelle nationale. Nous portons un intérêt particulier au trafic engendré par l'application pair à pair la plus populaire : eDonkey.

2.2 La topologie des réseaux pair à pair dans la littérature

Lorsque nous parlons de topologie, au moins deux définitions peuvent être avancées. Une première approche définit la topologie comme étant l'ensemble de paires qu'on pourrait atteindre en se connectant au système p2p. Une deuxième approche définit la topologie comme étant l'ensemble des paires présentant une activité donnée pendant une certaine durée. L'activité des pairs peut être liée, par exemple, au routage ou aux échanges de données. Notons que la prise en compte de l'activité des pairs donne forcément une description plus complète du système étudié.

Nous avons choisi alors d'adopter la seconde approche. Plus, l'étude est concentrée plus particulièrement sur l'activité liée aux échanges de données. Ce choix est justifié par la nature même du réseau eDonkey. Le réseau eDonkey est un réseau partiellement décentralisé. En effet, rappelons que l'indexation dans le réseau eDonkey est faite grâce à un ensemble de serveurs d'indexation géographiquement décentralisés. Le système eDonkey est alors dépourvu de tout mécanisme de routage. La caractérisation de la topologie du réseau eDonkey revient alors à déterminer une cartographie de l'écoulement des volumes dans ce réseau. Cette cartographie permet dans une seconde étape d'évaluer la capacité du système eDonkey à profiter, d'une façon optimale, de la topologie physique sous-jacente.

Une utilisation optimale de la topologie physique est utile pour le fournisseur d'accès (coût de peering, trafic inter FAI,..). L'utilisateur profite aussi de cette utilisation optimale du réseau de son fournisseur d'accès. En effet, l'optimisation des échanges dans le réseau pair à pair aurait un impact réactivité et sur les temps de téléchargement.

Les auteurs de [11] étudient la topologie du réseau eDonkey en analysant, au niveau d'un serveur d'indexation, les requêtes des paires. Pour ce faire, les auteurs retiennent pour chaque requête, la date de l'arrivée de la requête, du pair origine, l'identificateur du fichier demandé et enfin la liste de sources de ce contenu et qui sont susceptibles d'être contactées par le client demandeur. Toutes les informations collectées sont agrégées dans une structure de graphe. Les auteurs de [11] utilisent un graphe orienté avec deux types de nœuds. La première classe de nœuds est celle des pairs. La deuxième est celle des fichiers. Les liens orientés relient les nœuds de type différent. Il est clair que le premier objectif est de faire une analyse où se chevauchent la topologie sémantique et la topologie logique. Les auteurs notent la grande hétérogénéité entre les différents nœuds. Cette observation, généralement associé à la loi sans échelle signifie que la moyenne ne représente en aucun cas un comportement typique des nœuds. Bien que l'étude [11] ait permis de déceler quelques caractéristiques du réseau eDonkey, elle reste relativement limitée à un niveau d'analyse particulier. En effet, le but principal de cette étude est de caractériser des communautés sémantiques. L'existence de ces communautés

sémantiques est montré par [48]. En effet, l'application d'un algorithme de réduction de graphe permet d'obtenir des sous-graphes que les auteurs supposent représentatifs des communautés sémantiques. L'existence de ce type de communautés dans le réseau eDonkey a été confirmé dans [37, 36, 26].

2.3 Cadre expérimental et conventions

Dans ce document, nous considérons une trace de trois heures, collectée le 07 Mars 2005 entre 18h et 21h, dans les deux sens d'un lien OC3. Ce lien dessert une population d'utilisateurs ADSL estimée à plus de 2000 abonnés. Nous nous limitons à une classification applicative basée sur l'analyse des numéros de ports. Les flots TCP sont alors identifiés comme étant des flots eDonkey si le numéro de port source ou/et destination correspond à l'un des numéros de port communément réservés à l'application eDonkey (ex. 4662, 4661). L'analyse d'une trace moins récente que celle-ci (collectée sur un lien GE en Novembre 2004), a révélé une utilisation intensive des numéros de ports 5662, 14662 et le 14662 chez les utilisateurs eDonkey. Après une analyse de la couche applicative, nous avons vérifié que ces numéros de port sont effectivement utilisés par eDonkey. Nous tenons compte de cette observation en considérant dorénavant ces numéros de ports voisins comme des numéros de ports usuels de l'application eDonkey.

Avant d'aborder les aspects géographiques, nous commençons par présenter brièvement la nomenclature adoptée dans ce rapport. Nous rappelons que le sens descendant est le sens qui part du réseau dorsal vers la population d'utilisateurs ADSL étudiée. Le sens montant est le sens qui remonte des utilisateurs ADSL vers le réseau dorsal. Nous sommes également amenés à rappeler la notion d'adresse interne et externe. Le premier type d'adresse comporte toutes les adresses de destination du sens descendant ainsi que les adresses de source du sens montant. Ce premier lot d'adresses identifie grossièrement les utilisateurs ADSL desservis par le lien OC3 étudié. Le biais introduit par les mécanismes d'allocation dynamique des adresses IP est quasi inexistant dans le cas d'une étude qui s'intéresse à la localisation géographique à l'échelle d'un pays ou même à l'échelle d'une ville. Par ailleurs, une granularité plus fine pourrait avoir des répercussions sur la fiabilité des résultats obtenus. Ces répercussions pourraient être évitées grâce à une réduction de la fenêtre temporelle. En effet, une trace de courte durée garantit une certaine bijectivité entre l'ensemble des adresses internes d'un côté et l'ensemble des utilisateurs ADSL de l'autre. Cette assertion est également valable pour les adresses externes qui sont par définition les adresses IP de destination du sens montant ainsi que les adresses IP de source du sens descendant.

En utilisant ces définitions, nous menons dans la suite une étude phénoménologique de la localisation géographique du trafic ADSL avec un zoom particulier sur la composante eDonkey de ce trafic. Le reste des composantes du trafic, qui sont d'une importance non négligeable, serviront comme référence afin d'extraire les spécificités du trafic eDonkey. Mais avant de présenter les résultats sur la localisation géographique du trafic ADSL, nous allons commencer par justifier le choix de la méthode de localisation des adresses géographique adoptée.

2.4 Méthodes de localisation géographique des adresses IP

La problématique de la localisation des adresses IP n'est pas le premier objectif de ce document, toutefois nous allons faire un bref survol du spectre des méthodes utilisées dans le monde académique afin de nous aider à situer l'approche retenue pour la localisation des

adresses externes dans le présent document. Le principal constat qui découlerait d'une étude plus profonde des différentes méthodes de localisation serait sans doute le classique mais inévitable compromis entre le coût et la fiabilité. Le terme coût est ici un terme générique qui réunit une myriade de paramètres. Pour résumer, nous pouvons synthétiser les techniques de la localisation en mettant en évidence trois axes ou approches qui sont complémentaires.

La première famille de techniques consiste à utiliser les données publiques des organismes gestionnaires. Ces organismes prennent en charge l'attribution des plages d'adresses IP aux différents FAI (fournisseurs d'accès Internet).

Une deuxième famille de techniques de localisation se base sur les informations fournies par les sites web où les utilisateurs sont amenés à remplir des champs relatifs à leur localisation géographique (ex : sites de commerce en ligne). Ce type de techniques peut paraître simpliste mais explique entre autres les prix réduits que proposent les prestataires de service de la géo-localisation utilisant ces méthodes. A titre d'exemple, nous pouvons citer MaxMind qui garantit malgré tout une fiabilité dépassant en moyenne les 80%. Ce pourcentage varie bien évidemment en fonction de la granularité que nous souhaitons obtenir (pays, région, ville). Une granularité très fine serait une exigence très difficile à satisfaire par ce type de techniques.

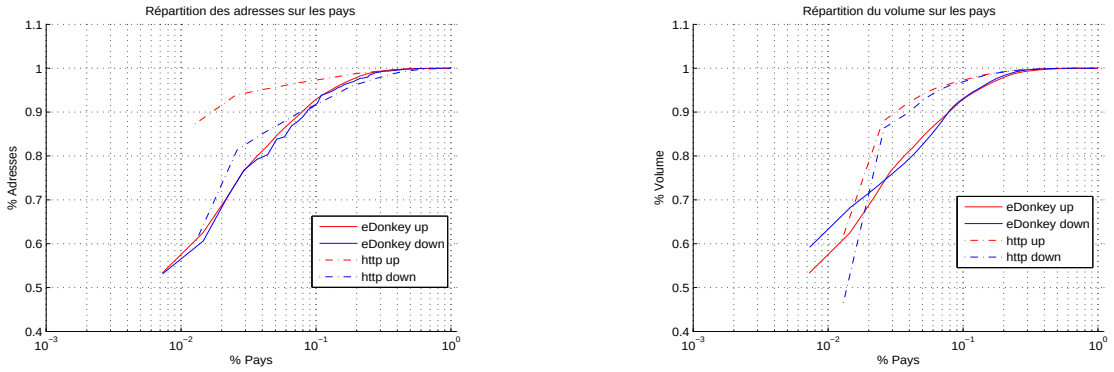
Afin de pallier les limitations de la famille précédente, des techniques plus ou moins complexes existent et peuvent être utilisées. Ces techniques se basent sur des mesures de temps de réponse et assument en conséquence l'existence d'une corrélation entre les délais et la distance géographique. Ces mesures de délai et a priori de distance, alimentent des processus de triangulation. Ces processus sont également alimentés par une base d'adresses balises dont la localisation géographique est bien résolue. Cette famille utilise alors le protocole IP dans le but de localiser les adresses. En plus des temps de réponse, d'autres paramètres issus du protocole IP s'avèrent utiles pour la localisation géographique telle que le traceroute. Cette famille semble être la plus fiable des trois familles présentées ci-dessus. Cette performance explique l'adoption de ce type de techniques par les leaders du marché de la localisation géographique. Cependant les produits fournis par ces industriels retournent le résultat d'un croisement et d'une compilation de l'ensemble des informations issues des trois familles de techniques.

Dans la présente étude, nous avons opté pour la première approche qui est une approche fondée sur les données publiques d'adresses. Cette approche reste efficace pour les grandes échelles (la granularité par pays). L'utilisation de cette approche pour la localisation des adresses externes de notre trace permet d'avoir des taux de non résolution qui dépassent de peu le 1% dans les pires cas (1.25% pour les adresses externes eDonkey, seulement 0.24% pour les adresses externes http). Néanmoins, ces taux ne donnent aucune information sur les taux de fausse résolution. Rappelons à ce niveau le cas connu d'AOL, qui est un fournisseur présent dans plusieurs pays du monde, mais qui fait passer ses utilisateurs par un proxy situé aux États-Unis. Une analyse auxiliaire montre que la contribution du système autonome associé à AOL au volume total qu'on observe sur le lien étudié reste négligeable et n'introduit qu'un faible biais sur les résultats de la localisation géographique des adresses externes.

2.5 Les premières observations sur la localisation

L'analyse du trafic eDonkey montre que les adresses externes observées sur les trois heures de capture sont localisées géographiquement dans 137 pays. Ce nombre est seulement de 77 pays pour le web. Le trafic eDonkey est alors plus réparti géographiquement que le trafic http. Ce premier constat semble être la traduction de la différence intuitive entre, d'un côté

l'architecture client/serveur comme le web et l'architecture pair à pair d'un autre côté représentée ici par l'application eDonkey. La question est de savoir comment les adresses externes sont réparties sur ces pays. L'idée a priori que nous avons sur les architectures laisse présager que le trafic web est concentré sur un ensemble restreint de pays par opposition à un trafic eDonkey uniformément réparti sur les pays. En effet, nous considérons les réseaux p2p comme des réseaux plats avec des pairs répartis dans le monde d'une façon équitable. L'étude de la répartition cumulative des adresses sur les pays confirme cette intuition mais seulement dans le cas du trafic web.



(a) Adresses (b) Volume

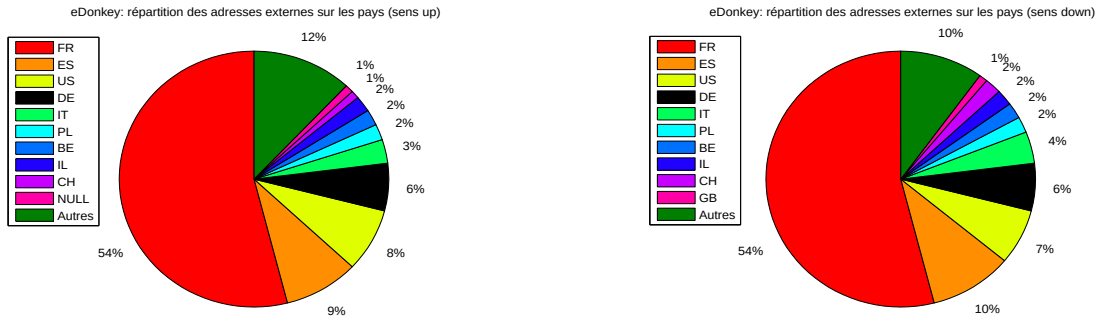
FIG. 2.1 – La répartition du volume et des adresses sur les pays

Dans les figures 2.1(a) et 2.1(b) où sont représentées les répartitions cumulatives du trafic web et eDonkey, nous observons une fois de plus la manifestation de la loi de Pareto. En effet, plus de 95% des adresses externes appartiennent à moins de 10% de l'ensemble des pays que nous avons pu dénombrer (ce qui représente environ 8 pays pour le web et 13 pays pour eDonkey). Cette observation n'est pas restreinte à la répartition des adresses externes sur les pays. La loi de Pareto est également applicable lorsqu'on s'intéresse à la cinématique des volumes. Dans 2.1(b) on peut observer qu'un dixième des pays dénombrés représente l'origine ou la destination de plus de 95% du trafic web observé sur les trois heures contre seulement 90% pour le trafic eDonkey. Cette légère différence n'est pas pour autant déterminante et ne garantit en aucun cas la parfaite répartition du trafic eDonkey sur les pays comme le laisse entendre le terme pair à pair. Ce terme est devenu un symbole d'un réseau régi par la loi de l'aléa où les pairs se découvrent mutuellement dans un cadre purement coopératif grâce à des serveurs d'indexation.

L'existence de ces serveurs d'indexation a fait du système eDonkey un système décentralisé hybride. Ce type de système adopte une approche similaire au système historique Napster. Mais contrairement à ce dernier, le serveur central d'indexation est scindé en une multitude de serveurs répartis géographiquement. Dans le système eDonkey, les serveurs d'indexation peuvent être installés par n'importe quelle personne voulant mettre à disposition la puissance de sa machine. La fermeture de l'un de ces serveurs, dans le cadre d'une action judiciaire, n'aura qu'un effet restreint sur le réseau eDonkey. Notons ici que plusieurs serveurs populaires ont émergé. A titre d'exemple, le serveur belge Razorback monopolise à lui seul près de 25% du nombre total des clients du système eDonkey estimé à environ 4 Millions d'utilisateurs en 2005. Ce serveur d'indexation a été récemment mis hors service suite à une action judiciaire

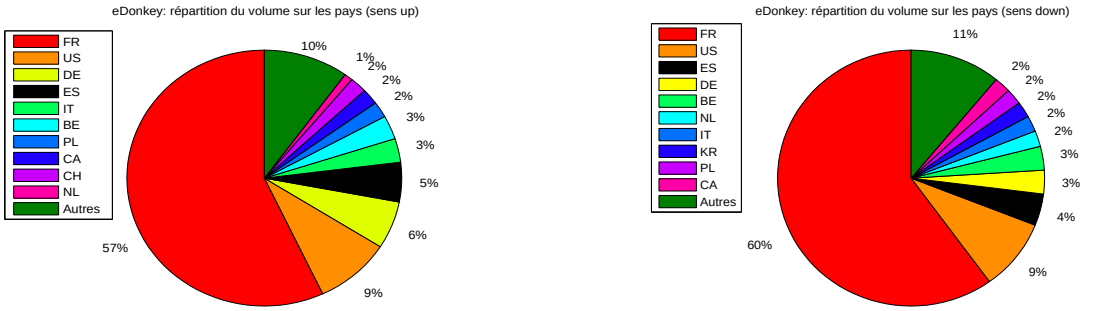
menée à son encontre et nous menons actuellement des études pour connaître l'impact de la disparition de ce serveur sur le trafic eDonkey.

Le réseau eDonkey est un réseau pair à pair avec un mécanisme d'indexation très centralisé (il l'est moins avec l'arrêt du serveur Razorback mais d'autres serveurs prendront certainement le relais). Cet aspect centralisé peut expliquer a priori ce que l'on constate conjointement sur le trafic web et eDonkey. Cependant l'aspect centralisé est seulement réservé à l'indexation et les serveurs eDonkey ne participent à la distribution de contenu que dans des cas exceptionnels qui restent sans effet sur le comportement général du réseau pair à pair. A ce niveau de l'étude, nous sommes incapables de corréler l'aspect centralisé de l'indexation dans le réseau eDonkey avec le fait que le trafic de ce dernier soit concentré sur un nombre restreint de pays.



(a) sens montant (b) sens descendant

FIG. 2.2 – Répartition des adresses sur les pays



(a) sens montant (b) sens descendant

FIG. 2.3 – Répartition du volume sur les pays

Les figures 2.2(a), 2.2(b), 2.3(a) et 2.3(b) apportent de nouveaux éléments qui donnent une explication plausible de ce phénomène. Nous remarquons d'emblée que la France est l'origine (et la destination) de plus de la moitié du trafic eDonkey sur la plaque ADSL considérée. Les proportions restent les mêmes lorsqu'on regarde la répartition des adresses externes eDonkey. Plus de la moitié des adresses eDonkey externes (c'est à dire ayant reçu/envoyé un paquet eDonkey de/vers l'un des utilisateurs ADSL concerné par l'étude) sont des adresses localisées

géographiquement en France. La préférence linguistique et culturelle est de mise dans le réseau eDonkey. Les utilisateurs ADSL qu'on observe communiquent et s'échangent des données principalement avec des utilisateurs français. Cette observation revient essentiellement au type de contenus échangés dans ce réseau (musique, films) dont le caractère culturel et assez souvent linguistique sont incontestables. Les raisons linguistiques sont plus ou moins confirmées par les proportions relatives à la Belgique qui arrive, certes dans le palmarès des dix premiers pays, mais loin derrière des pays comme à titre d'exemple l'Espagne ou l'Allemagne. Le français est bien évidemment loin d'être la langue la plus répandue dans ces deux pays. Par ailleurs, l'application eDonkey arrive en tête du palmarès des applications pair à pair les plus utilisées en Espagne et en Allemagne. En effet, il est de notoriété publique que cette application a eu (et continue d'avoir) du succès en Europe contrairement au système coopératif BitTorrent qui se place juste derrière eDonkey en Europe mais qui arrive en tête aux États-Unis en nombre d'utilisateurs. Les figures 2.2(a), 2.2(b), 2.3(a) et 2.3(b) prouvent ces classement avec environ 80% du trafic eDonkey qui part vers ou provient des pays européens. Nous pouvons alors nous poser la question sur les raisons de cette disparité entre les États-Unis et l'Europe au niveau de la popularité des applications eDonkey.

Les raisons les plus plausibles sont essentiellement issues de disciplines tels que la sociologie, la linguistique ou encore l'histoire. Des groupes de recherche se sont penchés sur la notion des réseaux sociaux qu'on peut retrouver dans les blogs ou dans les systèmes de partage de fichiers. D'autres travaux s'intéressent au clustering sémantique dans ce type de réseau en décelant des communautés qui partagent les mêmes centres d'intérêt. Mais cette famille de raisons n'exclut pas les raisons techniques. En effet, le taux de pénétration du haut débit ainsi que les débits offerts par les FAI jouent un rôle déterminant dans la répartition du trafic ou dans la popularité de telle ou telle application comme nous allons le constater lors de l'analyse du trafic à destination des liens de peering nationaux.

2.6 La géolocalisation à l'échelle nationale

2.6.1 Contexte expérimental et définitions

Comme nous l'avons remarqué dans la section précédente, la contribution du trafic national d'eDonkey dépasse la barre de 50% du trafic eDonkey global. La compréhension de la répartition du trafic eDonkey sur le plan national est devenue alors une étape essentielle pour la prédiction de l'impact de ce trafic sur le réseau d'un opérateur d'étendue nationale (et internationale) tel que France Telecom. Pour ce faire, nous procédons dans cette section à une analyse de différents types de données remontées par six sondes passives (les sondes OTARIE) qui couvrent d'une façon quasi complète le territoire français. Nous donnerons ensuite, quelques détails sur le type de données remontées par ces sondes. Par ailleurs, notons ici qu'un intérêt particulier est porté à la répartition du trafic sur les villes.

Par opposition à la section précédente, les applications sont identifiées ici par une analyse de la charge utile des paquets. En conséquence, chaque application est identifiée grâce à une signature applicative qui lui est spécifique. Les sondes, posées sur des liens équivalents au lien décrit dans la section précédente, remontent des données quotidiennes sur le trafic par application dans les sens montant et descendant.

Afin d'observer la répartition à l'échelle nationale du trafic ADSL, nous regroupons le trafic de chaque application suivant la destination non pas finale mais suivant une destination intermédiaire. Cette destination intermédiaire représente le point de sortie par laquelle un

paquet quitte (ou entre dans) le réseau dorsal de France Telecom. Il est important d'insister sur l'absence de toute corrélation entre le point de sortie et les mécanismes de routage mis en œuvre par l'opérateur. Ces mécanismes sont transparents dans cette étude et la répartition géographique du trafic est la pure conséquence du comportement des utilisateurs et des applications en question accompagné des mécanismes de l'application elle-même.

Pour chaque site, nous observons la répartition du volume sur les nœuds de collecte présents globalement dans toutes les grandes agglomérations urbaines. Ces nœuds de collecte, comme leur nom l'indique, agrègent le trafic issue de plusieurs plaques ADSL. Ces plaques couvrent géographiquement toute la région associée à la ville du nœud de collecte correspondant.

Nous indexons les destinations par des numéros de 1 à 41 (voir le tableau 2.1). Les destinations numérotées de 1 à 39 représentent les villes associées aux nœuds de collecte du RBCI à l'exception de la destination 'FTM-WAP' (N° 11) et la destination 'Inconnu' (N° 16). La destination 'Inconnu' représente le trafic dont nous sommes incapables de déterminer le point de sortie du RBCI. Elle représente également le trafic dont nous connaissons le point de sortie du RBCI mais sans pouvoir le localiser géographiquement. C'est essentiellement le cas du trafic issu du service Gigatransit. Ce service est proposé aux entreprises et leur permet d'avoir une connectivité Internet haut débit. Les bases de données dont nous disposons ne fournissent pas, dans le plupart des cas, des informations sur la localisation géographique des routeurs dédiés à ce service. En conséquence, le trafic Gigatransit est dans le plupart des cas associé à la destination 'Inconnu'. La destination 'FTM-WAP' comporte une partie du trafic des clients du mobile que nous sommes incapables de localiser géographiquement. Nous avons choisi de ne pas associer cette destination avec la destination 'Inconnu'. Finalement, nous avons les deux dernières destinations (40 et 41) : la première destination représente l'ensemble des clients de FAI internationaux. Les clients de FAI nationaux autre que Wanadoo sont regroupés au sein de la 41ème destination (OTIP).

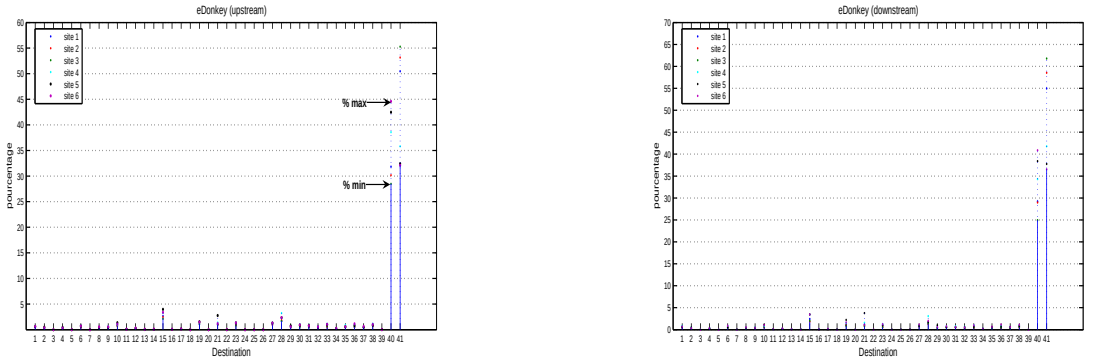
RBCI								Autres	
n°	Destination	n°	Destination	n°	Destination	n°	Destination	n°	Destination
1	Amiens	11	FTM-WAP	21	Lyon	31	Poitiers	40	International
2	Annecy	12	Grenoble	22	Marigot	32	Reims	41	FAI-Nationaux
3	Avignon	13	Guadeloupe	23	Marseille	33	Rennes		
4	Bayonne	14	Guyanne	24	Mayotte	34	Reunion		
5	Besancon	15	IDF	25	Metz	35	Rouen		
6	Bordeaux	16	Inconnu	26	Montpellier	36	Strasbourg		
7	Brest	17	INTERNAL	27	Nancy	37	Toulon		
8	Caen	18	LeMans	28	Nantes	38	Toulouse		
9	Clermont-Ferrand	19	Lille	29	Nice	39	Tours		
10	Dijon	20	Limoges	30	Orleans				

TAB. 2.1 – Correspondance entre numéros et destinations.

2.6.2 Premières observations

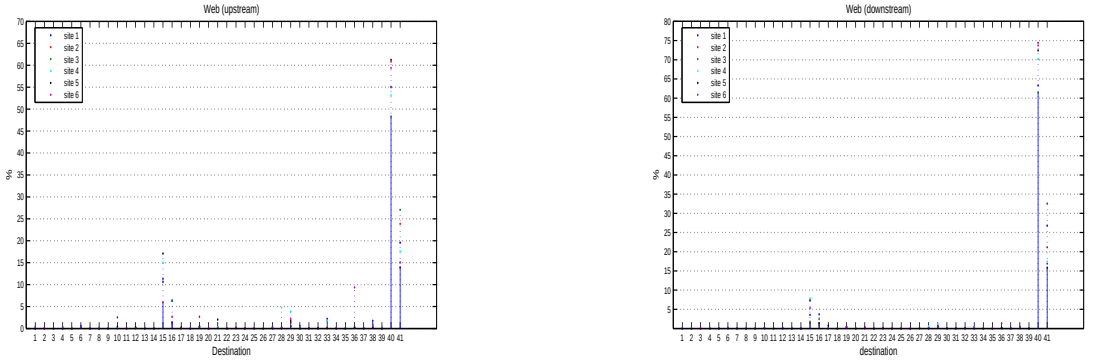
Dans les figures 2.4(a), 2.4(b), 2.5(a), 2.5(b), 2.6(a) et 2.6(b), nous représentons les répartitions géographiques associées à trois applications (eDonkey, BitTorrent et web). Le choix de ces trois applications est lié aux observations faites sur la composition du trafic ADSL (voir le figures 2.7(a) et 2.7(b)). Tout d'abord, nous avons choisi d'analyser l'application eDonkey

qui demeure l'application pair à pair la plus populaire en France. Ensuite, nous nous sommes intéressé au web étant donné que le trafic engendré par cette application arrive en seconde place en termes de contribution au volume global. Finalement, nous avons choisi d'analyser l'application BitTorrent bien qu'elle soit classée derrière Gnutella. L'application BitTorrent est populaire aux États-Unis et commence à émerger en France. La spécificité de cette application réside dans sa philosophie de collaboration très différente des autres réseaux pair à pair. Le trafic du réseau Gnutella est également traité dans le cadre de l'analyse de la stabilité temporelle des matrice de trafic.



(a) sens montant (b) sens descendant

FIG. 2.4 – Les destinations du trafic eDonkey

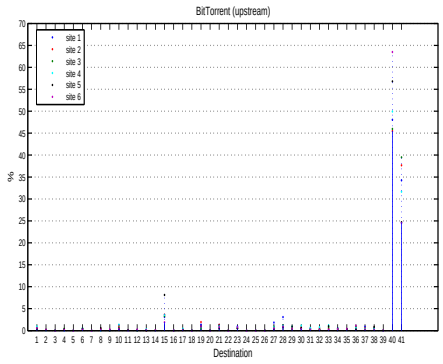


(a) sens montant (b) sens descendant

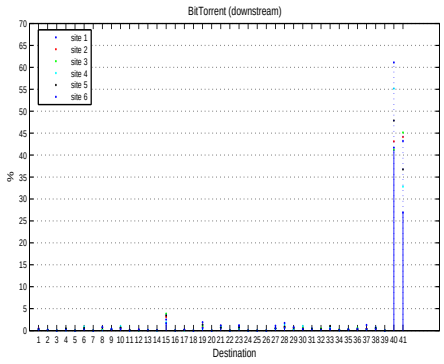
FIG. 2.5 – Les destinations du trafic web

Nous remarquons d'emblée une distribution bimodale des répartitions géographiques. Pour eDonkey, la majorité des destinations ont des proportions qui dépassent à peine la barre des 3%. Par contre, deux remarquables "singularités" capitalisent conjointement plus de 80% du trafic total d'eDonkey. Les deux principaux pics sont localisés sur les destinations 40 et 41 (resp. 'International' et 'FAI-Nationaux').

Avec des termes de mécanique newtonienne, nous qualifierions ces deux destinations de planètes extrêmement massives exerçant sur le trafic une force attractive prépondérante. Cette métaphore adhère à l'image que laisse deviner le modèle gravitationnel (gravity model). Ce

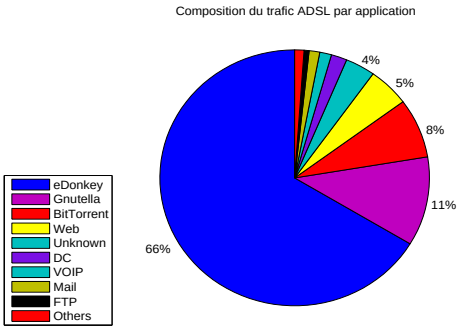


(a) sens montant

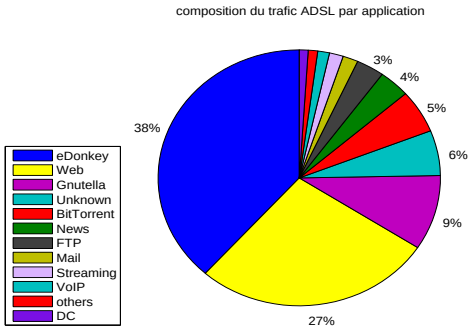


(b) sens descendant

FIG. 2.6 – Les destinations du trafic BitTorrent



(a) sens montant



(b) sens descendant

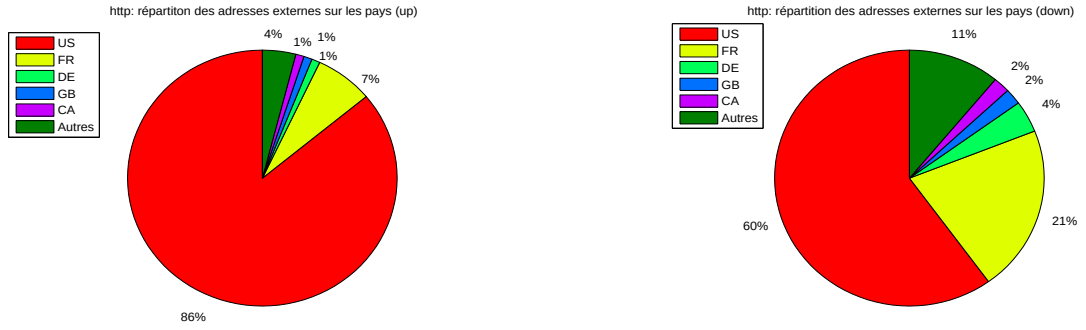
FIG. 2.7 – Composition du trafic ADSL par application (Lyon, le 03 Janvier 2006)

modèle formalise les distributions de la matrice de trafic dans le but d'alimenter les algorithmes d'estimation itératifs par des matrices initiales relativement proches de la réalité. Le modèle prend une forme théorique très analogue à la loi universelle de gravitation.

L'observation est commune aux trois applications avec une légère apparition d'un pic autour de la quinzième destination pour le trafic web. Ce pic existe également pour l'application BitTorrent mais reste relativement moins important que celui du web. Ce nouveau pic correspond à la région d'Île de France.

Nous déduisons alors que la majorité du trafic (eDonkey ainsi que les autres applications) provient ou part vers des clients d'autres opérateurs nationaux (destination n° 40 FAI-Nationaux) et vers des clients d'opérateurs étrangers (destination n° 40 International). La proportion du trafic échangée entre les nœuds de collecte internes au RBCI est relativement modeste et ne dépasse pas, dans les meilleurs cas, la barre des 25% du trafic global.

L'agrégation des données quotidiennes a permis en conséquence d'appréhender la matrice de trafic pair à pair au sein du réseau national de France Telecom. La répartition du trafic eDonkey a contredit une fois de plus le principe des systèmes pair à pair. La topologie que nous avons inférée est de type étoilé. Ce type de topologie est d'habitude en cohérence avec la philosophie client/serveur. La majorité du trafic des clients des plaques ADSL observées est le résultat des échanges de contenus entre ces clients et l'ensemble des abonnées des opérateurs nationaux (autre que Wanadoo) et internationaux. Les liens de peering national et de transit international risquent en conséquence de se retrouver saturés par un trafic pair à pair certes peu agressif mais qui reste majoritaire en termes de volume.

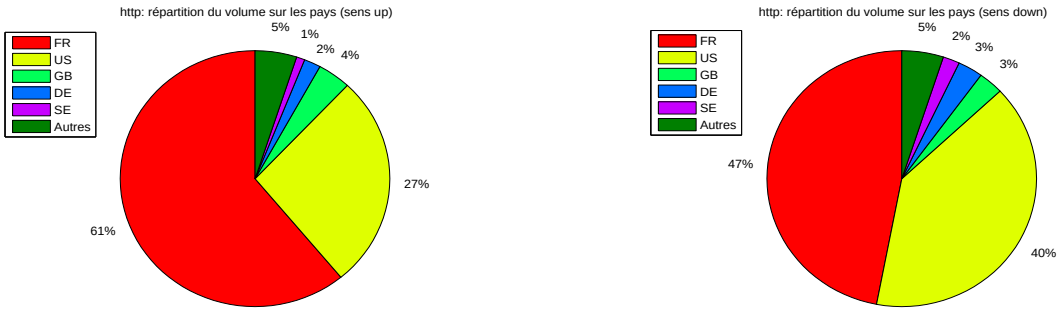


(a) sens montant

(b) sens descendant

FIG. 2.8 – Répartition des adresses externes du trafic web (Lyon, 07 Mars 2005)

Par ailleurs, le trafic web, plus agressif que le trafic pair à pair, exhibe le même phénomène de topologie étoilée avec un pic plus important au niveau de la destination 'International'. Une observation peu surprenante étant donné que le RBCI contient peu de serveurs web et que la majorité de ces serveurs sont situés aux États-unis. Les figures 2.8(a), 2.8(b), 2.9(a) et 2.9(b) expliquent la prépondérance du transit international sur les autres destinations. La répartition selon les pays du trafic web, montre que les États-Unis sont la destination privilégiée des internautes observés (plus de 80% des adresses externes sont géographiquement localisées aux États-Unis).



(a) sens montant

(b) sens descendant

FIG. 2.9 – Répartition du trafic web sur les pays en termes de volume Lyon, 07 Mars 2005)

2.6.3 Comparaison entre les grandes destinations

Dans le but d'estimer d'une façon plus exacte la proportion du trafic échangée au sein du RBCI, nous agrégeons l'ensemble des destinations 1 à 39 en une seule composante que nous appelons destination 'RBCI'. Cette destination représente en conséquence le trafic interne au RBCI. Cette proportion est comparée aux proportions du trafic échangées avec les destinations 'International' et 'FAI-Nationaux'. Les figures 2.10(a) et 2.10(b) représentent les résultats associés à l'application eDonkey.

La destination 'FAI-Nationaux' est largement prépondérante pour les trois premiers sites (1, 2 et 3). Par ailleurs, cette tendance est inversée lorsque nous regardons les trois derniers sites (4, 5 et 6) avec une prépondérance de la destination 'International'. La destination 'RBCI' est minoritaire dans les deux groupes de sites avec des proportions plus importantes pour les sites (4, 5 et 6).

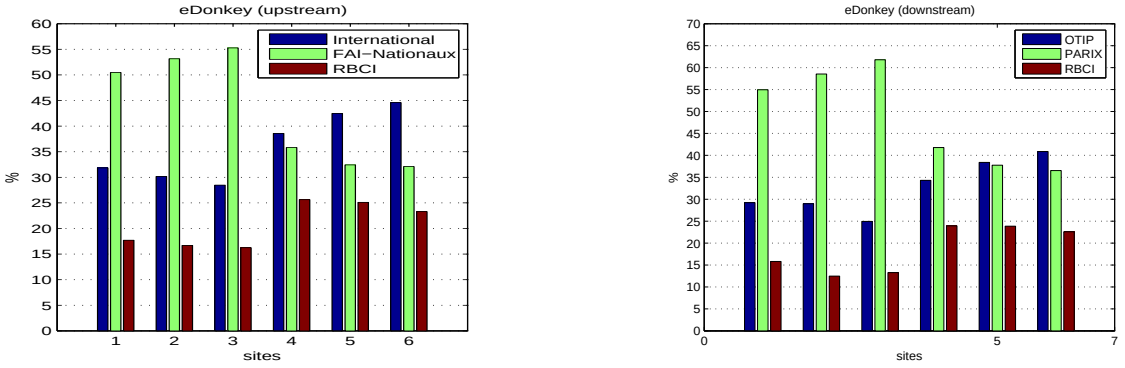
Cette observation révèle l'existence de deux types de sites. Le premier type de sites est représenté par les trois sites 1, 2 et 3. Les trois autres sites caractérisent le deuxième type. En effet, les trois premières sondes OTARIE sont posées sur des liens qui transportent majoritairement (en moyenne 95%) le trafic des clients ADSL CIPA (Collecte IP sur ATM). Ces clients appartiennent à des fournisseurs d'accès Internet autres que Wanadoo dont le transport de trafic est assuré par France Telecom (modèle fermé). Le deuxième type de site observe le trafic provenant des DSLAM réservés aux clients Wanadoo (voir tableau 2.2).

	n° site	Ville
Sites CIPA	Site 1	Toulouse
	Site 2	Lille
	Site 3	Dijon
Sites WOO	Site 4	Nantes
	Site 5	Lyon
	Site 6	Strasbourg

TAB. 2.2 – Sites OTARIE CIPA et Wanadoo

Les clients CIPA communiquent plus avec la destination 'FAI-Nationaux' que les clients Wanadoo. Ces derniers communiquent avec la destination 'International' comme nous pouvons

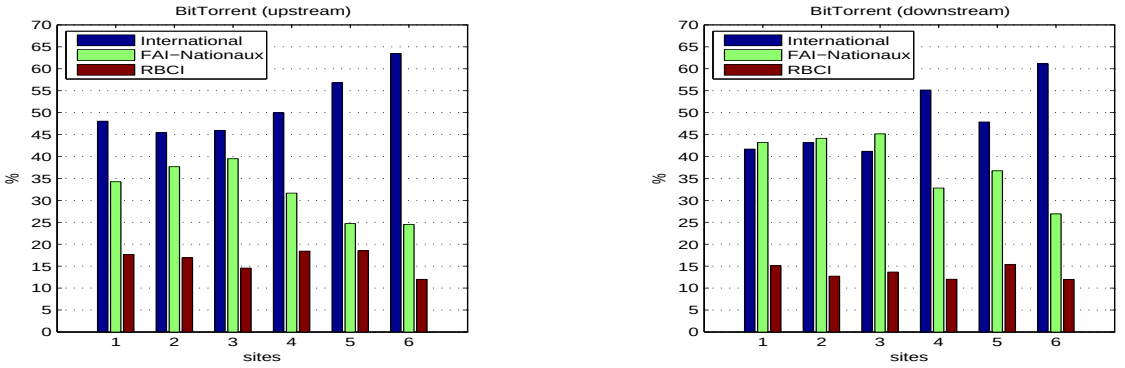
l'observer dans les figures 2.10(a) et 2.10(a) (équité entre 'International' et 'FAI-Nationaux' dans le sens montant, 'International' est majoritaire dans le sens descendant).



(a) sens montant

(b) sens descendant

FIG. 2.10 – Les destinations du trafic eDonkey



(a) sens montant

(b) sens descendant

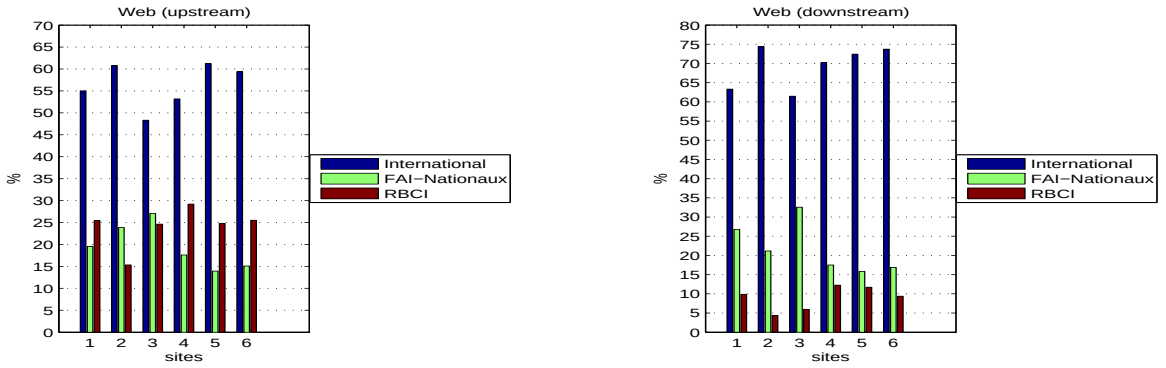
FIG. 2.11 – Les destinations du trafic BitTorrent

Pour les deux autres applications (BitTorrent et web), la destination 'international' arrive en tête comme le montrent les figures 2.11(a), 2.11(b), 2.12(a) et 2.12(b) (pour BitTorrent, la destination 'FAI-Nationaux' arrive légèrement en tête pour les sites CIPA dans le sens descendant).

2.6.4 La localisation géographique du trafic dans le RBCI

Afin d'approfondir nos connaissances sur la répartition géographique du trafic au sein du RBCI, nous nous intéressons maintenant à cette destination. L'analyse de la destination 'RBCI' révèle deux types d'informations.

L'analyse des données issues des sites Wanadoo nous révèle la répartition géographique du trafic interne au RBCI. Par ailleurs, l'analyse des données issues des sites CIPA nous fait découvrir la répartition du trafic issu d'autres opérateurs nationaux et qui pénètre dans le RBCI. En fait, le trafic des clients CIPA est transporté dans des tunnels L2TP jusqu'aux



(a) sens montant

(b) sens descendant

FIG. 2.12 – Les destinations du trafic web

points de peering nationaux (PARIX) indépendamment de sa destination finale. Au niveau de ces points de peering, le trafic est délivré au PoP (Point of Presence) du FAI approprié. Prenons l'exemple du trafic d'un client CIPA qui part à destination d'un client Wanadoo. Les paquets du client CIPA sont acheminés vers PARIX où ils sont délivrés au FAI. Ce dernier se charge alors de l'injection de ces paquets dans le RBCI. Le trafic d'un client CIPA vers une destination située chez un FAI étranger est également traitée de la même manière. Le trafic est acheminé tout d'abord vers le FAI associé au client CIPA. Le FAI se charge alors du transport de ce trafic vers ses propres points de peering internationaux (vers OTIP ou autres réseaux de transit international).

Dans les figures 2.13 et 2.14, nous représentons la répartition de la composante 'RBCI' du trafic eDonkey (clients Wanadoo et clients CIPA) sur les différentes destinations du RBCI (nous gardons la même indexation pour les 39 destinations). En plus du pic principal sur la destination 15 (IDF, Île de France), des pics secondaires sont visibles (28 Nantes, 23 Marseille, 19 Nantes, 27 Nancy). Nous observons également que la répartition géographique dans le RBCI reste la même indépendamment du site ou du type de clients observés (CIPA ou Wanadoo).

Malgré l'existence de ces pics, le trafic eDonkey dans le RBCI est réparti d'une manière relativement uniforme par rapport au trafic web. Dans les figures 2.15(a), 2.15(b), 2.16(a) et 2.16(b), nous représentons la répartition du volume sur les destinations du RBCI triées dans un ordre décroissant de contribution en termes de volume. Les 4 plus importantes destinations représentent plus de 80% du volume du trafic web. En revanche, il faut agréger le trafic des 16 destinations les plus importantes pour atteindre ce même pourcentage lorsque nous nous intéressons au trafic eDonkey.

Afin d'identifier les plus importantes destinations du trafic Web, nous représentons dans les figures 2.17 et 2.18 la répartition géographique du volume associé à cette application sur les 39 destinations du RBCI. Pour les sites CIPA, deux destinations sont prépondérantes : la destination 15 'IDF' et la destination 16 'Inconnu'. Elles représentent près de 70% du trafic global.

Le trafic vers destination 'Inconnu' est composé de deux types de trafic. Le premier type de trafic est le trafic issu du service Gigatransit. Sur l'ensemble du volume échangé avec la destination 'Inconnu', le trafic du service Gigatransit ne représente que 40% dans les meilleurs des cas. Le deuxième type de trafic est un trafic destiné à des adresses introuvables dans

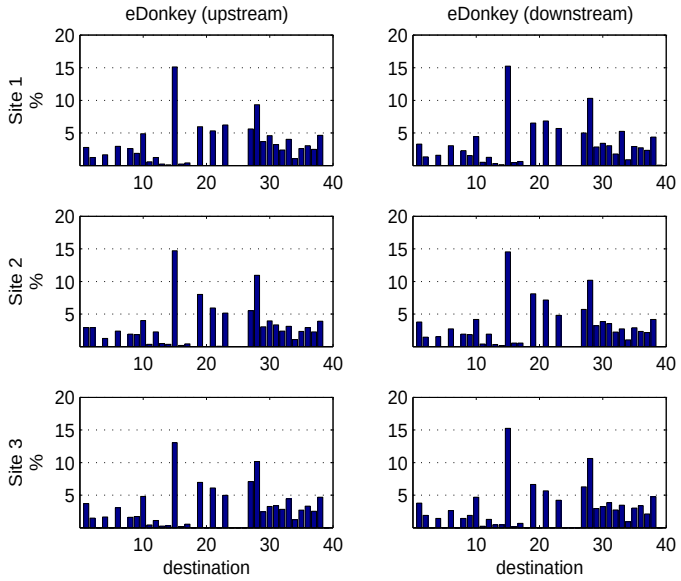


FIG. 2.13 – Répartition du trafic eDonkey dans le RBCI (sites CIPA)

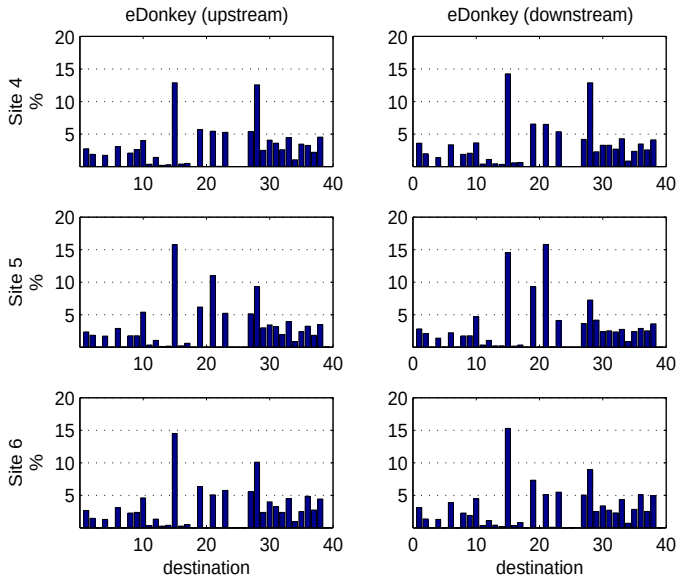
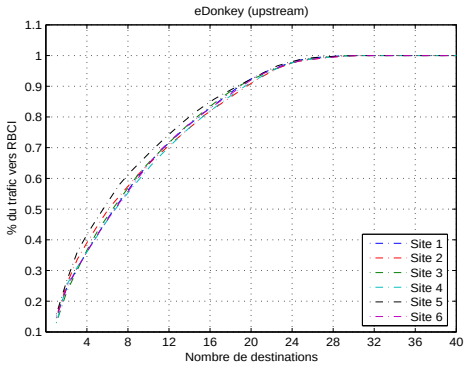
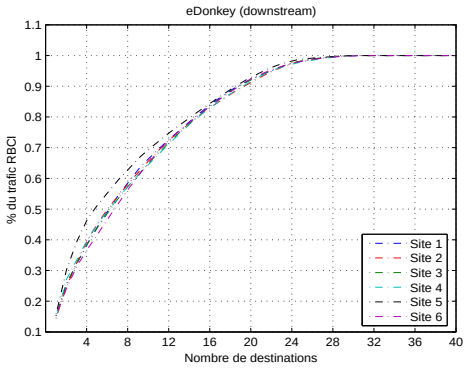


FIG. 2.14 – Répartition du trafic eDonkey dans le RBCI (sites Wanadoo)

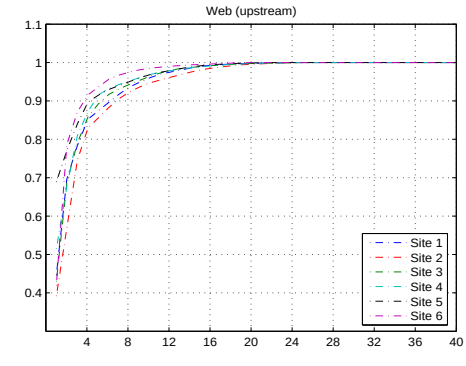


(a) sens montant

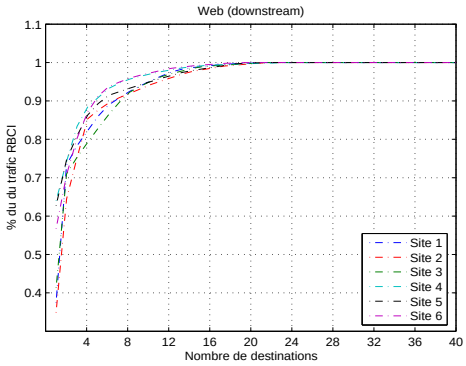


(b) sens descendant

FIG. 2.15 – Répartition du trafic eDonkey au sein du RBCI



(a) sens montant



(b) sens descendant

FIG. 2.16 – Répartition du trafic web au sein du RBCI

les tables de routage. En d'autres termes, ce trafic est le trafic que nous sommes incapables de router dans le réseau. Ce problème de routage concerne seulement le trafic des clients Wanadoo. En effet, les paquets issus des clients CIPA seront injectés dans le tunnel L2TP même si leur destination est absente dans les tables de routage.

L'importance de la proportion du deuxième type de trafic (destination 'Inconnu') pour les sites CIPA peut être liée à un trafic propre aux FAI nationaux qu'ils échangent avec leurs clients situés dans des zones non dégroupés. Ce trafic interne n'a pas à être routé à l'extérieur du réseau du FAI concerné (exp. échanges avec serveurs DNS privés, Webmail, Webnews etc...).

L'analyse des données issues des sites Wanadoo montre que la destination 'IDF' capitalise près de 50% du volume global. Par ailleurs, nous observons une chute de la proportion de la destination 'Inconnu' avec un trafic essentiellement issu du service Gigatransit (99% du trafic web vers la destination 'Inconnu' est de type Gigatransit). Cette observation confirme ce que nous avons conclu pour les clients CIPA sur l'importance du trafic vers des destinations non routables par rapport au trafic issu du service Gigatransit.

Par ailleurs, en regardant la composition par application du trafic vers la destination 'Inconnu' (composante Gigatransit et composante non routable), nous remarquons que le trafic issu du service Gigatransit est typiquement un trafic d'entreprise. En effet, nous notons une prédominance du web en termes de volume. Le trafic 'non routable' est essentiellement du trafic p2p pour les clients Wanadoo. Pour les clients CIPA, le trafic 'non routable' (de point de vue France Telecom et qui emprunte les tunnels L2TP) est essentiellement composé de trafic web.

Finalement, nous revenons à l'application web pour noter l'augmentation du trafic web local. Le trafic web local est le trafic échangé avec le nœud de collecte sur lequel la sonde est montée.

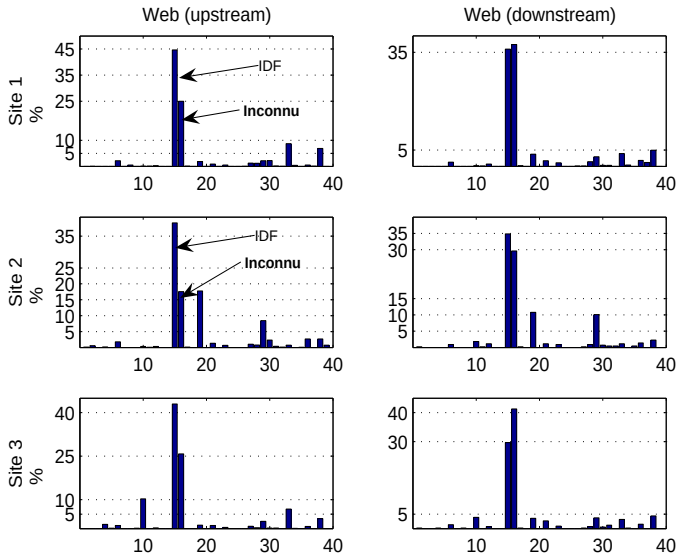


FIG. 2.17 – Répartition du trafic web dans le RBCI (sites CIPA)

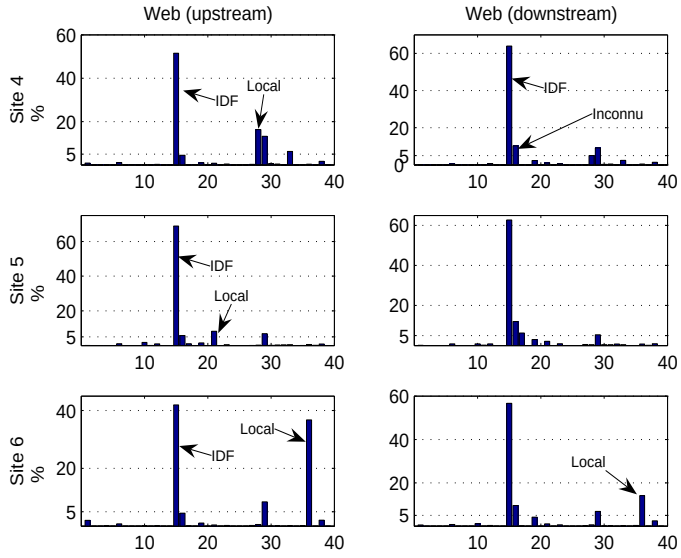


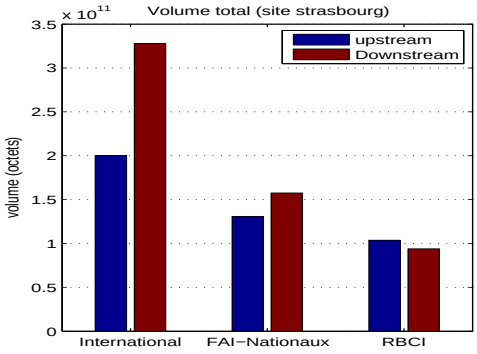
FIG. 2.18 – Répartition du trafic web dans le RBCI (sites Wanadoo)

2.6.5 La symétrie des volumes

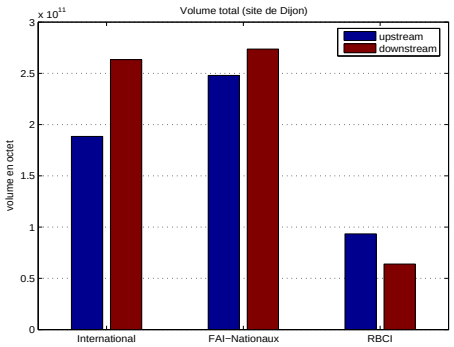
L'analyse de la répartition en termes de pourcentage du volume global a permis de caractériser la topologie en étoile dans le RBCI mais ne fournit aucune information sur le sens réel de l'écoulement des volumes dans cette structure arborescente. Afin d'identifier les fournisseurs de contenus parmi ceux qui jouent uniquement le rôle de consommateur, il est nécessaire de raisonner en termes de volume et non pas en termes de pourcentage. Dans les figures 2.19(a) et 2.19(b), nous représentons le degré de symétrie des volumes entre le site observé et les trois destinations 'International', 'FAI-Nationaux' et 'RBCI' (agrégation des volumes échangés avec tous les nœud de collecte du RBCI). Globalement, le RBCI est consommateur par rapport à la destination 'International'. Ce rôle de serveur que joue la destination 'International' se comprend à travers les deux figures 2.21(a) et 2.21(a) qui montre l'absence de toute symétrie entre le RBCI et la destination 'International' lorsqu'on s'intéresse au trafic web. En revanche, cette asymétrie disparaît lorsqu'on observe le trafic pair à pair (eDonkey et BitTorrent). Nous observons alors une symétrie entre les sens montant et descendant du trafic avec un léger avantage pour le sens montant (2.20(a), 2.20(b), 2.22(a) et 2.22(b)).

2.7 Stabilité temporelle de la matrice de trafic

Pour conclure l'étude de la localisation géographique du trafic ADSL à l'échelle nationale, nous nous intéressons dans cette partie à l'évolution temporelle de la matrice de trafic. Les sections précédentes ont montré que les matrices des trafics eDonkey, web ou BitTorrent gardent en gros les mêmes caractéristiques indépendamment du lieu géographique de l'observation. Cependant, cela est vrai du moment où nous observons le même type de clients (CIPA ou Wanadoo). Parallèlement à l'étude de la stabilité spatiale de nos observations, nous avons procédé à une étude sur la stabilité temporelle. Pour ce faire, nous avons analysé l'évolution dans le temps des matrices de trafic relatives à deux sites que nous supposons être représen-

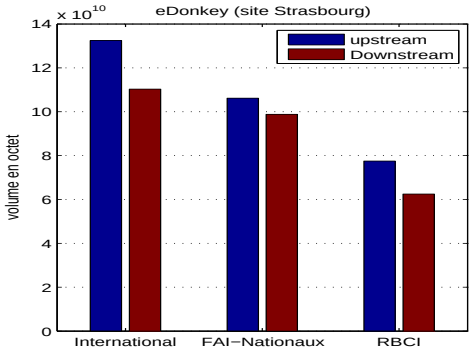


(a) Site 6

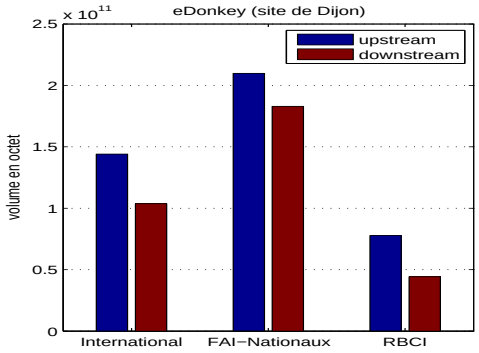


(b) Site 3

FIG. 2.19 – Symétrie du trafic total

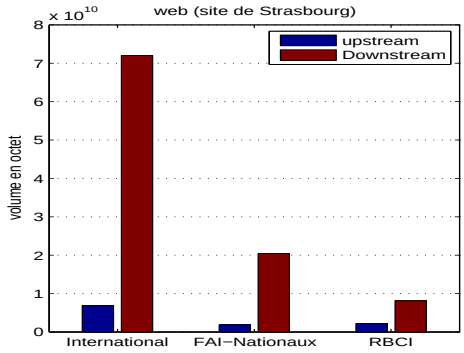


(a) Site 6

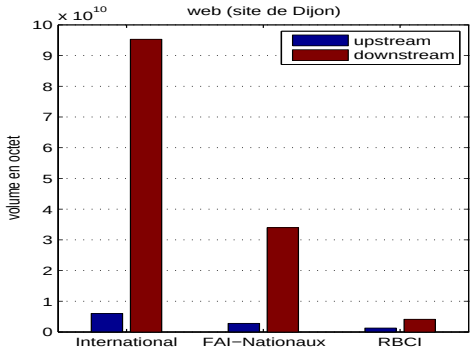


(b) Site 3

FIG. 2.20 – Symétrie du trafic eDonkey

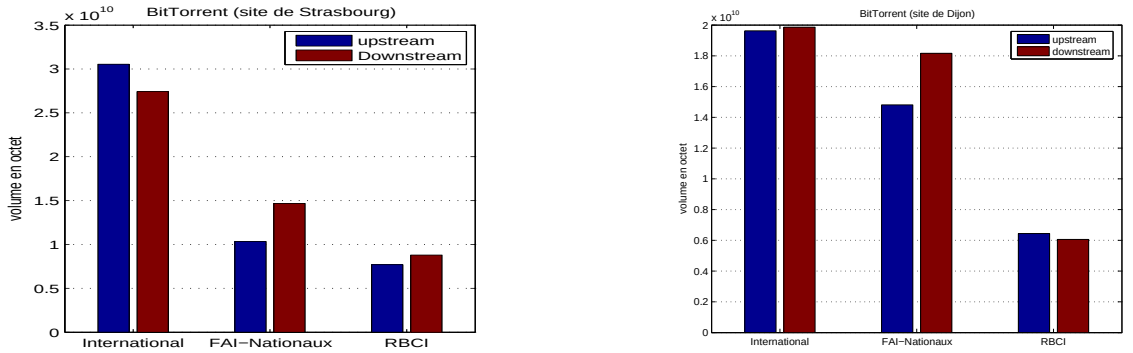


(a) Site 6



(b) Site 3

FIG. 2.21 – Symétrie du trafic web



(a) Site 6

(b) Site 3

FIG. 2.22 – Symétrie du trafic BitTorrent

tatifs. Nous avons gardé en conséquence un représentant de chaque type de sites. Le site 3 représente désormais les sites CIPA, et le site 6 pour les sites Wanadoo.

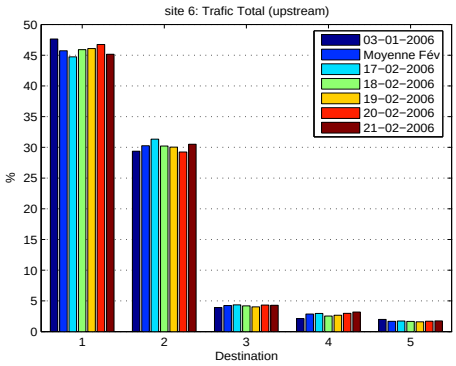
L'analyse de l'évolution temporelle prend en considération deux échelles. La première traite de l'évolution à long terme (45 jours) de la matrice du trafic. La analyse est liée à l'évolution de cette matrice sur une durée de 5 jours consécutifs.

Nous représentons dans l'ensemble des figures de la figure 2.23(a) jusqu'à la figure 2.32(b) l'évolution temporelle des matrices de trafic relatives aux applications les plus importantes du trafic ADSL. La matrice de trafic global est également représentée (2.23(a) et 2.23(b) pour le site 6 et 2.24(a) et 2.24(b) pour le site 3). Dans ces figures, nous représentons l'évolution des proportions des cinq premières destinations du premier jour de l'analyse. Le pourcentage moyenné sur les cinq jours de l'observation est également représenté avec une mesure de ce même pourcentage réalisée 45 jours auparavant (Janvier 2006). Les matrices de trafic sont relativement stables pour toutes les applications indépendamment du type de clients que l'on observe.

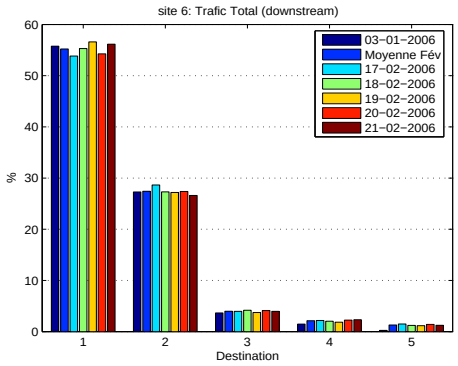
Par ailleurs, c'est l'application eDonkey qui donnent les matrices de trafic les moins variables au cours du temps (à court et à long terme). La stabilité de la matrice de trafic eDonkey se répercute sur la matrice du trafic global étant donnée la contribution prépondérante de cette application dans le volume total observé. Les matrices de trafic des autres applications sont très légèrement variables par rapport à celle du trafic eDonkey. Cette différence est de faible importance. Notons aussi qu'aucune observation particulière ne peut être faite sur les taux de variabilité entre les petites et grandes destinations. Finalement, nous remarquons que les observations faites sur les mesures de Janvier demeurent valables 45 jours après et permettent de se prononcer sur la stabilité temporelle de la matrice de trafic dans le réseau national de France Telecom.

2.8 Le trafic dans le réseau de transit de France Telecom

Le réseau OTIP (Open Transit IP), identifié par le numéro du système autonome 5511, assure l'interconnexion du RBCI avec Internet. Il possède une étendue mondiale grâce à des points de présence sur quatre des cinq continents. Le rôle de transit qu'assure le réseau OTIP explique en partie l'importance de la proportion du trafic du RBCI échangée avec cette des-

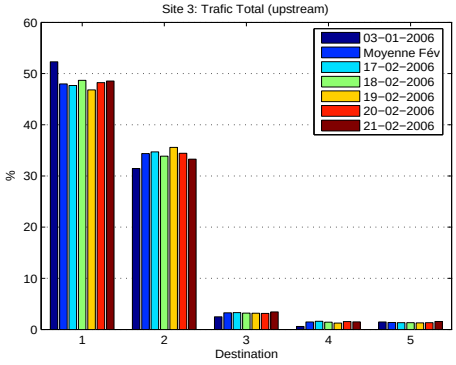


(a) sens montant

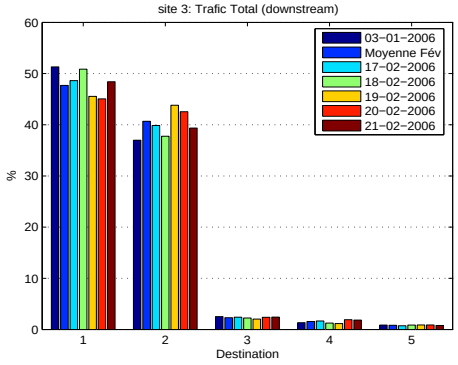


(b) sens descendant

FIG. 2.23 – Stabilité de la matrice de trafic total (site 6)

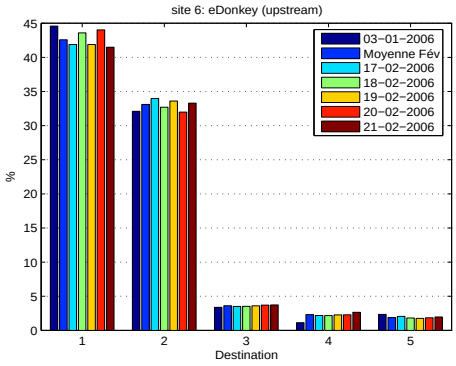


(a) sens montant

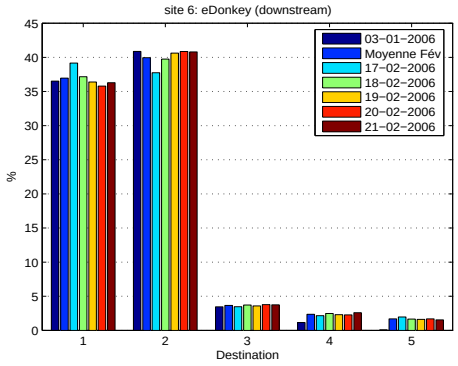


(b) sens descendant

FIG. 2.24 – Stabilité de la matrice de trafic total (site 3)

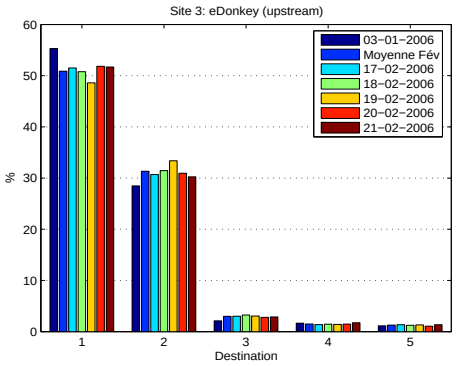


(a) sens montant

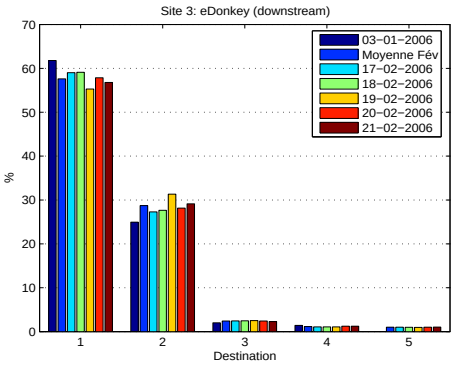


(b) sens descendant

FIG. 2.25 – Stabilité de la matrice de trafic eDonkey (site 6)

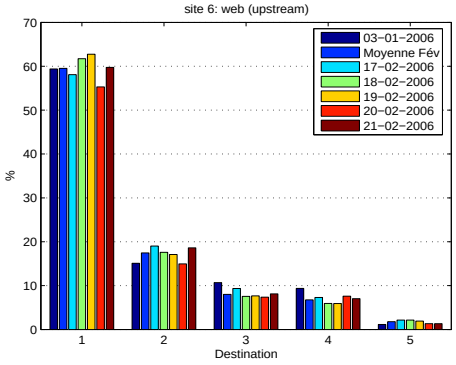


(a) sens montant

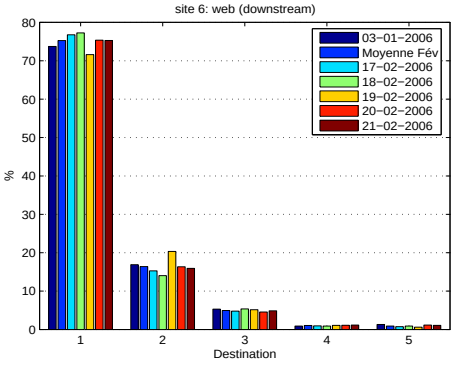


(b) sens descendant

FIG. 2.26 – Stabilité de la matrice de trafic eDonkey (site 3)

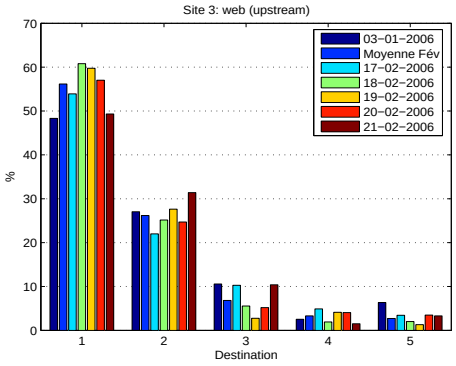


(a) sens montant

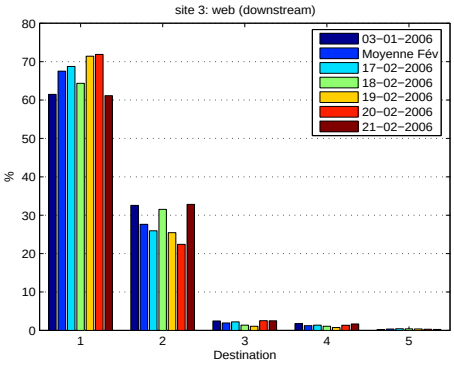


(b) sens descendant

FIG. 2.27 – Stabilité de la matric de trafic web (site 6)

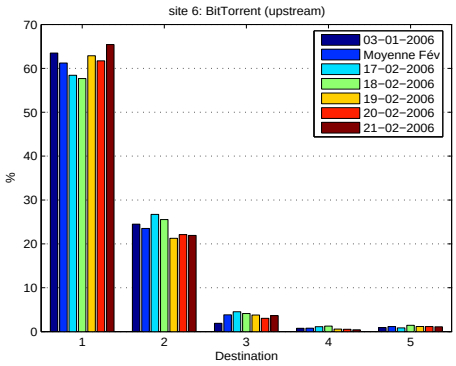


(a) sens montant

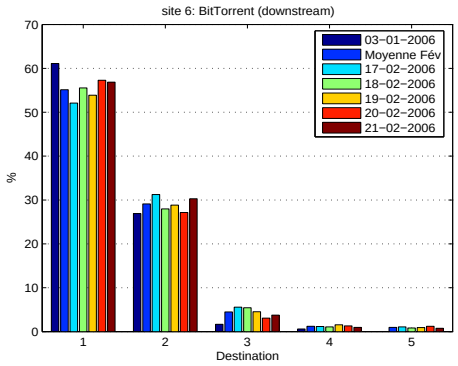


(b) sens descendant

FIG. 2.28 – Stabilité de la matric de trafic web (site 3)

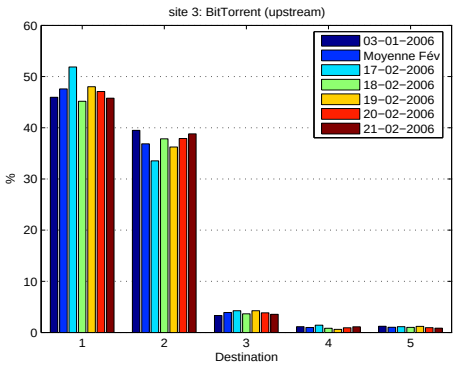


(a) sens montant

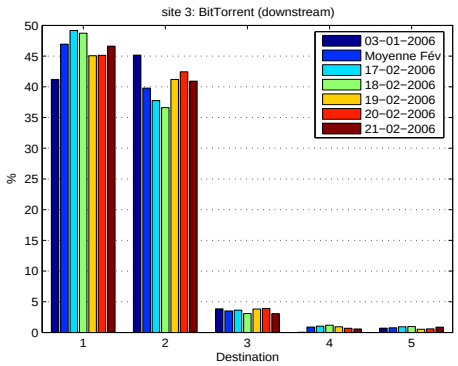


(b) sens descendant

FIG. 2.29 – Stabilité de la matric de trafic BitTorrent (site 6)

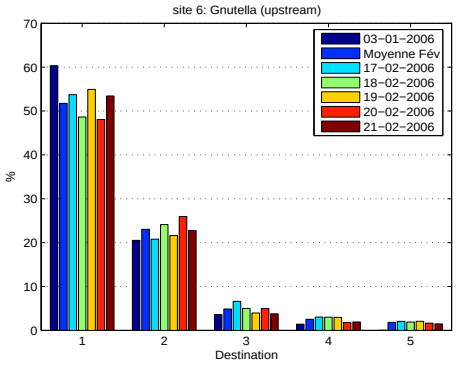


(a) sens montant

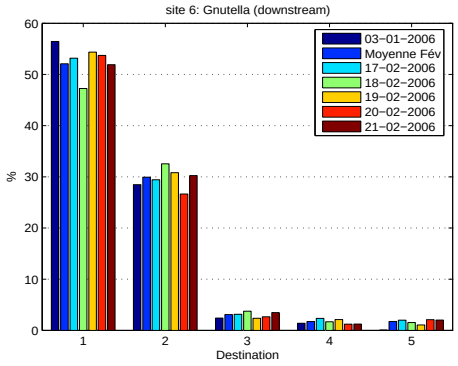


(b) sens descendant

FIG. 2.30 – Stabilité de la matric de trafic BitTorrent (site 3)

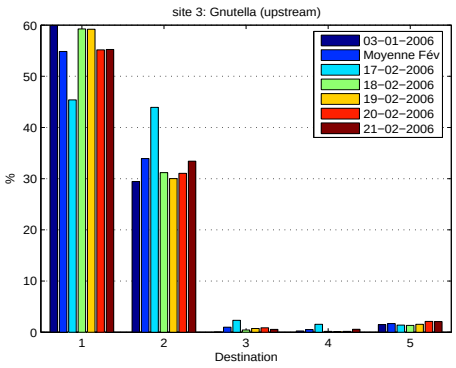


(a) sens montant

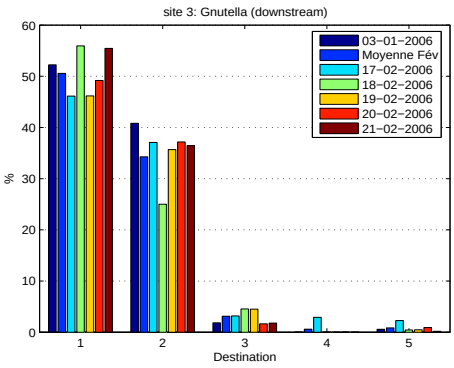


(b) sens descendant

FIG. 2.31 – Stabilité de la matric de trafic Gnutella (site 6)



(a) sens montant



(b) sens descendant

FIG. 2.32 – Stabilité de la matric de trafic Gnutella (site 3)

tination. Afin d'approfondir nos connaissances sur les répartitions géographiques de cette composante du trafic, nous nous intéressons dans cette section à l'étude du trafic au sein du réseau de transit IP de France Telecom.

Pour ce faire, nous utilisons les enregistrements Netflow remontés par quelques dizaines de routeurs du réseau OTIP. L'activation de la fonctionnalité Netflow sur un routeur permet d'obtenir des données décrivant le trafic qui passe par celui-ci. Ces données se limitent au niveau flot, qui comprend le quintuplet formé par l'adresse de source, l'adresse de destination, le numéro de port de source, le numéro de port de destination et enfin le protocole associé (TCP ou UDP). Les données Netflow ne contiennent aucune information sur la charge utile des paquets. Nous nous limitons alors à une classification des applications basée seulement sur les numéros de ports de source et de destination. Il est communément connu qu'une proportion de plus en plus conséquente du trafic eDonkey passe par des numéros de port non standards. Pour cerner le biais que peut introduire ce trafic eDonkey déguisé, nous analysons dans un premier temps la proportion du trafic eDonkey clair par rapport au volume total du trafic vu dans le réseau OTIP. Ceci permet de juger si cette composante claire est représentative de la totalité du trafic eDonkey.

Dans la figure 2.8, nous observons que la proportion du trafic eDonkey clair par rapport au volume total appartient à l'intervalle [25%, 40%] (31% en moyenne). Cette constatation nous permet de considérer la partie claire du trafic eDonkey comme représentative. En moyenne, près de la moitié du trafic eDonkey passe par les ports standard selon la sonde OTARIE. Nous retrouvons la partie déguisée du trafic eDonkey dans la partie labélisée 'Unknown'.

Dans les figures 2.34(a) et 2.34(b), nous donnons l'évolution sur une période de quatre jours des contributions des quatre applications majoritaires du trafic du réseau OTIP. Ces contributions sont données en termes de pourcentage et en termes de volume. Nous remarquons d'emblée le comportement journalier commun à l'ensemble des applications. Cependant, le contraste entre l'activité diurne et nocturne dépend énormément de l'application observée. Par exemple, pour une chute de 70% du trafic web entre le jour et la nuit, le trafic eDonkey ne chute que de 30%. Cette observation permet de mettre en relief l'une des caractéristiques du trafic eDonkey et du trafic p2p en général. Ce trafic ne dépend pas énormément du temps et affiche une stabilité temporelle entre le jour et la nuit par rapport au trafic web. En effet, les échanges de chunks dans le réseau eDonkey se font généralement d'une façon automati-

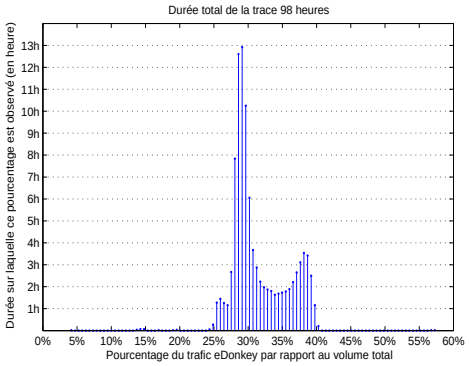
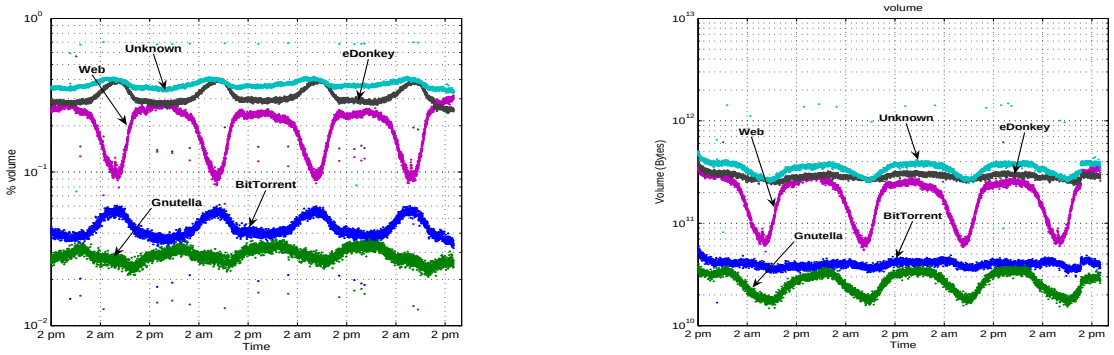


FIG. 2.33 – La proportion du trafic eDonkey clair sur une durée de 98 heures



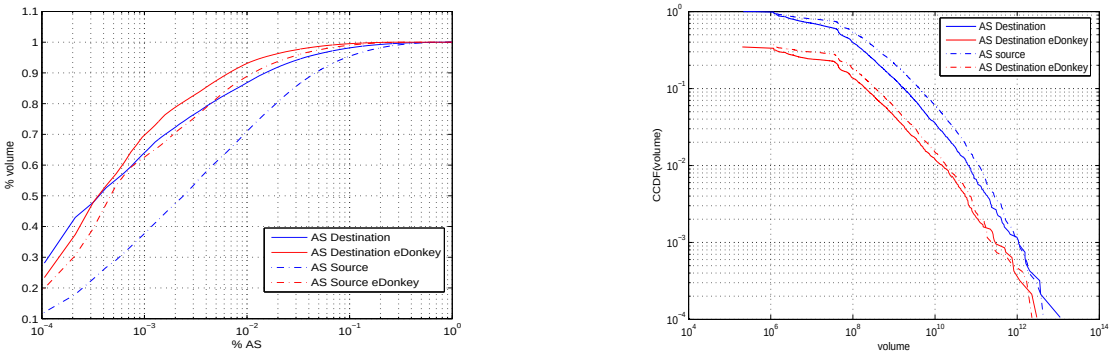
(a) Pourcentage

(b) Volume

FIG. 2.34 – Principales composantes du trafic dans le réseau OTIP

sée contrairement à la phase de recherche des fichiers à télécharger qui s'apparente à de la navigation web avec le modèle de transaction requête/réponse. Ceci confère au trafic de signalisation, pendant la phase de recherche, une certaine ressemblance avec le trafic web. Les deux types de trafic affichent alors une certaine dépendance au facteur humain. L'utilisateur est un facteur qui conditione le comportement du trafic web à travers une myriade d'éléments (préférence culturelle ou linguistique, heure d'activité humaine ou encore l'impatience). Ce parallélisme explique plusieurs observations sur le trafic web comme par exemple la chute de l'activité nocturne du web. Ce dernier phénomène devrait être moins intense dans un réseau d'une étendue internationale comme le réseau OTIP du fait que le décalage horaire permet de couvrir le manque d'activité nocturne de l'Europe par une activité diurne soutenue aux États-Unis et vice-versa. L'importante chute entre les volumes observés la nuit et le jour laisse présager que les plus gros clients de l'OTIP sont géographiquement localisés en Europe.

Nous nous intéressons maintenant à la manière avec laquelle les volumes s'écoulent au sein du réseau OTIP. Nous nous intéressons plus précisément à la matrice de trafic AS/AS perceptible à partir d'un réseau d'une étendue international comme le réseau OTIP. L'information sur les numéros de système autonome de source et de destination est fournie par les enregistrements Netflow. Notons que ces numéros désignent les systèmes autonomes de source et de destination auxquels appartiennent les deux adresses IP (de source et de destination) du flot. Ces adresses peuvent appartenir alors à des systèmes autonomes non adjacents au réseau OTIP.



(a) Répartition du volume sur les systèmes autonomes

(b) CCDF du volume par AS

FIG. 2.35 – Principales composantes du trafic dans le réseau OTIP

Dans une capture qui dure une heure, réalisée le 02 mars 2006 entre 15h et 16h, nous dénombrons 134.899 couples de systèmes autonomes (9542 AS de source contre 9440 AS de destination). Dans la figure 2.35(a), on remarque que la répartition du volume sur les systèmes autonomes n'est pas uniforme. Plus de 70% du volume sont issus ou partent vers environ 1% de la population des systèmes autonomes vus durant la capture.

L'examen approfondi de ces 1% de systèmes autonomes, résumé dans le tableau 2.3, montre que les principaux clients du réseau OTIP sont, à un premier niveau d'analyse, des systèmes autonomes européens. A un deuxième niveau d'analyse, nous remarquons que les systèmes autonomes de filiales de France Telecom (TPSA, UNI2, Wanadoo Pays Bas) sont bien présents dans la liste des dix premiers systèmes autonomes en termes de volume reçu ou envoyé (AS de source / AS de destination). Le RBCI est le premier client de l'OTIP en capitalisant 28% du

volume lorsqu'il est système autonome de destination et seulement 12% lorsqu'il est système autonome de source. Notons ici que le couple RBCI et TPSA reçoit près de la moitié du volume cumulé sur la période d'observation lorsqu'ils sont systèmes autonomes de destination. L'analyse des enregistrements Netflow issus de quelques dizaines de routeurs du réseau OTIP nous a permis de construire une matrice de trafic AS/AS. Les résultats que nous obtenons sur l'écoulement des volumes entre AS sont fiables malgré le biais que pourrait introduire l'échantillonnage pratiqué par Netflow. Cependant, des études antérieures [6] ont montré que ce biais est quasi inexistant lorsqu'on s'intéresse aux statistiques sur les volumes. En plus, l'échantillonnage vu d'une autre perspective, permet de fiabiliser les résultats sur les volumes cumulés. En effet, l'échantillonnage 1/N permet de réduire la probabilité de compter deux fois un même paquet d'un flot passant par deux routeurs.

La matrice de trafic obtenue dresse une image réelle de l'écoulement des volumes dans le réseau de transit international de France Telecom et ceci selon des critères de répartition géographique (source et destination). Nous arrivons à identifier à l'issue de cette étude les plus grands clients du réseau OTIP. L'analyse part sans aucune information sur les adjacences. En effet, le fait de considérer les champs AS de source ou de destination des enregistrements Netflow signifie que nous nous intéressons aux systèmes autonomes auxquels appartiennent les deux adresses IP et non aux systèmes adjacents qui délivre à l'OTIP les paquets issus des transactions entre ces deux adresses. Cependant, nous remarquons que l'étude de la matrice AS/AS revient en fin de compte à une étude de l'état du peering du réseau OTIP étant donné que les AS source et AS destination qui contribuent le plus en termes de volume sont des clients directs de ce réseau de transit. Par ailleurs, les termes clients et peering restent ici une notion vague car si la relation entre l'OTIP et un AS d'une filiale de France Telecom est de type fournisseur/client, la relation de l'OTIP avec les autres AS est à définir (exp. l'AS Proxad du FAI free).

	AS de source			AS de destination		
	Volume	eDonkey	Degré	Volume	eDonkey	Degré
1	RBCI	RBCI	RBCI	RBCI	RBCI	RBCI
2	TPSA	UNI2	TPSA	TPSA	TPSA	TPSA
3	UNI2	TPSA	UNI2	UNI2	UNI2	Woo PB
4	Telefonica Es	Telefonica Es	Woo PB	Telefonica Es	Telecom Italia	UNI2
5	Woo PB	Telecom Italia	0	NTL	ONO Cable	NTL
6	Telecom Italia	Auna Telecom	Mobistar	Proxad	Telefonica Data	0
7	AOL Transit	Woo PB	Sonatel	Telefonica Data	Auna Telecom	Proxad
8	Telecom Italia Fr	AOL Transit	IAM (Maroc)	Telecom Italia	DT	Mobistar
9	LLNW (CDN)	ASN Infostrada	33774	ONO Cable	AOL Transit	IAM Maroc
10	NTL	TELE2 AB	NTL	DT	Woo PB	33774

TAB. 2.3 – Le classement des AS de source et de destination selon le volume total, le volume eDonkey et le degré.

2.9 Conclusions et perspectives

- Le trafic eDonkey reste essentiellement en France (peering national et nœuds de collecte)
- L'existence d'une préférence linguistique et culturelle est en cohérence avec les observations faites sur le clustering sémantique dans d'autres travaux de recherche.

- La loi de Pareto est de mise lorsqu'on traite la répartition sur les pays et la répartition sur les points de sortie du réseau national de FT.
- Une proportion assez modeste du trafic eDonkey se propage entre les nœuds de collecte. La destination 'International' et la destination 'FAI-Nationaux' (l'ordre dépend du type de la population observée) absorbent la quasi totalité du trafic eDonkey. Pour le trafic web, la destination 'International' est largement majoritaire par rapport aux autres destinations.
- L'opérateur Free capitalise une importante proportion du trafic de peering. Le profil des abonnés de cet opérateur est plus ou moins responsable du fort pourcentage du trafic avec la destination 'FAI-Nationaux' (ce genre de constat rejoint très certainement la voie d'analyse sociologique).
- Les observations faites sur les différentes matrices de trafic sont indépendantes du lieu d'observation pourvu que l'on observe un même type d'utilisateurs.
- Cette invariance spatiale s'accompagne d'une stabilité temporelle à long et à court terme.
- Est-il possible de retrouver la philosophie pair à pair comme on l'appréhendait (topologie aléatoire et contribution de tous les pairs) dans le réseau de transit international de France Telecom ?
- La loi de Pareto est observable dans le réseau OTIP. En effet une faible proportion d'AS capitalise la quasi totalité du trafic qui traverse le réseau de transit de France Telecom.
- Les plus grands AS clients (de destination ou de source) de l'OTIP appartiennent à des opérateurs européens et sont en général des filiales de France Telecom.
- Finalement, notons que l'arrêt du serveur d'indexation principal du réseau eDonkey (Razorback) n'a eu aucun effet sur la matrice de trafic de cette application. je pensions que l'arrêt du serveur principal du réseau eDonkey allait purger les tables d'indexation de ce réseau (en cassant les milliers voir millions d'enregistrements associant tel contenu à telles sources) et que cette action de nettoyage aurait des impacts sur la matrice de trafic de l'eDonkey (par exemple, si les freenautes sont plus cités comme source de contenus, cette association prend fin avec l'arrêt du serveur d'indexation, et du coup on verra par exemple une augmentation du trafic Intra-RBCI). La matrice de trafic de l'eDonkey n'a subi aucun impact (de notre point d'observation, la cinématique du trafic eDonkey est restée la même et ceci en focalisant l'observation sur les destinations majoritaires, les parts ont été stables et n'ont pas changé malgré l'arrêt du serveur Razoback Le trafic pair à pair n'a pas vu sa contribution chuter après la fermeture de ce serveur).
  Le système eDonkey a une approche d'indexation relativement centralisée. Cet incident a montré que les serveurs eDonkey sont loin d'être le talon d'Achille de ce système. Cette robustesse s'explique par le fait que le système est entretenu par ses utilisateurs et n'importe quel utilisateurs peut installer son propre serveur. C'est cette répartition de responsabilité qui garantit la robustesse et la survie de ce système (cette responsabilité est paraît-il plus répartie que le trafic). Suite à l'arrêt de Razorback, le réseau Kad (l'approche DHT d'eDonkey basée sur le protocole Kademlia) a fait émerger. Aujoud'hui, pas moins de 95% des utilisateurs d'emule ont un client Kad actif. Des sondages sur des forums d'utilisateurs emule, estiment que 85% des utilisateurs emule, utilisent Kademlia.

Chapitre 3

La modélisation des réseaux p2p

Sommaire

3.1 Introduction

Nous présentons dans ce chapitre quelques approches de la littérature pour la modélisation des systèmes p2p. Nous présentons également la théorie du champ moyen à travers ses divers domaines d'application.

3.2 État de l'art de la modélisation des réseaux p2p

En 2000, Jordan Ritter, considéré comme le co-fondateur de Napster, esquissait dans un article informel quelques équations mathématiques dans le but de montrer l'incapacité du jeune réseau pair-à-pair Gnutella à passer à l'échelle [71]. Dans sa première version, le réseau Gnutella fonctionnait avec une architecture totalement décentralisée. L'indexation s'effectuait alors grâce à une inondation de requêtes vers l'ensemble des servents (une contraction de serveur et client) voisins. Ces voisins vont à leur tour relayer les requêtes vers d'autres servents. En comparaison avec une indexation centralisée, ce mécanisme d'indexation a tendance à surcharger le réseau. Cependant, les développeurs de Gnutella appréhendaient ce problème et ils ont instauré en conséquence un nombre de sauts maximal à partir duquel la requête n'est plus propagée. Cette restriction du champ de propagation implique une limitation de la capacité du réseau Gnutella à satisfaire les requêtes de ses utilisateurs. Néanmoins, le réseau Gnutella est un réseau plus robuste que le vulnérable Napster. En effet, le serveur central de ce système était un véritable talon d'Achille et son arrêt a fait disparaître le pionnier des réseaux pair à pair. La mort du système centralisé Napster a suscité un engouement parfois irrationnel pour les réseaux pair à pair décentralisés bien représentés à cette époque par Gnutella dans sa première version.

Dans ce contexte d'admiration totale pour les systèmes décentralisés, Jordan Ritter a écrit un article [71] où il exprime en termes d'équations mathématiques relativement simples la montée rapide des débits dès que le nombre d'utilisateurs augmente. Malgré ses hypothèses simplificatrices, le travail de Jordan Ritter, reste à notre connaissance, la première approche théorique dans l'étude des systèmes de partage de fichiers pair à pair.

Les premiers travaux académiques de modélisation se feront attendre et vont mettre quelques années avant d'émerger sporadiquement à partir de 2003, année de la publication d'un article sur la modélisation des réseaux pair à pair par l'équipe de Towsley et Kurose de l'université du Massachusetts [29]. Cependant, ce temps de latence ne signifie en aucun cas la stagnation de la recherche dans le domaine du pair à pair. Au contraire, ces années ont vu foisonner, mûrir et parfois mourir une myriade de mécanismes (méthode d'indexation, politique de sélection de pairs, découpage des fichiers) qui une fois injectés dans les systèmes pair à pair réels offraient des performances assez satisfaisantes.

Grâce à ces performances, les applications pair à pair ont connu ces dernières années une réussite indiscutable. Cette réussite a transformé les réseaux pair à pair en un véritable phénomène technique. Ce phénomène, qui affecte les réseaux de transmission modernes, a intéressé plusieurs chercheurs qui l'ont abordé sous plusieurs angles d'approche. Citons les travaux effectués sur la caractérisation des réseaux pair à pair ou ceux qui se penchent sur la mesure du trafic réel des systèmes pair à pair. Notons que les campagnes de mesures sont toujours d'actualité dans un contexte où les applications pair à pair capitalisent plus de la moitié de la bande passante dans les réseaux de cœur des plus grands opérateurs des télécommunications.

Mais cette gourmandise pour la bande passante n'est pas la seule caractéristique des systèmes pair à pair. En effet, l'observation de ces systèmes réels montre des systèmes d'une extrême complexité. Cette complexité est attendue étant donné que les systèmes en question se présentent comme une gigantesque nébuleuse constituée de pairs très hétérogènes et sollicitée par des humains aux comportements très divers. Afin de prédire le comportement de ces systèmes complexes, évaluer leur impact sur les réseaux des opérateurs ou encore évaluer les performances qu'ils offrent aux utilisateurs, leur modélisation est sans doute une tâche nécessaire dans le processus de la compréhension de ce phénomène.

3.2.1 Les files d'attente au service de la modélisation

Réseau fermé de files d'attente En 2003, les auteurs de [29] présentent dans leur article ce qui va être à notre connaissance la première tentative de modélisation des réseaux pair à pair. En effet, dans le but de trouver des résultats fondamentaux sur la performance des réseaux pair à pair, les auteurs partent des travaux antérieurs sur l'architecture client/serveur. Il s'inspirent alors des multiples modèles théoriques proposés dans ce domaine [55]. L'équipe de Towsley propose une plateforme théorique que ses concepteurs considèrent comme générique. Cette plateforme est basée sur le formalisme des réseaux de files d'attente fermés. Cette boîte à outils théorique permet de décrire le fonctionnement de base des réseaux pair à pair.

Ce fonctionnement de base englobe quatre états triviaux dans lesquels un pair donné peut se retrouver. Un utilisateur est initialement dans un état **déconnecté**. Après une certaine durée, cet utilisateur se connecte au réseau pair-à-pair. L'utilisateur passe alors dans un état d'**inactivité** (Think time), un état qui dépend énormément du comportement de l'utilisateur. L'utilisateur lance ensuite une requête pour un fichier donné passant ainsi vers un état dit de **services communs**. Cet état modélise la procédure de recherche de sources de contenu dans le réseau pair-à-pair.

Une hypothèse faite sur la procédure d'indexation stipule que le temps mis pour répondre à une requête est indépendant du type d'utilisateur ou du fichier. Cependant, la réponse à une requête peut être positive ou négative. Chaque fichier possède sa propre probabilité d'échec. L'échec d'une requête fait repasser l'utilisateur dans un état d'inactivité. Par ailleurs, une requête réussie fait entrer l'utilisateur dans un état de **téléchargement**. Chaque fichier a ses propres temps de téléchargement. Une fois le temps de téléchargement terminé, deux cas de figure sont possibles pour la suite. L'utilisateur peut éventuellement entamer un nouveau processus de téléchargement (état d'inactivité, services communs et puis téléchargement). L'utilisateur peut également choisir de passer dans l'état déconnecté.

Une fois l'espace d'états caractérisé, les paramètres de ce réseau de file d'attente sont évalués grâce à des hypothèses de fonctionnement du système pair à pair (indexation centralisée, loi de Zipf comme loi de popularité des fichiers). Le choix du mode d'indexation sert à modéliser l'état des services communs.

Pour paramétrer l'état d'inactivité, le raisonnement par rapport aux utilisateurs est indispensable. Le comportement de l'utilisateur sera déterminant pour le paramétrage de cet état. Un Free rider par exemple est assez souvent un utilisateur très pressé pour télécharger, il passe donc peu de temps dans l'état d'inactivité.

Finalement, les auteurs considèrent un réseau de files d'attente avec un nombre de classes de clients quelconque. La résolution est faite numériquement grâce à une approche analytique déjà existante (bottleneck analysis [24, 47]) et à l'utilisation d'une extension pour manipuler plusieurs classes d'utilisateurs.

Les auteurs évaluent le débit (client par seconde) en un point de référence dans le réseau fermé de file d'attente. Ce débit sert comme critère de comparaison entre les différentes méthodes d'indexation. Ce critère permet d'évaluer également l'impact des freeloaders sur le système. Cet impact se révèle minime. Cependant, ces résultats sont seulement vérifiés dans le cadre des hypothèses relativement restrictives des auteurs. En effet, ceux-ci supposent par exemple que le nombre de fichiers demeure constant malgré les téléchargements successifs. Cette hypothèse est loin d'être vraie dans les systèmes réels où la réplication du contenu demeure un critère déterminant pour les performances offertes par les réseaux pair à pair.

Nous pouvons citer dans ce cadre les travaux faits par Li Zou et Mostafa Ammar [90] qui raisonnent par rapport à un fichier donné et analysent le comportement de l'ensemble des pairs vis à vis du téléchargement de ce fichier. Ce raisonnement est bien adapté à la philosophie des réseaux coopératifs que représente BitTorrent. D'ailleurs, l'ensemble des travaux de recherche sur la modélisation adopte ce type de raisonnement et s'intéressent particulièrement aux réseaux similaires à BitTorrent. Cependant, cet acharnement sur ce réseau coopératif n'exclut pas une éventuelle utilisation des modèles BitTorrent pour la modélisation d'autres types de réseaux de partage de contenu (exemple eDonkey). En effet, il est judicieux, ne serait ce que dans une étape préliminaire de modélisation, de réduire la complexité d'un réseau comme eDonkey en supposant qu'un seul fichier est partagé sachant que même sous cette condition le système demeure d'une extrême complexité. D'ailleurs, Li Zou et Mostafa Ammar ont affaire à un modèle dont la résolution analytique est impossible lorsqu'ils essaient de modéliser le processus de propagation du fichier. En effet, leur démarche consiste à considérer une agrégation de pairs en interaction où chaque pair est représenté par une chaine de Markov détaillée. Ces chaînes de Markov avec un nombre d'états fini sont régies par un diagramme de transition. La formalisation du problème donne naissance à un système algébrique complexe que les méthodes d'analyse numérique existantes auront sans doute du mal à résoudre malgré le fait que la matrice résultante soit creuse (dans le cas d'un pair avec un espace d'états de taille 6, pour pouvoir résoudre le système, le nombre de clients ne doit pas dépasser 7 ! ! !).

L'approche proposée pour contourner cette complexité sera de mise pour l'ensemble des travaux de recherche qui vont suivre. L'esprit de cette approche consiste à raisonner par rapport au nombre de pairs qui se trouvent dans un état défini. Il est évident que cette approche rejoint celle proposée par l'équipe de Towsley. D'ailleurs, la résolution du système du modèle moins complexe obtenu confirme les résultats des chercheurs de l'université de Massachusetts sur l'impact des freeloaders sur le système.

Réseau ouvert de files d'attente Contrairement aux chercheurs de l'université de Massachusetts, des travaux plus récents ont utilisé le formalisme des réseaux ouverts de files d'attente. Citons dans ce cadre les travaux de Ramachandran et Sikdar [69] qui utilisent un réseau ouvert de files d'attente pour modéliser, dans un premier temps, le réseau de cœur formé d'un ensemble de routeurs interconnectés selon une topologie réelle ou simulée. Les auteurs évaluent le temps moyen pour le téléchargement d'un fichier.

Le formalisme des files d'attente est également utilisé dans un second temps pour modéliser les pairs (seulement dans leur rôle de serveur de contenu) pourvus de capacités de téléchargement et ayant des comportements (degré de coopération par exemple) différents. Le paramétrage du réseau ouvert de files d'attente modélisant les routeurs interconnectés est fait grâce à l'établissement d'un système linéaire (équation sur les taux d'arrivées au niveau de chaque routeur). Les taux d'arrivées obtenus apparaissent comme une fonction des données du

problème qui sont en l'occurrence les taux d'arrivée du trafic externe et la matrice de routage. Cependant, le paramétrage ne se limite pas à une résolution de système linéaire. Une étude est également menée sur la variabilité des solutions obtenues. Cette étude se base essentiellement sur des résultats issus de la théorie des files d'attente [85]. La modélisation débouche sur une évaluation de la latence globale du réseau de routeurs grâce à des résultats de W. Whitt [85] datant des années 80.

La latence du réseau de routeurs représente une partie de la durée totale de téléchargement d'un fichier. En effet, il faut ajouter deux éléments temporels très importants étant donné qu'ils sont proprement associés au réseau pair à pair étudié. Le premier élément est relatif à la durée moyenne de la procédure d'indexation. Le deuxième dépend du temps de téléchargement effectif du fichier en question. Il est évident que ce dernier délai est le critère qui conditionne le plus la "cinétique" globale du processus de téléchargement.

Pour évaluer ce délai, des résultats classiques issus des modèles PSQ (Processor Sharing Queue) [18] sont utilisés. L'analogie avec ce cas de figure est possible à partir du moment où un pair télécharge à partir de plusieurs serveurs de contenu (qui sont également des pairs). Les serveurs de contenu, capables de desservir simultanément m clients, sont modélisés par une file sans attente avec m serveurs. Pour conclure avec cette approche de modélisation, notons que le modèle s'intéresse particulièrement aux performances perçues par les pairs sans passer par une analyse de performance globale du système pair à pair et de sa stabilité ou de sa survie comme ce sera le cas dans les modèles fluides que nous présentons dans la suite de ce document.

3.2.2 Les modèles fluides

Si nous prenons à titre d'exemple le réseau eDonkey, les médias spécialisés (l'observatoire du pair à pair Slyck [79] par exemple) parlent de quelques millions d'utilisateurs connectés simultanément. Ces quelques millions d'utilisateurs ne sont pas perpétuellement connectés au réseau pair à pair : ils se connectent, interrompent leur téléchargement, se déconnectent, se reconnectent et choisissent de reprendre ou pas leur téléchargement. Cette volatilité caractérisant les systèmes pair à pair est une entrave sérieuse aux méthodes classiques d'analyse de performances tels que les modèles markoviens. Cette approche nécessite un large espace d'état qui devient assez rapidement très coûteux en termes de temps de résolution ou même de simulation.

Cependant, inspirés par la réussite des modèles fluides pour les réseaux de paquets, des chercheurs ont proposé des modèles pour les systèmes pair à pair dans le but de réduire l'espace d'états des systèmes. En 2004, X. Yang et Gustavo De Veciana de l'université du Texas, proposent un modèle fluide [88] dans le but d'évaluer les capacités du service pendant les régimes transitoire et permanent. La notion de capacité de service pendant le régime permanent rejoint les objectifs exprimés par Ramachandran et Sikdar [69] qui se soucient des performances perçues par un utilisateur. En revanche, la capacité du service en régime transitoire est un indice global du système évaluant ses capacités à absorber l'explosion des demandes lorsqu'il commence avec un nombre limité de serveurs.

Les auteurs utilisent les résultats classiques de la théorie des processus de branchement [32, 5]. L'évolution d'une population est régie par plusieurs paramètres comme le nombre de fils par nœud parent ou la distribution de la durée de génération des fils. Le formalisme des processus de branchement, très utilisé dans la biologie, s'avère applicable dans le cas des réseaux pair à pair. En effet, un serveur de contenu (seed dans le jargon BitTorrent)

dessert plusieurs clients simultanément (4 leechers pour BitTorrent). Ces clients deviennent des serveurs après un temps aléatoire (distribution des temps de téléchargement). Parmi les nouveaux serveurs qui apparaissent dans une génération donnée, une proportion décide de ne pas coopérer et ils deviennent en conséquence des serveurs stériles.

La modélisation permet de montrer que le nombre de serveurs augmente exponentiellement pendant le régime transitoire. Le parallélisme (upload vers clients multiples) ralentit cette évolution exponentielle étant donné que nous gardons la même capacité d'upload mais nous la partageons sur plusieurs clients. Toutefois, le parallélisme reste la meilleure garantie pour la survie du système pendant le régime transitoire quand une grande majorité de clients refuse de partager le contenu qu'ils ont téléchargé.

Quant au régime permanent, les auteurs adhèrent aux méthodes markoviennes. Pour ce faire, ils définissent comme état du système, le couple formé par, d'un coté, le nombre de clients qui sont en phase de téléchargement et de l'autre coté, le nombre de clients ayant fini de télécharger mais qui demeurent tout de même connectés au réseau pair à pair. Le système est alimenté par de nouveaux clients voulant télécharger le contenu en question. L'ensemble des clients qui téléchargent finiront tôt ou tard par récupérer la totalité du contenu et deviendront en conséquence des serveurs. En revanche, ces derniers peuvent à leur tour quitter le système ou ne plus partager le contenu. Ce mode de fonctionnement revient bien évidemment à un diagramme de transition d'une chaîne de Markov. Pour résoudre cette chaîne de Markov et déterminer sa distribution stationnaire, les auteurs tronquent d'une façon appropriée l'espace des états. Les résultats obtenus sont aussitôt validés par des données réelles récupérées sur un tracker BitTorrent.

Des chercheurs de l'université chinoise de Hong Kong vont dans le même sens et modélisent le système BitTorrent avec le même formalisme Markovien [81]. En revanche, ils choisissent de scinder l'étape dans laquelle le pair télécharge, en N étapes. Chaque étape intermédiaire caractérise le taux de progression dans le téléchargement du fichier (avec un pas d'avancement égal à $\frac{1}{N}$). A titre d'exemple, l'étape X_{N-1} indique le nombre de pairs qui téléchargent le dernier morceau. Grâce à l'utilisation de résultats sur les modèles de chaînes de Markov à temps continu [82], les auteurs arrivent à résoudre leur modèle. Cette résolution permet d'obtenir des résultats sur la répartition du taux de progression du téléchargement du fichier. D'après les résultats obtenus, la majorité des pairs se retrouvent assez souvent dans deux situations bloquantes. La première situation est celle de l'amorçage, la seconde situation est relative au dernier morceau. La distribution des pairs pendant le régime stationnaire est en forme de U montrant ainsi les limites des mécanismes BitTorrent mettant fin à une série d'articles élogieux en faveur du réseau BitTorrent. En effet, le système de coopération BitTorrent est assez souvent présenté comme le système optimal à travers des approches théoriques et des modèles mathématiques alimentant cette assertion [10, 68].

Les auteurs de [10] prouvent dans le cadre d'une étude comparative que la philosophie de réplication de BitTorrent permet d'offrir des performances nettement meilleures que l'ensemble des mécanismes proposés. Pour mener cette étude comparative, les auteurs fixent comme critère le temps nécessaire pour répliquer le contenu sur N pairs distincts.

Les auteurs de [68] se sont inspirés du modèle proposé par De Veciana [88] pour élaborer leur propre modèle fluide et pour montrer ainsi la "perfection" des mécanismes BitTorrent. Mais contrairement au modèle proposé par De Veciana, les auteurs de [68] contournent l'étude de la chaîne de Markov en utilisant un modèle déterministe. En effet, un bilan sur la dynamique de la population (clients et serveurs) permet d'établir deux équations différentielles ayant comme fonctions inconnues le nombre de clients et le nombre de serveurs. Une étude de la

stabilité permet de se prononcer sur l'existence d'un état d'équilibre. La stabilité dépend des contraintes que doit respecter le système. Dans ce cas de figure, les contraintes sur les débits montant et descendant vont faire basculer le système entre deux états. La méconnaissance de l'origine du goulot d'étranglement du système (le débit montant ou le débit descendant) est caractéristique des systèmes dits linéaires commutés (Switched Linear System) traités par Liberzon et Morse [51].

Les auteurs contournent ce cas de figure complexe en supposant que le débit descendant est infini et que le débit montant est tout le temps le goulot d'étranglement. Cette hypothèse permet de garantir la stabilité du système. Ce résultat leur permet d'écrire un système d'équations à deux inconnues après un passage à la limite du système d'équations différentielles. La résolution du système d'équations à deux inconnues permet de montrer la bonne adaptation à l'échelle des systèmes pair à pair. En effet, les expressions analytiques de ce modèle montrent que le temps de téléchargement d'un fichier est indépendant du taux d'arrivée des nouvelles requêtes (nouvelles demandes de téléchargement). Notons finalement que les auteurs étudient également le passage du modèle déterministe vers un modèle fluide stochastique. Pour ce faire, ils utilisent une approximation gaussienne pour estimer la variance autour des valeurs fournies par le modèle déterministe. En fait, le processus aléatoire est composé d'une partie déterministe et d'une partie aléatoire (processus d'Ornstein-Uhlenbeck) dont la covariance vérifie l'équation de Lyapunov [4].

D'autres travaux sur la modélisation des systèmes pair à pair utilisent l'approche théorique fluide. Les auteurs de [66, 62] tiennent compte de l'hétérogénéité des débits d'accès et analysent les performances du système sous ce critère. Le système d'équations différentielles de [68] devient un système de quatre inconnues. En effet, dans [66] le nombre de downloaders à l'instant t, noté $x(t)$, se scinde en deux composantes $x_l(t)$ et $x_h(t)$. La première composante représente le nombre de downloaders avec une grande bande passante. La deuxième représente le nombre de downloaders avec une faible bande passante.

Les auteurs de [62] intègrent un autre niveau d'hétérogénéité dû à l'existence de plusieurs classes de downloaders. Ces downloaders ont des capacités homogènes mais contribuent au système selon une politique d'allocation de ressource particulière. Les auteurs de [62] supposent que les downloaders d'une classe donnée répartissent leur capacité d'upload d'une façon non équitable sur les différentes classes. Dans [62], seulement deux classes sont considérées. En partant de ces hypothèses, les auteurs de cet article évaluent la durée moyenne de téléchargement pour chacune des classes pendant le régime stable. L'existence d'un tel régime dépend des conditions initiales. Le modèle développé dans [62] est un modèle fluide déterministe inspiré du modèle proposé par Srikant [68]. En revanche, les auteurs considèrent un cas relativement pessimiste. Les auteurs supposent que les pairs quittent le système après avoir terminé leur téléchargement. Sous toutes ces hypothèses, les auteurs analysent la stabilité du système et l'existence d'un point d'équilibre en utilisant la même démarche que dans [68].

En gardant le formalisme des modèles fluides, les auteurs de [50] s'inspirent des modèles de diffusion des épidémies pour modéliser la propagation d'un contenu dans un réseau pair à pair. Ce type de modèles est un outil de prédiction pour les biologistes et les aide à définir par exemple des stratégies de vaccination. Dans le domaine de la biologie, les modèles dits SIR sont généralement les plus utilisés. Le modèle SIR (pour *Susceptibles*, *Infectives* et *Removed*) tient son nom des trois états possibles d'un individu appartenant au système étudié. Par analogie aux systèmes biologiques, les auteurs de [50] utilisent cette notion pour les systèmes p2p.

A un instant t, les individus peuvent être inactifs (état I comme *Idle*) et se réveillent suivant une loi de poisson. Une fois actif, un pair commence à télécharger le contenu, il est donc dans un

état de téléchargement (état D comme *Downloading*). Ce pair finit de télécharger le contenu et devient alors un pair qui ne fait que partager le contenu (état S comme *Sharing*). Nous parlons alors d'un modèle *IDS*. Théoriquement, cette approche est analogue aux études des modèles fluides mais contrairement à celles là, les auteurs de [50] s'intéressent à la phase transitoire du processus de la diffusion du contenu. L'intérêt est essentiellement porté sur l'impact de la probabilité p qui représentant le pourcentage des pairs qui choisissent de partager le contenu. Ces pairs ne sont pas exactement des freeriders car, même s'il ne partage pas le contenu, un pair repasse forcément par l'état I en attendant de lancer une nouvelle requête pour un contenu. En effet, le processus de diffusion se fait en boucle jusqu'à ce que tous les pairs deviennent des seeders (l'état S de sharing). Les auteurs observent que le nombre de pairs dans l'état I décroît exponentiellement même si la probabilité p est inférieure à 1. Une probabilité p égale à 1 garantit une diffusion rapide du contenu.

Les auteurs de [38] usent du formalisme des processus de branchement. Ce formalisme, issu également du domaine de la biologie, a servi pour modéliser les deux étapes de downloading et de sharing du modèle *IDS* (*Idle*, *Downloading* et *Sharing*). Les processus de branchement a déjà été utilisé par [32, 5] pour modéliser le réseau BitTorrent. Contrairement aux travaux [32, 5], les auteurs de [38] se sont concentré sur l'impact de la probabilité de partage comme dans [50]. En effet, les auteurs de [38] ont pris en considération des arbres binaires Markoviens (à un instant donné, un client partageant un contenu ne sert qu'un seul client) et ont exprimé la probabilité d'extinction du système en fonction de la probabilité de partage p_{share}. Rappelons que la probabilité d'extinction d'un processus de branchement est la probabilité qu'un instant n existe tel que $Z_n = 0$ où Z_t représente le cardinale de la population à l'instant t. Dans [38], les branches de l'arbre binaire grandissent selon un processus de type phase.

Pour conclure cette partie, nous citerons finalement les travaux de [28, 49]. Imprégnés par la culture des graphes du web, les auteurs de [28, 49] utilisent la théorie des mariages stable pour modéliser les systèmes coopératifs. Cette théorie a été introduite depuis les années soixante par Gale et Shapley. Le problème consiste à considérer deux groupes hétérosexuels (un groupe de femmes et un groupe d'hommes) et chercher à trouver une configuration stable. Une configuration stable signifie que tous les membres des couples formés sont satisfaits de leur partenaire et ne veulent pas le changer. Gale et Shapley ont montré que ce problème admet au moins une configuration stable. Le problème des Roommates est un problème analogue, dans le sens ou tout le monde cherche à trouver le meilleur camarade (donc partenaire). En d'autres termes, les deux groupes (hommes et femmes) sont remplacés par un seul groupe formé d'un ensemble de pairs. Contrairement au problème des mariages, le problème des Roommates n'admet pas nécessairement de configuration stable. L'algorithme de Irving permet de dire si la solution stable existe ou pas. En outre, l'algorithme fournit cette configuration stable. Une généralisation du problème des Roommates consiste à supposer qu'un pair donné peut avoir simultanément b partenaires. Ce type de problème est dit le b-Matching. Le nombre de collaborations simultanées constitue le premier des trois paramètres les plus importants d'un problème de b-matching. Les deux autres paramètres sont le graphe d'acceptation et le système de notation des pairs. Intuitivement, nous comprenons que le système de notation représente la base de la notion de préférence. En effet, un pair donné va préférer les pairs les mieux notés. Dans le cas d'un réseau pair à pair par exemple, les pairs préfèrent collaborer avec des pairs dotés d'une grande capacité d'upload. La capacité d'upload détermine alors le système de notation. Le troisième paramètre est le graphe d'acceptation. Pour un pair donné, ce graphe détermine la liste des pairs avec lesquels ce pair est prêt à collaborer. Les auteurs de [28, 49] montrent l'existence d'une configuration stable dans le cas où le graphe des initiatives

est acyclique. Le graphe des initiatives représente l'ensemble des tentatives d'un pair pour se mettre avec un meilleur partenaire (rupture de liens et formation d'autres). La convergence vers la configuration stable est alors garantie si cette suite de tentative ne ramène pas le système à la configuration initiale. Les auteurs analysent également la vitesse de convergence mais mettent en évidence aussi deux phénomènes intéressants : la clusterisation et la stratification. Le phénomène de clusterisation se traduit par l'apparition de cliques dans le graphe de la configuration stable (ce graphe est dit graphe de collaboration et ce n'est qu'un sous graphe du graphe d'acceptance). Le deuxième phénomène est la stratification. Ce phénomène désigne l'apparition de communautés possédant les mêmes capacités d'upload. Les auteurs montrent que le premier phénomène tend à disparaître dés que le nombre de connexions possibles par peer devient légèrement variable d'un pair à un autre (par exemple une distribution gaussienne de la variable b de moyenne b_0 et d'écart type σ). En effet, en augmentant σ, on décèle un changement de comportement de la taille moyenne de cliques : cette taille moyenne devient subitement très élevée. En revanche, le phénomène de stratification persiste et la disparité entre les pairs d'une même clique demeure négligeable.

3.2.3 Des équations pour le pair à pair

Le modèle fluide A un instant t, le système pair à pair est caractérisé par deux paramètres. Le premier, noté $x(t)$, représente le nombre de clients qui sont en phase de téléchargement d'un fichier. Le deuxième paramètre est le nombre de seeders noté $y(t)$. Les clients arrivent dans le système avec un taux λ et quittent le système avant de finir le téléchargement avec un taux θ. Même si un client peut partager les morceaux de fichier qu'il a pu télécharger, on suppose que seulement une partie η de l'ensemble des clients accepte de coopérer. Les seeders quittent le système avec un taux de départ γ. Finalement, on note μ (resp. c) le débit montant (resp. descendant) des pairs. Nous supposons que les pairs ont les mêmes débits. Grâce à un bilan sur les deux populations, nous pouvons définir ainsi deux équations différentielles :

$$\frac{dx}{dt} = \lambda - \theta x(t) - min\{cx(t), \mu(\eta x(t) + y(t))\}$$

$$\frac{dy}{dt} = min\{cx(t), \mu(\eta x(t) + y(t))\} - \gamma y(t)$$

Pendant le régime permanent, on suppose que $x(t)$ et $y(t)$ se stabilisent autour d'une valeur constante, en conséquence on a :

$$\frac{dx}{dt} = \frac{dy}{dt} = 0$$

Les deux équations différentielles s'écrivent alors comme suit :

$$0 = \lambda - \theta x(t) - min\{cx(t), \mu(\eta x(t) + y(t))\}$$
$$0 = min\{cx(t), \mu(\eta x(t) + y(t))\} - \gamma y(t)$$

La résolution de ce système d'équations donnent les deux solutions suivantes :

$$\bar{x} = \frac{\lambda}{\beta(1 + \frac{\theta}{\beta})}$$

$$\bar{y} = \frac{\lambda}{\gamma(1 + \frac{\theta}{\beta})}$$

où $\frac{1}{\beta} = max\{\frac{1}{c}, \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})\}$
Grâce à la loi de Little, on détermine le temps moyen de téléchargement T :

$$T = \frac{1}{\beta + \theta}$$

Pour passer du modèle déterministe au modèle stochastique, on suppose que les solutions aléatoires varient autour des solutions déterministes de la façon suivante :

$$\begin{pmatrix} x_s(t) \\ y_s(t) \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + \sqrt{\lambda} \begin{pmatrix} \hat{x}(t) \\ \hat{y}(t) \end{pmatrix}$$

Où $\hat{\mathbf{X}} = \begin{pmatrix} \hat{x}(t) \\ \hat{y}(t) \end{pmatrix}$ est un processus de Ornstein-Uhlenbeck et qui est la solution de l'équation différentielle stochastique suivante :

$$\frac{d\hat{\mathbf{X}}}{dt} = \mathbf{A}\hat{\mathbf{X}}(t) + \mathbf{B}\frac{d\hat{\mathbf{W}}}{dt}$$

où les composants de $\mathbf{W}$ sont des mouvements Browniens et les matrices $\mathbf{A}$ et $\mathbf{B}$ sont des matrices conditionnées par les paramètres du système étudié. L'approximation est Gaussienne et $\hat{\mathbf{X}}_s$ est alors un vecteur Gaussien dont la variance pendant le régime permanent est la solution, notée Σ, de l'équation de Lyapunov :

$$\mathbf{A}\Sigma + \Sigma\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = 0$$

Les auteurs de [22] s'inspirent de ce modèle pour étudier les performances d'une solution de distribution de contenu basée sur la philosophie BitTorrent. Pour ce faire, un nouveau type de pairs est intégré dans le réseau. Ces pairs représentent le régime minimal de distribution de contenu. En effet, un distributeur de contenu intègre des pairs permanents dotés d'une grande capacité d'upload afin d'assurer le service même dans les scénarios les moins optimistes.

Les auteurs de [34], partent de la même formulation mais intègrent une nouvelle hypothèse sur les arrivées poissonniennes. En effet, les mesures empiriques de [67] sur le réseau BitTorrent montrent qu'au cours du temps, un fichier perd de sa popularité. En conséquence, le taux de requêtes pour un fichier donné fléchit au cours du temps. Les auteurs de [34] partent de ces observations et supposent que le taux d'arrivées des clients dans un système pair à pair décroît exponentiellement :

$$\lambda = \lambda_0 e^{-\frac{t}{\tau}}$$

avec τ le paramètre d'atténuation que les auteurs de [34] déterminent empiriquement à partir de leurs traces.

Le modèle des files d'attente Le modèle des files d'attente proposé par l'équipe de Towsley sert essentiellement à comparer plusieurs politiques d'indexation utilisées dans les systèmes pair à pair (indexation centralisée, indexation distribuée avec inondation,indexation distribuée avec une table de hachage). Pour ce faire, les auteurs de [29] proposent un réseau fermé de files d'attente dont les paramètres dépendent des hypothèses faites sur le mode de fonctionnement.

Le système contient M fichiers différents et les clients appartiennent à C classes distinctes. Au début, le client est dans une phase dite *offline*. Pour un client appartenant à une classe c,

cette phase dure en moyenne $\frac{1}{\lambda_{off}^{(c)}}$. Le client passe ensuite dans une phase qu'on appelle phase de *cogitation* et qui dure en moyenne $\frac{1}{\lambda_{idle}^{(c)}}$ pour un client de la classe c. Une fois cette phase terminée, le client passe dans la phase *recherche*. La durée moyenne de cette phase dépend du nombre de clients en ligne, noté $N_a = (N_a^{(1)}, \ldots, N_a^{(C)})$. Cette durée moyenne est notée μ_{N_a} et s'exprime différemment en fonction de la politique d'indexation adoptée. Dans le cas d'une indexation centralisée, μ_{N_a} est égale à une constante définie par la capacité du serveur central d'indexation. Pour une indexation décentralisée basée sur l'inondation, l'expression de μ_{N_a} est établie en fonction de la connectivité β des pairs et du TTL, noté τ. Pour une architecture basée sur les DHT (Distributed Hash Table), la durée moyenne est réduite par un facteur d'échelle par rapport à la durée moyenne calculée pour l'indexation par inondation. Ce facteur d'échelle est égal à $\log(N_a)$.

Notons que la durée moyenne μ_{N_a} est indépendante de la popularité du fichier. En revanche, le résultat de cette recherche dépend de cette popularité. En effet, une requête faite pour un fichier très populaire a plus de chance d'être satisfaite qu'une requête pour un fichier qui l'est moins. Les auteurs définissent en conséquence la probabilité $q_f(N_a, i)$ qui représente la probabilité d'échec des requêtes pour le ième fichier le plus populaire (le fichier $i = 1$ représente le fichier le plus populaire). En plus, la popularité d'un fichier détermine le nombre de requêtes qui lui sont destinées. Les auteurs définissent alors la probabilité de lancer une requête pour un fichier d'indice i. Cette probabilité est notée p_i. Afin d'évaluer cette probabilité, les auteurs font appel à la loi de Zipf qui est une loi souvent utilisée pour modéliser la popularité des contenus. Cette probabilité est égale à $\frac{K}{i^\alpha}$ où K est un paramètre de normalisation ($K = \frac{1}{\sum_{j=1}^{M} \frac{K}{i^\alpha}}$) et α est un facteur d'échelle de la loi de Zipf. Si la requête pour un fichier échoue, le client concerné par cet échec repasse de nouveau dans un état de cogitation. Par contre, si la requête réussit, le client entame la phase de téléchargement. Le téléchargement d'un fichier d'indice i dure en moyenne $\mu_f(N_a, i)$. Ce temps de service moyen reste constant au cours du temps et les auteurs ne tiennent pas compte de la réplication du fichier. En revanche, les auteurs expriment le temps moyen de téléchargement en fonction de la popularité du fichier. Par exemple, dans le cas d'une seule classe, le paramètre $\mu_f(N_a, i)$ s'écrit alors :

$$\mu_f(N_a, i) = \frac{N_a^{(1)} \times H \times K}{i^\alpha}$$

où N_a représente le nombre de clients en ligne et H la contribution d'un seul pair au système.

Une fois le téléchargement fini, le client passe dans la phase *offline* avec une probabilité $p_{off}^{(i)}$ où i est la classe à laquelle appartient le client. Par ailleurs, ce client peut revenir avec une probabilité $1 - p_{off}^{(i)}$ à la phase cogitation pour lancer une nouvelle requête. L'importance de la classe est indiscutable quand il s'agit de distinguer entre la classe des "freeriders" et la classe des "non freeriders". Les membres de la première classe sont plus agressifs en termes de nombre de requêtes. En plus , un freerider télécharge intensément sans partager de contenu. Il passe peu de temps dans la phase de cogitation mais passe plus de temps dans la phase *offline*.

Pour évaluer les performances des systèmes pair à pair, les auteurs se placent en un point de référence situé à la sortie des files d'attente qui modélisent la phase de téléchargement. Ils estiment alors le nombre de téléchargements réussis par unité de temps pour une classe donnée. Cette grandeur est majorée par :

$$T^{(c)} \leq \begin{cases} \frac{N^{(c)}}{\sum_{k \in Q} D_k^{(c)}}, & \text{sans congestion ;} \\ \frac{N^{(c)}}{\sum_{k \in Q} D_k^{(c)} + V_\beta^{(c)} W_\beta}, & \text{avec congestion.} \end{cases}$$

où $D_k^{(c)}$ est le produit du taux de visite de la file k, noté V_k, et du temps moyen de service dans cette file. Par ailleurs, W_β représente le temps d'attente moyen dans la file d'engorgement (bottleneck) β. Ce critère de performance est exprimé en fonction des paramètres du système présentés plus haut. Un autre critère, noté $R_{time}^{(c)}$, relatif au temps moyen nécessaire pour faire une requête et télécharger le fichier correspondant.

3.2.4 Des équations pour des problèmes analogues

Une approche théorique pour l'étude de la pollution La pollution consiste à injecter dans un système pair à pair une ou plusieurs versions corrompues d'un fichier donné. A cause du manque de la vigilance des utilisateurs, la version corrompue se propage. Afin de comprendre le mécanisme de prolifération de cette copie corrompue, plusieurs études théoriques ont été menées. Les auteurs de [46] élaborent une série de modèles fluides pour étudier la prolifération de la pollution dans les systèmes pair à pair.

Soit $\nu(t)$ l'ensemble des versions (bonnes et mauvaises) présentes à un instant t. Le nombre de copies d'une version $v \in \nu(t)$ est $n_v(t)$. La probabilité de choisir une version donnée dépend du nombre de copies de chaque version. Pour modéliser cette probabilité deux approches sont possibles. La première met en avant le nombre de copies pour chaque version. La deuxième ne tient pas compte du nombre de copie. La deuxième approche consiste à choisir de façon uniforme parmi celles existantes à l'instant t.

A l'état initiale, on suppose que les pollueurs introduisent volontairement N versions polluées. Ces versions polluées vont rester dans le système. On note par $x(t)$ le nombre de bonnes copies à l'instant t et par $y(t)$ le nombre de mauvaises copies à l'instant t. Il y a M pairs qui veulent télécharger le fichier. A un instant t, il y a $M - x(t)$ pairs qui veulent télécharger le fichier. La probabilité de choisir une version polluée est la probabilité $p(t)$ définie selon la première approche (Copy Centric Model) par :

$$p(t) = \frac{y(t) + N}{x(t) + y(t) + N}$$

La résolution de ce système passe par une approximation fluide à cause de la complexité de la résolution du modèle Markovien. Le nombre de bonnes copies augmente dans deux situations. La première survient quand un pair qui n'a téléchargé aucune copie télécharge une bonne copie. Le nombre de bonnes copies peut également augmenter quand un pair avec une mauvaise copie télécharge une bonne copie. La formalisation de ces deux situations donne l'équation différentielle suivante :

$$\frac{dx}{dt} = [M - x(t) - y(t)]\mu(1 - p(t)) + y(t)\mu(1 - p(t))$$

De la même manière, on écrit l'équation différentielle qui régit le nombre de mauvaises copies :

$$\frac{dy}{dt} = [M - x(t) - y(t)]\mu p(t) - y(t)\mu(1 - p(t))$$

Les solutions de ce système d'équations différentielles dépendent des conditions initiales. Le taux de pollution initial est déterminant dans le comportement du système. Le délai d'épuration du système dans le cas où l'attaque s'arrête est également étudié.

Le deuxième cas étudié est celui du VCM (Version Centric Model) où l'utilisateur choisit la version à télécharger indépendamment de sa popularité. Les auteurs supposent l'existence d'un nombre constant g de bonnes versions. Par ailleurs, le nombre de versions polluées dépend du temps. A un instant t, il est égal à $w(t)$. La probabilité $p(t)$ s'écrit alors ainsi :

$$p(t) = \frac{w(t)}{g + w(t)}$$

Nous obtenons alors le système à 3 inconnues suivant :

$$\frac{dx}{dt} = [M - x(t)]\mu\frac{g}{g + w(t)}$$

$$\frac{dy}{dt} = [M - x(t)]\mu\frac{w(t)}{g + w(t)} - y(t)\mu$$

$$\frac{dw}{dt} = [M - x(t)]\mu\frac{w(0)}{g + w(t)}$$

Grâce à la relation affine entre $x(t)$ et $w(t)$, le système d'équation différentielle se réduit à une équation différentielle linéaire usuelle. L'hypothèse d'un temps de téléchargement nul est loin d'être vrai dans les systèmes réels malgré la montée rapide des débits d'accès. Les auteurs tiennent compte de cette réalité et enrichissent leur modèle fluide en introduisant une nouvelle phase de téléchargement dans leur diagramme de transition. Les débits d'accès n'ont aucune influence sur l'évolution du taux de pollution.

Les systèmes de caches pair à pair Les systèmes de caches pair à pair font partie de la famille des systèmes de caches distribués. Les utilisateurs des systèmes de caches pair à pair partagent leur cache pour former ainsi un grand système coopératif. Le système Squirrel [40] est un exemple de cette approche. Les auteurs de [19] proposent un modèle fluide pour ce type de systèmes. La quantité fluide $X(t)$ représente maintenant la quantité des objets présents dans le système de caches.Les pairs arrivent dans le système avec un taux λ et partent avec un taux μ. Les auteurs supposent que la population des clients est fini (N clients). Le nombre de clients à l'instant t, noté $N(t)$ ($N(t) \in \{0, 1, 2, \ldots, N\}$) est alors un processus de naissance et de mort. Entre deux événements (connexion d'un client ou déconnexion) qui surviennent à deux instants T_n et T_{n+1}, on peut écrire l'équation différentielle suivante :

$$\frac{dX(t)}{dt} = \sigma N_n(1 - \frac{X(t)}{c}) - \theta X(t)$$

où N_n représente le nombre de clients connectés juste après l'événement survenu à l'instant T_n ($N_n = N(T_n+)$). Le taux de requêtes est noté σ. Le terme $\frac{X(t)}{c}$ est une estimation de la probabilité de la satisfaction d'une requête (c est le nombre total d'objets existants dans l'univers). En conséquence, le terme $1 - \frac{X(t)}{c}$ représente la probabilité d'échec d'une requête en supposant que la popularité est uniforme. Ce cas de figure signifie que les objets requis n'existent pas dans le système et doivent être téléchargés. Cette situation implique une augmentation de la quantité du fluide. Cette quantité de fluide peut également diminuer à cause

de la durée de vie des documents. En effet, les pages web dans les caches ont une durée de validité assez limitée car les serveurs web peuvent changer assez souvent le contenu de ces pages. Un objet dans le cache a une durée de vie moyenne égale à $\frac{1}{\theta}$ (TTL, *Time to Live*) .

3.3 Un modèle pour la formation d'un réseau eDonkey

Les différentes campagnes de mesures que nous avons faites pour étudier le trafic eDonkey ont révélées une forte activité de signalisation. Cette activité de signalisation met en évidence la gigantesque taille de la communauté eDonkey. En effet, la signalisation implique un très grand nombre de pairs. En revanche, seulement un pourcentage réduit de cette communauté eDonkey participe à de réels transferts de données.

Dans cette section, nous proposons un premier modèle mathématique qui montre la manière avec laquelle le réseau eDonkey se construit. Ce modèle mathématique arrive à expliquer la forte activité de signalisation que nous observons dans le réseau eDonkey réel. Par ailleurs, le premier objectif de la modélisation est de comprendre l'évolution de la formation du réseau à partir du moment où nous mettons en partage un objet donné (fichier ou chunk). Contrairement à la plupart des travaux de recherche, nous nous concentrons sur le régime transitoire. En effet, les travaux [29] et [68] s'intéressent au régime stable des réseaux pair à pair.

3.3.1 L'expansion du réseau sans freeriders

Soit Soit donc un objet donné que nous mettons en partage dans une communauté composée de N pairs. Dans cette section, nous traitons le cas d'un réseau sans freeriders. Nous supposons alors que tous les pairs acceptent de mettre en partage l'objet immédiatement après l'avoir téléchargé. En outre, nous avons quelques hypothèses supplémentaires sur le comportement du système et des pairs :

1. Les pairs s'enregistrent dans les serveurs d'indexation du réseau eDonkey comme étant serveur de l'objet en question immédiatement après la fin de son téléchargement,
2. Lorsque n pairs font désormais partie du système étudié, l'arrivée des requêtes des pairs restants ($(N-n)$ pairs) est un processus de poisson avec un taux $(N-n)\lambda$ où λ est un réel positif,
3. La durée de téléchargement de l'objet à partir d'un pair suit une loi exponentielle de paramètre μ où μ est un réel positif. L'objet a bien sûr une taille déterminée mais le temps de téléchargement peut être aléatoire et change d'un pair à l'autre en fonction des PCs des pairs, du nombre d'uploads en parallèle, etc., et surtout des capacités de transmission du lien montant (dépendant du type d'abonnement et de raccordement, ADSL, FTTH, etc.). Pour simplifier, nous supposons que le temps d'upload est exponentiellement distribué même si une loi plus concentrée autour de la moyenne serait plus adéquate mais bien plus difficile à analyser théoriquement,
4. Un pair qui joue le rôle de serveur partage équitablement sa capacité d'upload entre les pairs qu'ils l'ont choisi. La politique de service est alors une politique à temps partagé (Processor Sharing),
5. Un pair qui veut télécharger l'objet choisit aléatoirement un et un seul serveur.

En plus de ces hypothèses, nous supposons également que les pairs qui sont devenus serveurs utilisent une capacité de transmission constante sans interruption, par exemple pour

servir un autre objet dans une autre communauté. En d'autres termes, nous supposons l'existence d'un seul objet en partage et que la communauté eDonkey observée n'interagit avec aucune autre communauté qui partage un autre objet. Ceci nous permet de considérer la communauté autour d'un objet en isolation.

Tenant compte de l'ensemble de ces hypothèses, nous étudions dans la suite la formation du réseau eDonkey. Pour ce faire, nous décrivons l'état du système étudié par les trois variables suivantes :

N_s : le nombre de serveurs c'est à dire le nombre de pairs qui sont en possession de l'objet en question,

N_q : le nombre de pairs qui téléchargent l'objet à partir d'un serveur,

N_a : le nombre de serveurs qui servent au moins un client.

Notons ici que le nombre de pairs dans le système est exactement égal à N_s+N_q-1. Comme cela a été dit plus haut, l'état du système est décrit par les trois variables (N_s, N_a, N_q). Ces variables sont mises à jour avec les événement qui affectent le système étudié. Ces événement sont au nombre de deux : le premier est l'arrivée d'un nouveau pair dans le système et le deuxième correspond à la fin d'un téléchargement. Le premier événement se produit avec une probabilité p dont l'expression est la suivante :

$$p = \frac{(N - (N_s + N_q - 1))\lambda}{(N - (N_s + N_q - 1))\lambda + N_a\mu}$$

Lors de son arrivée dans le système, un pair choisit aléatoirement un serveur parmi les N_s existants (un serveur est choisi avec une probabilité $1/N_s$). Le deuxième événement correspond à la naissance d'un nouveau serveur d'objet. Cet événement se produit alors avec une probabilité $1 - p$.

Nous observons l'état de ce système aux instants où ces deux événements se produisent. Le système est alors à temps discret. Pour décrire l'état de ce système, nous notons respectivement par $N_s(t)$, $N_q(t)$, $N_a(t)$ les valeurs prises par N_s, N_a et N_q aux instants $t = 1, 2, 3, \ldots$. Le processus $(N_s(t), N_q(t), N_a(t))$ évolue alors de la manière suivante : à l'instant t et à partir d'un état (i, j, k) le système passe, à l'instant $t + 1$ à un état :

- $(i + 1, j - 1, k)$ avec une probabilité $(1 - p(t))$ multipliée par la probabilité que le pair devenu serveur, laisse derrière lui au moins un client dans le serveur,
- $(i + 1, j - 1, k - 1)$ avec une probabilité $(1 - p(t))$ multipliée par la probabilité que le pair devenu serveur, ne laisse derrière lui aucun client dans le serveur,
- $(i, j + 1, k)$ avec une probabilité $p(t)k/i$ (le nouveau pair rejoint un serveur avec des clients en téléchargement),
- $(i, j + 1, k + 1)$ avec une probabilité $p(t)(1 - k/i)$ (le nouveau pair rejoint un serveur avec aucun client en téléchargement),

où :

$$p(t) = \frac{(N - (N_s(t) + N_q(t) - 1))\rho}{(N - (N_s(t) + N_q(t) - 1))\rho + N_a(t))}$$

avec $\rho = \lambda/\mu$ qui représente la charge de la file PS avec un taux d'arrivée égal à λ et un taux de service égal à μ. Notons que le processus $(N_s(t), N_q(t), N_a(t), t = 1, \ldots, N)$ est non markovien. En effet, nous notons que les transitions dépendent du nombre de clients rattachés aux serveurs.

Afin d'évaluer les performances du système étudié, nous nous intéressons aux valeurs prises par les N_s, N_a et N_q quand l'ensemble des N pairs ont rejoint le système. L'aptitude du système

à disséminer l'objet assez rapidement peut être évaluée par l'ordre de grandeur des paramètres N_s et N_a. En effet, le fait d'avoir un nombre de serveur N_s de même ordre que N indique que l'objet s'est rapidement propagé parmi les pairs et que la majorité de ces pairs joue le rôle de serveurs. Dans ce cas, le nombre de serveurs N_a non libres est très petit.

En revanche, les performances du système sont jugées comme étant faibles lorsque le nombre de serveur N_s est petit et que le nombre N_q (pairs qui sont en train de télécharger l'objet) est de même grandeur que N. Dans ce cas, un nombre réduit de pairs partagent l'objet. Ceci produit une dissémination lente de l'objet dans le système. En d'autres termes, le système se retrouve dans un état de congestion où les pairs attendent une longue durée avant de voir leur téléchargement fini.

Cependant, la congestion ne dure pas éternellement car les pairs, dont le nombre est fini, finissent par récupérer l'objet en question. Au moment de la formation du réseau, nous observons un phénomène d'agglomération sur un nombre réduit de serveurs (formation d'un cœur). En revanche, aucune congestion n'est observée quand $N_a \ll N$ et nous observons un phénomène d'expansion du réseau.

Dans la figure 3.1 obtenue par simulation, nous représentons l'évolution de la fraction du nombre de clients qui n'ont pas encore fini leur téléchargement en fonction de la charge ρ. Rappelons que cette fraction est évaluée au moment de l'arrivée du dernier des N clients. Nous remarquons alors que cette fraction tend vers 1 quand la charge ρ tend vers 1. Lorsque le nombre de pairs N_q est de même ordre que N, nous pouvons affirmer que le système est congestionné et que ses performances sont plutôt médiocres.

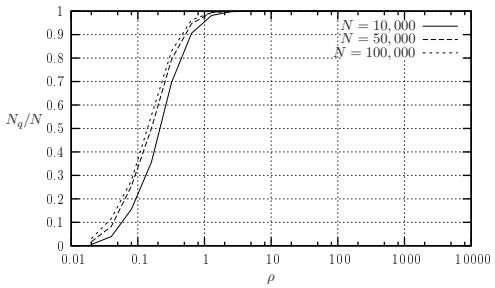


FIG. 3.1 – Fraction de pair en cours de téléchargement en fonction de la charge.

La dégradation des performances du système est confirmée par la figure 3.2 où nous représentons le rapport entre le nombre de serveurs occupés N_a et le nombre de serveurs N_s. L'évolution de ce rapport en fonction de ρ indique que pour une forte charge, les serveurs sont majoritairement des serveurs occupés qui sont en train de servir au moins un client. En conséquence, les pairs mettent plus de temps à télécharger l'objet.

Nous nous intéressons à présent à l'évolution du nombre de serveurs N_s en fonction de ρ. Sur la figure 3.3, nous observons que le nombre de serveurs décroît lorsque nous augmentons la charge ρ.

Finalement, nous nous intéressons à l'évolution du nombre de serveurs occupés, noté N_a, en fonction de la charge ρ. Ces serveurs représentent en quelque sorte le cœur du réseau eDonkey. En effet, lors de la formation du réseau, un ensemble de pairs complètent leur téléchargement et deviennent aussitôt des serveurs. Les nouveaux pairs vont ainsi s'accumuler sur ces serveurs primaires. Sur la figure 3.4, nous avons représenté l'évolution du ratio $N_a/\sqrt{N}$ en fonction de

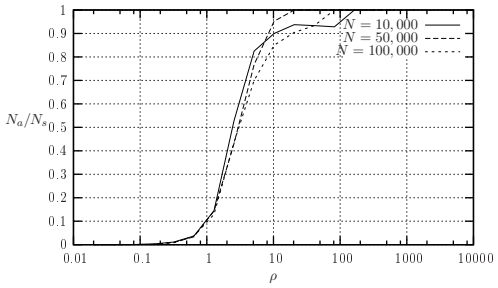


FIG. 3.2 – Fraction entre le nombre de serveurs occupés et le nombre de serveurs en fonction de la charge.

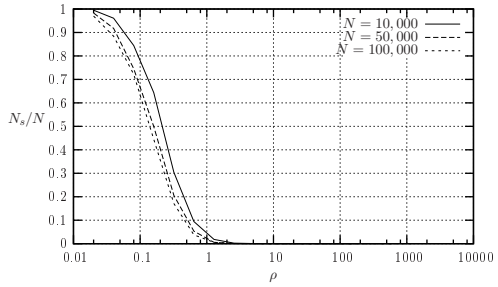


FIG. 3.3 – Fraction de serveurs par rapport à la population totale en fonction de la charge.

la charge ρ. Nous observons que le nombre serveurs occupés croît avec $\sqrt{N}$. Lorsque ρ croît, le nombre de serveurs occupés croît également mais lorsque ρ atteint une valeur voisine de 1, le nombre de serveurs occupés décroît rapidement. Dans le cas où nous avons une expansion du réseau, la taille de ce cœur du réseau est très petit par rapport au nombre total de la population. Dans un régime d'agglutination, les transferts de données se font à partir d'un nombre réduit de pairs congestionnés. Dans ce cas de figure, une très forte activité de signalisation sur l'état des files prend place dans le réseau.

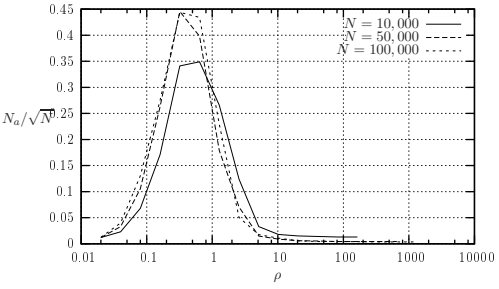


FIG. 3.4 – Fraction de serveurs occupés par rapport à à la racine de N

Il est évident que la charge ρ joue un rôle déterminant dans le mécanisme de formation du réseau. En effet, quand la charge ρ est inférieure à 1, un phénomène d'expansion se produit.

Le système est efficace et réussit à propager l'objet d'une façon rapide. Dans ce régime d'expansion, nous assistons également à une forte activité de signalisation qui se produit dans le réseau à cause des différents mécanismes de sélection des sources de données.

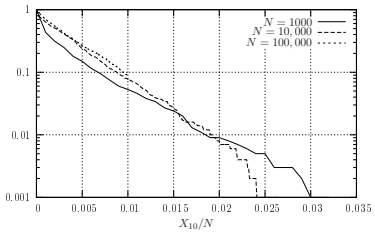
Par opposition au phénomène d'expansion, une charge ρ supérieure à 1 provoque un phénomène d'agglutination sur un cœur de nœuds congestionnés. Ce noyau est de petite taille. Cette agglutination des pairs sur un petit nombre de serveurs détériore les performances du système. En effet, l'objet ne se propage pas assez dans la communauté. En conséquence, les pairs sont contraints d'attendre plus de temps avant de récupérer l'objet en question. Mais malgré cela, les pairs finissent par rapatrier cet objet car la population reste tout de même de taille finie. Il faut néanmoins nuancer le propos quand on considère l'interaction entre plusieurs communautés formées autour du partage de plusieurs objets. Ces différentes communautés peuvent réduire les capacités de transmission des serveurs et les phénomènes de congestion augmentent. On se trouve alors dans une situation figée avec des temps de téléchargement excessivement longs, comme observé dans la pratique. En effet, les utilisateurs d'eDonkey se plaignent de la lenteur des téléchargements de ce logiciel (malgré la richesse de sa bibliothèque numérique). C'est la taille des files d'attente au niveau des serveurs qui est montré du doigt. Ce phénomène est aggravé par les débits d'upload pas très élevé.

Pour le modèle considéré, nous concluons alors que la charge ρ joue un rôle prépondérant dans l'apparition du phénomène d'agglutination. Cependant, l'hypothèse faite par rapport à la loi de service reste relativement optimiste. En fait, un temps de service suivant une loi exponentielle (dont le coefficient de variation = 1), atténue l'intensité du phénomène d'agglutination. En effet, il est connu dans la littérature [42] que le comportement des files PS dans un régime de surcharge est fortement lié à la loi de service et plus particulièrement au degré de variabilité de cette loi. Il serait intéressant alors d'étudier le comportement du système avec des temps de service plus variables.

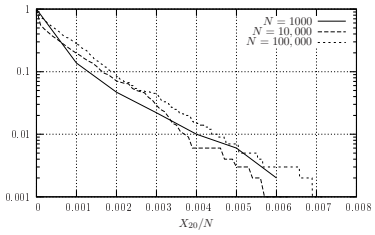
Mais avant d'évaluer l'effet d'une loi plus variable sur le comportement du système étudié, nous nous intéressons tout d'abord aux comportements limites des trois variables d'états N_a, N_s et N_q quand la taille de la population N devient assez grande. Les variables N_a, N_s et N_q normalisées d'une façon appropriée possèdent des limites quand N devient assez grand. De plus, nous considérons le vecteur aléatoire $X^{[N]} = (X_1^{[N]}, X_2^{[N]}, \ldots)$ où $X_j^{[N]}$ représente le nombre de clients rattachés au serveur j dans un régime surchargé ($\rho \geq 1$). Nous notons par $M^{[N]}$ le vecteur $X^{[N]}/N$. Sur les figures 3.5(a), 3.5(b) et 3.5(c), nous représentons les lois marginales de $M^{[N]}$ pour $j = 10$, 20 et 30 (c'est à dire, la distribution de $X_j^{[N]}/N$ pour ces valeurs de l'indice j) pour une charge $\rho = 1.2$. Nous remarquons que ces lois marginales tendent vers des limites ce qui signifie que la distribution $M^{[N]}$ tend vers une mesure déterministe μ définie dans $\mathbb{N}^{\mathbb{N}}$. Cette observation correspond à un comportement typique de ce qu'on appelle la théorie du champ moyen.

3.3.2 L'expansion du réseau avec free riders

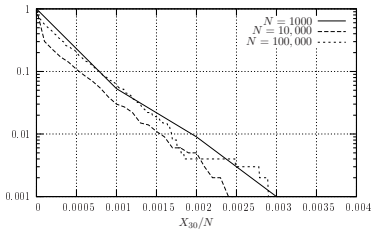
Après avoir étudier l'impact de la charge ρ sur l'apparition de différents phénomènes, nous nous intéressons dans cette partie à l'étude de l'impact free riders. Contrairement à la partie précédente, nous supposons maintenant qu'une proportion de la population des serveurs qui viennent de naître refuse de partager l'objet. Plus précisément, nous supposons qu'un pair qui vient de terminer le téléchargement de l'objet refuse de le partager et quitte immédiatement le système avec une probabilité f. En d'autres termes, un pair accepte de partager l'objet téléchargé avec une probabilité $1 - f$.



(a) $j = 10$



(b) $j = 20$



(c) $j = 30$

FIG. 3.5 – Convergence des marginales de la mesure $M^{[N]}$ quand N tend vers l'infini pour $\rho = 1.2$.

Dans la figure 3.6, nous représentons l'évolution des ratios N_s/N et N_q/N en fonction de la charge ρ pour différentes valeurs de la probabilité f pour $N = 100,000$. Nous remarquons que les freeriders détériorent les performances du système étudié. En effet, pour de grandes valeurs de la probabilité f, nous notons une augmentation de la proportion des pairs qui n'ont pas encore terminé leur téléchargement. En plus, le seuil à partir duquel le phénomène de congestion apparaît se déplace vers des charges inférieures à 1.

Ce rôle négatif que jouent les freeriders est confirmé lorsque nous analysons l'évolution du ratio N_a/N en fonction de la charge ρ pour différentes valeurs de la probabilité f. Sur la figure 3.7, nous remarquons que le maximum est atteint pour des valeurs de charge ρ de plus en plus petite au fur et à mesure que nous augmentons la probabilité f. Cette observation signifie que le noyau de serveurs congestionnés se forme pour des charges plus modérées. Cette conclusion sur l'impact des freeriders sur les performances du système sont moins optimiste que les résultats de [29]. En effet, les auteurs de [29] montrent que les freeriders peuvent bénéficier du système sans que les performances de ce dernier se dégradent. La différence entre les deux conclusions réside dans le fait que les auteurs de [29] analysent un système dans un régime stable et non pas dans un régime transitoire surchargé comme dans la présente étude.

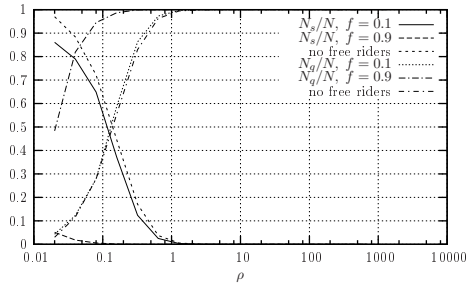


FIG. 3.6 – Ration entre le nombre de pairs attachés aux serveurs et le le nombre de serveurs en présence des freeriders.

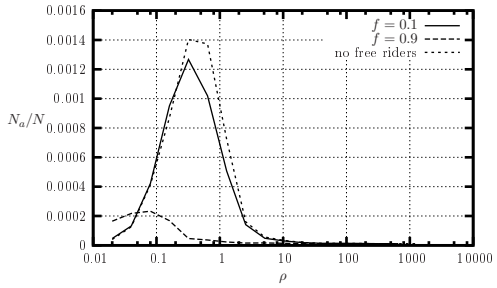


FIG. 3.7 – Ratio entre le nombre de serveurs occupés et le nombre total de la population en présence des freeriders.

3.4 Conclusion

Contrairement aux approches de modélisation présentées au début de ce chapitre, le modèle que nous proposons se focalise sur le régime de formation d'un simple réseau p2p. Ce modèle se base sur la simulation et nous fournit une explication plausible de la forte activité de signalisation que nous observons dans le système eDonkey réel grâce à la dichotomie souris/éléphant. Ce modèle met en évidence l'impact de l'intensité des arrivées des requêtes dans le système p2p sur la dissémination de l'objet partagé par les pairs. En fonction de ce taux, le système peut se retrouver dans l'un des deux régimes suivants : le régime d'expansion et le régime de congestion.

Conclusion Générale

Dans cette thèse, nous avons étudié le trafic Internet issu de différents points de mesure dans le réseau opérationnel de France Telecom. Cette étude, qui est essentiellement axée sur le trafic eDonkey, a permis d'appréhender les caractéristiques de ce trafic majoritaire dans le réseau de France Telecom.

Dans le premier chapitre, nous avons traité la problématique de l'identification du trafic p2p et du trafic eDonkey en particulier. Cette problématique n'est pas une tâche évidente. En effet, l'approche classique s'est avérée incapable d'identifier la totalité du trafic p2p. Cette approche se base sur les ports standards pour l'identification des applications. Les problèmes juridiques engendrés par l'utilisation des réseaux p2p ont joué un rôle crucial dans l'obsolescence de cette approche classique d'identification. Afin d'étudier les caractéristiques du trafic p2p, nous avons évalué dans un premier temps la représentativité du trafic p2p clair. Cette étude préliminaire, basée sur les numéros de port, nous a permis de faire deux constatations importantes. La première est relative à la représentativité de la partie claire observée du trafic eDonkey. Cette constatation est basée sur des mesures faites en 2004 et 2005. Ces mesures montrent que les pairs eDonkey communiquent fréquemment avec des numéros de ports standards. La deuxième constatation est relative à ce que nous avons appelé ports voisins. Les pairs eDonkey utilisent comme port d'écoute des numéros de port très proches des numéros de port standard (14662, 40662, 5662, etc.). Ces deux constatations préliminaires nous ont permis d'extraire la partie claire du trafic p2p et de la juger comme représentative de la totalité du trafic. Cependant, avec les nouvelles versions des applications p2p étudiées, cette représentativité est remise en question. Afin de pallier ce problème d'identification, nous avons étudié et proposé plusieurs méthodes de reconnaissance du trafic p2p. Nous avons évalué la performance de ces méthodes grâce à des campagnes de mesures de trafic issues du réseau opérationnel de France Telecom. Nous avons conclu que les méthodes de reconnaissance du trafic p2p sont complémentaires et aucune méthode ne prévaut. En revanche, l'identification basée sur les signatures applicatives reste une méthode de référence. Cette méthode est à son tour menacée par les évolutions récentes des applications p2p. La communauté de développeurs des applications p2p est particulièrement active et elle n'a pas cessé de relever les défis. Cette communauté a toujours réussit à contourner toutes les actions techniques menées à l'encontre des réseaux p2p. Les actions juridiques et les méthodes techniques (DRM, bridage, etc.) ont été mis en œuvre afin à protéger un modèle économique classique de diffusion de contenus (musique, films). Mais ces actions n'ont jamais réussi à protéger ce modèle bien au contraire elles figurent parmi les éléments moteurs de l'évolution des réseaux p2p (passage des réseaux centralisés vers les réseaux décentralisés, apparition des DHT, brouillage des protocoles, etc.).

Dans le deuxième chapitre, nous avons analysé la topologie du réseau eDonkey dans les réseaux opérationnels de France Telecom. Afin de comprendre les aspects géographiques du trafic eDonkey, nous avons choisi de caractériser la matrice de trafic eDonkey dans le réseau

national de France Telecom. En effet, étant donné que la contribution du trafic national d'eDonkey dépasse la barre des 50% du trafic eDonkey global, la prédiction de l'impact de ce trafic passe par une analyse de la topologie dans le RBCI (réseau national de France Telecom).

Pour ce faire nous avons procédé par une analyse des données remontées par six sondes passives (les sondes OTARIE). Ces six sondes couvrent d'une façon quasi complète le territoire français. L'analyse de l'ensemble de ces données montre que la majorité du trafic (eDonkey ainsi que les autres applications) provient ou part vers des clients d'autres opérateurs nationaux et vers des clients d'autres opérateurs étrangers. La proportion du trafic p2p échangée entre les nœuds de collecte internes au RBCI reste relativement modeste et ne dépasse pas dans les meilleurs des cas la barre des 25% du trafic global. L'agrégation des données quotidiennes a permis en conséquence d'appréhender la matrice du trafic p2p au sein du réseau national de France Telecom. La matrice du trafic eDonkey contredit le principe des réseaux P2P.La topologie que nous avons inférée est de type étoilé. Ce type de topologie est d'habitude en cohérence avec la philosophie client/serveur. En plus du réseau national de France Telecom, nous avons également analysé la matrice AS/AS dans le réseau de transit international de France Telecom. Pour ce faire, nous avons utilisé les enregistrements NetFlow envoyés par un certain nombre de routeurs dans le réseau de transit international. Nous arrivons à l'issue de cette étude, à identifier les sommets de la matrice AS/AS. Les sommets de cette matrice sont les plus grands clients du réseau OTIP. Ces clients sont bien évidement adjacents au réseau de transit de France Telecom.

Le troisième chapitre s'est focalisé sur la problématique de la modélisation des réseaux p2p. Les mesures sur la topologie du p2p mettent en évidence une topologie simple de type étoilé. Cette constatation est due à une myriade d'explications issues de domaines très variés (techniques, sociales, etc.). Ces observations nous ont poussés à regarder le mécanisme de formation de ce type de réseau. Contrairement aux modèles des réseaux P2P proposés dans la littérature, notre modèle s'intéresse au régime de formation du réseau p2p. Ce modèle se base sur la simulation et nous fournit une explication plausible de la forte activité de signalisation que nous observons dans le système eDonkey réel. Il met en évidence également l'importance du taux d'arrivées des requêtes dans le système p2p sur la dissémination de l'objet partagé par les pairs. En fonction de ce taux, le système peut se retrouver dans l'un des deux régimes suivants : le régime d'expansion et le régime de congestion.

A l'issue de cette étude, nous pouvons conclure par quelques réflexions sur les différentes problématiques traitées. Dans un avenir proche, la métrologie occuperait un rôle prépondérant dans le processus de dimensionnement d'un réseau opérationnel. A l'heure actuelle, le rôle de la métrologie est relativement limité et restreint à une tâche de constatation et d'observation. En effet, les capacités de transport offertes par les équipements du marché grandissent chaque jour. Cette augmentation des capacités s'accompagne d'une baisse remarquable du coût de la bande passante offerte par ces équipements. Face à cette offre, les opérateurs des télécoms n'ont pas hésité à choisir la carte du surdimensionnement. Ce choix, facilement mis en œuvre, leur garantit jusqu'à aujourd'hui une exploitation stable du réseau opérationnel. En conséquence, l'augmentation de la charge d'un lien donné se résout tout simplement par une duplication de la capacité de ce lien. Mais cette stabilité est aujourd'hui en péril avec les nouvelles exigences de l'utilisateur de demain. Cet utilisateur a fait des concessions sur la qualité de service lorsqu'il téléchargeait les contenus non libres de droits et cela représentait pour lui le prix de la gratuité. Mais cette gratuité peut être remise en question si le modèle d'une licence globale s'impose. A l'heure actuelle, la majorité des acteurs de la diffusion des contenus refusent d'adhérer à ce modèle de rémunération et préfèrent protéger leur modèle classique à travers

des protections techniques appuyées par des actions juridiques. Mais historiquement, le monde de la diffusion des contenus a toujours vécu ce processus qui commence par l'indifférence vis-à-vis de la nouvelle technologie de distribution de contenus mais qui finit par son acceptation (la radio, VHS, K7, etc.). L'instauration d'une licence globale pourrait aggraver la situation d'un opérateur car les pirates d'aujourd'hui deviendront des utilisateurs exigeants par rapport à la qualité de service. En plus, une telle licence éliminera l'inhibition des utilisateurs réticents à l'heure actuelle et de nouveaux pairs s'ajouteront à la nébuleuse p2p.

Cependant, une licence globale mettra fin aux duels techniques entre la communauté des développeurs des applications p2p et les concepteurs de méthodes d'identification et de limitation du trafic p2p. Avec une éventuelle légalisation du p2p, la communauté des chapeaux blancs ("pirates de la bonne cause"), actuellement en position de force, ne serait plus contrainte de camoufler ce trafic. En plus, les deux concurrents d'aujourd'hui pourront joindre leurs efforts pour garantir une amélioration de la qualité de service offerte par les réseaux p2p. D'un autre côté les efforts de l'identification seront plus concentrés sur des problèmes plus critiques tels que la lutte contre la cyberpédophilie sur les réseaux p2p. Dans ce contexte, un opérateur, garant dans un avenir proche de la neutralité de ses réseaux, sera amené à intégrer cette dynamique en faisant appel à ses métrologues. En effet, la métrologie occupera un rang primaire chez un opérateur car le trafic p2p qui devient légalisé mais identifiable, chargera de plus en plus les liens du réseau et le surdimensionnement sera sans doute un jour obsolète. La métrologie accompagnera ce phénomène et sera une étape clé du processus du dimensionnement.

En plus du trafic p2p, d'autres types de trafic commencent à émerger et risquent de déstabiliser le choix du surdimensionnement. Les adeptes du streaming vidéo sont de plus en plus nombreux (les prédictions parlent d'1 Milliard d'utilisateurs sur les sites de streaming à l'image du modèle Youtube et Dailymotion pour 2012). Ces applications, bien qu'elles soient considérées comme représentatives de la tendance web 2.0, restent fidèles à une philosophie client/serveur. En effet, le fonctionnement des principaux sites n'est pas basé sur les principes p2p. Pour visionner une vidéo, un utilisateur interroge un serveur (ou une ferme de serveurs) bien que la vidéo soit affectée à un utilisateur donné. Nous pouvons imaginer une évolution vers une approche p2p (quelques spécialistes commencent à parler du web 3.0) à l'image des applications de la télévision diffusé en p2p (PPlive, Joost, etc.).

Si l'approche client/serveur ne sollicite que le sens descendant de l'accès de l'utilisateur, une migration vers un modèle de streaming p2p augmenterait la charge du sens montant (à l'heure actuelle et pour le streaming vidéo, ce sens est sollicité lors de l'upload des vidéo ce qui est le cas d'une minorité d'utilisateur, le phénomène de Pareto est plus marqué pour Dailymotion et Youtube). Cette migration vers un modèle p2p assurera un équilibre entre le sens montant et le sens descendant et remettra en cause l'asymétrie des accès ADSL et justifiera une migration plus rapide vers les accès optiques. Le FTTH serait plus adapté à ce que l'on constate aujourd'hui sur le trafic de partage de fichiers p2p et ce que l'on peut constater très probablement sur d'autres applications s'inspirant de la philosophie p2p (pensons par exemple à une éventuelle émergence des hologrammes, qui pourrait être une véritable killer application justifiant le passage vers un accès optique).

La démocratisation du FTTH pourrait exporter la congestion des liens d'accès vers le cœur du réseau de l'opérateur. Les débits offerts par cette technologie pourrait être un vrai appel d'air et induira une augmentation de la demande du débit par utilisateur (télévision HD, multipostes, télévision 3D, etc). Les accès FTTH chez France Telecom pourrait avoir un impact sur la topologie des réseaux p2p et soulagera en conséquence les liens de peering. Toutes ces réflexions ne sont que des spéculations déduites de l'observation de l'état actuel

des choses tout en faisant des analogies avec évolutions de technologie antérieures. Sous ces conditions, la métrologie devrait continuer sa mission d'analyse des nouveaux phénomènes du trafic et de caractérisation de leurs comportements en attendant de reprendre une place plus importante avec la fin de l'ère du surdimensionnement.

Annexe A

eDonkey : histoire, protocole et trafic

En 2000, Jed McCaleb a créée le client eDonkey2000 et marque ainsi le début de l'histoire du réseau eDonkey. Ce système d'échange de fichiers ressemble à la base au célèbre Napster, pionnier des systèmes pair à pair. Cependant, les développeurs du reseau eDonkey ont su remédier aux faiblesses de Napster. En effet, l'unique serveur d'indexation de Napster a été remplacé dans eDonkey par une multitude de serveurs géographiquement répartis. Cette décentralisation géographique des serveurs eDonkey est accompagnée d'une décentralisation des responsabilités. Les serveurs eDonkey peuvent être installés par n'importe quel individu ce qui rend la tâche des actions légales relativement difficile. En effet, la fermeture de l'un de ces serveurs d'indexation n'aura qu'un effet restreint sur la survie du réseau. En février 2006, la saisie du très populaire serveur belge *Razorback* n'a eu aucun effet sur la part du trafic eDonkey ni sur sa répartition géographique malgré le fait que ce serveur d'indexation sert près du quart des utilisateurs eDonkey. Ce serveur communiquait simultanément avec près d'un million de clients.

Ce chiffre confirme la popularité du réseau eDonkey dans le monde et surtout en Europe. Cette popularité est essentiellement due à la capacité de ce système à étaler le téléchargement sur une longue durée. Le réseau eDonkey se caractérise également par la richesse de la "bibliothèque numérique" formée par ses utilisateurs. Notons ici que l'évolution de la capacité des ordinateurs pour visionner de la vidéo a été un fait déterminant dans l'émergence du réseau eDonkey.

Le système eDonkey est connu par les noms de ses clients. Nous pouvons citer à titre exemple emule, eDonkey2000 ou Mldonkey. Le client emule est devenu le client le plus populaire du réseau eDonkey et représente plus de 90% des clients installés. Quant aux serveurs, deux types se sont imposés : eDonkey2000 et Lugdunum. A titre d'information, le serveur Razorback utilisait Lugdunum. Ce type de serveur interagit avec des clients emule.

A.1 Le protocole eDonkey

A.1.1 Identificateur Fichier

Dans le réseau eDonkey, les fichiers ne sont pas identifiés par leurs noms. Un fichier est identifié par une clé de hachage calculée à partir de son contenu. Pour ce faire, le fichier est découpé en parties de 9.28 Mo. La fonction de hachage sera appliquée sur chacune des parties de fichier. L'identificateur du fichier sera alors la combinaison de ces différents résultats. Etant donné qu'on se base sur le contenu, l'identificateur du fichier a une validité étendue sur tout

le réseau et permettra ainsi une indexation plus rigoureuse. En plus de ce rôle d'indexation, l'identificateur de fichier est un moyen pratique de détecter des versions ou des parties corrompues d'un fichier donné. Le client peut vérifier l'intégrité d'une partie du fichier et peut éventuellement demander au réseau sa retransmission d'une partie en cas de problème.

A.1.2 Identificateur client

L'identificateur client est attribué par le serveur eDonkey lors de l'établissement de sa connexion avec le serveur. Cet identificateur reste valide tout au long de la session TCP. Pour un client donné, l'identificateur du client peut prendre deux niveaux possibles. L'identificateur attribué par le serveur d'indexation peut être de bas ou de haut niveau en fonction de l'aptitude du client à recevoir des connexions TCP entrantes. Par exemple, un client aura un identificateur client de bas niveau s'il est situé derrière un pare-feu refusant des connexions entrantes. Citons aussi l'exemple des clients qui se connectent sur le même serveur d'indexation en utilisant la même adresse IP (Network Address Translation).

L'identificateur client de bas niveau est compris entre 1 et 16 Millions. Par contre, un identificateur de haut niveau est supérieur à 16 Millions. Un client avec une adresse IP fixe aura toujours le même identificateur de haut niveau sur n'importe quel serveur eDonkey étant donné que l'identificateur est une fonction déterministe de l'adresse IP. Par contre, un identificateur de bas niveau peut changer d'un serveur à un autre ou d'une session à une autre. Un client qui n'est connecté à aucun serveur a un identificateur client nul.

A.1.3 Identificateur Utilisateur

L'identificateur client est attribué au début de chaque session contrairement à l'identificateur utilisateur qui est unique pour toutes les sessions. Cet identificateur est utilisé pour gérer le mécanisme du crédit. Un utilisateur qui se montre généreux comme fournisseur de fichiers, aura un traitement de privilégié lorsqu'il est consommateur. Pour éviter toute usurpation des identités utilisateurs visant à profiter des crédits d'autrui, un mécanisme d'authentification sécurisé est implémenté. Le mécanisme est basé sur l'algorithme de chiffrement RSA.

A.1.4 Communication Client-serveur TCP

Le client eDonkey est préconfiguré avec une liste de serveurs d'indexation (fichier point mat). Une fois lancé, le client eDonkey essaie d'établir une connexion avec l'un des serveurs de cette liste. Dès que l'une des tentatives de connexion aboutit, les autres tentatives sont abandonnées. La majorité des clients maintiennent une connexion TCP avec un et un seul serveur (le client MLDonkey établit des connexions TCP avec plusieurs serveurs, des extensions de clients permettent des connexions avec d'autres serveur en UDP). La connexion client serveur sera maintenue durant toute la session. Sur cette connexion transitera un premier message *Login* du client vers le serveur. Dans ce message, le client envoie l'identificateur utilisateur et son port d'écoute.

Le serveur tentera d'établir une connexion expérimentale sur ce port d'écoute afin de savoir si le client accepte bien des connexions entrantes. De succès ou de l'échec de cette connexion sera attribué l'identificateur client par le serveur. Si le serveur arrive à joindre le client sur son port d'écoute, le client aura un identificateur de haut niveau. Dans le cas contraire, le client aura un identificateur de bas niveau. Les serveurs d'indexation peuvent avoir dans leur implémentation une limitation logicielle du nombre de clients. Aussitôt cette limite logicielle

atteinte, le logiciel refusera les clients pourvu d'un identificateur client de bas niveau. Bien entendu, une limitation matérielle existe et une fois que cette borne est dépassée, le serveur n'accepte plus de client.

Cette phase d'amorçage est suivie par une phase de mise à jour des deux interlocuteurs (le client et le serveur). Le client fournit sa liste de fichiers qu'il partage. Pour ce faire le client envoie un message *offerFiles*. Dans ce message, le client spécifie le nombre de fichiers partagés dans un champ dédié de 4 octets. Une borne inférieure ou supérieure peut être instaurée par le serveur. La littérature parle d'une borne supérieure de 200 fichiers.

Ce champ est suivi d'une concaténation de descriptifs fichier. Un descriptif fichier contient une multitude de paramètres spécifiques au fichier. Quelques paramètres sont indispensables tels que l'identificateur du fichier, son nom, sa taille, l'identificateur client (initialisé à zéro dans le cas d'un client ayant un identificateur de bas niveau). D'autres paramètres sont facultatifs et servent à affiner la description du fichier. Parmi ces paramètres, nous pouvons citer le format du fichier, le codec utilisé ou encore la durée (pour les fichiers média).

Le client demandera la mise à jour de sa liste de serveurs d'indexation en envoyant un message *GetListServers*. Le serveur répond alors par un message *ListOfServers*. Un premier champ du message spécifie le nombre d'enregistrements dans la liste de serveurs envoyé. Ce premier champ est suivi d'une concaténation d'enregistrements. Chaque enregistrement identifie un serveur d'indexation en indiquant son nom, son adresse IP et son port d'écoute.

Une fois les mises à jour terminées, l'utilisateur peut initier une phase recherche de fichiers. L'utilisateur tape une chaîne de caractères dans un champ de l'IHM du client afin de décrire approximativement le fichier qu'il désire télécharger (un titre de chanson par exemple). Cette chaîne de caractère sera encapsulée avec d'autres paramètres intitialisés par l'utilisateur (type du fichier demandé, taille maximale, taille minimale,...) dans un message *SearchRequest*. Ce message est envoyé vers le serveur d'indexation.

A titre de réponse, le serveur envoie un ensemble de propositions proches de la requête du client. Une proposition contient outre l'identificateur fichier des informations sur sa disponibilité (nombre de sources). D'autres paramètres peuvent figurer dans la proposition (débit, durée pour les fichiers multimedia,...). L'utilisateur prend connaissance des propositions et choisit les fichiers qu'il désire télécharger.

Le client eDonkey envoie alors un message *GetSources* pour chaque fichier sélectionné par l'utilisateur. Le client demande à travers ce message la liste des adresses des sources disponible ainsi que leurs ports d'écoute.

A.1.5 Communication Client-Client TCP

Le client obtient une liste de sources à partir de son serveur d'indexation. Il contacte les sources de cette liste. Dans la suite de cette section, nous notons par A, le client qui veut récupérer le contenu et par B l'un des clients figurant dans la liste des sources.

Le client A initie une connexion TCP avec le client B sur le port d'écoute de ce dernier. Une fois la connexion établie, le client A envoie un message *Hello*. Dans ce message, le client A envoie son identificateur client et utilisateur, son port d'écoute ainsi que l'adresse IP et le port d'écoute de son serveur de rattachement. Dans le message *HelloAnswer* envoyé par le client B à titre de réponse, on retrouve les mêmes champs.

Le client A envoie au client B l'identificateur du fichier dans un message *FileRequest*. Le client B peut répondre par deux types de message. Si le client B ne possède plus le fichier ou ne le met plus en partage, il répond par un message *FileNotFound* et la connexion TCP est

rompue. Dans le cas où le fichier figure encore dans la liste de partage du client B, ce dernier répond par un message *FileRequestAnswer*.

Dans ce cas, le client A demande le déclenchement du téléchargement en envoyant un message *StartUploadRequest* au client B. Si le client A est le seul à vouloir télécharger le fichier, il sera aussitôt servi par le client B. Pour ce faire, le client B répond par un message *AcceptUploadRequest*.

Dans le cas où il y a plusieurs clients en concurrence, le mécanisme de file d'attente implémenté au niveau du client B gère ce télescopage. Le client B indique au client A sa position dans la file d'attente et rompt la connexion TCP. Le client B établira une connexion TCP avec le client A dès que ce dernier atteindra la tête de la file d'attente du fichier. Le premier message envoyé par le client B sur cette connexion sera alors *AcceptUploadRequest*. Par ailleurs, il est possible que le client A ait pu se procurer de la totalité du fichier à partir d'autres sources avant d'être recontacté par le client B. Il peut donc annuler sa demande de téléchargement en envoyant un message *CancelTransfer* et la connexion est alors rompue.

Lorsque les deux interlocuteurs terminent leur phase de signalisation, le transfert de fichier peut commencer. Le client A envoie alors un *RequestFileParts* au client B dans lequel il spécifiera, l'identificateur du fichier et aussi la position de la partie désirée dans le fichier (pointeur début et pointeur fin). Dans ce message *RequestFileParts*, le client A peut demander simultanément trois parties différentes du fichier. Plusieurs paramètres conditionnent le choix de la partie à télécharger. Le client A va répartir le téléchargement sur les différentes sources auxquelles il est connecté.

Chaque partie sera envoyée dans une série de paquets du client B vers le client A. Le premier paquet contiendra l'entête qui identifie la totalité du transfert de la partie. Cet entête contient la taille (en octets) de la totalité de la partie envoyée et pas seulement la taille du paquet courant. Avec l'identificateur du fichier, on trouvera également les offsets des données contenues dans le paquet courant.

La connexion TCP entre le client A et le client B peut transporter également des messages de type *ViewSharedFiles* ou *HashsetRequest*. Le message *ViewSharedFiles* sert à demander au client d'afficher sa liste de fichiers partagés. Cette requête peut être acceptée ou refusée. Dans le premier cas, le client interrogé répond par un message *ViewSharedFilesAnswer* contenant la liste des fichiers partagés (les mêmes informations envoyées par le client au serveur lors de la phase de mise à jour). Dans le deuxième cas, le client répond par un *ViewSharedDenied*. Le message *HashsetRequest* sert quant à lui à échanger les résultats de hachage de chaque partie du fichier. En réponse, le client interrogé répond par un *HashsetAnswer*.

A.1.6 Communication Client-Client UDP

Entre deux clients, la communication se fait essentiellement sur TCP. Par ailleurs, le protocole UDP peut être utilisé pour insérer un client qui ne figure pas dans la file d'attente d'un client fournisseur. Périodiquement, un client A cherchant à intégrer une file d'attente au niveau d'un client B, va envoyer un *ReAskFile*. Dans ce message, l'identificateur fichier est spécifié ainsi que le nombre de sources susceptibles de fournir (au client A) le fichier indiqué. Le client B peut nier l'existence du fichier désiré dans sa liste de fichiers partagés par un message *FileNotFound*. Le client B peut également annoncer que la file est pleine en envoyant un *QueueFull*. En revanche, le client B peut admettre le client A dans sa file d'attente. Il lui renvoie alors son rang dans cette file. Le rang est envoyé dans un message *ReAskFileAck*.

A.1.7 Communication Client-Serveur UDP

En UDP, la communication entre le client et le serveur a essentiellement deux fonctions. La première fonction est de vérifier périodiquement la validité de la liste des serveurs. Pour ce faire, un client met un nombre aléatoire dans des messages de type *StatusRequest* ou *DescriptionRequest*. Le serveur doit répondre avec le même nombre aléatoire. Un temporisateur est déclenché lors de l'envoi de ces messages. Si le temporisateur s'expire, le serveur est considéré comme mort et est rayé de la liste. Le deuxième but consiste à faire des recherches de sources parallèlement à celles faites avec le serveur principal. Ces recherches se font avec d'autres serveurs et le mécanisme s'apparente à celui décrit pour la communication serveur Client en TCP.

A.1.8 Communication Serveur-Serveur UDP

La communication entre serveurs ne se fait qu'en UDP. Grâce à ce protocole, un serveur fraîchement connecté, pourra annoncer sa présence (*Announce*). Le protocole UDP sert également dans les échanges de messages de battement de cœur (*Ping Pong*) ou pour échanger les listes de serveurs d'indexation.

A.1.9 Quelques mécanismes du système eDonkey

Nous décrivons dans la suite quelques mécanismes mis en oeuvre par eDonkey :

Mécanisme de callback Un client ayant un identificateur client de bas niveau ne peut pas recevoir de connexion entrante. Mais ce client communique la liste de ses fichiers partagés à son serveur. Ses fichiers se retrouvent ainsi indexés par le serveur. Ce client peut être en conséquence une source de ces fichiers. Pour permettre aux clients désirant télécharger des fichiers de ce client, un mécanisme de Callback a été mis en place. L'idée est de passer par la connexion TCP entre le serveur et le client *Low ID* (établie à l'initiative du client) pour demander à ce dernier d'ouvrir une connexion sortante vers le client *High ID*. Ce mécanisme n'est pas bien toléré au niveau du serveur à cause de la charge supplémentaire qu'il engendre. Il faut noter également que le tunneling entre deux serveurs n'existe pas.

Système de crédit Le crédit est une notion locale qui n'a de signification qu'entre deux clients donnés. Le credit sert pour le mécanisme de préemption dans la file d'attente. C'est un mécanisme d'incitation au partage. Le credit d'un client distant s'exprime en fonction du cumul des données qu'il a envoyé au client considéré et le cumul des données qu'il a reçues. Le premier cumul est noté e et le deuxième est noté r. Nous avons :

$$Credit = min(\frac{2e}{r}, \sqrt{e+2})$$

Pour permettre à un nouveau client de récupérer assez rapidement du contenu qu'il pourrait partager, le credit est initialisé avec la valeur maximale.

Mécanisme de préemption Un score est calculé au niveau de la source de contenu. Le client ayant le plus grand score va bénéficier du téléchargement

$$Score = \frac{Credit * P_f * T}{100}$$

où $P_f \in [0.2, 1.8]$ est la priorité du fichier (cette priorité est fixée par l'utilisateur qui partage le fichier). La file d'attente est ordonnée et le client avec le plus grand score

sera placé en tête de la file. Cette arrivée déclenchera le processus de téléchargement. Un client qui arrive en tête verra son score augmenté.

Échange de Sources L'échange de sources peut se faire entre clients. Chaque 10 minutes, le client interroge un client pris au hasard parmi la liste de sources dont il dispose. Si le fichier est rare (moins de 40 sources), le client interroge l'ensemble des sources de sa liste. Parmi les nouvelles sources obtenues, le client ne gardera que les sources disposant de parties manquantes.

Bannissement Un client réitère sa demande de téléchargement toutes les 10 à 20 minutes pour s'assurer que son correspondant est encore une source valide. Si un client envoie un peu trop de requêtes, il sera banni par les sources.

Annexe B

La théorie du champ moyen dans la littérature

B.1 La théorie du champ moyen dans la littérature

La théorie du champ moyen est une méthode d'approximation issue du monde de la mécanique statistique. En effet, le modèle de Curie-Weiss est l'une des diverses approches visant à caractériser le comportement limite des différentes grandeurs physiques d'un réseau de particules. Ce dernier modèle montre que les fluctuations du potentiel au niveau d'une particule deviennent négligeables lorsqu'on fait tendre le nombre de particules en interaction vers l'infini. Le potentiel se stabilise alors autour d'une valeur moyenne induite par l'influence de l'ensemble des particules. Le réseau de particules en interaction devient équivalent à un système de particules indépendantes soumises à un champ moyen. La méthode du champ moyen est très liée au domaine de la mécanique statistique qui reste historiquement son premier champ d'application. D'ailleurs, Sherrington et Kirkpatrick vont proposer une version modifiée du modèle de Curie-Weiss mais qui reste liée à la théorie du champ moyen. Au milieu des années 80, Parisi est devenu une référence dans ce domaine d'application en proposant une résolution du modèle de Sherrington et Kirkpatrick. La démonstration de Parisi est une approche non rigoureuse et le monde de la physique statistique cherche depuis à lui donner une justification.

Cependant, la méthode d'approximation du champ moyen ne s'est pas restreinte aux domaines de transitions de phase et de ferromagnétisme. D'autres domaines ont usé de cette méthode d'approximation. Les champs d'application vont encore se diversifier et les particules élémentaires du domaine pionnier vont continuer à prendre de nouvelles formes et intégrer de nouveaux concepts. Nous trouverons l'approximation du champ moyen dans le domaine de la biologie moléculaire où cette théorie a servi à estimer la loi du nombre moyen de mutation au cours d'une réaction PCR (Polymerase Chain Reaction) [64, 65, 84]. Cette réaction consiste à créer une grande quantité de copies d'une portion d'une molécule d'ADN à partir d'un nombre très réduit d'exemplaires de cette molécule. Dans des conditions expérimentales à fort taux de mutation, l'un des objectifs des chercheurs de biologie est d'estimer le taux de mutations après un nombre de cycles donné. L'approximation du champ moyen intervient lorsqu'on part d'une condition initiale stipulant que le nombre d'exemplaires est infini.

Comme la biologie moléculaire, les sciences cognitives vont puiser à leur tour dans cette théorie du champ moyen. Les segments d'ADN vont se transformer alors en neurones [23]. La théorie du champ moyen a permis de caractériser le comportement d'un réseau de neurones

connectés aléatoirement. L'analyse d'un réseau de très grande taille permet de formaliser des comportements naturels et d'ouvrir la voie à des études sur l'apprentissage. La théorie du champ moyen est un outil théorique très souple comme le prouve la diversité de ses domaines d'application. La théorie est un outil capable de formaliser le comportement des systèmes naturels (particules, ADN, neurone). Cependant, la théorie se prête à l'analyse de systèmes artificiels.

Dans les années 90, les approximations de type champ moyen ont servi dans le domaine du traitement de l'image. Les pixels ont remplacé ainsi les particules élémentaires de la physique statistique où les valeurs de spin céderont la place à la nuance de gris ou de couleur. L'approximation du champ moyen sert alors à donner une estimation simple au champ markovien. Un système de variables indépendantes remplace ainsi le champ de Markov avec ses interactions complexes. Cette approximation a servi pour la ségmentation des images i.e. la détection des surfaces [63, 30] ou pour la détection des contours [89]. La théorie du champ moyen a servi également à la détermination de la loi de distribution des longueurs de chemin dans le modèle de réseau des petits mondes de Watts et Storgatz [57]. Les réseaux Bayésiens et les réseaux de Neurones vont profiter de la souplesse de cette théorie [74, 75] dans le but d'approximer des distributions de probabilité dans différents scénarios. L'application de cette théorie sur des applications plus proche du domaine des réseaux informatiques vont apparaître au début de ce nouveau millénaire [41, 54, 53, 12].

Les auteurs de [54, 53] s'intéressent aux interactions entre N connexions TCP pendant la phase d'évitement de congestion de la version RENO de TCP. Ces N connexions partagent une file FIFO munie du mécanisme RED (Random Early Detection). Ce mécanisme consiste à détecter la congestion avant qu'elle ne survienne en marquant ou en détruisant à l'avance des paquets avec une probabilité qui croît avec la longueur de la file d'attente. Le comportement idéal du mécanisme RED est de répartir de manière équitable les pertes de paquets. Les auteurs analysent la convergence de la taille des fenêtres de N sessions. L'ensemble des N tailles de fenêtre constitue alors l'état du système étudié. Les auteurs analysent alors la convergence de ce vecteur lorsque le nombre de sessions concurrentes tend vers l'infini. L'analyse porte également sur l'évolution de la taille de la file d'attente par connexion. Pour évaluer la performance du mécanisme RED, les auteurs analysent la variabilité des débits par connexion. En partant d'un modèle fluide, les auteurs utilisent le processus empirique du système afin d'étudier son comportement asymptotique (nombre de connexions N infini). Dans les conditions de stabilité, le modèle converge vers le champ moyen.

La théorie du champ moyen est également utilisée dans le domaine des réseaux sans fil. Les auteurs de [12] prouvent le découplage entre des sources qui partagent une ressource (canal de transmission) quand le nombre de ces sources tend vers l'infini. L'accès à la ressource partagée se fait aléatoirement et d'une façon décentralisée (chaque source est autonome et décide l'instant de l'accès à la ressource). Le protocole d'accès étudié est de type Backoff. L'état du système est défini par les N probabilités de Backoff. Ces probabilités évoluent au cours du temps en fonction des événements (collision, transmission réussie). Les auteurs démontrent l'indépendance de ces probabilités après une renormalisation quand N est grand. Sans cette normalisation, les probabilités vont tendre naturellement vers 0.

B.2 Le champ moyen, une approche théorique

La théorie du champ moyen reste une approche formelle qui peut être présentée indépendamment du contexte dans lequel elle est appliquée. Nous présentons dans la suite une forme générique de la théorie du champ moyen. Soit $X = \{X_1, \ldots, X_N\}$ un vecteur aléatoire à espace d'états discret $H = H_1 \bigotimes \cdots \bigotimes H_N$. On note $P(X)$ une distribution de probabilité définie sur l'ensemble H. Dans plusieurs domaines, cette distribution est numériquement difficile à estimer. En conséquence, on introduit une distribution $Q(X)$ définie sur le même espace d'état H sous quelques contraintes simplificatrices. La distribution $Q(X)$ est l'approximation du champ moyen si elle obéit aux trois contraintes suivantes :

Distance entre P(X) et Q(x) minimale Pour mesurer la distance entre $P(x)$ et $Q(x)$, on utilise la distance $D(P||Q)$ dite Kullback-Leibler et définie par [20] :

$$D(P||Q) = \sum_{X \in H} P(X) \log \frac{P(X)}{Q(X)} \equiv \langle \log \frac{P(X)}{Q(X)} \rangle_{P(X)}$$

. La distribution $Q(X)$ doit être la plus proche possible de la distribution $P(X)$ selon la distance de Kullback-Leibler.

Propriété de factorisation Les variables $X_1, \ldots, X_N$ sont indépendantes par rapport à la distribution Q et nous avons ainsi :

$$Q(X_1, \ldots, X_N) = \prod_{i=1}^{N} Q(X_i)$$

Contrainte de normalisation Comme toute distribution de probabilité, nous avons la contrainte de normalisation que nous pouvons exprimée de la manière suivante :

$$\sum_{X \in H} Q(X) = 1$$

La première contrainte découle du principe variationnel issu de la mécanique statistique qui cherche à minimiser l'énergie libre. La minimisation de cette énergie revient à minimiser la distance de Kullback-Leibler citée plus haut.

Sans la deuxième contrainte, la distribution qui minimise la distance de Kullback-Leibler est la distribution de Gibbs P_G définie par :

$$P_G(X) = W^{-1} exp(-H(X))$$

où H est le Hamiltonien définie comme une fonction d'énergie associée à chaque état et qui s'écrit de la forme suivante :

$$H(X) = \sum_{c \in C} V_c(X_c)$$

Les fonctions V_c sont des fonctions de potentiel qui sont associées à une clique c. Avec des termes de la physique statistique, la clique c représente un groupe de particules en interaction. Le terme W est une constante de normalisation, appelée la fonction de partition. Cependant, la propriété de factorisation reste une propriété importante et qui représente l'objectif essentiel de l'approximation en champ moyen. L'approximation en champ moyen se présente alors comme un problème d'optimisation sous contrainte. Les auteurs de [35] proposent une résolution itérative de ce problème d'optimisation.

Annexe C

Quelques Notions sur les graphes et les réseaux

C.1 Quelques Notions sur les graphes et les réseaux

Degré d'un nœud C'est le nombre de liens associés à ce nœud. La moyenne des degrés des nœuds d'un graphe est appelée densité. La distribution des degrés est définie par :

$$p_k = Pr[Degr\ = k]$$

Dans les réseaux réels, cette distribution est souvent approchée par une loi de puissance :

$$p_k \sim k^{-\alpha}$$

avec un paramètre $\alpha \in]2, 3[$.
Ces réseaux sont dits réseaux sans échelle (une loi de puissance vérifie $f(ax) = af(x)$). On définit aussi le degré maximal k_{max} comme le degré qui vérifie

$$np_{k_{max}} = 1$$

où n est le nombre total de nœuds dans le graphe. En d'autres termes, il y a moins qu'un nœud ayant un degré supérieur à k_{max}. Dans le cas d'un réseau sans échelle on a

$$k_{max} \sim n^{\frac{1}{\alpha}}$$

Coefficient de clustering Le coefficient de clustering ou de transitivité est la probabilité que deux voisins d'un sommet pris au hasard soient eux-mêmes voisins. Quantitativement, on définit le coefficient de clustering de la manière suivante :

$$C = \frac{6 * Nombre\ de\ triangles}{Nombre\ de\ chemins\ de\ longueur\ 2}$$

Un triangle d'arrêtes met en liaison trois nœuds. Les réseaux réels sont caractérisés par un fort coefficient de clustering.

Distance La distance entre deux sommets est la longueur du plus court chemin qui les relie (en termes de nombre de liens). Entre deux nœuds il peut exister plusieurs chemins de même longueur.

Diamètre Le diamètre d'un réseau est la plus grande distance entre deux nœuds quelconques du réseau :

$$Diametre = max_{i,j} d_{i,j}$$

avec $d_{i,j}$: distance entre les nœuds i et j.

L'effet petits-mondes Ce phénomène a été observé pour la première fois dans les réseaux sociaux. Le phénomène se manifeste par l'observation suivante : deux personnes prises au hasard sont liées par une chaîne de connaissance relativement courte. Dans les expériences de Milgram qui est le premier à avoir observer ce phénomène dans les réseaux sociaux, on arrivent à vérifier empiriquement qu'en moyenne, cette chaîne de connaissances a une longueur située entre cinq et six. Le même phénomène a été observé ensuite dans le reste des réseaux réels. Pour formaliser le phénomène, on parle d'une décroissance logarithmique de la distance moyenne en fonction de la taille du graphe étudié. En d'autres termes, la distance n'évolue pas aussi rapidement que la taille du système. Le système se manifeste aussi par un fort coefficient de clustering. Par analogie aux réseaux sociaux, cela revient à dire qu'il y a une forte probabilité que l'ami de mon ami soit aussi mon ami.

C.2 La modélisation dans le monde des graphes

La modélisation des réseaux n'est pas un domaine tout récent. Les études empiriques menées sur une grande variété de réseaux réels ont permis de révéler de nombreuses propriétés. La modélisation a tenu compte de ces constatations en partant dans deux directions différentes. La première direction a tendance à créer des modèles mathématiques qui présentent des propriétés proches de la réalité. La deuxième direction tente d'expliquer l'apparition des ces propriétés en modélisant leurs mécanismes d'évolution. Les modèles issus de cette tendance sont appelés les modèles d'expansion.

C.2.1 Les graphes aléatoires

Les graphes aléatoires poissonniens

Le modèle des graphes aléatoires poissonniens date des années soixante. Ce modèle aléatoire élaboré par Erdös et Rényi est relativement simple. Initialement, on dispose de n nœuds sans aucun lien. Avec une probabilité p, on cable chaque couple du graphe. On parle ainsi d'un graphe $G_{n,p}$. Le calcul de la distribution des degrés montre que cette dernière peut être approchée par une distribution de poisson d'où le nom du modèle :

$$P_k = \frac{z^k e^{-z}}{k!}$$

avec $z = p(n-1)$ et cela en supposant qu'un nœud peut avoir une boucle sur lui même. L'étude théorique montre l'existence de deux régimes. L'apparition de ces deux régimes est conditionnée par le choix de la probabilité p. Pour une petite valeur de p, on observe l'apparition de plusieurs composants de petite taille dans le graphe. Une grande valeur de la probabilité p implique l'apparition d'un composant géant englobant une très grande proportion des nœuds du système. La réalité des petits-mondes est vérifiée pour ce modèle par rapport au premier

critère de phénomène à savoir l'évolution de la distance moyenne en fonction de la taille du réseau. En effet, si l est la distance moyenne dans le réseau, on a :

$$z^l \approx n$$

ce qui signifie une décroissance logarithmique de la distance moyenne en fonction de la taille du réseau. Par contre, le modèle reste incapable de fournir un grand coefficient de clustering du fait que la probabilité p ne tient pas compte du voisinage. En plus, la distribution poissonnienne des degrés n'est pas la distribution observée dans les réseaux réels.

Le modèle de Molloy et Reed

Afin de rendre le modèle Erdös et Rényi plus réaliste au niveau de la loi de distribution des degrés, le modèle de Molloy et Reed semble être une bonne alternative. Ce modèle permet de générer un graphe aléatoire ayant la distribution voulue. Ce modèle configurable part initialement de n nœuds. Chaque nœud sera pourvu d'un certain nombre de demi arrêtes qui respecte la distribution de degré qu'on désire avoir. L'algorithme de génération du graphe consiste à relier aléatoirement les demi-arrêtes. L'algorithme peut générer un graphe relativement complexe. Cette complexité peut être réduite en éliminant les multi-arrêtes et les boucles tout en gardant le plus grand composant connexe. La complexité du graphe est en conséquence réduite mais on dévie de la distribution qu'on s'est fixée au début. Ce modèle peut être configuré avec la distribution en loi de puissance observable dans les réseaux réels. Le paramètre de décroissance de la loi de puissance conditionne le comportement du graphe qu'on génère. Le choix de ce paramètre déterminera la présence ou non d'un composant géant et de sa proportion dans le graphe. Le modèle de Molloy et Reed s'applique bien aux calculs formels vu sa simplicité mais comme le modèle d'Erdös et Rényi, il présente un faible coefficient de clustering.

Le modèle de Watts-Strogotaz, le modèle des petits mondes

Aucun des deux modèles vus plus haut n'a pu résoudre le problème de la transitivité. Le modèle des petits mondes de Strogatz est parmi les premiers modèles qui ont aborder cette problématique. Ce modèle part de l'idée que les connexions entre les nœuds se font selon une préférence de proximité. En partant d'une structure initiale (anneau sur lequel on a répartie uniformément L nœuds), on relie chaque nœud avec les voisins situés à moins de k sauts. La deuxième étape consiste à prendre une proportion p des nœuds et de les recâbler de nouveau. L'opération de recâblage se fait d'une façon linéaire. Pour un nœud donné, on réoriente un lien avec une probabilité p vers un nœud choisi aléatoirement parmi les restants, tout en évitant le dédoublement des liens et les boucles. Le choix d'une probabilité p nulle nous donnera un graphe régulier avec un coefficient de clustering qui tend vers une constante proche de 1 quand k est grand. Le fort coefficient de clustering est pénalisé par l'absence de l'effet petit monde qu'on retrouve dans les réseaux réels. Cet effet est bien présent dans le cas où on a un graphe complètement aléatoire ($p = 1$) mais le coefficient de clustering devient relativement petit. Un compromis au niveau du choix de la probabilité p s'impose alors pour avoir l'effet petits mondes et un fort coefficient de clustering. Même si le choix de la probabilité permet de bien caler le modèle sur la réalité des réseaux réels, la distribution des degrés reste relativement éloignée des lois en puissance observées empiriquement. Le modèle des petits mondes a fait l'objet de quelques changements dans le souci de simplifier son traitement mathématique. Monasson, Newman et Watts ont élaboré un modèle où on accepte les boucles et le dédoublement des liens.

Les graphes aléatoires exponentiels

Les graphes aléatoires exponentiels ou les modèles $p*$ ont été élaborés par Strauss et Holland afin de combler les lacunes des premiers modèles incapables de retrouver l'effet des petits mondes qui se manifeste par un fort coefficient de clustering. Par analogie avec les lois de la physique statistique et les fameuses lois de Boltzman, un modèle a été élaboré. Ce modèle est loin d'être expressément analytique d'où le recours aux simulations pour retrouver les principaux paramètres de ce modèle.

C.2.2 Les modèles d'expansion

Le modèle de Price

Historiquement, le premier modèle d'expansion s'intéressait à l'étude des citations dans les articles scientifiques et tentait d'expliquer le phénomène de la loi de puissance qui est un phénomène commun à tous les réseaux réels. En effet, les graphes formés par l'étude de citations sont eux aussi des graphes sans échelle (distribution en loi de puissance). Les graphes de citations sont des graphes avec des liens orientés. Un lien orienté d'un nœud i vers un nœud j signifie que le papier i a cité le papier j dans sa bibliographie. De ce fait, le nombre de liens sortant d'un certain nœud restera constant par opposition au nombre de liens entrant qui lui évoluera au cours du temps. Price est parti des travaux de Simon qui a introduit la notion de l'effet Matthieu (les riches deviennent plus riche). C'est ce que Price a appelé la cumulation d'avantages et qui est devenu plus tard avec Barabasi et Albert l'attachement préférentiel. L'idée de base est de supposer que la probabilité qu'un nouveau nœud ait un lien sortant avec un vieux nœud est proportionnelle au degré entrant de ce dernier. En partant de cette hypothèse, on peut être confronté à un problème surtout pour les nœuds fraîchement inclus dans le graphe. Ces nœuds qui commencent souvent sans liens entrants (càd avec un degré entrant nul) ne pourront pas avoir de nouveaux liens entrants si on reste dans le cadre de l'hypothèse initiale d'attachement préférentiel. Une légère modification de cette hypothèse permet de contourner ce problème. La modification consiste à commencer non avec un degré nul mais avec un degré constant k_0. Généralement, on prend $k_0 = 1$ ce qui laisse entendre que la publication du papier est considérée comme une première citation. Le calcul de la distribution donne des résultats relativement fidèles à la réalité avec une loi de puissance :

$$p_k \sim k^{-(2+\frac{1}{m})}$$

avec $m = \sum_k kp_k$ supposé être constant dans le modèle de Price. C'est une loi de puissance avec un paramètre $\alpha = 2 + \frac{1}{m}$ qui ne dépend pas du degré initial k_0. Le modèle de price a été élaboré dans les années soixante. Vu la limitation en ressources calculatoires, Price n'a pas pu faire des simulations de son modèle et le modèle est resté inconnu par la communauté scientifique. Ce modèle est redécouvert par Barabasi et Albert quelques décennies plus tard.

Le modèle de Barabasi et Albert

Le modèle de Barabasi et Albert s'inspire du modèle de Price en adoptant la notion de rattachement préférentiel. Le modèle de Price traite des liens orientés. Avec le modèle de Barabasi et Albert les liens deviennent non orientés ce qui résout le problème du degré initial. Par ailleurs, une telle hypothèse fait émerger un problème de fiabilité de représentation surtout lorsqu'on s'intéresse aux graphes du Web et des citations. Le modèle de Barabasi et

Albert présente des concessions par rapport au réalisme mais tout cela est fait dans un but de simplicité mathématique. La distribution des degrés est une loi de puissance :

$$p_k = \frac{1}{k^3}$$

D'autres travaux présentent des résultats relatifs au modèle de Barabasi et Albert. On peut trouver par exemple des résultats concernant l'évolution du degré moyen d'un nœud en fonction de son âge : le degré moyen d'un nœud augmente avec l'âge. Ce résultat n'est pas confirmé dans le cas du web. Cela ne met pas en doute la fiabilité du modèle, le web est trop compliqué pour être modélisé par un simple modèle. D'autres résultats s'intéressaient à la corrélation observée entre les degrés des nœuds adjacents. D'autres travaux visent à généraliser ce modèle en acceptant par exemple un k_0 négatif. Krapivsky prend un modèle avec une relation non linéaire entre la probabilité d'attachement et le degré des nœuds :

$$Pr[attachement] \sim k^{\gamma}$$

Pour $\gamma < 1$, on retrouve une loi de puissance. Pour $\gamma > 1$, un phénomène de condensation s'établit. Pour $\gamma = 1$, on est dans le cas d'une relation linéaire étudiée par Barabasi et Albert.

Modèle de la copie des nœuds

Le modèle de Barabasi et Albert est simple mais comporte quelques lacunes surtout par rapport à la modélisation du graphe du web. Le modèle de Kleinberg avec sa notion de copie de nœuds figure parmi les principales alternatives au modèle de Barabasi et Albert pour la modélisation de l'expansion du réseau WEB. Le modèle consiste à supposer que le graphe évolue en ajoutant aléatoirement des nœuds d'une façon linéaire ou exponentielle [stochastic models for the web graph]. Parallèlement à ce mécanisme, un deuxième est mis en place : la copie des nœuds. On prend un nœud N qui existe déjà et on fixe un entier m. On choisit aléatoirement un autre nœud N'. On prend au hasard m liens du nœud N' et on génère les mêmes liens à partir du nœud N. Dans le cas où le nombre de liens du nœud N' est insuffisant, on prend la totalité de ses liens et on continue la même procédure avec un nouveau nœud pour copier à la fin exactement m liens. Kelinberg envisage également un mécanisme de suppression de liens. Le modèle donne une distribution selon une loi de puissance.

Bibliographie

[1] A. Abimbola, Q. Shi et M. Merabti. « Using Intrusion Detection to Detect Malicious Peer to Peer Network Traffic ». Dans *PAM2004*, 2004.

[2] Advestigo. « http ://www.Advestigo.com/ ».

[3] Allot. « http ://www.allot.com/ ».

[4] L. Arnold. *Stochastic Differential Equations : Theory and Applications.* John Wiley, 1974.

[5] K.B. Athreya et P. E. Ney. *Branching Processes.* 1972.

[6] N. Ben Azzouna. « *Etude des méthodes d'échantillonnage des flux pour la mesure dans les réseaux larges bande* ». Thèse de doctorat, 2004.

[7] L. Bernaille, R. Teixeira et K. Salamatian. « Early Application Identification ». Dans *Conference on Future Networking Technologies*, Lisbonne, Portugal, December 2006.

[8] Laurent Bernaille et Renata Teixeira. « Early Recognition of Encrypted Applications ». Dans *PAM 2007*, Louvain-la-neuve, Belgium, April, 2007.

[9] Laurent Bernaille, Renata Teixeira, Ismael Akodjenou, Augustin Soule et Kavé Salamatian. « Traffic Classification on the fly ». *ACM SIGCOMM Computer Communication Review*, 36(2), 2006.

[10] E. W. Biersack, P. Rodriguez et P. Felber. « Performance Analysis of Peer-to-Peer Networks for File Distribution ». Dans *QofIS'04*, Barcelona, Spain, 2004.

[11] S. Le Blond, M. Latapy et J. L. Guillaume. « Statistical analysis of a p2p query graph based on degrees and their time evolution ». Dans *IWDC 04*, Kolkata, India, 2004.

[12] C. Bordenave, D.R. McDonald et A. Proutière. « Random Multi-access Algorithms- A mean Field analysis ». Rapport Technique, 2005.

[13] J. Borland. « RIAA threat may be slowing file swapping ».

[14] Marc Boullé. « A Bayes Optimal Discretization Method for Continuous Attributes ». *Journal of Machine Learning*, 65 :131–165, 2004.

[15] Marc Boullé. « A Bayes Optimal Approach for Partitioning the Values of Categorical Attributes ». *Journal of Machine Learning Research*, 6 :1431–1452, 2005.

[16] Marc Boullé. « Une méthode optimale d'évaluation bivariée pour la classification supervisée ». Dans *EGC 2007*, Namur, Belgique, Januray, 2007.

[17] Marc Boullé. « Regularization and Averaging of the Selective Naïve Bayes classifier ». Dans *IJCNN 2006*, Vancouver,Canada, July, 2006.

[18] D. Burman. « Insensitivity in queing systems ». *Advances in Applied Probability*, 13 :846–859, 1981.

[19] F. Clévenot et P. Nain. « A Simple Fluid Model for the Analysis of the squirrel Peer-to-Peer Caching System ». Dans *Infocom 2004*, Hong Kong, China, 2004.

[20] T. M. Cover et J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[21] T. D. Dang, M. Perényi, A. Gefferth et S. Molnár. « On the Identification and Analysis of P2P Traffic Aggregation ». Dans *IFIP NETWORKING 2006*, Coimbra, Portugal, May 15-19, 2006.

[22] S. Das, S. Tewari et L. Kleinrock. « The Case for Servers in a Peer-to-Peer World ». Dans *ICC 2006*, Istanbul, Turkey, 2006.

[23] E. Dauce. « *Adaptation Dynamique et Apprentissage dans des Réseaux de Neurones Récurrents Aléatoires* ». Thèse de doctorat, 2000.

[24] D. L. Eager et K. C. Sevcik. « Bound Hierarchies for multiple-class queing networks ». *Journal of the ACM*, 33 :179–206, 1986.

[25] J. Erman, M. Arlitt et A. Mahanti. « Traffic Classification Using Clustering Algorithms ». Dans *Sigcomm' 06*, Pisa, Italy, September, 2006.

[26] F. Le Fessant, S. B. Handurukande, A.-M. Kermarrec et L. Massoulié. « Clustering in Peer-to-Peer File Sharing Workloads ». Dans *3rd International Workshop on Peer-to-Peer Systems (IPTPS)*, San Diego, USA, 2004.

[27] Filetopia. « http ://www.filetopia.org/ ».

[28] Anh-Tuan Gai, Fabien Mathieu, Fabien de Montgolfier et Julien Reynier. « Stratification in P2P networks, Application to BitTorrent ». Dans *International Conference on Distributed Computing Systems 2007*, Toronto, Canada, 2007.

[29] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose et D. Towsley. « Modeling Peer-to-Peer Sharing Systems ». Dans *Infocom 2003*, San Francisco, USA, 2003.

[30] D. Geiger et F. Girosi. « Parallel and deterministic algorithms from Mrfs :Surface Reconstruction ». *IEEE Transactions on Pattern Analysis and Machine intelligence*, 1 :401–412, 1991.

[31] Yimming Gong. « Identifying P2P users using traffic analysis », juillet 2005.

[32] G. R. Grimmet et D. R. Stirzaker. *Probability and Random processes*. 1995.

[33] K. P. Gummadi, Richard J. Dunn, S. Saroiu, S. D. Gribble, H.M. Levy et John Zahorjan. « Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload ». Dans *Proceedings of the 19th ACM Symposium of Operating Systems Principles (SOSP)*, Bolton Landing, NY, USA, 2003.

[34] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding et X. Zhang. « Measurments, Analysis and Modeling of BitTorrent-like Systems ». Dans *IMC 2005*, Berkeley, CA, USA, 2005.

[35] M. Haft, R. Hofmann et V. Tresp. « Model-independent mean field theory as a local method for approximate propagation of information ». Rapport Technique, 1997.

[36] S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant et L. Massoulié. « Exploiting Semantic Clustering in the eDonkey P2P Network ». Dans *11th ACM SIGOPS European Workshop (SIGOPS)*, Leuven, Belgium, 2004.

[37] S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant, L. Massoulié et S. Patarin. « Peer Sharing Behaviour in the eDonkey Network and Implications for the Design of Server-less File Sharing Systems ». Dans *EuroSys 06*, Leuven, Belgium, 2006.

[38] S. Hautphenne, K. Leibnitz et M. Remiche. « Extinction Probability in Peer-to-Peer File Diffusion ». *ACM SIGMETRICS Performance Evaluation Review Special issue on Performance*, 34 :3–4, 2005.

[39] Pew Internet et American Life Project. « Sharp Decline in music file swappers ». 2004.

[40] S. Iyer, A. Rowstron et P. Druschel. « Squirrel : A decentralized, peer-to-peer Web cache ». Dans *PODC 2002*, Monterey, California, 2002.

[41] P. Jacquet, Y. Suhov et N. Vvedenskaya. « Dynamic routing in the mean-field approximation ». Rapport Technique.

[42] A. Jean-Marie et P. Robert. « On the transient behavior of the processor sharing queue ». *Queing Systetems, Theory an Applications*, 17 :129–136, 1994.

[43] T. Karagiannis, Andre Broido, M. Faloutsos et K. Claffy. « Transport Layer Identification of P2P Traffic ». Dans *IMC'04*, October 2004 Taormina Italy.

[44] T. Karagiannis, Andre Broido, M. Faloutsos, K. Claffy et N. Brownlee. « Is P2P dying or just hiding ? ». Dans *IEEE*, October 2004.

[45] T. Karagiannis, K. Papagiannaki et M. Faloutsos. « BLINC :Multilevel Traffic Classification in the Dark ». Dans *ACM SIGCOMM*, Philadelphia, USA, 2005.

[46] R. Kumar, D.D. Yao, A. Bagchi, K. W. Ross et D. Rubenstein. « Fluid Modeling of Pollution Proliferation in P2P Networks ». Dans *Sigmetric06*, Saint-Malo, France, 2006.

[47] E. D. Lazowska, J. Zahorjan, G. S. Graham et K. C. Sevcik. *Quantitative System Performance : Computer System Analysis Using Queuing Network Models*. Prentice-Hall, Inc, 1984.

[48] S. Le-Blond, J. L. Guillaume et M. Latapy. « Clustering in P2P exchanges and consequences on performances ». Dans *4-th International workshop on Peer-to-Peer Systems IPTPS'05*, Ithaka, New York, USA, 2005.

[49] Dmitry Lebedev, Fabien Mathieu, Laurent Viennot, Anh-Tuan Gai, Julien Reynier et Fabien de Montgolfier. « On Using Matching Theory to Understand P2P Network Design ». Dans *International Network Optimization Conference 2007*, Spa, Belgium, 2007.

[50] K. Leibnitz, T. Hossfeld, N. Wakamiya et M. Murata. « Modeling of Epidemic Diffusion in Peer-to-Peer File-Sharing Networks ». Dans *BioAdit'06*, Osaka,Japan, 2006.

[51] D. Liberzon et A. Morse. « Basic problems in stability and design of switched systems ».

[52] Audible Magic. « http ://www.audiblemagic.com/ ».

[53] D.R. McDonald et J.Reynier. « Mean Field Convergence of a model of multiple TCP connections through a buffer implementing RED ». Rapport Technique, 2003.

[54] F. Baccelli D. McDonald et J. Reynier. « A Mean-field Model for Multiple TCP connections through a buffer Implementing RED ». Rapport Technique, 2002.

[55] D. A. Menascé, V. A. F. Almeida et L. W. Dowdy. *Capacity Planning for Web Services : metrics, models and methods*. Prentice-Hall, Inc, 2001.

[56] A. W. Moore et D. Zuev. « Internet Traffic Classification Using Bayesian Analysis Techniques ». Dans *SIGMETRICS'05*, Banff, Canada, 2005.

[57] M. Newman, C. Moore et D. Watts. « Mean-field solution of the small-world network model ».

[58] NIMI. « http ://www.ncne.org/nimi/ ».

[59] S. Ohzahata, Y. Hagiwara, M. Terada et K. Kawashima. « A Traffic Identification Method and Evaluations for a Pure P2P Application ». Dans *PAM2005*, August 2005.

[60] D.J. Parish, K. Bharadia, A. Larkum, I. W. Philips et M.A. Olivier. « Using packet size distributions to identify real-time networked applications ». Dans *IEEE*, August 2003.

[61] Pcube. « http ://www.p-cube.com/ ».

[62] F. Perronnin, P. Nain et K. Ross. « Multiclass P2P Networks : Static Resource Allocation for service Differentiation and Bandwidth Diversity ». *Elseiver Science*, 2005.

[63] N. Peyrard. « *Approximations de Type Champ Moyen des Modèles de Champ de Markov Pour La ségmentation de Données Spatiales* ». Thèse de doctorat, 2001.

[64] D. Piau. « Processus de Branchement en Champ Moyen ». *Advances in Applied Probability*, 33 :391–403, 2001.

[65] D. Piau. « Immortal Branching Markov Processes :Averaging Properties and PCR Applications ». *The Annals of Probability*, 0 :1–28, 2002.

[66] F. L. Piccolo et G. Neglia. « The Effect of Heterogenous Link Capacities in BitTorrent-Like File Sharing Systems ». Dans *HOT-P2P'04*, Volendam, The Nederlands, 2004.

[67] J. Pouwelse, P. Garbacki, D. Epema et H. Sips. « The BitTorrent P2P file-Sharing system : Measurement and analysis ». Dans *IPTPS'05*, Ithaca, NY, USA, 2005.

[68] D. Qiu et R. Sirkant. « Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer networks ». Dans *ACM SIGCOMM 2004*, Portland, USA, 2004.

[69] K. Ramachandran et B. Sikdar. « An analytic Framework for modeling p2p networks ». Dans *Infocom 2005*, Miami, USA, 2005.

[70] M. Ripeanu et I. Foster. « Internet Traffic Classification Using Bayesian Analysis Techniques ». Dans *IPTPS 02*, Cambridge, 2002.

[71] Jordan Ritter. « Why Gnutella Can't Scale. No, Really », 2000.

[72] M. Roughan, S. Sen, O. Spatscheck et N. Duffield. « Class-of-Service Mapping for QoS : A statistical Signature-based Approach to IP Traffic Classification ». Dans *IMC'04*, Taormina, Italy, 2004.

[73] S. Saroiu, P. Gummadi et S. D. Gribble. « Measurement study of Peer to Peer file Sharing Systems ». Dans *Mutilmedia Computing and Networking, MMCN '02, San Jose, USA*, 2002.

[74] L. Saul, T. Jaakkola et M. Jordan. « Mean Field Theory for Sigmoid Belief Network ». *Artificial Intelligence Research*, 4 :61–71, 1996.

[75] L. Saul et M. Jordan. « Exploiting Tractable Substructurs in Intractable Networks ». Advances of Neural Information Processing Sytems, 1995.

[76] S. Savage. « Sting : a TCP-based Network Measurement Tool ». Dans *USENIX Symposium on Internet Technologies and Systems*, 1999.

[77] S. Sen, O. Spatscheck et D. Wang. « Accurate Scalable In-Network Identification of P2P Traffic Using Application Signatures ». Dans *WWW2004 New York USA*, volume NJ, May 2004.

[78] S. Sen et J. Wang. « Analyzing Peer to Peer Traffic Across Large Networks ». *IEEE/ACM Transactions on Networking*, 12(2) :219–232, 2004.

[79] Slyck.

[80] C. Soldani. « Peer to Peer Behaviour Detection by TCP Flows Analysis ». Rapport de DEA, University of Liège, 2004.

[81] Y. Tian, D. Wu et K. W. Ng. « Modeling, Analysis and Improvement for BitTorrent-Like File Sharing Networks ». Dans *Infocom 2006*, Barcelona, Spain, 2006.

[82] K. S. Trivedi. *Probability and Statisticswith Reliability, Queing and Computer Science Applications*. John Wiley and sons, 2002.

[83] K. Tutschku. « A Measurement-based Traffic Profile of the eDonkey FileSharing Service ». 2003.

[84] G. Weiss et A. Von Haeseler. « Modeling the Polymerase Chain Reaction ». *Computational Biology 2*, 1 :49–61, 1995.

[85] W. Whitt. « The Queuing Network Analyzer ». *Bell System Technical Journal*.

[86] Winny. « http ://www.nynode.info/ ».

[87] C. V. Wright, F. Monrose et G. M. Masson. « On Inferring Application Protocol Behaviors in encrypted Network Traffic ». *Journal of Machine Learning Research*, 7 :2745–2769, 2006.

[88] X. Yang et G. De Veciana. « Service capacity of p2p Networks ». Dans *Infocom 2004*, Hong Kong, China, 2004.

[89] J. Zerubia et R. Chellappa. « Mean Field Approximation Using Compound Gauss-Markov random field for Edge Detection and Image Estimation ». *IEEE Transactions on Neural Networks*, 4 :703–709, 1993.

[90] L. Zou et M. H. Ammar. « A File-Centric Model for Peer-to-Peer File Sharing system ». Dans *ICNP 03*, Atlanta, USA, 2003.