UPMC
SORBONNE UNIVERSITÉS

habilitation à diriger des recherches
de l'université Pierre et Marie Curie

Spécialité
Informatique

Présentée par
Clémence Magnien

Intégrer mesure, métrologie et analyse pour l'étude des graphes de terrain dynamiques

Soutenue le 6 juillet 2010,
devant le jury composé de :

Patrice Abry, DR, CNRS	Examinateur
Vania Conan, Chercheur, Thales communications	Examinateur
Eric Fleury, Professeur, ENS Lyon	Examinateur
Pierre Fraigniaud, DR, CNRS	Examinateur
Eric D. Kolaczyk, Professeur, Boston University	Rapporteur
Guy Melançon, Professeur, Université Bordeaux I	Rapporteur
Jean-Jacques Pansiot, Professeur, Université de Strasbourg	Rapporteur
Franck Petit, Professeur, Université Pierre et Marie Curie	Examinateur

Remerciements

Depuis le début de ma thèse, j'ai eu la chance de rencontrer et d'interagir avec un grand nombre de personnes différentes. Collaborer avec elles, ou simplement discuter, a été un très grand apport, à la fois sur le plan scientifique car elles m'ont énormément appris, et sur le plan humain, car elles ont contribué à tisser un cadre de travail extrêmement agréable.

Ces personnes sont trop nombreuses et mon cerveau trop petit pour que je puisse espérer les citer toutes nommément. J'exprime donc tous mes remerciements à mes collègues (toutes catégories professionnelles confondues) du LIAFA et du LIX (les deux laboratoires où j'ai effectué ma thèse), du CREA (où j'ai été recrutée en tant que chargée de recherche CNRS), du laboratoire J.-V. Poncelet (où j'ai effectué un séjour d'un an), du LIP6, mon laboratoire actuel, et à tous les autres.

Je réserve une pensée spéciale pour les stagiaires et les doctorants que j'ai encadrés. Cela a été des expériences à chaque fois différentes et toujours enrichissantes. Travailler avec eux a été un plaisir, et je les remercie pour la confiance qu'ils m'ont accordée.

J'ai rejoint le LIP6 en janvier 2007, en même temps que Matthieu Latapy, afin d'y développer une activité de recherche sur les graphes de terrain. Cette activité s'est concrétisée par la création de l'équipe *Complex Networks*, à laquelle je suis ravie et fière d'appartenir. Je tiens donc à remercier l'ensemble de ses membres, avec qui je travaille (et fais des pauses) au quotidien, pour leur ouverture, leurs réflexions, leur bonne humeur et leur soutien. Ce mémoire leur est dédié.

Mes interactions dans le laboratoire ne sont pas restreintes aux membres de mon équipe, et je saisis cette occasion pour exprimer à l'ensemble de mes collègues du LIP6 mon plaisir à travailler parmi eux. Je me suis sentie bien dans le laboratoire dès mon arrivée, et la richesse des thématiques des différentes équipes en fait un cadre de travail très profitable.

Finalement, je remercie infiniment Eric D. Kolaczyk, Guy Melançon et Jean-Jacques Pansiot d'avoir accepté d'être rapporteurs pour ce mémoire, de s'y être plongés et de m'avoir fait part de leurs commentaires. Je remercie également l'ensemble des membres de mon jury de m'avoir fait l'honneur d'accepter de donner leur avis sur mon travail.

INTRODUCTION

On peut modéliser de nombreux objets issus du monde réel par des graphes. Citons par exemple la topologie de l'internet (au niveau des routeurs ou des systèmes autonomes), les graphes du web (ensembles de pages web et liens hypertextes entre elles), les réseaux métaboliques (réactions entre protéines au sein d'une cellule), les connexions dans le cerveau, les réseaux sociaux comme les réseaux d'amitié, de communication ou de collaboration, et les réseaux de transport.

Depuis une dizaine d'années, un grand nombre de travaux s'intéresse à ces objets, suite à la découverte que, bien qu'ils soient issus de contextes différents, ils se ressemblent au sens de certaines propriétés statistiques [121, 16]. Ces travaux ont étudié une grande variété de graphes et fait des observations importantes dans ces divers contextes. D'autre part, le fait qu'ils se ressemblent a fait émerger plusieurs questions *transversales*, c'est-à-dire s'appliquant à l'ensemble de ces graphes. On peut citer en particulier l'étude de leur structure en communautés, c'est-à-dire en groupes de nœuds fortement reliés les uns aux autres et faiblement liés à des nœuds d'autres groupes [120, 39, 36, 19], l'introduction de diverses propriétés statistiques pour leur description, ou encore leur robustesse face à des pannes ou des attaques [8, 29, 22, 75]. Ceci a montré qu'il est effectivement pertinent de considérer ces objets comme un ensemble cohérent. C'est pourquoi on les désigne sous le terme général de *graphes de terrain* (*complex networks* en anglais).

Il est possible de regrouper les différentes questions liées à l'étude des graphes de terrain en grandes familles de problématiques [63]. En particulier, les graphes de terrain ne sont pas donnés *a priori* et la connaissance des nœuds et liens qui les composent passe par une opération de *mesure*, dont la nature dépend du graphe étudié. Remarquons que dans l'immense majorité des cas, la mesure ne peut espérer capturer l'intégralité du graphe, en particulier en raison de sa taille et de différentes autres contraintes. Il a été montré que l'échantillon collecté peut avoir des propriétés statistiques différentes de celles du graphe de départ, et que donc la mesure induit un biais dans les observations. La *métrologie* vise à étudier ce biais, le corriger et/ou proposer des méthodes capables de capturer certaines propriétés de manière fiable. La taille des graphes de terrain est en général très grande (on parle dans beaucoup de cas de centaines de milliers, voire de millions ou de milliards de nœuds et de liens), ce qui rend impossible, une fois les données collectées, de comprendre leur structure par une observation directe. L'*analyse* vise à décrire cette structure, par l'introduction de propriétés statistiques (distribution

des degrés, coefficient de *clustering*, etc.) ou structurelles (hiérarchie de communautés par exemple) qui résument l'information et en soulignent les principales caractéristiques. On peut citer également les problématiques liées à la modélisation, l'algorithmique, et les phénomènes ayant lieu sur des graphes de terrain (par exemple des phénomènes de diffusion), que je ne détaille pas car elles sortent du contexte de ce mémoire.

Ces différentes problématiques ne sont bien sûr pas complètement indépendantes les unes des autres. La mesure et la métrologie sont par exemple un préalable indispensable à l'analyse rigoureuse. Réciproquement, l'identification de propriétés d'intérêt par l'analyse peut conduire au développement d'opérations de mesure ciblées pour capturer ces propriétés. Cependant, cette séparation des problématiques permet d'avoir une compréhension synthétique du domaine et des différentes contributions, et de positionner de nouveaux travaux par rapport à l'existant.

Finalement, même si beaucoup reste à faire, le domaine des graphes de terrain a acquis une certaine maturité, et a conduit à un ensemble de notions et de techniques pertinentes pour l'étude de n'importe quel graphe de terrain, qui sont des guides précieux pour tout chercheur entamant l'étude d'un tel graphe.

La plupart des graphes de terrain sont dynamiques, c'est-à-dire que leur structure évolue au fil du temps par l'ajout et/ou le retrait de nœuds et/ou de liens, à des fréquences plus ou moins grandes [1]. La grande majorité des travaux qui les ont étudiés les ont cependant considérés comme statiques, c'est-à-dire qu'ils ont considéré un instantané d'un graphe, capturé à un instant donné. Ceci est naturel, car entamer d'emblée l'étude de la dynamique sans connaissances préalables de la structure est une tâche extrêmement ardue, et l'étude des graphes statiques a produit un grand nombre de résultats importants. La dynamique est cependant une composante fondamentale des graphes de terrain, et le domaine a aujourd'hui atteint une maturité suffisante pour aborder son étude.

Ceci s'est traduit par l'apparition, depuis quelques années, de travaux sur la dynamique des graphes de terrain. Certains se sont intéressés à des cas particuliers, comme les contacts entre personnes mesurés à l'aide de capteurs ou de téléphones *bluetooth* par exemple [21, 23, 27, 99, 113], les échanges pair-à-pair [67, 45, 68], la topologie de l'internet [65, 81, 76, 87, 85, 60], des réseaux biologiques [14, 91, 116, 105], des réseaux de citations d'articles [72], et plusieurs types de réseaux sociaux en ligne [30, 106, 100]. D'autres ont abordé des questions transversales. On peut citer par exemple des méthodes permettant de manipuler des graphes dynamiques [20, 57], la détection ou le suivi de communautés dynamiques [15, 72, 84], l'étude ou la formation de motifs spécifiques [117, 24, 61], le dessin de graphes dynamiques [37, 40], ou des études plus générales [11, 64, 73, 74, 7, 80].

Ma recherche est centrée sur les graphes de terrain et j'ai principalement contribué à l'étude de leur dynamique, sous les angles de la mesure, de la métrologie et de l'analyse (chapitres 1, 2 et 3 de ce mémoire respectivement). J'ai contribué à la mesure de l'activité dans le système pair-à-pair *eDonkey* et de la topologie de l'internet. Dans ce deuxième cas, j'ai participé à la création d'une nouvelle approche permettant d'étudier certains aspects de la dynamique avec une granularité temporelle beaucoup plus fine que ce qui

1. J'insiste sur le fait que je traite dans ce mémoire les dynamiques qui modifient la structure du graphe, c'est-à-dire ajoutent et suppriment des nœuds et des liens, et non des dynamiques qui se produisent *sur* des graphes, comme des diffusions épidémiques par exemple. Dans ce deuxième cas, ce sont les états des nœuds qui évoluent au fil du temps.

était possible auparavant. J'ai étudié dans plusieurs contextes les biais liés à la dynamique induits dans la mesure. Je me suis notamment intéressée à une dynamique particulière, le *load balancing* (équilibrage de charge) sur l'internet, et montré qu'elle peut fausser les données observées. Je me suis également intéressée à la dynamique de la mesure elle-même. Une opération de mesure d'un graphe de terrain est en effet loin d'être un processus instantané, et étudier la façon dont l'échantillon collecté évolue au fur et à mesure qu'elle progresse permet de tirer des conclusions sur la fiabilité des propriétés mesurées. Enfin, j'ai traité le cas de propriétés intrinsèquement dynamiques, et montré que le fait que la durée de mesure soit finie induit un bais dans leur caractérisation. Ceci est une question générale de métrologie dynamique. Enfin, mes travaux sur l'analyse de la dynamique se situent dans deux contextes différents : celui d'une dynamique *subie* par le graphe, à savoir la manière dont un graphe résiste à des suppressions de nœuds causées par des pannes et des attaques, et la dynamique d'un graphe qui évolue au fil du temps, à savoir une partie de la topologie de l'internet.

J'ai choisi de présenter mes travaux selon un découpage lié aux problématiques plutôt qu'aux objets d'étude. Ceci rend parfois la présentation un peu difficile mais permet à mon avis de mettre mes travaux en perspective, et de mettre en évidence le caractère général de certains d'entre eux.

Ce plan montre bien que la mesure, la métrologie et l'analyse sont toujours des familles de problématiques pertinentes dans le contexte de l'étude de graphes de terrain dynamiques. En présentant mes conclusions, je montrerai cependant que la prise en compte de la dynamique rend ces problématiques beaucoup plus liées qu'il n'y paraît au premier abord, à un point tel qu'il est difficile de dissocier l'analyse de la métrologie et de la mesure. En particulier, l'étude de certaines propriétés de la dynamique n'a pas de sens si on ne prend pas en compte la façon dont les données ont été collectées. Ceci ne remet pas en cause la distinction entre ces problématiques, mais indique à mon avis qu'il y a des principes fondamentaux sous-jacents à la dynamique qui restent à découvrir.

Bien entendu, je n'ai pas réalisé seule les travaux que je présente dans ce mémoire, et je suis profondément redevable à tous mes collaborateurs (non seulement ceux avec qui j'ai co-rédigé des articles, mais également ceux avec qui j'ai eu l'occasion d'avoir des discussions et des échanges d'idées fructueux). Sans leur aide, leur soutien et leurs intuitions, j'aurais été incapable d'effectuer une synthèse raisonnée de ces travaux, que je considère comme le principal apport de ce mémoire.

CHAPITRE 1

MESURE

Dans le cas général, les graphes de terrain ne sont pas connus *a priori*. La *mesure* est l'opération qui permet d'acquérir des informations sur les nœuds et liens présents dans le graphe. La méthode de mesure dépend naturellement de l'objet étudié. Pour la topologie de l'internet, on emploie souvent des outils tels que `traceroute`, qui permet d'obtenir (en principe) la route entre une machine source et une machine destination ; pour les graphes du web, on utilise des techniques similaires à des parcours en largeur, consistant à télécharger une page web, puis toutes les pages vers lesquelles elle pointe, et ainsi de suite ; pour les échanges pair-à-pair, il faut définir des méthodes permettant d'interroger le système sur les pairs connectés et les échanges qu'ils effectuent ; etc.

Dans tous les cas, on est confronté à la taille de l'objet étudié et à des contraintes techniques telles que l'espace de stockage ou la bande passante disponible (au niveau de l'outil de mesure mais également des systèmes interrogés). Dans le cas général, il est impossible de mesurer l'objet entier.

On peut distinguer deux grandes familles de méthodes permettant d'acquérir de l'information sur un graphe de terrain dynamique. Dans un cas, il s'agit de collecter des événements ponctuels. Par exemple, si l'on s'intéresse au graphe où deux personnes sont reliées si l'une téléphone à l'autre, chaque appel correspond à un lien dans ce graphe, existant au moment de l'appel ; de même, capturer les messages de mise à jour d'un routeur BGP permet d'avoir des informations sur les changements dans le graphe de routage entre systèmes autonomes (AS). Dans l'autre cas, il s'agit de répéter périodiquement une exploration du graphe. Dans les deux cas, l'architecture de mesure a des limites et il faut établir un compromis entre la fréquence de mesure, sa durée, la quantité d'information récoltée et le nombre de points de mesure ; de plus la mesure impose une charge importante au système étudié, ce qui peut entraîner une dégradation de la qualité des données obtenues [1], mais également être interprété comme une attaque ou même nuire à son fonctionnement [2].

Une opération de mesure a donc en général plusieurs paramètres importants, comme le nombre de points de mesure, la fréquence de mesure et la quantité d'information

1. Il est par exemple connu que certains routeurs peuvent arrêter temporairement de répondre aux sondes si l'outil `traceroute` est utilisé à une fréquence élevée [66].

2. Par exemple, télécharger un grand nombre de pages simultanément sur un site web peut le saturer.

recherchée. Il est important de tester ces paramètres de manière rigoureuse, en évaluant l'impact des différents choix possibles. Ceci a un coût important, mais c'est à ce prix que le processus de mesure est maîtrisé et que l'on obtient des données de qualité, permettant des analyses fiables.

Je présente dans la section 1.1 les mesures que j'ai effectuées dans le système *eDonkey*, par deux méthodes différentes [6, 9]. J'ai également contribué au développement d'une nouvelle approche pour mesurer et étudier la dynamique de la topologie de l'internet [65], que je présente dans la section 1.2.

1.1 Mesure du système pair-à-pair *eDonkey*

Les systèmes *pair-à-pair* (*peer-to-peer*) sont beaucoup utilisés pour échanger des fichiers de manière décentralisée entre utilisateurs : les utilisateurs (ou *pairs*) se connectent au système, proposent aux autres utilisateurs certains fichiers qui sont en leur possession, et téléchargent des fichiers fournis par d'autres utilisateurs.

Il existe plusieurs types de systèmes pair-à-pair. Certains sont complètement décentralisés, comme par exemple *Gnutella* ou *Kad*. D'autres gardent un aspect centralisé ou semi-centralisé. Dans ce cas, un ou plusieurs serveurs indexent les fichiers et/ou les utilisateurs fournissant ces fichiers. C'est le cas notamment des systèmes *eDonkey* et *BitTorrent*.

L'étude des échanges de fichiers ayant lieu dans ces systèmes est importante pour plusieurs raisons. Tout d'abord, comprendre les usages permet de concevoir des protocoles efficaces, car adaptés au type d'échanges qu'ils auront à gérer en pratique. Ensuite, ces usages présentent un intérêt sociologique (comprendre les comportements des utilisateurs et leurs goûts) et économique (savoir quels types de fichiers sont les plus demandés, suivre l'évolution de cette demande, etc.). Enfin, ces systèmes peuvent servir à échanger des contenus illicites, notamment des contenus à caractère pédopornographique. Être capable de détecter automatiquement les utilisateurs impliqués dans ces échanges et de comprendre leurs usages est donc une question clé pour les forces de l'ordre.

De nombreux travaux ont effectué des mesures de l'activité sur des systèmes pair-à-pair, suivant diverses approches. Il est possible de mesurer une partie de cette activité en créant un client pair-à-pair qui se connecte au système et enregistre de manière passive toute l'activité qui transite par lui [3, 5, 51, 52, 104, 26]. Ceci est particulièrement adapté pour les systèmes complètement décentralisés, dans lesquels les clients se chargent tous d'une partie de l'indexation et du partage des fichiers, ainsi que du maintien de la connexion des pairs au système. Une autre approche consiste à déployer des clients qui font des requêtes au système afin de récupérer des informations sur les pairs présents et les fichiers qu'ils échangent [97, 49, 69, 79, 122, 90]. Il est également possible de découvrir itérativement les pairs connectés dans le système *Gnutella* [97, 110]. Quand un système est centralisé ou semi-centralisé, les échanges peuvent être capturés au niveau d'un serveur, ce qui fournit une information riche sur tous les clients connectés et les fichiers qu'ils partagent pendant la période de mesure [45, 67, 68, 54]. Enfin, il est également possible de capturer le trafic internet, sur un routeur par exemple, et de l'analyser pour y détecter le trafic pair-à-pair [71, 101, 96, 56, 115].

Mes travaux dans ce domaine concernent le système *eDonkey*, qui est l'un des sys-

tèmes les plus utilisés. Il s'agit d'un système semi-centralisé : plusieurs serveurs indexent les fichiers disponibles. Lorsqu'un utilisateur souhaite acquérir un fichier, il envoie une requête contenant des mots-clés décrivant le fichier recherché (par exemple un nom d'artiste ou un titre de film). Le serveur répond par une liste de descriptions de fichiers correspondant à cette requête. L'utilisateur peut ensuite choisir un ou plusieurs fichiers dans cette liste et envoyer une requête au serveur pour connaître des fournisseurs pour ces fichiers. Le serveur envoie une liste de fournisseurs, et son rôle s'arrête là : l'utilisateur contacte ensuite directement les fournisseurs pour effectuer l'échange du fichier [3].

Il est possible de mesurer l'activité sur ce système de plusieurs façons, et j'ai participé à des mesures effectuées selon les méthodes suivantes :

- auprès d'un serveur : il s'agit de capturer les requêtes reçues par un serveur ainsi que les réponses qu'il fournit ;
- par un client effectuant des requêtes auprès d'un ou plusieurs serveurs : on peut ainsi découvrir des fichiers correspondant à un mot-clé donné, ainsi que leurs fournisseurs ;
- par un client prétendant proposer des fichiers (*honeypot*) : en envoyant au serveur une liste de fichiers qu'il prétend avoir, un client peut ensuite enregistrer des requêtes d'autres utilisateurs pour ces fichiers.

Remarquons que dans chacun de ces cas on s'intéresse à l'*activité* des utilisateurs (les fichiers fournis et demandés), et jamais au *contenu* des fichiers eux-mêmes, que nous ne téléchargeons pas.

Ces trois manières de mesurer sont complémentaires : la mesure sur serveur capture l'intégralité de l'activité sur ce serveur, mais est coûteuse et ne permet pas d'observer les autres serveurs ; les clients effectuant des requêtes ne voient qu'une (très petite) partie de l'activité globale, mais peuvent étudier une thématique ciblée sur un grand nombre de serveurs. De plus, cette approche est beaucoup simple à mettre en place qu'une mesure sur serveur. Enfin, la mesure par *honeypot* permet d'observer les échanges entre utilisateurs, ce qui n'est pas le cas avec les autres méthodes. Ces méthodes permettent toutes trois de récolter des informations riches sur l'activité dans le système *eDonkey*. Je détaille ci-dessous les mesures effectuées auprès d'un serveur, et les mesures par *honeypot*.

Avant d'aller plus loin, je dois insister sur le fait que collecter des données sur les échanges ayant lieu dans de tels systèmes soulève d'importantes questions légales et éthiques. En effet, ces données concernent des échanges effectués par des personnes réelles et contiennent donc des informations personnelles. De plus elles concernent souvent des échanges de fichiers illégaux. Pour cette raison, toutes les données décrites ici sont complètement anonymisées au moment de la collecte. Les adresses IP sont remplacées par des entiers de manière cohérente (une même adresse correspond toujours au même entier et réciproquement). Les fichiers, et notamment leurs noms, peuvent également correspondre à des données personnelles [10, 4]. C'est pourquoi nous anonymisons tant leur identifiant que leur nom. Enfin, les mots-clés entrés par les utilisateurs sont également anonymisés.

1.1.1 Mesure sur un serveur

Nous avons effectué une mesure en continu du trafic sur un serveur *eDonkey* important, pendant une durée de dix semaines [6]. Nous avons observé pendant cette mesure

3. Ceci est une schématisation du fonctionnement exact du système, qui est plus complexe. Voir par exemple [58] pour plus de détails.

89 884 526 utilisateurs (identifiés par leur adresse IP) et 275 461 212 fichiers distincts, ce qui fait de cette trace, publiquement disponible [34], la plus grosse donnée existante sur les échanges pair-à-pair.

Les principales difficultés dans cette mesure étaient liées d'une part au décodage du trafic IP en messages *eDonkey*, et d'autre part à l'anonymisation à la volée des données.

Après des discussions avec l'administrateur du serveur visant à s'assurer que nous ne perturberions pas le fonctionnement du serveur, nous avons décidé de mettre en place un programme de copie du trafic qui le renvoyait vers une machine de capture dédiée. Nous avons utilisé *libpcap*, une bibliothèque standard de capture de trafic. Ceci a entraîné des pertes de paquets. Vu la difficulté de reconstituer des flux TCP dans ce contexte, nous avons uniquement étudié le trafic UDP, qui représente environ la moitié du trafic global.

Il s'agissait ensuite de décoder les paquets UDP pour en extraire les messages *eDonkey*. Ce trafic est engendré par une grande quantité de clients différents, parfois peu fiables, ayant leur propre interprétation du protocole. De plus, certains paquets étaient mal formés et/ou ne correspondaient pas au protocole *eDonkey*. Décoder ce trafic hétérogène s'est donc avéré très complexe, et a requis une grande quantité de travail manuel.

La difficulté de l'anonymisation tient à l'énorme quantité de données que nous avons eu à gérer. Nous n'avons pas utilisé de techniques classiques, telles que des fonctions de hachage. En effet, il est possible dans ce cas de construire un dictionnaire inverse, en particulier pour les adresses IP qui sont relativement peu nombreuses : il suffit pour cela de calculer le *hash* de chaque adresse. Nous avons fait donc le choix d'anonymiser les adresses IP, les identifiants de fichiers et les chaînes de caractères par un entier représentant leur ordre d'apparition dans nos traces, ce qui de plus facilite énormément la manipulation des données *a posteriori*.

Étant donnée l'énorme quantité de messages *eDonkey* à anonymiser, il n'était pas possible de stocker la correspondance entre un objet et son anonymisation dans des structures de données classiques : il nous fallait en effet effectuer des millions d'insertions et des milliards de recherches, tout cela en un espace compact et en temps réel.

Pour les adresses IP, nous avons créé un tableau de 2^{32} entiers permettant de stocker les identifiants de toutes les adresses IP possibles. Ceci a représenté un fort coût en mémoire vive (le tableau occupait 16 Go), mais a rendu l'anonymisation très efficace.

Il n'a pas été possible d'utiliser cette méthode pour les identifiants de fichiers, qui font 128 bits. Nous avons choisi de stocker les identifiants observés et leur anonymisation dans des tableaux triés, afin de réduire au maximum l'espace utilisé. L'insertion dans un tableau trié ayant un coût très élevé, nous avons choisi de diviser l'ensemble des identifiants en 65 536 tableaux, chacun correspondant à une valeur des deux premiers octets de l'identifiant.. Bien que les identifiants des fichiers soient des *hash* de leur contenu, nous avons constaté qu'ils n'étaient pas répartis uniformément. Ceci indique qu'on est en présence de nombreux identifiants fabriqués qui ne correspondent pas à des fichiers [70], ce qui a compliqué notre tâche : si l'on n'y prend pas garde, on divise l'ensemble des identifiants en tableaux de tailles très inégales, ce qui ralentit le processus. Nous avons au final choisi deux octets au hasard pour indexer nos tableaux.

Nous avons également anonymisé les chaînes de caractères présentes dans les données. Au final, nous avons été capables de décoder le trafic UDP en temps réel, en utilisant 24 Go de mémoire vive au total. Les données anonymisées ont ensuite été formatées en

fichiers XML, afin d'être facilement exploitables.

Afin d'illustrer la richesse de ces données, nous présentons ci-dessous quelques analyses préliminaires. On définit la *popularité* d'un fichier comme le nombre d'utilisateurs différents ayant demandé ou fourni ce fichier. La distribution des popularités des fichiers, que nous ne présentons pas ici, est hétérogène : bien qu'un très grand nombre de fichiers aient une popularité inférieure ou égale à 10, 46 d'entre eux ont une popularité supérieure à 50 000.

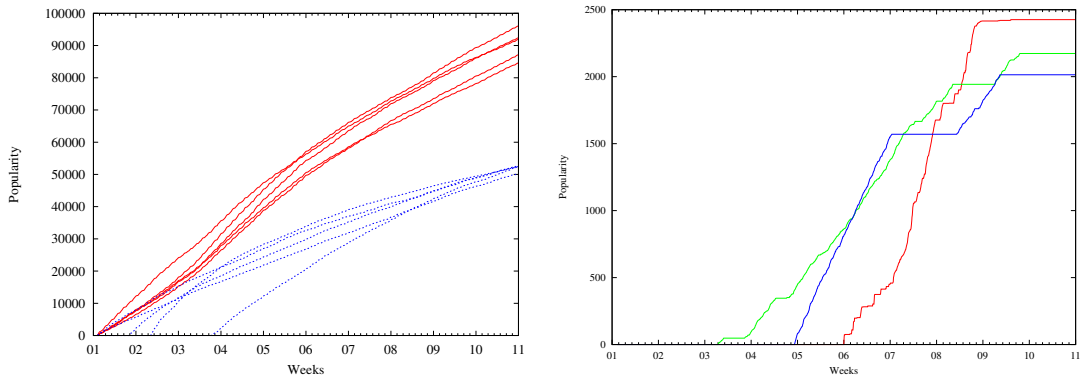

FIGURE 1.1 : Gauche : évolution au fil du temps de la popularité des cinq fichiers les plus populaires et des cinq les moins populaires, parmi les 46 fichiers ayant une popularité supérieure à 50 000. Droite : évolution au fil du temps de la popularité de quelques fichiers représentatifs, parmi ceux dont on observe l'apparition et la disparition.

La figure 1.1 (gauche) présente l'évolution au fil du temps de la popularité de quelques uns de ces 46 fichiers. On voit clairement qu'elle grandit en continu dès que le fichier apparaît. On voit également que certains fichiers apparaissent relativement tard dans nos mesures (jusqu'à trois semaines après le début de la capture).

On voit donc que les fichiers populaires le restent pendant une longue période. Ceci implique que, bien que l'on puisse observer l'apparition de fichiers populaires, on n'observe pas leur disparition dans nos données. Afin d'étudier cet aspect, nous avons retenu les fichiers les plus populaires parmi ceux qu'on n'observe ni dans la première ni dans la dernière semaine de mesure. Parmi ces fichiers, seulement 26 ont une popularité supérieure à 2 000. La figure 1.1 (droite) présente l'évolution de la popularité de quelques uns d'entre eux. On voit que cette évolution est très différente de celle observée pour les fichiers les plus populaires. Leur apparition est suivie d'une période relativement courte durant laquelle ils sont téléchargés par un grand nombre d'utilisateurs, puis ils disparaissent de notre mesure. Remarquons que cela ne signifie pas nécessairement qu'aucun utilisateur n'est plus intéressé par ces fichiers : il est possible que plus aucun fournisseur ne soit présent par exemple, auquel cas ils n'apparaîtront pas dans les listes de fichiers fournies par le serveur, et ne seront donc demandés par aucun utilisateur. Ceci pourrait également expliquer les plateaux observés sur certaines courbes.

En conclusion, nous avons effectué une mesure de longue durée et obtenu des données extrêmement riches. Une grande quantité de travail a été nécessaire afin de décoder le trafic UDP en messages *eDonkey* intelligibles. De plus, une étude préalable a été nécessaire afin de mettre au point une stratégie de décodage et d'anonymisation efficace.

1.1.2 Mesure par *honeypot*

Nous avons expérimenté la possibilité d'effectuer des mesures par *honeypot*. Le but était ici de tester la faisabilité de ces mesures et d'identifier les valeurs pertinentes pour les paramètres [9]. Les données obtenues (qui représentent plusieurs centaines de milliers de pairs et de fichiers, observés sur une période d'un mois) sont disponibles sur demande.

Notre *honeypot* est un client *eDonkey* qui se connecte à un serveur, et annonce à ce serveur des fichiers (qu'en fait il ne possède pas). Quand un autre pair effectue une requête pour l'un de ces fichiers, le serveur peut donc lui indiquer notre *honeypot* comme source. Certains pairs contactent ensuite le *honeypot* pour télécharger un fichier. Notre *honeypot* stocke ces messages. De plus, il demande à chaque pair qui le contacte la liste des fichiers qu'il partage, ce qui représente une information intéressante et lui permet de plus d'annoncer éventuellement de nouveaux fichiers. Nous insistons sur le fait que le *honeypot* ne possède pas les fichiers qu'il annonce, et qu'aucun contenu n'est jamais transmis aux pairs.

De nombreux paramètres ont une influence sur ces mesures : les fichiers annoncés, leur nombre, le nombre de *honeypots*, la durée de la mesure, et ainsi de suite. Afin d'évaluer l'influence de ces paramètres et de voir ce qu'il est possible de faire avec cette approche, nous avons effectué des mesures avec des paramètres différents :

- la mesure *distribuée* a utilisé 24 *honeypots* situés sur des nœuds PlanetLab [89], pendant une durée d'un mois. Tous les *honeypots* annonçaient les quatre mêmes fichiers (un film, une chanson, une distribution de linux et un texte). Ils étaient tous connectés à un même serveur ;
- la mesure *gloutonne* a été effectuée depuis seulement un *honeypot*, pendant une durée de deux semaines. Elle visait à annoncer le plus de fichiers possible. La mesure s'est faite en deux phases : pendant la première journée, le *honeypot* a demandé leur liste de fichiers partagés à tous les pairs qui l'ont contacté, et a ajouté ces fichiers à sa propre liste de fichiers annoncés ; les journées suivantes, il n'a plus annoncé de nouveaux fichiers et a simplement enregistré les requêtes qu'il a reçues et les listes de fichiers envoyées par les pairs qui l'ont contacté.

Comme nous avons effectué des mesures distribuées, il n'était pas possible d'anonymiser au moment de la mesure les adresses IP des pairs en les remplaçant par un entier représentant l'ordre dans lesquelles nous les avons observées. Cela n'aurait en effet pas permis de savoir si un même pair avait contacté deux *honeypots* différents. Nous avons donc choisi d'utiliser une anonymisation en deux étapes. Dans un premier temps, chaque *honeypot* a anonymisé les adresses IP à la volée en les remplaçant par un *hash* (tous les *honeypots* ont utilisé la même fonction). Les données ont ensuite été centralisées par un moniteur chargé de remplacer les *hash* par des entiers, dans leur ordre d'apparition.

La table 1.1 résume les caractéristiques principales des données obtenues. On voit que notre méthode permet d'obtenir de grandes quantités de données, sur des centaines de milliers de pairs et de fichiers.

Nous avons mené une étude sur l'impact des différents paramètres de mesure sur les données obtenues. Nous avons tout d'abord étudié l'impact de la durée de mesure. La figure 1.2 (gauche) présente le nombre de pairs distincts observés par la mesure distribuée en fonction de la durée de la mesure, ainsi que le nombre de nouveaux pairs découverts

	distribuée	**gloutonne**
Nombre de *honeypots*	24	1
Durée (en jours)	32	15
Nombre de fichiers annoncés	4	3 175
Nombre de pairs observés	110 049	871 445
Nombre de fichiers observés	28 007	267 047

TABLE 1.1 : Statistiques élémentaires des données récoltées par nos mesures *honeypot*.

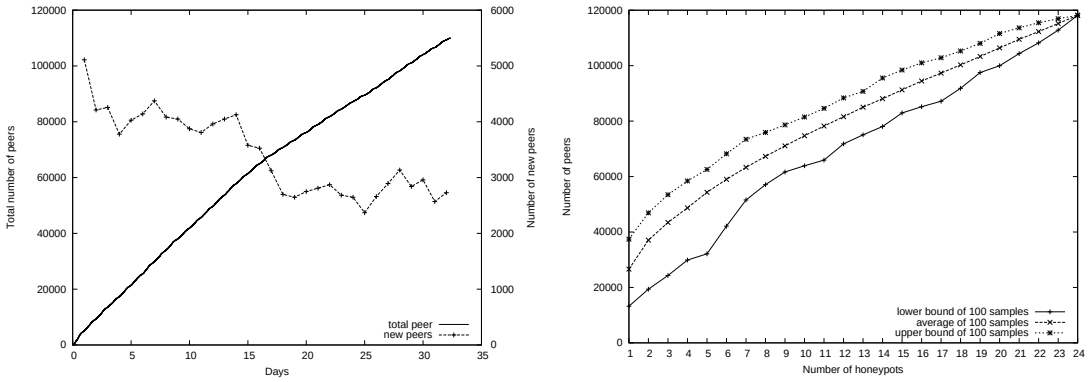

FIGURE 1.2 : Gauche : nombre de pairs distincts observés par la mesure distribuée (axe vertical de gauche) et de nouveaux pairs observés chaque jour (axe vertical de droite) en fonction du temps écoulé depuis le début de la mesure (en jours). Droite : nombre de pairs distincts observés à la fin de la mesure, en fonction du nombre n de *honeypots* utilisés. Pour chaque n, nous avons choisi 100 ensembles aléatoires de n *honeypots*, et nous présentons les valeurs moyennes, minimales et maximales observées sur ces 100 ensembles.

chaque jour. On observe que le nombre de pairs distincts observés augmente rapidement en fonction de la durée de la mesure, et de manière quasiment linéaire.

Lorsque l'on étudie le nombre de pairs découverts chaque jour, on observe qu'il décroît au fil du temps. Ceci est probablement dû au fait que la popularité des fichiers annoncés diminue, ou alors que la plupart des pairs intéressés par ces fichiers les ont déjà acquis (ou ont en tout cas déjà contacté l'un de nos *honeypots*). Il est également possible que le fait que nos *honeypots* ne fournissent aucun contenu ait été détecté, et que certains aient été placés sur liste noire. Il faut tout de même constater que cette décroissance est légère, et qu'au bout de 30 jours de mesure on observe encore plus de 2500 *nouveaux* pairs chaque jour. Ceci montre donc qu'il est pertinent d'effectuer des mesures pendant de longues durées, car cela permet de découvrir un nombre important de nouveaux pairs.

Nous avons utilisé la mesure distribuée pour étudier l'impact du nombre de *honeypots* sur les mesures. Il n'est en effet pas évident que l'utilisation de beaucoup de *honeypots* soit nettement plus efficace que de n'en utiliser qu'un petit nombre. La question à laquelle nous souhaitons répondre est donc : étant donné un nombre n de *honeypots*, quel gain obtient-on quand on utilise un *honeypot* supplémentaire ?

Pour un nombre n de *honeypots* fixé (entre 1 et 24), nous étudions donc le nombre de pairs distincts observés avec seulement n *honeypots* choisis au hasard. Comme le choix de ces n *honeypots* influe grandement sur le résultat, nous répétons cette opération pour 100 ensembles aléatoires de n *honeypots* (l'idéal serait d'étudier les 2^{24} sous-ensembles

de nos 24 *honeypots*, mais ce n'est pas faisable), et traçons dans la figure 1.2 (droite) les valeurs minimales et maximales observées, ainsi que la moyenne. La moyenne représente ce que l'on peut s'attendre à observer en pratique. Les valeurs minimales et maximales donnent une idée d'à quel point on peut s'éloigner de la moyenne.

On observe que les courbes obtenues sont proches les unes des autres, sauf lorsque l'on utilise un très petit nombre de *honeypots* (l'un de nos *honeypots* a observé à lui seul 37 000 pairs, alors qu'un autre n'en a vu que 13 000). D'autre part, on voit clairement qu'il y a un gain important à utiliser un nombre relativement important de *honeypots*. Ce gain est cependant particulièrement visible au début de la courbe, et devient moins élevé lorsque le nombre de *honeypots* augmente. Si l'on augmentait indéfiniment le nombre de *honeypots*, on atteindrait au bout d'un moment un point où chaque *honeypot* supplémentaire n'ajouterait que peu d'information (alors qu'établir des *honeypots* a un coût non négligeable).

Enfin, nous avons utilisé une technique similaire pour étudier l'impact du nombre de fichiers annoncés sur la quantité de pairs observés. La mesure gloutonne a utilisé un *honeypot* qui annonçait un grand nombre de fichiers. Nous avons sélectionné les cent fichiers les plus populaires (c'est-à-dire ceux pour lesquels le plus de pairs avaient fait des requêtes), et nous avons étudié, en fonction de n, le nombre de pairs qui auraient été observés si l'on n'avait annoncé que n fichiers parmi ces fichiers populaires. Nous avons observé que le nombre de pairs découverts augmente linéairement, avec une pente importante, en fonction du nombre de fichiers annoncés. Ceci montre qu'annoncer un grand nombre de fichiers est pertinent en termes de quantité d'information collectée.

Au final, cette méthode est complémentaire des autres approches, avec des avantages et des inconvénients. Elle permet de se concentrer sur un sujet particulier, ne requiert pas la coopération d'un administrateur de serveur et/ou d'un fournisseur d'accès, et est relativement légère à mettre en place. En contrepartie, elle donne une vision très partielle de l'activité du système.

Nous avons cependant été confrontés à un autre problème. Vus depuis l'extérieur, nos *honeypots* apparaissent comme des fournisseurs pour les fichiers qu'ils annoncent. Lors des mesures de test que nous avons effectuées et décrites ci-dessus, nos *honeypots* annonçaient plusieurs fichiers sous *copyright*. Ils ont été détectés par des organismes détenteurs du *copyright*, qui nous ont contactés pour nous demander de cesser de diffuser ces fichiers. Bien que nous ne diffusions en réalité aucun fichier, poursuivre ces mesures en annonçant des fichiers sous *copyright* s'est donc avéré problématique, ce qui a sérieusement restreint le type d'activité observable. Nous souhaitions également étudier les échanges de fichiers pédophiles, dans le cadre d'un projet de lutte contre ce type d'activité. Après des discussions avec les forces de l'ordre impliquées dans cette lutte, il s'est avéré que nos *honeypots* interféreraient avec leur infrastructure de surveillance. De telles mesures peuvent donc difficilement être menées hors des institutions légales.

Pour ces raisons, et bien que la mesure par *honeypot* se soit révélée prometteuse, nous n'avons pas effectué de mesure à grande échelle au-delà de celle décrite ici.

1.2 Un radar pour l'internet

Certains projets effectuent des mesures de la topologie de l'internet de manière périodique. C'est le cas en particulier du projet CAIDA, à travers son projet de mesure Skitter [102], récemment remplacé par Archipelago[4] [12], et du projet *iPlane* [53]. Le principe est d'utiliser l'outil `traceroute` ou une variante afin d'explorer périodiquement les routes entre un ensemble de moniteurs et de destinations. Les projets DIMES [32] et Traceroute@home [114] effectuent aussi des mesures successives des routes entre un ensemble de moniteurs et de destinations. Leur particularité est qu'il s'agit d'efforts collaboratifs, dans lesquels un grand nombre de participants peut installer des moniteurs de mesure, ce qui permet d'établir des moniteurs répartis dans un grand nombre d'endroits dans le monde. Ceci permet de mesurer la topologie de manière plus complète et vise à supprimer, ou du moins à diminuer grandement, le biais dans la topologie ainsi mesurée.

Il est aussi possible d'utiliser une particularité du protocole multicast : on peut interroger les routeurs qui l'implémentent pour connaître la liste de leurs voisins. Cette possibilité a été explorée dans [85]. À partir d'un ensemble de routeurs multicast connus au départ, leurs voisins ont été explorés de manière itérative, permettant de découvrir l'ensemble des routeurs multicast accessibles à partir de l'ensemble de départ. Cette opération a été répétée tous les jours pendant plus de quatre ans. Au final, les données contiennent, pour chaque journée, l'ensemble des voisins des routeurs multicast découverts jusqu'à présent.

On peut également s'intéresser à la topologie vue au niveau des systèmes autonomes (ou AS). Dans ce cas, il est possible d'enregistrer sur un routeur de bordure les messages de mise à jour des chemins. Ceci permet de capturer une partie des changements des liens entre AS [25, 41, 59]. Le projet *Route Views* de l'université de l'Oregon [95] sauvegarde périodiquement les tables de routage d'un certain nombre de routeurs de bordure. Chaque table de routage contient les chemins, en termes d'AS, depuis le routeur correspondant vers toutes les destinations possibles. L'enregistrement périodique de ces tables de routage permet donc de connaître la dynamique de ces chemins.

Ces efforts de mesure représentent un apport fondamental pour l'étude de la dynamique de la topologie de l'internet. Nous avons déjà dit qu'il est difficile, voire impossible, de capturer la topologie dans son intégralité à un moment donné. Ces mesures ne capturent pas non plus la dynamique de la topologie dans son intégralité, d'une part parce que certains nœuds et liens ne sont pas vus, d'autre part parce que la fréquence de mesure ne permet pas d'observer tous les changements.

Nous proposons ici une nouvelle approche, complémentaire de celles décrites ci-dessus : nous nous concentrons sur ce qu'une machine voit de la topologie autour d'elle-même, que nous appelons *vue ego-centrée*. Étant donnés une machine moniteur et un ensemble de destinations, une vue ego-centrée est constituée de l'ensemble des routes entre ce moniteur et ces destinations. Il s'agit d'un objet bien défini, que l'on peut mesurer de manière efficace avec des outils tels que `traceroute`. En répétant la mesure de manière périodique, on peut mesurer la dynamique d'une vue ego-centrée avec une fréquence élevée. C'est ce que nous appelons une mesure *radar* [65].

4. Archipelago améliore Skitter entre autres en permettant de l'installer sur un grand nombre de moniteurs de mesure, et en apportant une amélioration à l'outil `traceroute` introduite dans [13, 118], que je décris dans la section 2.1.

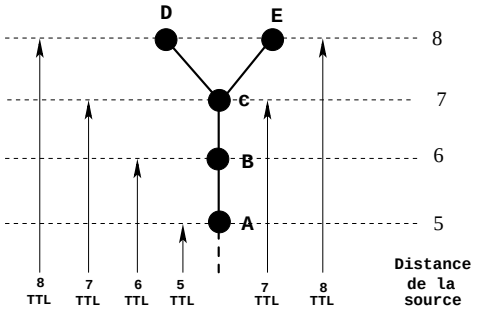

FIGURE 1.3 : Illustration du principe de `tracetree`. Des paquets envoyés vers D et E avec un TTL 8 permettent de découvrir ces deux machines. Les paquets envoyés avec un TTL 7 découvrent tous les deux C, et on arrête d'envoyer des sondes vers E. B et A sont découverts avec des paquets de TTL 6 et 5 envoyés vers D.

Je décris ci-dessous l'outil que nous avons implémenté pour mesurer une vue ego-centrée, appelé `tracetree`. Cet outil a de nombreux paramètres, et nous avons testé leur influence de manière rigoureuse. Je décris ces tests, ainsi que les mesures que nous avons effectuées.

1.2.1 L'outil `tracetree`

Pour mesurer une vue ego-centrée, on peut utiliser l'outil `traceroute` depuis le moniteur et mesurer séparément la route vers chaque destination. Cette approche présente cependant des inconvénients majeurs [33] : la charge imposée aux nœuds n'est pas équilibrée, et il y a une grande redondance dans les données obtenues. Ceci implique de plus que les informations obtenues ne sont pas homogènes (les destinations ne sont vues qu'une fois, alors que les machines proches du moniteur, qui sont sur la route d'un grand nombre de destinations, sont vues plusieurs fois), et qu'il est donc difficile de les analyser rigoureusement.

Il est possible d'éviter ces problèmes en effectuant une mesure d'arbre à l'envers [33, 78] : étant donné un ensemble de destinations, on commence par découvrir les derniers liens des chemins vers les destinations, puis le lien précédent, et ainsi de suite. Quand deux chemins ou plus se rencontrent (c'est-à-dire découvrent un même nœud), on arrête d'envoyer des sondes vers toutes les destinations correspondantes sauf une. La figure 1.3 illustre ce principe. Son implémentation présente plusieurs subtilités que nous ne détaillerons pas ici. Pour plus de détails voir [65].

L'outil assure au final que les données obtenues forment un arbre, ce qui simplifie l'analyse. Il envoie un seul paquet par lien, ce qui est optimal, et assure également que chaque lien n'est vu qu'une fois, ce qui fournit une vue homogène, et équilibre la charge imposée par la mesure.

Nous avons comparé notre outil avec `traceroute` en termes de nombre d'adresses IP observées et de nombre de paquets envoyés. Ces comparaisons ont montré que `traceroute` ne capture pas significativement plus d'informations que `tracetree` : il ne permet de voir que 3% d'adresses IP supplémentaires, et les dynamiques observées sont quasiment identiques. Par contre, `traceroute` envoie deux fois plus de paquets que `tracetree`. Les mesure `tracetree` imposent donc une moindre charge au réseau que `traceroute`, ce qui

permet de les répéter avec une fréquence plus élevée.

1.2.2 Influence des paramètres

L'outil `tracetree` permet d'effectuer des mesures de type radar : étant donnés un moniteur et un ensemble de destination, il s'agit de lancer `tracetree` périodiquement pour mesurer la vue ego-centrée correspondante. Une mesure radar est donc constituée de *passes* successives, chaque passe correspondant à une mesure `tracetree`.

Un grand nombre de paramètres (à commencer par le choix du moniteur et des destinations) peuvent influencer les données obtenues. Par exemple, effectuer la mesure à une fréquence très élevée permet de capturer finement la dynamique, mais cause une charge importante qui fait que certains routeurs arrêteront probablement de répondre à nos sondes.

Comme les paramètres sont nombreux, il est impossible de tester toutes leurs combinaisons. Pour évaluer de façon rigoureuse leur influence, nous avons proposé l'approche suivante. Nous avons tout d'abord choisi un ensemble de paramètres qui semblent *a priori* raisonnables. Ces paramètres sont les suivants : 3 000 destinations pour chaque moniteur, un *timeout*[5] de $2s$ et un délai de 10 minutes entre deux passes consécutives. Nous avons ensuite effectué des mesures avec ces paramètres depuis plusieurs moniteurs en parallèle. Sur certains moniteurs, dits *de contrôle*, nous n'avons pas changé ces paramètres ; sur d'autres, dits *de test*, nous avons alterné entre des périodes de mesure avec les paramètres de base et des périodes où nous changions l'un des paramètres. Les moniteurs de contrôle permettent de savoir si les changements observés sur les moniteurs de test sont dus aux changement de paramètres ou bien à des événements sur le réseau. Nous illustrons ci-dessous cette approche pour quelques paramètres représentatifs.

La figure 1.4 (gauche) présente l'impact du délai entre deux passes consécutives. On voit clairement que le fait de réduire ce délai n'a pas d'impact significatif sur le comportement observé. En particulier, bien que la courbe du nombre d'adresses IP observées ait une plus grande résolution temporelle après la réduction du délai, ses variations sont similaires. De plus, le moniteur de contrôle permet de voir que la valeur de base du délai est pertinente, car la diminuer ne met pas en évidence une dynamique différente.

La figure 1.4 (droite) montre l'impact du nombre de destinations. Comme on pouvait s'y attendre, augmenter ce nombre entraîne l'observation de plus d'adresses IP. Cependant, augmenter le nombre de destinations peut imposer une trop forte charge sur le réseau, et entraîner une perte d'efficacité relative : comme certains routeurs répondent aux sondes avec un taux limité, les surcharger les rend invisibles pour nos mesures. En simulant des mesures avec 3 000 ou 1 000 destinations à partir de mesures avec 10 000 destinations, on voit effectivement que le nombre d'adresses IP vues est plus faible que celui que l'on observe en mesurant directement avec ces nombres de destinations. Ceci confirme qu'utiliser un tel nombre de destinations entraîne une perte d'efficacité relative (le moniteur de contrôle prouve que ceci n'est pas dû à un changement simultané dans la topologie). Il est important de remarquer que ceci ne se produit pas dans les simulations de mesures avec 1 000 destinations à partir de mesures avec 3 000 destinations, ce qui montre que la charge imposée par l'utilisation de 3 000 destinations est raisonnable.

5. C'est-à-dire le temps au delà duquel on cesse d'attendre une réponse pour une sonde envoyée.

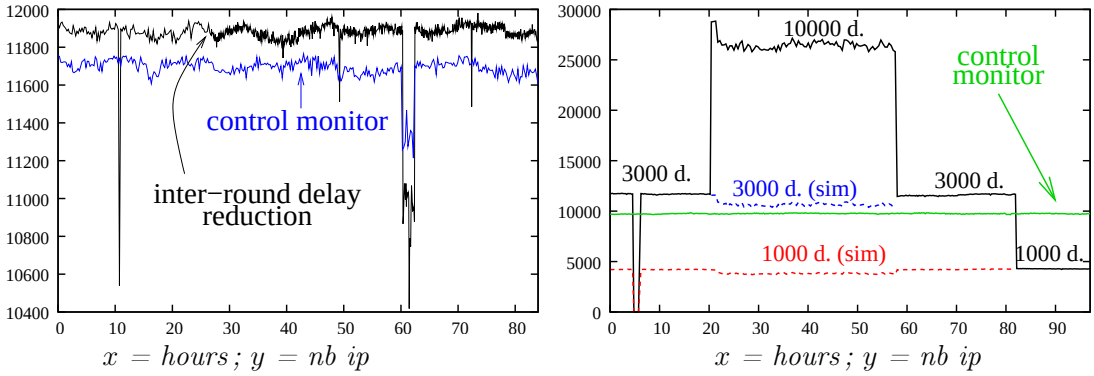

FIGURE 1.4 : **Impact des paramètres de mesure.** Nombre d'adresses IP vues à chaque passe en fonction du temps (en heures). **Gauche : impact du délai entre les passes.** La courbe du bas correspond à un moniteur de contrôle ; l'autre moniteur démarre avec les paramètres de base, et environ 27 heures plus tard nous réduisons le délai entre les passes de 10 à 1 minute. **Droite : impact du nombre de destinations.** La courbe proche de $y = 10\,000$ correspond à un moniteur de contrôle. L'autre courbe en traits continus correspond à un moniteur qui démarre avec un ensemble de destinations D de taille 3 000 (valeur de base), change pour un ensemble D' de 10 000 destinations contenant D environ 20 heure plus tard, revient à D environ 40 heures plus tard, et finalement passe à un sous-ensemble D'' de D de taille 1 000 un peu plus de 20 heures plus tard. Les lignes en pointillés correspondent à des simulations de ce que nous aurions observé depuis ce moniteur en utilisant D à partir des mesures utilisant D' (obtenues en éliminant tous les nœuds et liens qui sont sur des chemins vers des destinations n'appartenant pas à D), et de ce que nous aurions observé en utilisant D'' à partir des mesures effectuées avec D et D'.

Nous avons également étudié les autres paramètres, ainsi que d'autres observables (comme le nombre de sondes sans réponse, ou le nombre de réponses reçues après le *timeout*). Nous en avons conclu que les paramètres de base conviennent bien à nos besoins. Soulignons que ces paramètres permettent d'effectuer des mesures radar à une fréquence très élevée, d'une passe tous les quarts d'heure environ. Les projets existants, qui visent à effectuer une cartographie aussi exhaustive que possible de la topologie de l'internet, ont des fréquences de mesure de l'ordre d'une passe par jour. Ceci montre bien l'intérêt de notre approche pour l'étude de la dynamique.

1.2.3 Mesures

Nous avons utilisé l'outil `tracetree` pour effectuer des mesures à grande échelle. Nous avons utilisé des destinations choisies aléatoirement parmi des adresses IP valides qui ont répondu à un paquet ICMP Echo Request (`ping`).

Nous avons utilisé deux jeux de paramètres. D'une part, nous avons effectué des mesures avec les paramètres de base décrits ci-dessus, qui correspondent à une mesure vers 3 000 destinations à une fréquence d'une passe tous les quarts d'heure environ. Nous avons effectué des mesures en continu pendant plusieurs semaines depuis plus d'une centaine de moniteurs répartis dans le monde (principalement des moniteurs PlanetLab [89]).

D'autre part, nous avons isolé un autre jeu de paramètres pertinent, qui utilise un nombre de destinations moins important (1 000 au lieu de 3 000), mais qui effectue des mesures à un rythme d'une passe toutes les cinq minutes environ. Nous avons effectué des

mesures de longue durée (plusieurs mois) avec ces paramètres depuis un nombre restreint de moniteurs. Les données récoltées constituent une contribution importante pour l'étude de la dynamique de la topologie, sur laquelle nous reviendrons dans la section 3.2.

Les données obtenues, ainsi que le programme `tracetree`, sont librement disponibles [92].

1.3 Conclusion

Nous nous sommes intéressés à la mesure de graphes de terrain dynamiques dans deux contextes : les échanges pair-à-pair et la topologie de l'internet. Dans les deux cas, nous avons effectué des campagnes de mesure à grande échelle, conduisant à des ensembles de données parmi les plus importants du domaine.

L'une des principales conclusions qui ressort de ces travaux est l'importance des paramètres de mesure. Nous avons vu que, dans la plupart des cas, les paramètres ont une forte influence sur les données obtenues. Afin de mener les mesures de manière rigoureuse, et d'obtenir des données qui puissent être utiles dans le plus de contextes possibles, il est donc important d'effectuer des analyses rigoureuses de l'influence des paramètres de mesure.

Le fait de rendre publiques les données que nous collectons est un élément important à nos yeux pour deux raisons. Tout d'abord, il s'agit d'un apport important dans le domaine, permettant à d'autres chercheurs d'étudier les graphes de terrain, ce qui valorise donc nos efforts de mesure. Ensuite, cela permet à d'autres chercheurs de *valider* les analyses ultérieures effectuées sur ces données. Il s'agit là d'un aspect essentiel dans une démarche scientifique.

Finalement, la *reproductibilité* est également un aspect fondamental d'une démarche scientifique. Lorsque l'on étudie des graphes de terrain, il n'est pas clair que l'on puisse appliquer ce principe : on a en effet affaire à des agents qui, dans le cas général, n'ont pas des comportements identiques d'un moment à l'autre, et donc deux mesures de la dynamique d'un même graphe de terrain effectuées à deux moments différents ne seront pas identiques. On peut cependant espérer observer une certaine *stationnarité* dans les systèmes étudiés : même si l'on n'observe pas exactement les mêmes utilisateurs et s'ils n'ont pas exactement le même comportement à deux périodes différentes, il est possible que les caractéristiques globales de ces comportements soient similaires. Je ferai plus ou moins l'hypothèse d'une telle stationnarité dans la suite de ce mémoire. Dans tous les cas, le fait de documenter de manière précise les méthodes de mesure employées et les choix effectués pour leurs paramètres est le seul moyen d'effectuer des mesures similaires à des périodes différentes, et donc de tester si cette stationnarité existe effectivement.

CHAPITRE 2

MÉTROLOGIE

La *métrologie*, dans notre contexte, consiste essentiellement en l'étude du biais introduit par une opération de mesure. Lorsque l'on a effectué des mesures sur un graphe de terrain, les données obtenues sont en effet incomplètes : il est absolument impossible dans l'immense majorité des cas d'espérer capturer *tous* les nœuds et liens. La métrologie tente alors d'analyser les rapports entre l'échantillon obtenu par la mesure et le graphe lui-même. En particulier, les propriétés de l'échantillon sont-elles les mêmes que celles du graphe de terrain ? En d'autres termes, l'échantillon est-il *représentatif* du graphe complet ?

De nombreux travaux se sont intéressés à la métrologie des graphes de terrain, en particulier dans le cas de l'internet, voir par exemple [42, 111, 2, 1, 28, 31, 62, 43, 44, 47, 88, 93, 64, 107]. La plupart de ces travaux se sont concentrés sur l'étude de la métrologie dans le cas de graphes de terrain statiques (ou considérés comme tels), c'est-à-dire n'évoluant pas au cours du temps. Il a en particulier été montré, dans le cas de l'internet, qu'effectuer des mesures de type `traceroute` depuis un petit nombre de moniteurs vers un grand nombre de destinations peut induire un biais important sur la distribution des degrés observée [62, 2, 1, 31].

Dans notre contexte, il est indéniable que la dynamique d'un graphe de terrain doit être étudiée par la métrologie : le fait que le graphe évolue pendant même qu'on le mesure joue certainement un rôle sur la fiabilité des données obtenues. Quelques travaux se sont intéressés à cette question. En premier lieu, dans le cas de l'internet, il a été montré que la dynamique du *load balancing* peut faire que des outils tels que `traceroute` rapportent des informations inexactes [50, 112, 103]. Les auteurs de [81] se sont intéressés à la dynamique de la topologie de l'internet vue au niveau des AS, notamment à la question de savoir si un changement *observé* dans la topologie correspondait à un événement réel ou non. Les auteurs de [108] se sont intéressés au problème du choix aléatoire et non biaisé de sommets dans un graphe dynamique, au moyen de marches aléatoire modifiées. Enfin, certains travaux ont constaté que le fait d'observer un système dynamique pendant un temps fini et/ou en effectuant des mesures de manière périodique pouvait entraîner un biais dans l'estimation de propriétés élémentaires telles que la durée de vie des éléments [94, 98, 109, 119].

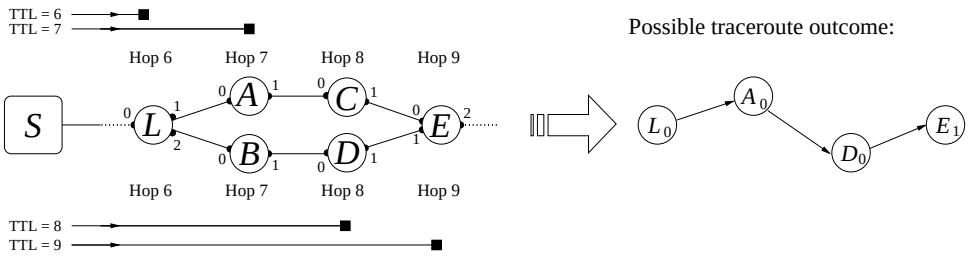

FIGURE 2.1 : Exemples de liens et nœuds manquants, ainsi que de faux liens, dans la topologie mesurée avec `traceroute`.

Je me suis intéressée aux liens entre métrologie et dynamique dans trois contextes différents. Tout d'abord, j'ai étudié comment une dynamique particulière, le *load balancing* sur l'internet, peut jouer un rôle dans les données observées [13, 118] (section 2.1). Ensuite, je me suis intéressée à la dynamique de la mesure elle-même : une opération de mesure d'un graphe de terrain est en effet loin d'être un processus instantané. J'ai étudié la façon dont les propriétés de l'échantillon mesuré évoluent pendant qu'il grandit, et montré que ceci permet dans certains cas de déterminer s'il est représentatif du graphe complet ou non [64] (section 2.2). Enfin, j'ai traité le cas de propriétés intrinsèquement dynamiques, en étudiant la durée de vie des nœuds dans un graphe de terrain. J'ai introduit une méthodologie permettant de savoir si la durée de mesure est suffisante pour caractériser une propriété donnée de manière fiable [17] (section 2.3).

2.1 Biais causé par le *load balancing*

Le *load balancing* est un procédé permettant aux administrateurs de réseaux d'augmenter la fiabilité et d'équilibrer l'utilisation des ressources. Un routeur effectuant du *load balancing* envoie les paquets pour une même destination sur des chemins différents, afin d'équilibrer la charge qu'il transmet sur ces différents chemins. Nous verrons que ceci peut fausser les mesures `traceroute`. Nous avons conçu un outil, `paris-traceroute`, qui évite une partie du *load balancing* et permet de mettre ce phénomène en évidence, ainsi que de le quantifier [13, 118].

2.1.1 `traceroute`, *load balancing* et `paris-traceroute`

Le *load balancing* fausse les données obtenues par `traceroute` : certains liens ne sont pas vus, et de plus, de faux liens sont observés. Ceci est illustré dans la figure 2.1. Dans cet exemple, L est un routeur effectuant du *load balancing* situé à distance 6 du moniteur S. À gauche, on voit la vraie topologie pour les routeurs situés à des distances du moniteur comprises entre 6 et 9. Les cercles représentent des routeurs, et chaque interface est numérotée. Les carrés noirs représentent des sondes envoyées avec des TTL allant de 6 à 9 [1]. Elles sont représentées au dessus de la topologie si L les a envoyées vers A, et

1. On rappelle que lorsqu'une sonde est envoyée avec un TTL h, elle s'arrête à une machine située à distance h du moniteur, qui y répond par un message d'erreur. Ceci permet à `traceroute` d'inférer la présence de cette machine. En faisant varier h, `traceroute` explore en principe un chemin entre le

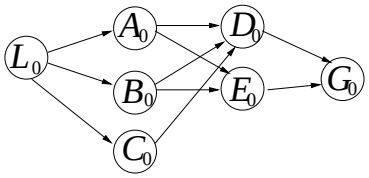

FIGURE 2.2 : Exemples de diamants. (L_0, D_0) est un diamant de taille 3, et (L_0, E_0), (A_0, G_0) et (B_0, G_0) sont des diamants de taille 2.

au-dessous s'il les a envoyées vers B. À droite, on voit un exemple de ce que `traceroute` peut voir. On ne voit pas les routeurs B et C, ni les liens qui leurs sont incidents tels que (L_0, B_0) ou (B_0, D_0). De plus, de faux liens peuvent être observés, tels que $(A_0,\ D_0)$.

Il existe plusieurs stratégies de *load balancing* pour décider du chemin sur lequel envoyer chaque paquet. Les deux qui nous intéressent ici sont la stratégie *par paquet*, qui consiste à les envoyer de manière équilibrée sur les différents chemins, et la stratégie *par flot* qui tente d'assigner les paquets qui appartiennent à un même flot à un même chemin, afin que ces paquets arrivent dans l'ordre à la destination. La plupart des routeurs considèrent que deux paquets appartiennent à un même flot s'ils ont les mêmes adresses et ports de source et de destination, et qu'ils utilisent le même protocole.

Dans les deux cas, les sondes envoyées par `traceroute` vers une même destination peuvent suivre des chemins différents, ce qui pose les problèmes décrits ci-dessus.

Pour remédier en partie à ces inconvénients, nous avons introduit un nouvel outil, `paris-traceroute`, qui ne souffre pas du *load balancing* par flot. Pour cela, l'outil crée les paquets sondes en manipulant les champs de l'en-tête de manière à ce que tous les paquets pour une même destination appartiennent à un même flot. Manipuler les en-têtes de cette manière est complexe et présente de nombreuses subtilités que nous ne présenterons pas ici. Pour plus de détails, voir [86, 13, 118]. Remarquons que l'outil `paris-traceroute` rencontre les mêmes problèmes que `traceroute` face au *load balancing* par paquet.

Afin d'évaluer l'influence du *load balancing* par flot sur les mesures effectuées avec `traceroute`, nous avons conduit en parallèle des mesures avec `traceroute` et `paris-traceroute` depuis un même moniteur vers un même ensemble de 5 000 destinations. Chacune des 1 465 passes de mesure a consisté à effectuer une mesure avec `traceroute` puis avec `paris-traceroute` vers chaque destination.

2.1.2 Artéfacts introduits par le *load balancing* par flot

Nous avons identifié trois types d'artéfacts créés par le *load balancing* par flot dans les données obtenues avec `traceroute`.

Les *boucles* (respectivement les *cycles*) sont des liens d'un nœud vers lui-même (respectivement des chemins d'un nœud vers lui-même de longueur supérieure à 1). Nous avons montré que de tels liens peuvent apparaître si le *load balancing* se produit entre des routes de longueurs inégales. Les boucles, correspondant à des différences de longueur de 1, sont beaucoup plus fréquentes que les cycles. Nous avons montré que près de 90% des boucles et 80% des cycles observées avec `traceroute` disparaissaient avec `paris-traceroute`.

moniteur et la destination [55].

Les artéfacts les plus fréquemment créés par le *load balancing* sont les *diamants*. Formellement, un diamant est une paire (h, t) d'adresses IP telles qu'il existe au moins deux adresses r_i distinctes appartenant à des chemins de la forme $M, \ldots, h, r_i, t, \ldots, D$. L'adresse h est la *tête* du diamant, t est sa queue, et l'ensemble des adresses r_i forme son *noyau*. La taille du diamant est le nombre d'adresses du noyau. La figure 2.2 présente un exemple de diamants.

Au total, on observe 28 231 diamants avec `traceroute`. La figure 2.3 (gauche) présente la distribution de leurs tailles. Si l'on considère séparément les routes vers chaque destination, on observe que $90,2\%$ des destinations entraînent l'apparition d'au moins un diamant, et que l'on en voit en moyenne $21,3$ pour chaque destination.

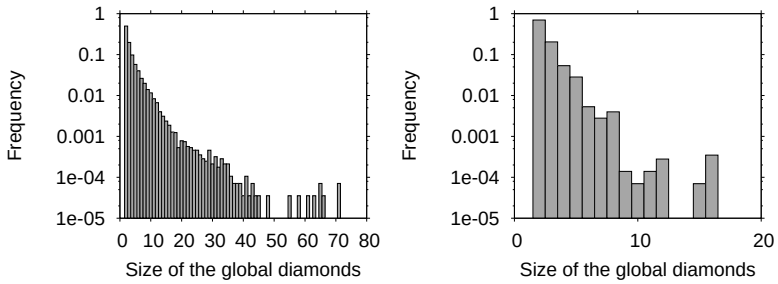

FIGURE 2.3 : Distribution des tailles des diamants observés avec `traceroute` (gauche) et `paris-traceroute` (droite).

Si l'on utilise `paris-traceroute`, le nombre de diamants chute fortement : $52,6\%$ d'entre eux disparaissent. La figure 2.3 (droite) présente la distribution des tailles des diamants observés avec `paris-traceroute`. On constate que non seulement le nombre de diamants diminue, mais leur taille également. Si l'on considère séparément les routes vers chaque destination, on observe que $89,6\%$ des destinations entraînent l'apparition d'au moins un diamant, et que le nombre moyen de diamants observés pour chaque destination n'est plus que de $9,7$.

Cette comparaison donne une idée de l'impact du *load balancing* par flot dans nos mesures effectuées avec `traceroute`. Environ la moitié des diamants disparaissent quand on utilise `paris-traceroute`, et sont donc fort probablement constitués de faux liens induits par du *load balancing* par flot.

Nous avons également observé un phénomène surprenant : quelques diamants sont observés avec `paris-traceroute` mais pas avec `traceroute` (ils représentent environ $3,26\%$ du total observé avec `traceroute`). Une des causes possibles de ce phénomène est que certains diamants sont observés avec une très faible fréquence dans nos mesures. Les diamants observés avec `paris-traceroute` mais pas avec `traceroute` correspondent donc peut-être à des diamants très rares, qui ont une faible probabilité d'être observés. Ceci est peut-être dû à des événements dans le réseau, comme des changements importants de routage, qui mèneraient à l'observation de structures éphémères. Nous reviendrons sur cet aspect dans la section 3.2 et dans la conclusion de ce mémoire.

2.1.3 Conclusion

Nos travaux ont montré que le *load balancing* est responsable d'un grand nombre d'artéfacts dans les mesures effectuées avec l'outil `traceroute`. En isolant ces artéfacts

grâce à l'outil `paris-traceroute`, nous avons à la fois pu montrer qu'ils jouent un rôle important, et fourni un outil permettant de faire des mesures sans souffrir de ce biais. Il faut toutefois remarquer que l'outil `paris-traceroute` est, comme `traceroute`, vulnérable au *load balancing* par paquet, et que les données obtenues avec cet outil contiennent par conséquent encore un certain nombre d'artéfacts.

De plus, cette étude n'a pas pris en compte le fait que des événements peuvent survenir dans la dynamique de la topologie, comme des changements de routage par exemple. Certaines observations indiquent en effet que certaines des différences observées entre les mesures effectuées avec `traceroute` et `paris-traceroute` sont causées par de tels événements et non par le *load balancing* par flot. Pour pouvoir étudier le biais induit par ces événements, et de manière plus générale par la dynamique de la topologie, il est nécessaire d'étudier cette dynamique. Nous reviendrons plus en détail sur son analyse dans la section 3.2, et sur le lien fondamental entre analyse et métrologie dans la conclusion de ce mémoire.

2.2 Estimer la pertinence des propriétés observées

Nous avons déjà signalé que les opérations de mesure des graphes de terrain statiques (ou considérés comme tels) pouvaient souffrir d'un biais, c'est-à-dire que les propriétés de l'échantillon ne sont pas celles du graphe original. La plupart des travaux qui se sont intéressés à cette question ont produit des résultats théoriques importants, principalement en simulant des explorations sur des modèles de graphes, puis en comparant les propriétés de l'échantillon obtenu à celles du graphe de départ.

En l'absence de méthodologie permettant de savoir en pratique si un échantillon est représentatif du graphe original, la plupart des travaux de mesure font (souvent de manière implicite) l'hypothèse suivante : la mesure passe par une phase initiale pendant laquelle l'échantillon obtenu n'est pas représentatif, mais, quand l'échantillon grandit, *on atteint un régime stable dans lequel les propriétés fondamentales cessent d'évoluer*. La plupart des travaux capturent donc un échantillon aussi grand que possible du graphe de terrain considéré, puis font l'hypothèse que cet échantillon est représentatif.

Nous avons introduit une méthodologie empirique permettant d'évaluer la pertinence de cette hypothèse pour des mesures réelles [64]. Elle consiste à traiter de grandes mesures, et à étudier les propriétés qui auraient été observées si la mesure s'était arrêtée avant que l'échantillon atteigne sa taille finale. Nous nous intéressons donc ici à la dynamique de la mesure et non à celle du graphe mesuré, que l'on suppose statique.

Nous avons appliqué cette méthode à quatre mesures à grande échelle de graphes de terrain, pour lesquelles nous connaissions le moment où chaque nœud et chaque lien a été découvert. Nous avons ensuite étudié l'évolution des propriétés de l'échantillon au fur et à mesure que sa taille augmentait. Le programme et les données utilisées sont disponibles [83].

Nous allons illustrer nos résultats sur quelques propriétés classiques. La figure 2.4 présente l'évolution du degré moyen de l'échantillon en fonction de sa taille (en nombre de nœuds). Dans le cas du graphe du web, les mesures atteignent un régime dans lequel le degré moyen de l'échantillon se stabilise, aux environs de 40. Il semble donc que l'on puisse avoir confiance en cette valeur, qui est au moins indépendante de la taille de l'échantillon.

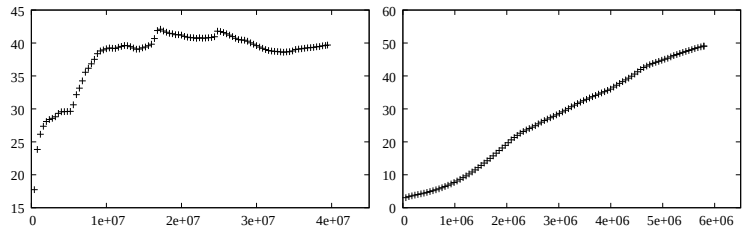

FIGURE 2.4 : Évolution du degré moyen en fonction de la taille de l'échantillon. Gauche : un graphe du web ; droite : échanges entre utilisateurs dans un système pair-à-pair.

Dans le cas des échanges pair-à-pair par contre, le degré moyen ne se stabilise absolument pas. Il est alors clair que l'on ne saurait avoir confiance dans la valeur du degré moyen de l'échantillon final : arrêter la mesure plus tôt ou plus tard aurait donné des valeurs très différentes. De plus, comme les échantillons que nous étudions sont de très grande taille, il y a peu d'espoir que cette valeur se stabilise si les mesures sont poursuivies plus longtemps.

Il faut noter que pour un graphe donné, le fait que l'on puisse avoir confiance dans une propriété n'implique pas que l'échantillon soit globalement représentatif. En effet, les courbes du coefficient de *clustering*[2], que nous ne présentons pas ici, présentent pour nos deux graphes étudiés un comportement inverse de celui du degré moyen : il est relativement stable pour le graphe des échanges pair-à-pair, et instable pour le graphe du web. Ceci montre qu'il est important de ne pas raisonner en cherchant à savoir si l'échantillon obtenu est globalement représentatif du graphe réel, mais en cherchant à savoir en quelles propriétés on peut avoir confiance.

Enfin, nous avons montré que, pour une même notion intuitive, certaines propriétés semblent plus faciles à estimer que d'autres. Le coefficient de *clustering* capture par exemple intuitivement la notion de *densité locale* : il représente la densité moyenne de liens dans le voisinage d'un nœud. Une propriété importante est que, pour la plupart des graphes de terrain, ce coefficient est plusieurs ordres de grandeur au-dessus de la densité du graphe. Une autre façon de capturer cette notion est d'étudier le rapport entre le coefficient de *clustering* et la densité, présenté dans la figure 2.5.

On voit que dans les deux cas cette quantité a tendance à se stabiliser quand la taille de l'échantillon grandit. Ceci indique que dans notre contexte, le rapport entre le coefficient de *clustering* et la densité pourrait être une propriété statistique plus pertinente que le coefficient de *clustering* lui-même. Il semble donc raisonnable de chercher à estimer ce rapport de manière fiable, plutôt que ces deux propriétés individuellement.

Pour conclure, nous avons mis au point une méthodologie *empirique* qui permet d'évaluer rigoureusement la pertinence des propriétés lors de mesures à grande échelle de graphes de terrain. Cette méthodologie permet d'identifier rigoureusement des cas où l'on ne peut pas avoir confiance en les propriétés observées, alors que dans les autres cas considérer les propriétés comme représentatives est raisonnable.

2. Le coefficient de clustering d'un nœud v est égal au nombre de liens présents entre ses voisins, divisé par le nombre de liens possibles entre ses voisins. Le coefficient de *clustering* du graphe est la moyenne de ce coefficient pour tous les nœuds de degré au moins 2.

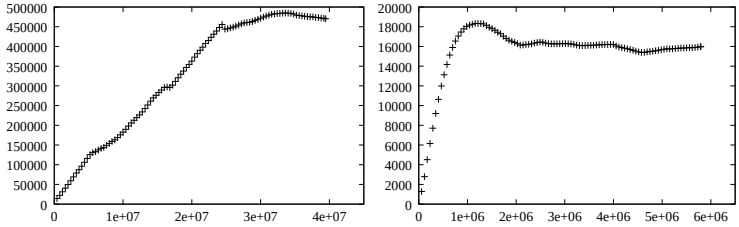

FIGURE 2.5 : Évolution du coefficient de *clustering* divisé par la densité, en fonction de la taille de l'échantillon. Gauche : un graphe du web ; droite : échanges entre utilisateurs dans un système pair-à-pair.

Il faut cependant souligner que, même si une propriété se stabilise pendant la mesure, elle peut tout de même être biaisée. En effet, il est possible qu'elle se mette à évoluer de nouveau si la mesure continue. C'est pour cela qu'il est important de disposer de jeux de données les plus importants possible. La propriété peut également souffrir d'effets de bord, c'est-à-dire être stable pendant que la mesure explore presque tout le réseau, puis se remettre à évoluer à la fin. Il est également possible que la méthode de mesure soit intrinsèquement biaisée. Il est par exemple globalement admis qu'explorer la topologie de l'internet en effectuant des mesures depuis un seul moniteur donnera toujours une image biaisée de la distribution des degrés, quelle que soit la quantité de données capturées. C'est pourquoi nos travaux sont complémentaires des travaux théoriques étudiant l'impact de la mesure sur les propriétés observées.

Enfin, dans la plupart des cas, le graphe considéré évolue pendant même qu'on le mesure, ce qui peut certainement fausser les propriétés, et ce d'autant plus que la mesure dure plus longtemps. Remédier à cela est une perspective importante, qui requiert au préalable une bonne compréhension de la dynamique, à laquelle nous nous intéressons dans la section 3.2.

2.3 Biais causé par la durée de mesure

Lorsque l'on observe un système dynamique, la période d'observation est par définition finie. Ceci suffit à induire un biais dans les observations. Les événements ayant lieu avant et après la période de mesure ne sont pas capturés, ce qui empêche de mesurer correctement plusieurs statistiques, comme les corrélations entre les événements (puisque certains sont capturés et d'autres non), leur fréquence, les durées de vie des nœuds et des liens, etc.

Ceci a déjà été constaté, et une méthodologie a été introduite pour remédier, dans une certaine mesure, à ce problème [94, 98]. De même, il a été constaté que le fait de mesurer un système en l'observant périodiquement peut causer un biais, notamment parce que cela ne permet pas d'observer certains événements courts [119].

Nous avons introduit une méthodologie, complémentaire de celles déjà existantes, permettant d'estimer l'importance du biais lié à la durée de la mesure dans la caractérisation statistique d'une propriété dynamique [17]. Cette méthodologie repose sur l'idée que plus la fenêtre d'observation est longue, plus la portion d'événements mal observés est petite par rapport aux événements observés correctement, et donc plus le biais est faible. Notre

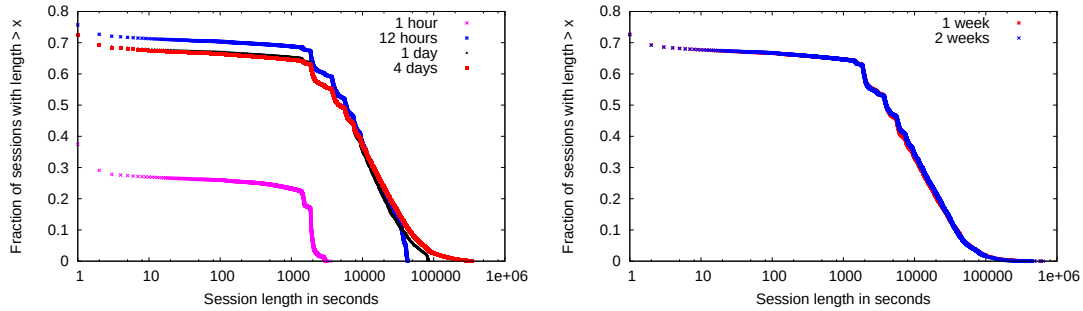

FIGURE 2.6 : Distributions cumulatives inverses des durées de vie observées pour différentes tailles de la fenêtre de mesure. Un point de coordonnées (x, y) indique qu'une fraction y des sessions ont une durée supérieure ou égale à x.

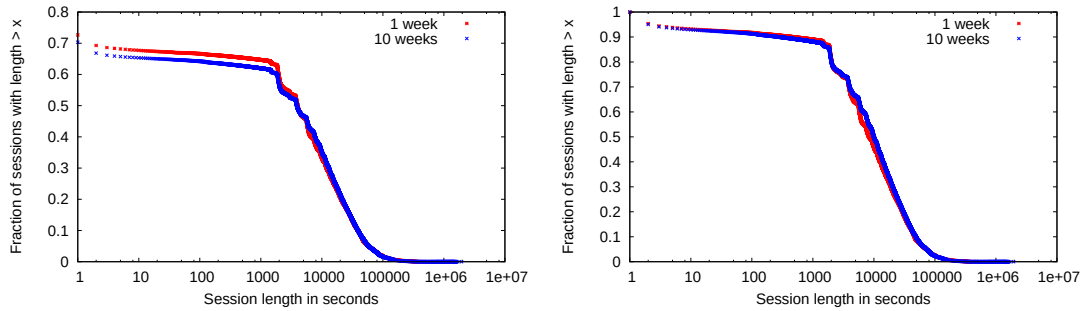

FIGURE 2.7 : Distributions cumulatives inverses des durées de vie observées pour des fenêtres de mesure de une et dix semaines. Gauche : distribution normalisée ; droite : distribution normalisée par le nombre de valeurs supérieures à 0.

approche consiste à simuler des fenêtres d'observation plus courtes que la mesure réelle, puis à étudier la façon dont la propriété étudiée évolue lorsque la longueur de la fenêtre d'observation augmente. Si, à partir d'une certaine longueur, la propriété devient stable, cela indique que le biais est éliminé. Cette méthodologie n'a bien sûr de sens que si le système, ou tout au moins la propriété étudiée, est stationnaire, c'est-à-dire que le fait de la caractériser sur un grand intervalle de temps a un sens. Remarquons que si ce n'est pas le cas, on n'arrivera pas à caractériser la propriété étudiée. On ne court donc pas le risque de croire avoir caractérisé une propriété alors qu'elle n'est en réalité pas stationnaire.

Nous avons appliqué cette méthodologie à l'étude de la durée de vie des utilisateurs dans un système pair-à-pair. Nous avons pour cela utilisé les données capturées sur un serveur *eDonkey* [6] décrites dans la section 1.1.1.

Avant toute chose, il était nécessaire de choisir une définition de *session*. En effet, la notion de connexion ou de déconnexion n'existe pas dans le protocole UDP d'*eDonkey*. Les seules informations que nous ayons sont les requêtes effectuées, et les réponses du serveur à ces requêtes. Nous considérons qu'une session est une suite de requêtes effectuées depuis la même adresse IP, dans laquelle deux requêtes consécutives ne sont jamais séparées par plus de trois heures. Voir [17] pour les motivations de ce choix.

La figure 2.6 présente la distribution cumulative inverse des durées de vie observées lorsque la longueur de la fenêtre de mesure varie entre une heure et deux semaines. On

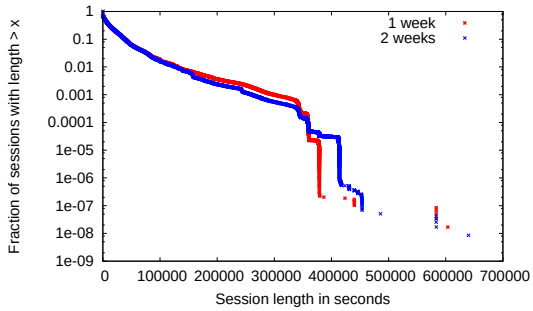

FIGURE 2.8 : Distributions cumulatives inverses des durées de vie observées pour des fenêtres de mesure de une et deux semaines. Échelle lin-log.

constate que la forme des distributions évolue lorsque la longueur de la fenêtre de mesure augmente. Pour des longueurs inférieures ou égales à un jour, les distributions s'arrêtent brusquement ; pour des longueurs supérieures, la queue de la distribution s'aplatit, après un coude situé à proximité de 100 000s, et l'on observe des valeurs *extrêmes* après ce coude. À partir de 4 jours, la forme de la distribution semble ne plus évoluer, et les distributions obtenues pour des fenêtres de mesure d'une ou deux semaines sont très similaires.

Cependant, lorsque la longueur de la fenêtre de mesure augmente à nouveau, on observe une légère différence entre les distributions obtenues. La figure 2.7 (gauche) montre les distributions des durées de vie observées pour des fenêtres de mesure de une et dix semaines. Il y a un écart entre ces distributions, causé par la fraction de sessions de durée nulle (qui n'apparaît pas sur la figure en raison de l'échelle logarithmique). Lorsque l'on normalise ces distributions par le nombre de sessions de durée supérieure à 0 (figure 2.7, droite), cet écart disparaît. Ceci montre que la forme de la distribution n'évolue plus, bien que la fraction de sessions de longueur nulle, elle, varie.

On a vu que les distributions n'évoluent plus *visuellement* lorsque la fenêtre de mesure dépasse une semaine. Il faut cependant considérer les observations visuelles avec prudence. À titre d'exemple, la figure 2.8 présente les mêmes distributions que la figure 2.6 (droite), mais avec une échelle linéaire pour l'axe des x et logarithmique pour l'axe des y. À première vue, les deux distributions semblent très différentes l'une de l'autre. On peut cependant constater qu'elles sont similaires pour au moins 99% des valeurs. Elles diffèrent seulement pour des valeurs supérieures à 150 000s, qui sont des valeurs *extrêmes*. Ceci montre que la partie *normale* des distributions n'évolue pas, et est donc correctement caractérisée. Les valeurs extrêmes, elles, ne peuvent pas êtres caractérisées de cette façon.

Afin de confirmer de manière plus formelle le fait que la distribution n'évolue plus lorsque la fenêtre de mesure atteint une semaine, nous avons utilisé la distance de Monge-Kantorowich [38], qui quantifie l'écart entre deux distributions. Nous avons calculé la distance entre la distribution obtenue pour la fenêtre de mesure complète (10 semaines) et pour des fenêtres de différentes tailles. Les valeurs obtenues tendent à confirmer que, au-delà d'une certaine longueur de la fenêtre de mesure (proche d'une semaine), la distribution obtenue est très proche de la distribution finale.

En conclusion, nous avons montré que cette approche permet dans une certaine mesure de caractériser de manière fiable une propriété dans un système dynamique. Elle permet de caractériser la partie normale de la distribution de la durée des sessions des utilisateurs,

tout en mettant en évidence le comportement différent des sessions de longueur nulle et de celles qui ont des valeurs extrêmes, ce que les méthodologies existantes ne permettent pas. Étudier séparément ces deux types de sessions est une suite intéressante de ces travaux.

Une perspective fondamentale consiste à effectuer des simulations, afin de confirmer nos résultats et de tester d'autres cas de figure. À plus long terme, on peut espérer obtenir des résultats formels, comme par exemple des bornes sur la durée de mesure suffisante pour caractériser correctement une propriété.

Il convient également d'appliquer cette méthodologie à d'autres propriétés que la durée de vie et à d'autres graphes de terrain dynamiques, afin d'assurer qu'elle est pertinente dans le cas général. Néanmoins, cette méthodologie ouvre la voie pour la métrologie de propriétés *dynamiques*, qui a été jusqu'ici très peu étudiée.

2.4 Conclusion

La métrologie est un processus très proche de la mesure. Dans la plupart des cas, une bonne connaissance du processus de mesure est en effet essentielle pour pouvoir étudier rigoureusement les biais qu'il entraîne. De même, de nombreuses questions de métrologie sont intimement liées au système étudié : par exemple, caractériser l'influence du *load balancing* n'a *a priori* de sens que dans le cas de la topologie de l'internet.

Remarquons que la question du paramétrage de la mesure, que nous avons traitée pour les mesures radar et *honeypot* (sections 1.2 et 1.1.2), peut être naturellement vue soit comme de la mesure soit comme de la métrologie. Vus comme de la mesure, les tests de paramètres visent à s'assurer que le système n'est pas surchargé, ainsi qu'à évaluer la pertinence d'effectuer des mesures largement distribuées, par rapport au coût du déploiement de l'infrastructure nécessaire. Vus comme de la métrologie, ils permettent d'évaluer à quel point les propriétés des données obtenues dépendent de l'infrastructure de mesure qui a été déployée. Concernant les analyses que nous avons effectuées sur la mesure par *honeypot*, il aurait suffit que nous nous intéressions à des propriétés moins élémentaires que les seuls nombres de pairs et de fichiers observés, comme par exemple le nombre de fichiers proposés par chaque pair, pour que ces analyses relèvent clairement du domaine de la métrologie.

Il ne semble donc pas *a priori* évident que faire la distinction entre mesure et métrologie soit pertinent. Cependant, certaines questions dépassent le cadre strict du système étudié et de la méthode de mesure employée. Tout d'abord, certaines techniques que nous avons établies pour tester l'influence des paramètres de mesure ont un caractère général. Par exemple, nous avons utilisé les techniques permettant d'évaluer l'apport d'utiliser plusieurs *honeypots*, décrites dans la section 1.1.2, dans d'autres contextes, à savoir la mesure de la topologie de l'internet [82], et la mesure pair-à-pair effectuée par des clients.

Certaines questions enfin dépassent franchement le cadre de la mesure. C'est le cas par exemple de l'étude de l'influence de la durée de mesure sur les observations qui termine ce chapitre (section 2.3). La méthodologie que nous avons introduite peut être appliquée telle quelle à d'autres propriétés dynamiques et à d'autres systèmes, ce qui en fait une méthode de métrologie à part entière.

CHAPITRE 3
ANALYSE

Le but de l'analyse d'un graphe de terrain est de le décrire, souvent au moyen soit de propriétés statistiques, telles que la distribution des degrés ou le coefficient de clustering, soit de propriétés structurelles, telles qu'une décomposition en communautés. Bien que beaucoup reste à faire, l'analyse des graphes statiques est aujourd'hui un domaine relativement mûr, et bon nombre d'outils et de notions applicables à n'importe quel graphe de terrain ont été définis.

Plus récemment, des travaux ont commencé à s'intéresser à l'analyse des graphes dynamiques. La plupart de ces travaux ont effectué une étude en profondeur de cas spécifiques. Le cas de réseaux de contacts entre individus, mesurés à l'aide de capteurs électroniques ou de téléphones *bluetooth* par exemple, a été beaucoup étudié [21, 23, 27, 99, 113]. D'autres cas, comme les échanges dans des réseaux pair-à-pair [67, 45, 68], la topologie de l'internet [81, 87, 85, 60], des réseaux biologiques [14, 91, 116, 105] ou de citations [72], et divers types de réseaux sociaux en ligne [30, 106, 100], ont également été étudiés.

Certains travaux ont abordé des questions générales, pertinentes pour l'étude de n'importe quel graphe dynamique. On peut citer des méthodes pour manipuler des graphes dynamiques [20, 57], la détection ou l'étude de l'évolution de communautés [15, 72, 84], l'étude de l'évolution ou de la formation de motifs spécifiques [117, 24, 61], et quelques travaux de portée plus générale [11, 64, 72, 73, 74, 7, 80]. Cependant, les méthodes et outils proposés sont loin d'être suffisants par rapport aux besoins, et l'essentiel reste à faire dans ce domaine.

J'ai contribué dans cette direction principalement dans deux contextes : j'aborde dans la section 3.1 la question de la robustesse des graphes de terrain face à des suppressions de nœuds ou de liens [46, 75], et j'étudie dans la section 3.2 la dynamique des vues ego-centrées de la topologie de l'internet [76], introduites dans la section 1.2.

3.1 Robustesse des graphes de terrain

La dynamique que nous étudions ici est une dynamique subie par le graphe, et non celle d'un graphe évoluant au fil du temps. La question est de savoir comment un graphe

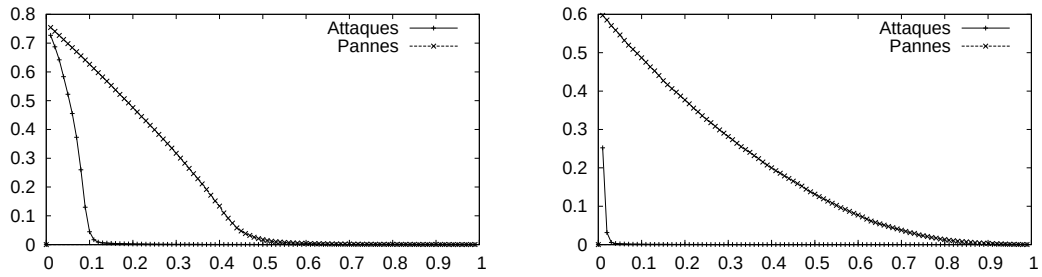

FIGURE 3.1 : Effet des pannes et des attaques. Gauche : sur des graphes aléatoires ; droite : sur des graphes sans échelle.

se comporte lorsque des pannes ou des attaques se produisent, entraînant la suppression de certains de ses nœuds ou de ses liens.

Les auteurs de [8] ont initialement cherché à savoir comment un graphe se comportait face à des *pannes* (simulées par des suppressions aléatoires de nœuds ou de liens), ou à des *attaques* malveillantes visant à endommager ce graphe. Plusieurs stratégies sont possibles pour simuler des attaques ; la plus connue, introduite dans [8], consiste à supprimer les nœuds par ordre décroissant de leur degré.

La question est alors de savoir quelles propriétés d'un graphe peuvent influencer sa capacité à résister aux pannes ou aux attaques. La propriété qui a été le plus étudiée dans ce contexte est la distribution des degrés. Afin d'étudier l'impact de cette distribution, on peut étudier la robustesse de graphes aléatoires choisis parmi ceux ayant une distribution des degrés données [18], et comparer ce que l'on obtient pour différentes distributions. Dans la suite, nous étudions deux types de graphes : des graphes aléatoires [1], qui ont une distribution des degrés en loi de Poisson, et des graphes aléatoires avec une distribution des degrés en loi de puissance, que nous appellerons *graphes sans échelle*.

Afin d'évaluer la résistance d'un graphe, un paramètre très couramment étudié est sa *connexité*, cruciale en particulier dans le cas de réseaux de communication : si après une suppression de nœuds ou de liens il subsiste dans le graphe une grande composante connexe, on estime que les nœuds de cette composante peuvent encore communiquer les uns avec les autres, et que le réseau a peu souffert. À l'inverse, si le graphe a été fragmenté en une grande quantité de petites composantes, on considère qu'il est détruit. L'étude de la taille de la plus grande composante connexe en fonction du nombre de nœuds supprimés permet donc d'obtenir une estimation de l'efficacité d'une stratégie de suppression de nœuds.

La figure 3.1 illustre ceci. Elle présente, pour des graphes aléatoires et des graphes sans échelle, la taille de la plus grande composante connexe en fonction du nombre de nœuds supprimés, lorsque l'on supprime les nœuds aléatoirement (pannes) ou par ordre décroissant de leur degré (attaques).

On peut constater plusieurs choses. Tout d'abord, les attaques sont plus efficaces que les pannes au sens où elle déconnectent le graphe plus vite. Quand on considère les graphes sans échelle, on voit que cette différence d'efficacité est très forte : pour les pannes, il faut supprimer quasiment tous les nœuds pour casser complètement la plus grande composante

1. au sens d'Erdös-Rényi [35], c'est-à-dire des graphes tirés aléatoirement avec probabilité uniforme parmi l'ensemble des graphes ayant un nombre de nœuds et de liens fixés.

connexe ; pour les attaques, à l'inverse, il suffit de supprimer une très faible fraction des nœuds pour que la taille de la plus grande composante connexe devienne quasiment nulle. Pour les graphes aléatoires, au contraire, même si les attaques sont plus efficaces que les pannes, les deux ont des effets *similaires* : dans les deux cas, après la suppression d'une fraction des nœuds strictement inférieure à un, la taille de la plus grande composante connexe devient quasiment nulle. Ces observations ont mené les auteurs de [8] à conclure que les graphes sans échelle sont beaucoup plus robustes que les graphes aléatoires face aux pannes, mais beaucoup plus sensibles aux attaques.

Ceci a engendré une grande quantité de travaux, tentant d'établir de manière formelle ces résultats empiriques, de comprendre ces phénomènes plus en profondeur, et/ou d'étudier d'autres types d'attaques ou d'autres critères pour évaluer la robustesse d'un graphe.

La plupart de ces travaux relèvent du domaine de la physique statistique. Ils utilisent donc des outils et des méthodes qui sont peu connus des informaticiens, comme les approximations continues ou les hypothèses de champ moyen. Notre première contribution dans ce domaine [46, 75] a consisté à présenter une revue de ces travaux destinée à un public d'informaticiens. Nous avons présenté une introduction aux méthodes spécifiques utilisées dans ces articles et détaillé les passages rapides. Nous avons également pris soin d'expliciter toutes les *approximations* présentes dans ces travaux. Une technique couramment utilisée consiste en effet à utiliser des approximations de champ moyen, permettant de simplifier les expressions formelles et d'aboutir à des résultats, dans des cas où il n'est pas possible (ou extrêmement difficile) d'effectuer des preuves formelles en utilisant les expressions exactes. Nous avons jugé important d'expliciter ces approximations, et d'identifier les parties des preuves qui reposent dessus et celles qui sont exactes.

Notre deuxième contribution a consisté à tenter de comprendre en profondeur l'impact des pannes et des attaques sur les graphes sans échelle. Les explications intuitives qui avaient été avancées étaient que leur grande résistance face aux pannes était due à la présence d'un faible nombre de nœuds de très fort degré. Ces nœuds assureraient la connexité du réseau, et comme ils sont peu nombreux, ils ne sont supprimés que tardivement lors de suppressions aléatoires. Réciproquement, ces mêmes nœuds sont la cause de l'extrême faiblesse de ces graphes face aux attaques : comme ces nœuds ont un très fort degré, leur suppression au début de l'attaque entraîne la suppression d'un très grand nombre de liens, ce qui conduit à la rupture rapide de la plus grande composante connexe.

Il est cependant établi [22] que, lorsque l'on supprime des liens de manière aléatoire (ce qui modélise des *pannes de liens*), alors l'effet est similaire à ce que l'on observe pour les pannes de nœuds : il faut supprimer quasiment tous les liens du graphe pour détruire la plus grande composante connexe. Le grand nombre de liens attachés aux nœuds de fort degré ne saurait donc être l'explication de l'efficacité des attaques : lorsque l'on supprime le même nombre de liens, mais choisis aléatoirement, la taille de la plus grande composante est réduite mais est loin de devenir nulle.

Pour aller plus loin, nous avons étudié le résultat sur lequel la plupart des travaux s'appuient pour calculer le moment où la plus grande composante connexe disparaît [77]. Il établit que, pour qu'un graphe aléatoire à distribution des degrés fixée p_k (où p_k représente

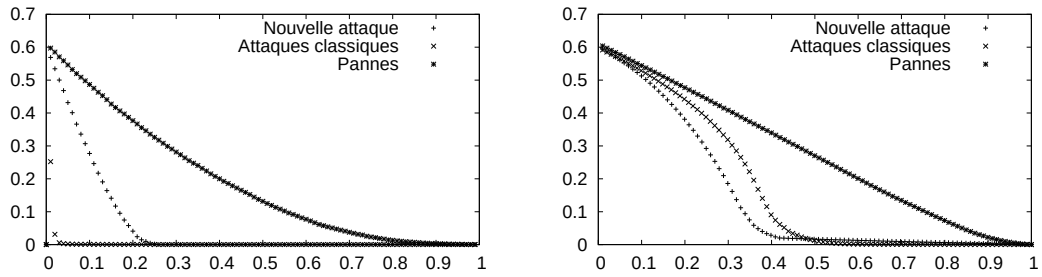

FIGURE 3.2 : Effet des nouvelles attaques sur des graphes sans échelle. Gauche : nouvelles attaques sur les nœuds ; droite : nouvelles attaques sur les liens.

la fraction de nœuds de degré k) ait une composante géante [2] il faut et il suffit que [3] :

$$\sum_{k\geq 0} k(k-2)p_k \geq 0.$$

Lorsque l'on examine cette somme, on voit qu'un seul de ses termes est négatif, celui qui correspond à p_1, la fraction de nœuds de degré 1. On peut donc ré-écrire cette expression de la manière suivante :

$$p_1 \leq \sum_{k\geq 2} k(k-2)p_k.$$

On voit ainsi que la fraction de nœuds de degré 1 joue un rôle clé dans la connexité d'un graphe. Ceci indique que toute stratégie visant à l'augmenter, ou de manière équivalente à diminuer le nombre de nœuds de degré supérieur à 2, mènera à la déconnexion du graphe. Nous avons donc introduit et étudié deux nouvelles stratégies d'attaque, une sur les nœuds et une sur les liens, afin d'approfondir ce point.

La première stratégie consiste à supprimer au hasard des nœuds *de degré supérieur ou égal à deux*. La figure 3.2 (gauche) présente l'évolution de la taille de la plus grande composante connexe en fonction de la fraction de nœuds supprimés, pour cette nouvelle attaque, comparée avec les pannes et les attaques classiques. On voit que la nouvelle attaque est loin d'être aussi efficace que les attaques classiques. Cependant, on constate un changement très important par rapport aux pannes : il suffit de supprimer environ 20% des nœuds pour détruire la plus grande composante connexe, alors que pour obtenir le même résultat avec des pannes il faut supprimer quasiment l'intégralité des nœuds.

Il faut bien noter que la nouvelle attaque est très similaire aux pannes : dans les deux cas les nœuds sont supprimés de manière aléatoire ; la seule contrainte supplémentaire est que l'on ne supprime pas de nœuds de degré 1 (qui ne peuvent pas détruire la plus grande composante connexe). Ceci montre donc que la différence d'efficacité entre les pannes et les attaques classiques est due en bonne partie au fait que les attaques classiques ne suppriment pas de nœuds de degré 1.

La deuxième stratégie d'attaque que nous avons introduite est une stratégie sur les liens. Elle consiste à supprimer aléatoirement des liens *dont les deux extrémités ont un degré supérieur ou égal à 2*. À nouveau, cette attaque est très similaire à des pannes (de liens).

2. C'est-à-dire une composante connexe dont le nombre de nœuds est linéaire en la taille du graphe.
3. Ceci est une simplification de l'énoncé exact du théorème.

La figure 3.2 (droite) présente l'évolution de la taille de la plus grande composante connexe quand les liens sont supprimés par cette attaque, comparée avec les pannes de liens et les attaques classiques considérées du point de vue des liens (c'est-à-dire la stratégie consistant à enlever d'abord les liens attachés aux nœuds de plus fort degré). On voit que, comme dans le cas précédent, cette nouvelle attaque est beaucoup plus efficace que les pannes de liens. De manière marquante, elle est même plus efficace que les attaques classiques. Ceci confirme bien le fait que l'efficacité des attaques classiques est due au fait qu'elles suppriment des liens entre des nœuds de degré supérieur à un.

Nous avons également effectué des simulations intensives des différents types de pannes et d'attaques sur des graphes aléatoires et sans échelle de différentes tailles et de différents degrés moyens et exposants. Ces simulations ont montré que l'on observe des différences relativement importantes entre le cas fini et les résultats théoriques, qui sont valables dans la limite où la taille du graphe tend vers l'infini. Ces différences rendent moins important l'écart entre les comportements des graphes aléatoires et des graphes sans échelle. Ceci, de même que l'étude des nouvelles stratégies d'attaques que nous avons introduites, permet une compréhension plus fine des différents phénomènes en jeu, et modère les conclusions préalablement obtenues sur la différence entre pannes et attaques.

3.2 Dynamique des vues ego-centrées de la topologie de l'internet

Dans cette section nous nous intéressons à la dynamique des vues ego-centrées de la topologie de l'internet, dont nous avons décrit la mesure dans la section 1.2. On s'attend à observer plusieurs composantes dans cette dynamique. Tout d'abord, nous avons vu dans la section 2.1 que le *load balancing* joue un rôle important dans les mesures effectuées avec des outils de type `traceroute`. Il s'agit d'une dynamique à court terme : sans aucun changement de la configuration des routeurs, on découvre des routes différentes quand on explore le chemin vers la même destination plusieurs fois de suite. De plus, l'internet est en perpétuelle évolution, par l'ajout et la suppression de machines, ainsi que par la création ou la suppression de liens entre routeurs et/ou AS. Ceci engendre naturellement une dynamique à plus long terme. Enfin, on s'attend à observer des *événements* particuliers dans la dynamique : pannes de routeurs, changements de configuration entraînant des changements de routage importants, etc. La détection de ces événements est une problématique à part entière, à laquelle j'ai contribué [48] mais que je n'aborderai pas ici.

Nous présentons ci-dessous nos études sur la dynamique des données obtenues par une mesure radar [76]. Nous avons étudié les données issues de plusieurs moniteurs, obtenues à des périodes différentes. Bien que l'on retrouve dans ces données des différences, les comportements observés étaient *qualitativement* les mêmes pour tous les moniteurs et toutes les périodes. Nous présentons donc ci-dessous une analyse de données représentatives, provenant d'une mesure de deux mois effectuée depuis un moniteur situé au Japon.

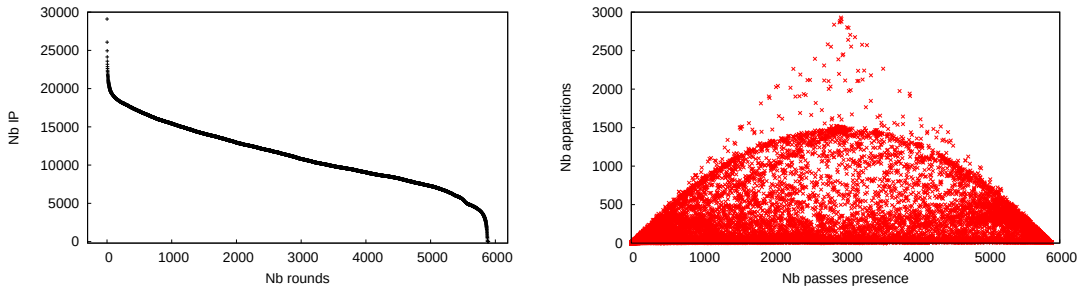

FIGURE 3.3 : Observations des adresses IP. Gauche : Distribution du nombre de passes où chaque adresse est observée. Droite : Corrélations entre le nombre d'observations et le nombre d'apparitions.

3.2.1 Caractérisation des adresses IP observées

Les premières études que nous avons effectuées ont rapidement mis en évidence le fait que l'ensemble des adresses IP observées à chaque passe varie. La question est alors de savoir comment il évolue. Pour étudier ceci nous avons compté, pour chaque adresse IP, le nombre de passes distinctes pendant lesquelles cette adresse a été observée. La figure 3.3 (gauche) présente la distribution cumulative inverse de cette quantité. Un point de coordonnées (x, y) sur cette courbe signifie que y adresses ont été observées dans au moins x passes. On observe deux zones denses, correspondant à des endroits où la pente de la distribution est forte : un grand nombre d'adresses n'apparaissent qu'un petit nombre de fois (plus de 3 000 adresses ont été observées une seule fois, sur un total de 29 100), et un nombre important d'adresses apparaissent quasiment à chaque passe (un peu plus de 2 000 adresses ont été vues dans au moins $95,5\%$ des passes). Entre les deux, on a une densité plus ou moins constante. Ceci permet de distinguer plusieurs classes d'adresses IP : certaines ne sont observées qu'une fois (ou un très petit nombre de fois) pendant toute la durée de la mesure. À l'inverse, d'autres sont extrêmement stables et sont observées à chaque passe ou presque. Entre ces deux extrêmes, tous les comportements sont représentés.

Pour aller plus loin, nous nous sommes intéressés à la stabilité des adresses observées : il y a une différence notable entre une adresse observée 10 fois dans 10 passes consécutives, et une autre observée dans 10 passes éloignées temporellement les unes des autres. La figure 3.3 (droite) présente les corrélations entre le nombre de passes où une adresse est observée (axe des x) et le nombre de fois où elle apparaît (axe des y). On considère qu'une adresse apparaît si elle est présente à une passe mais n'était pas présente à la passe précédente. Ainsi, une adresse observée dans x passes n'apparaîtra qu'une fois si ces passes sont consécutives, et x fois si ces passes sont distantes les unes des autres.

On observe une forme géométrique à première vue surprenante, mais qui s'explique simplement. Tout d'abord, le triangle est délimité par les droites $y = x$ et $y = n - x$, où n est le nombre total de passes : une adresse ne peut pas apparaître plus de fois qu'elle n'est présente, ni plus de fois qu'elle est absente. La parabole a pour équation $x(n-x)/n$. Elle correspond au nombre d'apparitions attendu en moyenne pour une adresse observée dans x passes différentes, si l'on suppose que ces passes sont choisies *au hasard*[4].

4. Dans ce cas, chaque passe correspond à une apparition si l'adresse est observée (ce qui arrive

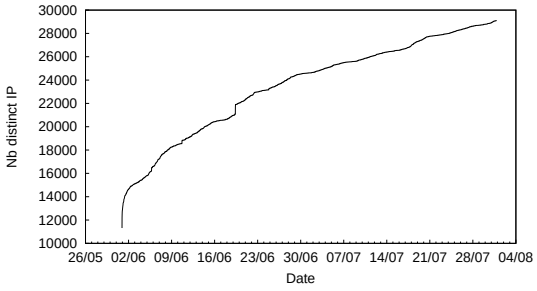

FIGURE 3.4 : Nombre d'adresses IP observées au moins une fois depuis le début de la mesure, en fonction du temps.

Cette courbe permet à nouveau de distinguer plusieurs catégories d'adresses IP. Celles qui sont proches de la parabole n'ont pas de comportement particulier : elles apparaissent beaucoup ou peu de fois, mais elles sont moyennement stables compte tenu du nombre de passes où elles sont observées. Au dessous de la parabole, proche de l'axe des x, on observe une forte densité d'adresses *stables* : leur nombre d'apparitions est plus faible qu'attendu, ce qui indique que, quand ces adresses sont observées, elles ont tendance à rester présentes pendant un grand nombre de passes consécutives. Enfin, les adresses situées au dessus de la parabole apparaissent un plus grand nombre de fois qu'attendu : elles ont tendance à clignoter. Le *load balancing* joue probablement un rôle pour un grand nombre de ces adresses.

3.2.2 Évolution des adresses IP observées

Nous étudions maintenant à quelle fréquence de nouvelles adresses IP sont observées dans nos mesures. La figure 3.4 présente le nombre d'adresses IP observées au moins une fois depuis le début de la mesure. On voit qu'il est en augmentation constante, à un rythme élevé. À titre d'indication, sur un total de 29 100 adresses observées au total pendant les deux mois de mesure, 1 118 ont été découvertes lors de la dernière semaine. La vitesse à laquelle on découvre de nouvelles adresses est bien plus élevée que ce à quoi on s'attendait. Il s'agit d'un phénomène général, et nous l'avons observé pour tous les moniteurs étudiés, et pour des durées de mesure allant jusqu'à six mois.

Nous nous sommes tout d'abord demandé s'il ne s'agissait pas d'un artéfact de mesure. Deux phénomènes simples auraient naturellement pu causer cette croissance du nombre d'adresses IP observées : le fait que certaines des destinations soient des adresses dynamiques, c'est-à-dire allouées à différents ordinateurs, possiblement situés à des endroits différents, à différents moments ; ou encore le fait que certains routeurs répondent avec des adresses IP différentes à chaque fois qu'on les interroge. Nous ne détaillons pas ici les méthodes que nous avons employées (pour plus de détails, voir [76]), mais nous avons montré que ces artéfacts ne sont pas la cause principale du phénomène observé (même s'il n'est pas exclu qu'ils y participent).

Pour aller plus loin, nous avons étudié le phénomène au niveau des systèmes autonomes

avec probabilité x/n) et si l'adresse n'a pas été observée pendant la passe précédente (avec probabilité $(n-x)/n$). En multipliant par n qui est le nombre total de passes, on obtient le nombre attendu d'apparitions.

(*Autonomous Systems* ou AS). Nous avons cherché à savoir si les adresses IP que l'on découvre appartiennent à de nouveaux AS, ou font partie d'AS déjà connus. Nous avons pour cela utilisé l'historique des tables de routage du projet *Route Views* [95], qui fournit l'ensemble des chemins depuis quelques moniteurs vers tous les autres AS, enregistrés de manière périodique. Nous avons montré que les deux phénomènes se produisent en même temps : beaucoup des adresses IP que l'on découvre appartiennent à des AS déjà observés, mais un nombre non négligeable de nouveaux AS sont découverts régulièrement.

La question est alors de connaître la *cause* de ces découvertes d'adresses IP (et d'AS) : s'agit-il d'adresses et d'AS nouveaux, c'est-à-dire étant apparus dans le réseau après le début de la mesure, ou deviennent-ils visibles suite à des changements de routage ?

L'historique des tables de routage fourni par *Route Views* permet de répondre à cette question pour les AS. En effet, comme une table de routage permet de router vers toutes les destinations, *a priori* tous les AS existant à un moment donné apparaissent dans une table de routage enregistrée à ce moment. Même si cela n'est pas toujours absolument vrai, le fait d'observer un AS dans une table de routage indique que cet AS existait au moment où cette table a été enregistrée. Au final, sur les 72 AS découverts après le début de la mesure, seulement 2 étaient nouveaux. Les 70 autres existaient depuis le début de la mesure, et sont donc devenus visibles à cause de changements de routage.

On ne peut pas utiliser la même approche pour les adresses IP : on ne dispose pas d'un historique des adresses IP, et quand on observe une nouvelle adresse, il est impossible de dire si elle était allouée plus tôt ou non.

Nous nous sommes donc intéressés au problème inverse, celui des disparitions d'adresses IP. Nous avons en effet constaté que les adresses IP disparaissent des mesures radar au même rythme que de nouvelles adresses sont découvertes. Étudier les disparitions permet donc de mieux comprendre les apparitions. De plus, cette approche a l'avantage qu'il est possible dans une certaine mesure de savoir si une adresse qui a disparu de nos mesures existe toujours sur l'internet, en particulier si elle répond à un `ping`.

Nous avons effectué une mesure dédiée pour étudier cette question. En parallèle d'une mesure radar normale, nous avons maintenu à jour une liste de toutes les adresses IP observées au moins une fois. La mesure envoyait périodiquement des `ping` à chaque adresse de cette liste : ceci permettait de s'assurer que les machines qui continuaient à répondre au `ping` après avoir disparu des mesure radar étaient toujours présentes sur l'internet (mais ne permettait pas de conclure sur celles qui ne répondaient pas au `ping`, une machine pouvant être configurée pour ne jamais y répondre par exemple).

La figure 3.5 (gauche) présente le nombre d'adresses IP qui ont disparu lors de cette mesure, c'est-à-dire le nombre d'adresses que l'on a observées au moins une fois, et que l'on n'observe plus jamais après le temps x. On voit effectivement que la forme de la courbe est similaire à celle de la figure 3.4, à une rotation de 180° près. Parmi les adresses disparues, on voit qu'un grand nombre ont répondu à un `ping` qui a été effectué à la fin de la mesure. La figure 3.5 (droite) présente la fraction des adresses qui ont répondu à ce `ping`, parmi celles qui ont disparu. On observe que cette fraction est stable, et proche de 80%. Ceci signifie que ces adresses sont toujours présentes sur l'internet, et que ce sont donc des changements de routage qui les ont fait disparaître de nos mesures. Ceci confirme que les changements de routage jouent un rôle très important dans la dynamique observée de la topologie de l'internet.

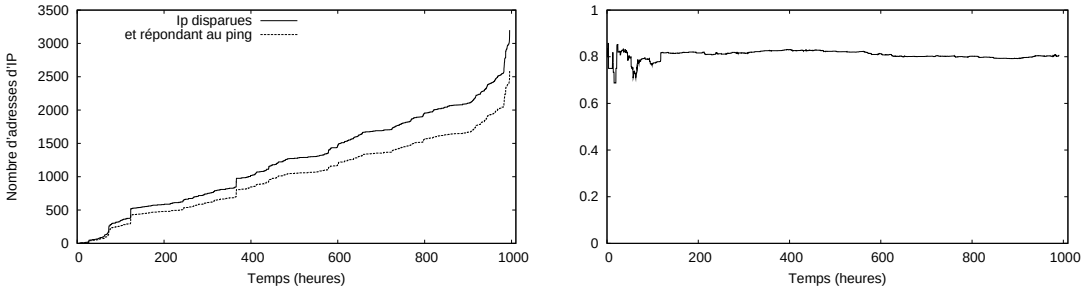

FIGURE 3.5 : Gauche : nombre d'adresses IP observées dans une mesure radar, qui ne sont plus jamais observées après le temps x (nombre total d'adresses, et nombre d'adresses qui ont répondu à un `ping` effectué à la fin de la mesure). Droite : fraction des adresses ayant répondu à un `ping` effectué à la fin de la mesure, parmi les adresses ayant disparu.

3.2.3 Conclusion

Nous avons étudié dans cette section certains aspects clés de la dynamique des vues ego-centrées de l'internet. D'une part, nous avons exhibé des caractéristiques importantes de cette dynamique. Nous avons mis en évidence le fait que les adresses IP ne jouent pas toutes le même rôle dans la dynamique, certaines étant très stables et d'autres très volatiles, certaines ayant tendance à clignoter et d'autres non. Ceci ouvre la voie à une caractérisation plus fine de la dynamique observée, en particulier en étudiant des propriétés de graphes plus élaborées. Nous nous sommes en effet limités ici à des statistiques sur la présence et l'absence de nœuds, et il est certain qu'étudier des propriétés telles que l'apparition de composantes connexes ou les corrélations entre la présence de liens apporterait beaucoup d'informations.

D'autre part, nous avons mis en évidence que la dynamique observée est beaucoup plus rapide que ce à quoi l'on s'attendait. Après avoir montré qu'il ne s'agissait pas d'un artéfact de mesure, nous avons montré que les changements de routage jouent un rôle très important dans cette dynamique.

Il est cependant clair que le *load balancing* joue un rôle non négligeable dans la dynamique observée. Ces deux facteurs intervenant en même temps, il est difficile de déterminer leur importance respective. Nous prévoyons d'étudier formellement cette question en modélisant les mesures radar par l'exploration d'un graphe aléatoire dynamique (la dynamique consistant à modifier à chaque étape un nombre donné de liens choisis au hasard). On peut simuler une mesure radar par un parcours en largeur depuis le moniteur vers les destinations. Un parcours en largeur explore les voisins d'un sommet donné les uns après les autres ; il est possible de simuler le *load balancing* en choisissant à chaque passe de manière aléatoire l'ordre dans lequel les voisins seront considérés. On peut ainsi étudier empiriquement l'influence de ces deux facteurs, et espérer les quantifier formellement. Des études préliminaires sur ce sujet ont montré que cette approche est prometteuse.

Enfin, les observations que nous avons présentées se rapportent à des mesures de durée *finie*, ce qui rejoint la problématique de l'étude de l'influence de la durée de mesure présentée dans la section 2.3. En particulier, des études préliminaires ont montré que la durée de la mesure a une influence sur la distribution du nombre de passes pendant lesquelles les adresses IP sont observées (figure 3.3, gauche). Une perspective importante est donc d'appliquer la méthodologie que nous avons introduite pour tester l'influence de

la durée de la mesure sur les statistiques étudiées dans cette section.

3.3 Conclusion

Nous avons mené une étude de la dynamique de graphes dans deux contextes différents. Dans le premier cas, un phénomène extérieur modifie la structure du graphe, et dans le deuxième il s'agit de la dynamique d'un graphe évoluant au cours du temps.

Bien que nous ayons étudié ces deux phénomènes indépendamment, il semble clair qu'il serait intéressant de les combiner : les graphes de terrain complètement statiques sont rares, et en pratique les phénomènes de pannes ou d'attaques ont lieu sur des réseaux dont la structure évolue en même temps. Étudier ces deux phénomènes indépendamment est cependant un préliminaire indispensable : leur complexité est telle qu'on ne peut espérer l'entamer sans une bonne compréhension préalable de leurs différentes composantes.

De plus, l'analyse de la dynamique de graphes a un intérêt indéniable, indépendamment de tout phénomène ayant lieu sur le graphe. Son but est de décrire cette dynamique et de faire des observations pertinentes. En ce sens, chaque travail d'analyse s'effectue sur un cas particulier et est spécifique à ce cas. On peut cependant chercher à définir des notions transversales, pertinentes à l'ensemble des graphes de terrain dynamiques. En effet, l'étude des graphes de terrain statiques a conduit à la définition d'un ensemble de notions pertinentes pour l'ensemble des graphes de terrain. L'expérience a montré que ces notions permettent d'entamer facilement l'analyse de n'importe quel graphe de terrain statique : lorsque l'on est confronté à un graphe particulier, étudier ces statistiques permet de se faire une bonne intuition générale sur l'objet étudié, et dans de nombreux cas d'isoler des aspects qui méritent un approfondissement.

Cet ensemble de notions n'existe pas à l'heure actuelle pour les graphes dynamiques. Dans les deux cas que nous avons étudiés, nous avons été freinés par le fait que nous devions à la fois étudier une dynamique particulière, sans intuition préalable sur ses caractéristiques, et inventer les notions pour l'étudier. Par exemple, dans le cas de la dynamique des vues ego-centrées, disposer d'un ensemble de notions reconnues nous aurait sans conteste aidés dans nos études et aurait probablement permis d'isoler d'autres facteurs intéressants.

Pour cette raison, une perspective majeure de nos travaux est de contribuer à créer cet ensemble de notions. Ceci passe par deux aspects. D'une part, il s'agit de compléter les études que nous avons effectuées en introduisant de nouvelles statistiques. De l'autre, il s'agit d'étudier ces statistiques sur plusieurs cas différents afin de valider leur pertinence dans le cas général.

CONCLUSION

J'ai présenté dans ce mémoire mes travaux sur les graphes de terrain dynamiques selon les trois axes de la mesure, la métrologie et l'analyse. Ceci a l'avantage, par rapport à une présentation structurée en fonction des objets étudiés, de permettre de dégager naturellement des questions et résultats généraux dans chacune de ces problématiques.

Ce découpage n'est cependant pas complètement satisfaisant. Nous avons déjà vu qu'il existe des liens naturels entre mesure, métrologie et analyse : par exemple, la mesure nourrit la métrologie et l'analyse, car elle est un préalable indispensable à toute étude ; le test des paramètres de mesure peut être naturellement vu soit comme de la mesure, si le souci est de s'assurer que le système de mesure est correctement paramétré, soit comme de la métrologie, si l'on se pose la question de savoir à quel point des paramètres comme la fréquence de mesure ou le nombre de points de mesure influent sur les propriétés observées de la dynamique.

Ces liens ne remettent toutefois pas en cause le fait qu'il s'agit en général de familles de problématiques bien distinctes. Je vais maintenant montrer que ceci n'est plus vrai dans certains cas, pour lesquels étudier la dynamique d'un graphe de terrain indépendamment de la manière dont les données ont été collectées n'a pas forcément de sens.

Revenons par exemple sur les travaux présentés dans la section 2.1 sur le biais introduit par le *load balancing* dans des mesures de type `traceroute`. L'idée était de comparer des mesures effectuées avec `traceroute` et `paris-traceroute`, ce dernier permettant de supprimer une partie du *load balancing*. Les liens observés avec le premier mais pas le deuxième étaient donc des artéfacts dus au *load balancing*, et en particulier on s'attendait à observer strictement moins de liens avec `paris-traceroute` qu'avec `traceroute`. Nous avions cependant fait une observation surprenante, que nous n'avions pas su expliquer à l'époque : nous avions observé un nombre faible mais non négligeable de liens avec `paris-traceroute` qui n'étaient pas visibles avec `traceroute`.

Il est maintenant facile de comprendre ce phénomène, suite aux travaux sur la dynamique des vues ego-centrées (présentés dans la section 3.2). Nous avions fait à l'époque l'hypothèse que la topologie observable avec `traceroute` était (raisonnablement) stable. Le *load balancing* fait que l'ensemble des nœuds et liens que l'on peut observer varie d'une passe de mesure à l'autre, mais nous supposions que l'ensemble observable était fixe et relativement restreint. Nous savons maintenant que ce n'est pas le cas, et que les vues ego-centrées évoluent très rapidement. En particulier, un nombre important de

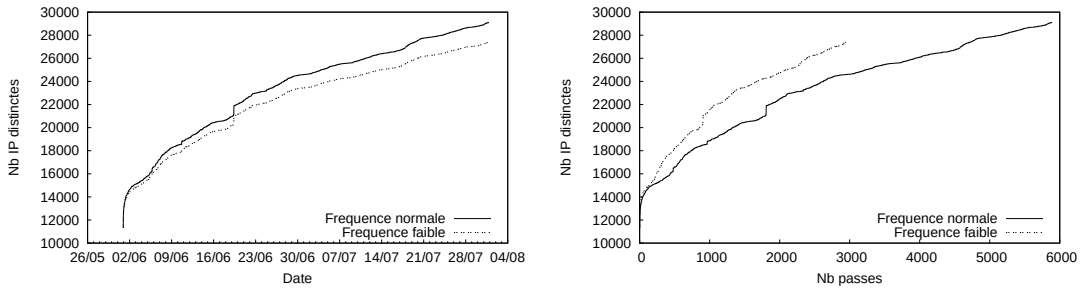

FIGURE 3.6 : Comparaison entre une mesure radar normale et une mesure radar à fréquence deux fois plus faible. Nombre d'adresses IP observées depuis le début de la mesure. Gauche : en fonction du temps écoulé depuis le début de la mesure ; droite : en fonction du nombre de passes effectuées.

nœuds ne sont observés qu'une seule fois, ou un très petit nombre de fois. Lorsque l'on effectue des mesures en parallèle avec `traceroute` et `paris-traceroute`, il est donc naturel d'observer certains nœuds et liens avec l'un des outils et pas l'autre. De même, certains des liens que nous avions classés parmi les artéfacts liés au *load balancing* parce que nous les avions observés avec `traceroute` mais pas avec `paris-traceroute` étaient des événements rares, et non des artéfacts.

Ceci n'enlève à mon avis rien à la valeur du travail effectué à l'époque, mais cela met en évidence un cas où une question de métrologie ne peut être complètement résolue sans travaux d'analyse.

Pour continuer dans cette direction, revenons sur la croissance continuelle du nombre d'adresses IP observées au cours du temps par une mesure radar. Nous avons vu qu'au moins deux facteurs jouent un rôle dans ce phénomène : le *load balancing* et une dynamique qui change les routes existantes entre le moniteur et les destinations (je ne m'intéresse pas ici à la question de savoir s'il s'agit uniquement d'une dynamique de routage entre nœuds existants, ou si des nœuds apparaissent ou disparaissent dans la topologie). Ces deux facteurs jouent des rôles différents. Des changements dans les routes causent une augmentation du nombre d'adresses observées d'autant plus grande qu'il s'écoule plus de temps entre le début et la fin de la mesure : plus la durée de la mesure est grande, plus il se produit de changements. Le *load balancing* au contraire cause une augmentation du nombre d'adresses observées d'autant plus grande que le nombre de passes effectuées est grand, indépendamment du temps qui s'écoule entre elles [5].

Pour tenter d'isoler ces deux facteurs, étudions l'influence de la fréquence de mesure sur le nombre d'adresses observées. Pour cela, considérons une mesure radar normale et simulons une mesure plus lente en ne considérant qu'une passe sur deux (la mesure lente a donc lieu à une fréquence d'une passe toutes les demi-heures, et la mesure rapide d'une passe tous les quarts d'heure). La figure 3.6 présente le nombre d'adresses IP découvertes par ces mesures, en fonction du temps écoulé depuis le début de la mesure (gauche), et du nombre de passes de mesure effectuées (droite).

Comme on pouvait s'y attendre, la mesure lente découvre moins d'adresses IP au fil du temps que la mesure plus rapide (figure 3.6, gauche). En un intervalle de temps fixe,

5. Dans les deux cas, ceci n'est valable que dans certaines limites sur le nombre de passes effectuées et l'intervalle de temps qui s'écoule entre elles.

effectuer plus de passes de mesure permet donc de découvrir plus d'adresses IP, ce qui montre bien qu'il faut effectuer plusieurs passes pour découvrir la totalité de ce qui est visible, indépendamment d'éventuels changements de routes. Ceci est dû à des facteurs tels que le *load balancing*. Inversement, la mesure lente découvre plus d'adresses IP *par passe* que la mesure rapide (figure 3.6, droite). Attendre plus longtemps entre deux passes consécutives permet de découvrir plus d'adresses IP à chaque passe, ce qui montre bien que les routes changent au fil du temps.

Ces observations sont importantes pour deux raisons principales. Tout d'abord, elles permettent de mettre en évidence le rôle joué par deux facteurs différents dans la dynamique des vues ego-centrées, ce qui est un résultat important pour l'étude de la topologie de l'internet. Mais surtout, elles montrent que l'analyse des données radar a des limites : il est par exemple illusoire de chercher à quantifier la vitesse d'évolution d'une vue ego-centrée en calculant la pente d'une courbe similaire à celles de la figure 3.6, car elle dépend intrinsèquement de la fréquence de mesure.

Dans ce cas, le découpage entre mesure, métrologie et analyse n'est pas complètement approprié, et l'on devine aisément qu'il ne s'agit pas d'un cas isolé et que la même constatation pourrait certainement être faite dans d'autres contextes. Dans ces cas, ne pas mettre de frontière entre ces trois problématiques permet de comprendre plus en profondeur les graphes étudiés. Ceci ne veut pas dire que ce découpage n'est pas pertinent dans le cas de la dynamique de graphes de terrain : il souligne la cohérence d'ensemble du domaine, et permet de le structurer. Je pense au contraire que ceci indique que la prise en compte de la dynamique fait émerger d'autres grandes problématiques transversales qui restent à identifier de manière ferme, et que l'influence de la fréquence de mesure est probablement l'une d'entre elles.

Mes travaux ouvrent plusieurs perspectives, à plus ou moins long terme. Parmi elles, deux me semblent particulièrement importantes. Il s'agit tout d'abord de l'influence de la fréquence et de la durée de mesure sur les propriétés observées de la dynamique d'un graphe de terrain. On a vu qu'il s'agit d'une question importante. D'un côté, je souhaite poursuivre les études que j'ai menées. En particulier, appliquer les analyses que j'ai effectuées sur l'influence de la durée de la mesure (section 2.3) et de sa fréquence (ci-dessus) à d'autres cas apportera certainement une bonne compréhension des principes sous-jacents. D'autre part, il est possible d'effectuer des simulations. Pour étudier l'influence de la durée de mesure sur la durée de vie observée des nœuds, on peut par exemple injecter une distribution de durée de vie connue, et simuler des mesures plus ou moins longues, ce qui permettrait de se faire une intuition de l'influence de la durée de mesure, et ce pour plusieurs types de distributions sous-jacentes. À terme, on peut espérer dériver des résultats formels indiquant par exemple, sous certaines conditions sur la distribution réelle des durées de vie des nœuds, quelle durée doit avoir la mesure pour observer cette distribution avec un intervalle de confiance donné.

Ensuite, les observations faites sur la dynamique des vues ego-centrées ouvrent des perspectives fondamentales sur la cartographie de l'internet. En effet, nous avons montré que les vues ego-centrées se renouvellent constamment, ce qui fait qu'une partie des informations observées devient très vite obsolète. En contrepartie, comme il existe plusieurs chemins d'un même moniteur vers une même destination, obtenir tous les chemins, quelle que soit la méthode utilisée, prend du temps. Il y a donc un compromis à faire entre

la quantité d'informations obtenue et le fait que ces informations soient à jour ou non, et étudier ce compromis est une question importante. Une autre approche consisterait à mettre au point des techniques permettant d'avoir plus ou moins confiance dans le fait que les nœuds et liens observés sont toujours présents, en fonction du nombre de fois où ils ont été observés précédemment par exemple.

À plus long terme, il existe un fort besoin de notions et d'outils standards pour l'analyse de la dynamique des graphes de terrain. Un certain nombre de travaux ont déjà étudié des cas particuliers, et quelques notions générales commencent à émerger. Pour aller dans cette direction, il sera nécessaire d'effectuer les mêmes analyses sur plusieurs graphes différents, ce qui permettra d'identifier les notions pertinentes dans le cas général. À ce sujet, une question importante reste en suspens : il a été montré que la plupart des graphes de terrain, si on les considère comme statiques, se ressemblent au sens de certaines propriétés, bien qu'ils proviennent de contextes très différents. La question est donc de savoir si l'analyse de leur dynamique mettra également en évidence de telles ressemblances.

Enfin, il existe un besoin avéré de modèles de graphes de terrain dynamiques, pour plusieurs applications. Comme je l'évoquais plus haut, l'état de l'art n'a pas abouti à l'heure actuelle à une bonne compréhension de la dynamique, ce qui rend difficile de proposer des modèles de graphes dynamiques réalistes pour un graphe particulier. Il est toutefois possible de proposer des modèles de dynamique aléatoire, et quelques propositions ont été faites en ce sens. Ceci est cohérent avec le fait que les graphes aléatoires au sens d'Erdös-Rényi étaient couramment utilisés pour modéliser divers graphes de terrain (statiques) avant que l'on découvre que ce n'était pas satisfaisant. Dans notre contexte, il semble donc naturel d'imaginer que l'identification de propriétés dynamiques pertinentes mènera à la proposition de modèles reproduisant ces propriétés, comme dans le cas statique.

BIBLIOGRAPHIE

[1] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore. On the bias of traceroute sampling ; or, power-law degree distributions in regular graphs. In *ACM Symposium on Theory of Computing (STOC2005)*, 2005.

[2] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore. On the bias of traceroute sampling. *Journal of the ACM*, 56(4) :1–28, 2009.

[3] William Acosta and Surendar Chandra. Trace Driven Analysis of the Long Term Evolution of Gnutella Peer-to-Peer Traffic. In *PAM*, pages 42–51, 2007.

[4] E. Adar. User 4xxxxx9 : Anonymizing query logs. In *Workshop on Query Log Analysis at the 16th World Wide Web Conference*, 2007.

[5] Eytan Adar and Bernardo A. Huberman. Free Riding on Gnutella. *First Monday*, 5, 2000.

[6] Frédéric Aidouni, Matthieu Latapy, and Clémence Magnien. Ten weeks in the life of an eDonkey server. In *Proceedings of HotP2P'09*, 2009.

[7] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics 74, 47*, 2002.

[8] R Albert, H Jeong, and A-L Barabási. Error and attack tolerance of complex networks. *Nature*, 406 :378–82, 2000.

[9] Oussama Allali, Matthieu Latapy, and Clémence Magnien. Measurement of eDonkey Activity with Distributed Honeypots. In *Proceedings of HotP2P'09*, 2009.

[10] Mark Alllman and Vern Paxson. Issues and etiquette concerning use of shared measurement data. In *IMC '07 : Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 135–140, New York, NY, USA, 2007. ACM.

[11] Juan A. Almendral and Albert Díaz-Guilera. Dynamical and spectral properties of complex networks. *New Journal of Physics*, 9 :187, 2007.

[12] Caida – archipelago project. `http://www.caida.org/projects/ark/`.

[13] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Traceroute anomalies : Detection and prevention in internet graphs. In *Proceedings of the international conference* ACM Internet Measurement Conference IMC'06, 2006.

[14] M.M. Babu, N.M. Luscombe, L. Aravind, M. Gerstein, and S.A. Teichmann. Structure and evolution of transcriptional regulatory networks. *Curr Opin Struct Biol.*, 14(3) :283–291, 2004.

[15] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks : Membership, growth, and evolution. In *Proc. 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2006.

[16] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286 :509–512, 1999.

[17] Lamia Benamara and Clémence Magnien. Estimating properties in dynamic systems : the case of churn in P2P networks. In *Proceedings of the Second International Workshop on Network Science for Communication Networks (NetSciCom 2010), in conjunction with IEEE INFOCOM*, 2010.

[18] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory (A)*, 24 :357–367, 1978.

[19] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statstical Mechanics : Theory and Experiment*, page P10008, 2008.

[20] B. Bui-Xuan, A. Ferreira, and A. Jarry. Evolving graphs and least cost journeys in dynamic networks. In *Proceedings of WiOpt'03 – Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks*, pages 141–150, Sophia Antipolis, March 2003. INRIA Press.

[21] R. Calegari, M. Musolesi, F. Raimondi, and C. Mascolo. CTG : A connectivity trace generator for testing the performance of opportunistic mobile systems. In *European Software Engineering Conference and the International ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE07)*, Dubrovnik, Croatia, 2007.

[22] D.S. Callaway, M.E.J. Newman, S.H. Strogatz, and D.J. Watts. Network robustness and fragility : Percolation on random graphs. *Physical Review Letters*, 85 :5468–5471, 2000.

[23] A. Chaintreau, J. Crowcroft, C. Diot, R. Gass, P. Hui, and J. Scott. Pocket switched networks and the consequences of human mobility in conference environments. In *WDTN*, pages 244–251, 2005.

[24] Jeffrey Chan, James Bailey, and Christopher Leckie. Discovering correlated spatio-temporal changes in evolving graphs. *Knowledge and Information Systems*, 16(1) :53–96, 2008.

[25] Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. The Origin of Power-Laws in Internet Topologies Revisited. In *IEEE Infocom*. IEEE, 2002.

[26] Thibault Cholez, Isabelle Chrisment, and Olivier Festor. A Distributed and Adaptive Revocation Mechanism for P2P Networks. In *Seventh International Conference on Networking (ICN 2008)*, 2008.

[27] A. Clauset and N. Eagle. Persistence and periodicity in a dynamic proximity network. In *DIMACS Workshop*, 2007.

[28] A. Clauset and C. Moore. Accuracy and scaling phenomena in internet mapping. *Phys. Rev. Lett*, 2005.

[29] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdown. *Physical Review Letters*, 85 :4626, 2000.

[30] Jean-Philippe Cointet and Camille Roth. Socio-semantic Dynamics in a Blog Network. In *IEEE SocialCom 09 Intl Conf Social Computing*, pages 114–121. IEEE CS, 2009.

[31] L. Dall'Asta, J.I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani. A statistical approach to the traceroute-like exploration of networks : theory and simulations. In *Workshop on combinatorial and Algorithmic Aspects of Networking (CAAN)*, 2004.

[32] Dimes website. `http://www.netdimes.org/new/`.

[33] B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Efficient algorithms for large-scale topology discovery. In *Proc. ACM SIGMETRICS*, Jun. 2005.

[34] *eDonkey* server data. `http://www-rp.lip6.fr/~latapy/tenweeks/`.

[35] P. Erdös and A. Rényi. On Random Graphs I. *Publ. Math. Debrecen*, 6 :290–297, 1959.

[36] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find community structures based on information centrality. *Physical Review E*, 70(5) :056104, 2004.

[37] Yaniv Frishman and Ayellet Tal. Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics*, 14 :727–740, 2008.

[38] Tryphon T. Georgiou, Johan Karlsson, and Mir Shahrouz Takyar. Metrics for Power Spectra : An Axiomatic Approach. *IEEE Transactions on Signal Processing*, 57(3) :859–867, March 2009.

[39] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12) :7821–7826, 2002.

[40] Carsten Görg, Peter Birke, Mathias Pohl, and Stephan Diehl. *Dynamic Graph Drawing of Sequences of Orthogonal and Hierarchical Graphs*, pages 228–238. Springer Berlin/Heidelberg, 2005.

[41] R. Govindan and A. Reddy. An Analysis of Internet Inter-Domain Topology and Route Stability. In *IEEE INFOCOM*. IEEE, 1997.

[42] Mark Granovetter. Network Sampling : Some First Steps. *The American Journal of Sociology*, 81(6) :1287–1303, 1976.

[43] J.-L. Guillaume and M. Latapy. Complex networks metrology. In *Complex systems*, 2005.

[44] J.-L. Guillaume and M. Latapy. Relevance of massively distributed explorations of the internet topology : Simulation results. In *IEEE infocom*, 2005.

[45] Jean-Loup Guillaume, Matthieu Latapy, and Stevens Le-Blond. Statistical analysis of a P2P query graph based on degrees and their time-evolution. In *IWDC*, pages 126–137, 2004.

[46] Jean-Loup Guillaume, Matthieu Latapy, and Clémence Magnien. Comparison of Failures and Attacks on Random and Scale-free Networks. In *Proc. of the 8-th International Conference on Principles of Distributed Systems (OPODIS'04)*, 2004.

[47] Jean-Loup Guillaume, Matthieu Latapy, and Damien Magoni. Relevance of massively distributed explorations of the internet topology : Qualitative results. *Computer Networks*, 50 (16) :3197–3224, 2006.

[48] Assia Hamzaoui, Matthieu Latapy, and Clémence Magnien. Detecting Events in the Dynamics of Ego-centered Measurements of the Internet Topology. In *Proceedings of International Workshop on Dynamic Networks (WDN), in conjunction with WiOpt 2010*, 2010. To appear.

[49] S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant, L. Massoulié, and S. Patarin. Peer sharing behaviour in the edonkey network, and implications for the design of server-less file sharing systems. In *EuroSys '06*, pages 359–371, New York, NY, USA, 2006. ACM.

[50] B. Huffaker, D. Plummer, D. Moore, and k. claffy. Topology discovery by active probing. In *Symposium on Applications and the Internet*, 2002.

[51] Daniel Hughes, Geoff Coulson, and James Walkerdine. Free Riding on Gnutella Revisited : The Bell Tolls ? *IEEE Distributed Systems Online*, 6(6), 2005.

[52] Daniel Hughes, James Walkerdine, Geoff Coulson, and Stephen Gibson. Peer-to-peer : Is deviant behavior the norm on P2P file-sharing networks ? *IEEE Distributed Systems Online*, 7(2), 2006.

[53] University of washington – iPlane project. `http://iplane.cs.washington.edu/data.html`.

[54] Mikel Izal, Guillaume Urvoy-Keller, Ernst W Biersack, Pascal A Felber, Anwar Al Hamra, and Luis Garces-Erice. Dissecting BitTorrent : five months in a torrent's lifetime. In *5th annual Passive & Active Measurement Workshop (PAM'2004)*, 2004.

[55] V. Jacobson. traceroute, February 1989. The most recent version is available at : `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`.

[56] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. Transport layer identification of P2P traffic. In *IMC '04 : Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134, New York, NY, USA, 2004. ACM.

[57] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In *Proc. 32nd ACM Symposium on Theory of Computing*, 2000.

[58] Y. Kulbak and D. Bickson. The eMule Protocol Speficiation. Technical report, Hebrew University of Jerusalem, 2005.

[59] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Origins of internet routing instability. In *INFOCOM*, pages 218–226, 1999.

[60] M. Lad, D. Massey, and L. Zhang. Visualizing Internet Routing Changes. *IEEE Transactions on Visualization and Computer Graphics, special issue on Visual Analytics*, 2006.

[61] Mayank Lahiri and Tanya Y. Berger-Wolf. Mining Periodic Behavior in Dynamic Social Networks. In *IEEE International Conference on Data Mining*, pages 373–382. IEEE, 2008.

[62] A. Lakhina, J. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. In *IEEE INFOCOM*, 2003.

[63] Matthieu Latapy. Grands graphes de terrain – mesure et métrologie, analyse, modélisation, algorithmique. Mémoire d'habilitation à diriger les recherches, UPMC, 2007. `http://www-rp.lip6.fr/~latapy/HDR/`.

[64] Matthieu Latapy and Clémence Magnien. Complex network measurements : Estimating the relevance of observed properties. In *Proceedings of IEEE Infocom*, 2008.

[65] Matthieu Latapy, Clémence Magnien, and Frédéric Ouédraogo. A radar for the internet. In *Proceedings of ADN'08 : 1st International Workshop on Analysis of Dynamic Networks, in conjonction with IEEE ICDM*, 2008.

[66] Farah Layouni, Brice Augustin, Timur Friedman, and Renata Teixeira. Origine des étoiles dans traceroute. In *Colloque francophone sur l'ingénierie des protocoles*, 2008.

[67] Stevens Le-Blond, Jean-Loup Guillaume, and Matthieu Latapy. Clustering in P2P exchanges and consequences on performances. In *IPTPS*, pages 193–204, 2005.

[68] Stevens Le Blond, Fabrice Le Fessant, and Erwan Le Merrer. Finding Good Partners in Availability-Aware P2P Networks. In *Stabilization, Safety, and Security of Distributed Systems (SSS)*, volume 5873 of *Lecture Notes in Computer Science*, pages 472–484, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[69] F. Le Fessant, S. Handurukande, A. M. Kermarrec, and L. Massoulié. Clustering in peer-to-peer file sharing workloads. In *3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, San Diego, CA, February 2004.

[70] U. Lee, M. Choi, J. Cho, M. Y. Sanadidi, and M. Gerla. Understanding pollution dynamics in P2P file sharing. In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, 2006.

[71] Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki. Deconstructing the Kazaa Network. In *WIAPP '03 : Proceedings of the The Third IEEE Workshop on Internet Applications*, page 112, Washington, DC, USA, 2003. IEEE Computer Society.

[72] E. A. Leicht, G. Clarkson, K. Shedden, and M. E. J. Newman. Large-scale structure of time evolving citation networks. *Eur. Phys J. B*, 59, 2007.

[73] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution : Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*, 1(1), 2007. Arxiv physics/0603229.

[74] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

[75] Clémence Magnien, Matthieu Latapy, and Jean-Loup Guillaume. Impact of Random Failures and Attacks on Poisson and Power-Law Random Networks. *ACM Computing Surveys*, 2010. To appear.

[76] Clémence Magnien, Frédéric Ouedraogo, Guillaume Valadon, and Matthieu Latapy. Fast dynamics in internet topology : preliminary observations and explanations. In *Proceedings of the Fourth International Conference on Internet Monitoring and Protection (ICIMP)*, 2009.

[77] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6 :161–179, 1995.

[78] T. Moors. Streamlining traceroute by estimating path lengths. In *Proc. IEEE International Workshop on IP Operations and Management (IPOM)*, Oct. 2004.

[79] G. Neglia, G. Reina, H. Zhang, D.F. Towsley, A. Venkataramani, and J.S. Danaher. Availability in bitTorrent systems. In *INFOCOM*, 2007.

[80] Mark Newman, Albert-Laszlo Barabási, and Duncan J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, Princeton, USA, 2006.

[81] R. Oliveira, B. Zhang, and L. Zhang. Observing the evolution of internet AS topology. In *ACM SIGCOMM*, 2007.

[82] Frédéric Ouédraogo and Clémence Magnien. Impact of sources and destinations on the observed properties of the internet topology, 2010. Submitted.

[83] Outils pour évaluer la pertinence d'un échantillon, programme et données. `http://www-rp.lip6.fr/~latapy/Measurement/`.

[84] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446 :664, 2007.

[85] Jean-Jacques Pansiot. Local and Dynamic Analysis of Internet Multicast Router Topology. *Annales des télécommunications*, 62 :408–425, 2007.

[86] `paris-traceroute`. Available at `http://paris-traceroute.net/`.

[87] S.-T. Park, D. M. Pennock, and C. L. Giles. Comparing static and dynamic measurements and models of the Internet's AS topology. In *IEEE Infocom*. IEEE, 2004.

[88] T. Petermann and P. De Los Rios. Exploration of scale-free networks. *Eur. Phys. J.B*, 38(2), 2004.

[89] Planetlab. `http://www.planet-lab.org/`.

[90] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips. The Bittorrent P2P File-sharing System : Measurements and Analysis. In *4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, 2005.

[91] Robert J J. Prill, Pablo A A. Iglesias, and Andre Levchenko. Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol*, 3(11) :1881–1892, 2005.

[92] Radar program and data. `http://www-rp.lip6.fr/~latapy/Radar`.

[93] P. De Los Rios. Exploration bias of complex networks. In *Proceedings of the 7th conference on Statistical and Computational Physics Granada*, 2002.

[94] D. Roselli, J. R. Lorch, and T. E. Anderson. A comparison of file system workloads. In *Proc. of USENIX Annual Technical Conference*, 2000.

[95] University of Oregon – Route Views Project. `http://www.routeviews.org/`.

[96] Walid Saddi and Fabrice Guillemin. Measurement based modeling of eDonkey peer-to-peer file sharing system. In *International Teletraffic Congress*, pages 974–985, 2007.

[97] S. Saroiu, P.K. Gummadi, and S.D. Gribble. A measurement study of peer-to-peer file sharing systems. In *MMCN*, 2002.

[98] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Systems*, 9 :170–184, 2003.

[99] A. Scherrer, P. Borgnat, E. Fleury, J.-L. Guillaume, and C. Robardet. Description and simulation of dynamic mobility networks. *Computer Network*, 52 :2842–2858, 2008.

[100] Fabian Schneider, Anja Feldmann, Balachander Krishnamurthy, and Walter Willinger. Understanding online social network usage from a network perspective. In *Internet Measurement Conference*, pages 35–48, 2009.

[101] Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Trans. Netw.*, 12(2) :219–232, 2004.

[102] Caida – Skitter project. `http://www.caida.org/tools/measurement/skitter/`.

[103] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM*, 2002.

[104] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. A global view of kad. In *Internet Measurement Conference*, 2007.

[105] Ralf Steuer, Adriano Nunes Nesi, Alisdair R. Fernie, Thilo Gross, Bernd Blasius, and Joachim Selbig. From structure to dynamics of metabolic pathways : application to the plant mitochondrial TCA cycle. *Bioinformatics*, 23(11) :1378–85, June 2007.

[106] Alina Stoica and Christophe Prieur. Structure of Neighborhoods in a Large Social Network. In *IEEE International Conference on Social Computing (SocialCom-09)*. IEEE, 2009.

[107] Michael P. H. Stumpf, Carsten Wiuf, and Robert M. May. Subnets of scale-free networks are not scale-free : sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12) :4221–4, 2005.

[108] D. Stutzbach, R. Rejaie, N.G. Duffield, S. Sen, and W. Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Transactions on Networking*, 2008.

[109] Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *Internet Measurement Conference*, pages 189–202, 2006.

[110] Daniel Stutzbach, Shanyu Zhao, and Reza Rejaie. Characterizing files in the modern Gnutella network. *Multimedia Systems*, 13(1) :35–50, 2007.

[111] F. Tarissan, M. Latapy, and C. Prieur. Efficient measurement of complex networks using link queries. In *Proceedings of IEEE International Workshop on Network Science For Communication Networks (NetSciCom'09)*, 2009.

[112] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. In search of path diversity in ISP networks. In *Proceedings of the Internet Measurement Conference (IMC)*, 2003.

[113] Pierre Ugo Tournoux, Jérémie Leguay, Marcelo Dias de Amorim, Farid Benbadis, Vania Conan, and John Whitbeck. The Accordion Phenomenon : Analysis, Characterization, and Impact on DTN Routing. In *Proceedings of the 28rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1116–1124. IEEE, 2009.

[114] Traceroute@home website. `http://www.traceroutеathome.net/`.

[115] K. Tutschku. A measurement-based traffic profile of the eDonkey filesharing service. In *ACM PAM*, 2004.

[116] A. Vázquez, R. Dobrin, D. Sergi, J. P. Eckmann, Z. N. Oltvai, and A. L. Barabási. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *PNAS*, 101 :17940–17945, 2004.

[117] A. Vázquez, J.G. Oliveira, and A.-L. Barabási. The inhomogeneous evolution of subgraphs and cycles in complex networks. *Physical Review E*, 71 :025103, 2005.

[118] Fabien Viger, Brice Augustin, Xavier Cuvellier, Matthieu Latapy, Clémence Magnien, Timur Friedman, and Renata Teixeira. Detection, understanding, and prevention of traceroute measurement artifacts. *Computer Networks*, 52 :998–1018, 2008.

[119] Xiaoming Wang, Zhongmei Yao, and Dmitri Loguinov. Residual-based estimation of peer and link lifetimes in P2P networks. *IEEE/ACM Transactions on Networking (TON)*, 17, 2009.

[120] S. Wasserman and K. Faust. *Social network analysis*. Cambridge University Press, Cambridge, 1994.

[121] D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684) :440–2, June 1998.

[122] M. Zghaibeh and K. Anagnostakis. On the impact of P2P incentive mechanisms on user behavior. In *NetEcon+IBC*, 2007.