THÈSE

dirigée par Matthieu Latapy
co-dirigée par Christophe Crespelle

présentée pour obtenir le grade de

DOCTEUR EN SCIENCES
DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

spécialité Informatique

Une approche pour l'estimation fiable des propriétés de la topologie de l'Internet

Élie Rotenberg

soutenue publiquement le 8 janvier 2015 devant le jury composé de

Rapporteurs :	Bertrand Jouve	Directeur de recherches, CNRS
	Jean-Jacques Pansiot	Professeur émérite, Université de Strasbourg
Examinateurs :	Clémence Magnien	Chargée de recherches, CNRS
	Pascal Mérindol	Maître de conférences, Université de Strasbourg
	Philippe Owezarski	Directeur de recherches, CNRS
Directeur :	Matthieu Latapy	Directeur de recherches, CNRS
Co-directeur :	Christophe Crespelle	Maître de conférences, UCBL

Remerciements

Mes premiers remerciements vont aux rapporteurs de cette thèse, Bertrand Jouve et Jean-Jacques Pansiot, et aux examinateurs, Clémence Magnien, Pascal Mérindol et Philippe Owezarski. Je tiens également à remercier mes encadrants de thèse, Matthieu Latapy et Christophe Crespelle, pour votre patience à mon égard. Je sais que je n'ai pas été un doctorant très facile à encadrer (je prépare cette litote depuis cinq ans), et vous avez fait preuve d'une opiniâtreté, d'une compréhension et d'une persévérance remarquables. Vous m'avez fait confiance même quand je n'avais plus confiance moi-même, vous m'avez canalisé dans les moments les plus chaotiques. Je suis fier d'avoir été encadré par vous. Vous allez pouvoir reprendre une vie normale…

Mes parents, Béatrice et Lionel, je ne souhaite pas me contenter de vous remercier, je profiterai de cette saynette en marge du théâtre familial pour déclarer que vous êtes les meilleurs parents du monde. En dépit des joutes formelles, de la névrose pathologique que vous m'avez soigneusement inculquée, vous m'avez également prodigué tout ce qu'un enfant peut rêver de ses parents, et même davantage. Le goût du savoir, de la curiosité, du travail, de l'accomplissement, ne se trouvent pas dans les gènes, mais dans l'éducation, et c'est bien à vous deux que je la dois en premier lieu.

Je voudrais remercier deux personnes que je considère comme mes mentors. Guy Alarcon, vous m'avez donné accès aux délices de l'abstraction et des approches formelles, vous avez mis le feu à une poudre qui depuis, éclaire mon chemin. Nathalie Collin, tu m'as toujours prodigué confiance, conseil et inspiration, et ces éléments ont été et sont toujours extrêmement moteurs dans ma vie.

Je remercie mon entourage le plus proche, à commencer par Marie, mais aussi mes frères et soeur et leurs compagnons respectifs, Benjamin et Anne-Solène, Déborah et Guillaume, Samuel et Constance. Me suppporter au quotidien, quand j'étais plus jeune et insupportable ou maintenant (je suis toujours insupportable), n'est pas une mince affaire, et que nous y ayons tous survécu mérite des féliciations et ma gratification. J'adresse également une mention particulière à Nicolas Bonifas et Antoine Viannay, deux de mes plus vieux amis, pour leur soutien dans les situations les plus rocambolesques.

Enfin, je souhaite remercier mes collaborateurs. Ceux du monde académique, de l'équipe Complex Networks (ou ex-), avec qui j'ai eu la chance de travailler, ou tout simplement discuter (que j'ai eu l'occasion d'assomer de ma cuistrerie) lors de mes apparitions : Fabien, Jean-Lou, Lionel, Raphaël, Daniel, Maximilien, Jordan, Sergey, Alice, Noé, Sébastien, Adrien, et tous les anciens de l'équipe. Et ceux du monde vidéo-ludique, de Millenium, qui m'ont vu mener une curieuse vie pendant plusieurs années, en particulier mon associé Cédric Page et sa femme Naïma.

Je m'excuse de ne pas pouvoir ici remercier les nombreuses personnes qui ont eu un impact déterminant dans ma vie, et donc dans mon travail. Si vous arrivez jusqu'à ce paragraphe, c'est probablement que vous méritez mes remerciements également. Bon courage pour la lecture de la suite.

Table des matières

Remerciements **3**

1 Introduction **11**
1.1 Préliminaires 13
1.2 Approches historiques 19
1.2.1 Génèse de la problématique 19
1.2.2 Modèles formels d'Internet 20
1.2.3 Cartographie d'Internet 23
1.3 Notre approche 24
1.4 Organisation 25

2 Mesure de la topologie logique **27**
2.1 Interprétation rigoureuse de TRACEROUTE 28
2.2 Primitive de mesure de bas niveau basée sur TRACEROUTE 34
2.3 Primitive de mesure de haut niveau basée sur TRACEROUTE 36
2.4 Estimation de la distribution de degré du cœur d'Internet au niveau logique 42
2.5 Échantillonage des cibles dans le cœur 45
2.6 Filtrage des résultats 46
2.7 Simulations 50
2.8 Mesure avec *Planetlab* 52
2.9 Protocole complet 56
2.10 Limites de l'approche 59
2.11 Conclusion 61

3 Mesure de la topologie physique **63**
3.1 Primitive de mesure de bas niveau basée sur UDP PING 64
3.2 Primitive de mesure de haut niveau basée sur UDP PING 67
3.3 Echantillonage rigoureux dans le cœur 68
3.3.1 Echantillonage d'adresses IP de routeurs du cœur 70
3.3.2 Correction du biais 71
3.4 Validation du principe 73
3.5 Evaluation d'un ensemble de moniteurs 77
3.5.1 Colocalisation des moniteurs 77

3.5.2 Diversité des observations 80
3.5.3 Convergence des résultats 83
3.6 Mesure réelle 83
3.6.1 Déroulement 84
3.6.2 Résultats 86
3.7 Protocole complet 89
3.8 Validation 90
3.8.1 Qualité de l'ensemble de moniteurs 90
3.8.2 Réinjection dans les simulations 92
3.9 Discussion et conclusion 95

4 Mesure des tables de routage 99
4.1 Structure des tables de transmission 99
4.2 Contraintes obtenues avec UDP Ping 101
4.3 Inférence 103
4.3.1 Schéma le plus spécifique 104
4.3.2 Schéma le moins spécifique 104
4.3.3 Schéma AS 106
4.4 Mesure réelle avec *Planetlab* 107
4.4.1 Conditions de la mesure 107
4.4.2 Résultats de la mesure 108
4.5 Conclusion 109

5 Conclusions et perspectives 113
5.1 Contributions 113
5.2 Perspectives 115
5.2.1 Approfondissement d'UDP Ping 115
5.2.2 Echantillonage orienté propriété 119
5.2.3 Nouveaux objets d'intérêt 120

Bibliographie 123

Table des figures

1.1 Couches d'Internet 14
1.2 Interfaces et voisins 16
1.3 Topologie physique et topologie logique d'Internet 18
1.4 Cartes du réseau *ARPANET*, 1969-1977[32] 20
1.5 Carte *best effort* du réseau *ARPANET*, 1977[32] 21
1.6 Carte *best effort* du réseau TCP/IP, 1985[32] 22
1.7 Croissance du nombre d'hôtes connectés au réseau, 1981-2012[37] 22

2.1 Fonctionnement de TRACEROUTE 29
2.2 Erreurs dans l'interprétation de TRACEROUTE 32
2.3 Erreurs dans l'interprétation de TRACEROUTE, même corrigée . . . 33
2.4 Observation d'une cible depuis un moniteur 34
2.5 Observation d'une cible depuis un moniteur, cas favorable 35
2.6 Observation d'une cible depuis un moniteur, cas défavorable . . . 35
2.7 Observation d'une cible depuis deux moniteurs 37
2.8 Voisin impossible à observer avec TRACEROUTE 38
2.9 Cœur et bord d'Internet 39
2.10 TRACEROUTE distribué vers une cible dans le bord 40
2.11 TRACEROUTE distribué vers une cible du cœur 41
2.12 Mesure de la liste des interfaces avec TRACEROUTE 43
2.13 Sélection des cibles du cœur par leur degré dans le cœur mesuré avec TRACEROUTE 47
2.14 Découpage de G_3 par rapport à une cible 49
2.15 Simulation de la topologie logique 52
2.16 Répartition des TRACEROUTE par moniteur 54
2.17 Répartition des TRACEROUTE par cible 55
2.18 Cibles atteintes par des moniteurs de longueur constante 56
2.19 Distribution du nombre d'interfaces tournées vers le cœur des voisins tournés vers le cœur de notre ensemble de cibles 57

3.1 En-têtes des paquets UDP PING 65
3.2 Choix de l'interface $m(t)$ par la cible $\bar{t}$ 66
3.3 Cas simplifié d'une cible à deux interfaces avec routage basique . 67
3.4 Cas d'une cible dans le cœur et d'une cible dans le bord 69
3.5 Filtrage des adresses de routeurs du cœur 72

3.6 Transformation de correction du biais 74
3.7 Observation de la distribution de degrés, simulation avec une loi de Poisson 75
3.8 Observation de la distribution de degrés, simulation avec une loi de puissance. 76
3.9 UDP Explore— nœud de branchement d'un moniteur 79
3.10 Moniteurs colocalisés 80
3.11 Nombre d'observations par cible et par moniteur 87
3.12 Distribution cumulative inverse du degré des routeurs du cœur . . 89
3.13 Évolution de la qualité de l'ensemble de moniteurs avec le nombre de classes de colocalisations 92
3.14 Convergence des fractions de cibles de degré k avec le nombre de classes de colocalisation 93
3.15 Distributions observées dans les simulations de validation 94

4.1 Impact du schéma d'inférence sur la taille des tables inférées . . . 109
4.2 Impact du nombre de moniteurs sur la taille des tables inférées (schéma le moins spécifique) 110
4.3 Impact du nombre de moniteurs sur la taille des tables inférées (schéma le plus spécifique) 111
4.4 Impact du nombre de moniteurs sur la taille des tables inférées (schéma *AS*) 112

Chapitre 1

Introduction

Internet est, à de nombreux égards, l'une des plus stupéfiantes constructions humaines. Ce réseau né à la fin des années 60 pour relier entre elles quelques unités logiques constitue aujourd'hui un gigantesque maillage, sans centre clairement identifié, qui permet des communications très rapides entre des centaines de millions de terminaux[12]. Plus qu'un réseau de télécommunication, Internet s'impose aujourd'hui comme *le* réseau unifié, qui sert de base à d'innombrables édifices applicatifs. Des couriers électroniques à la vidéophonie en passant par les jeux, les transactions bancaires et les encyclopédies en lignes, toutes ces fonctionalités avancées reposent, au dessus d'un empilement de couches d'abstraction, sur le réseau Internet. Pour fonctionner, elles délèguent la problématique de l'acheminement de l'information à Internet.

Cette confiance dans la robustesse et la fiabilité du réseau est justifiée par son histoire. Internet n'a jamais été "éteint", ni même sévèrement menacé dans son fonctionnement. Si des pannes ont temporairement perturbé son fonctionnement, cela n'est arrivé que ponctuellement et dans une mesure toute relative. Paradoxalement, pourtant, Internet n'a pas été imaginé et fabriqué pour être capable de desservir des milliards d'utilisateurs, qui chaque jour à travers le réseau envoient des centaines de milliards de courriers électroniques, visionnent des milliards de vidéos, effectuent des dizaines de millions d'appels en téléphonie Internet, des milliards de recherches, sur plus d'un milliard de sites web et pour un traffic qui totalise plusieurs milliards de GB de données[38]. En revanche, l'histoire a révélé que sa nature décentralisée permet une très grande souplesse et une très grande robustesse, ce qui a donc permis à de nombreux acteurs publics et industriels de participer à son développement avec relativement peu de gouvernance. De telle sorte que de très nombreux acteurs ont pu connecter leur réseau sur Internet, l'agrandissant du même coup, en y rajoutant leurs propre briques, parfois incompatibles avec les autres, mais sans remettre en cause l'intégrité d'un réseau intrinsèquement très robuste. Internet est, aujourd'hui, le résultat de son histoire riche et complexe, plutôt qu'un produit imaginé et conçu par une entité clairement définie et traçable.

Pour cette raison, s'il est indiscutable qu'Internet "fonctionne", il n'existe nulle part de carte complète du réseau. Certaines des propriétés les plus élémentaires de la structure du réseau Internet, qu'on appelle aussi la *topologie d'Internet*, telles que sa taille totale, la quantité d'information réelle qui y circule, ou même sa forme

générale, sont aujourd'hui inconnues ou au mieux incertaines et parfois contestées. Il est donc très difficile de raisonner ou de modéliser Internet dans sa globalité. Nous n'avons qu'une connaissance très empirique du socle fondamental d'une quantité toujours croissante de nos activités qui échappe assez largement à l'analyse. Une meilleure connaissance du réseau est essentielle à la fois pour pouvoir confirmer des acquis empiriques, en particulier concernant la fiabilité du réseau à l'égard des pannes, mais aussi pour rechercher des optimisations, notamment du routage, ou pour identifier des faiblesses, par exemple face à une attaque ciblée, physique ou logicielle. Plusieurs approches ont été explorées pour obtenir une connaissance plus approfondie de la topologie d'Internet, mais se sont heurtées à de nombreux obstacles à la fois théoriques et pratiques, et ont conduit à des résultats mitigés.

Cette thèse se positionne dans ce contexte, l'objectif central étant de développer une méthode de mesure fiable pour évaluer certaines des propriétés les plus importantes de la topologie d'Internet. Nous prendrons le soin de présenter précisément l'importance, la structure, et l'historique de nos objets d'intérêt, puisque la topologie d'Internet peut s'interpréter à différents niveaux d'abstraction que nous décrirons. Nous présenterons les approches historiques, leurs résultats et leurs limites, avant de présenter notre propre approche. Nous terminerons ce chapitre introductif par une présentation de l'organisation de cette thèse.

1.1 Préliminaires

Une partie de la difficulté des travaux autours de la topologie d'Internet repose sur la relation complexe entre les différents éléments du réseau, leur rôle exact, et la manière dont ils sont modélisés. Le modèle que l'on choisit de considérer dépend bien sûr de la nature du problème que l'on souhaite étudier et plusieurs modèles peuvent revendiquer le nom de "topologie d'Internet". Cette imprécision dans les définitions des objets étudiés est une source historique de confusion et d'erreurs d'interprétation [67]. L'une de nos contributions est de définir précisément les objets que nous étudions, et en particulier la relation entre ces objets et les outils de mesure que nous utilisons pour les observer, dans les approches historiques comme dans la nouvelle approche que nous proposons.

Internet permet de réaliser une interconnexion entre des entités de différentes natures utilisant des implémentations logicielles et des supports physiques très hétérogènes. Cette grande flexibilité s'appuie sur un découplage des différentes problématiques en plusieurs *couches* [6, 5, 8, 69] (**Figure 1.1**). La couche la plus basse à laquelle nous nous intéresserons, nommée *data-link layer* (liaison de données), ou encore L2 [*], correspond à une première abstraction du support physique de communication et un protocole d'encodage des messages qui dépend de ce support. Par exemple, *Ethernet* est un protocole de liaison de données permettant à deux entités reliées par un câble d'échanger des messages. Au dessus de cette couche se trouve la couche *Internet*, ou L3 [†], qui est caractéristique du réseau qui porte son nom. Elle se base sur la couche inférieure pour encoder des messages, sous la forme de *paquets*, formés d'un en-tête et d'un corps de message. L'en-tête contient des informations telles que les identifiants du destinataire et de l'expéditeur (sous le nom d'*adresses* IP) [53]. En décodant un paquet à partir des *frames* (trames) correspondantes, on peut donc savoir d'où vient et où doit aller ce paquet. Au dessus de cette couche se trouve encapsulée la couche *transport*, qui varie selon les besoins. Les plus courantes sur Internet sont UDP et TCP. Comme ceux de la couche inférieure, les paquets UDP [51] et TCP [54] sont pourvus d'en-tête qui contiennent des informations telles que le *port de destination* du paquet. Enfin, au dessus de la couche de transport se trouve la couche *application*, ou couche applicative, qui représente des messages spécifiques. HTTP [21], le protocole du Web, est l'un des protocoles qui se trouvent à ce niveau. Certains protocoles s'étalent à la fois sur la couche *transport* et la couche *application*, comme ICMP [54]. D'autres utilisent une couche intermédiaire supplémentaire, comme HTTPS [57], qui utilise comme intermédiaire le protocole de transport sécurisé TLS/SSL en dessous du protocole HTTP.

Les entités connectées à Internet sont toutes capables de communiquer en utilisant la couche *link*. Elles forment la *topologie physique d'Internet* ou *topologie*

∗. Pour *Layer 2*, en référence au modèle OSI.
†. Pour *Layer 3*, en référence au modèle OSI.

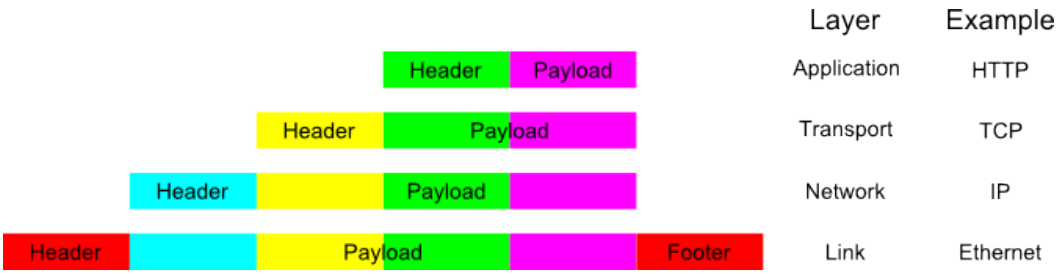


Figure 1.1 – *De haut en bas, les couches d'Internet de la plus abstraite à la plus concrète. Un paquet de la couche Application dispose d'un en-tête (*Header*) et d'une charge utile (*Payload*). Pour être transporté, ce paquet est encapsulé dans un paquet de la couche Transport, dont il forme lui-même la charge utile. Ce paquet de Transport est muni de son propre en-tête. Pour être envoyé sur le réseau, le paquet Transport est lui-même encapsulé dans un paquet Réseau, dont il forme la charge utile, et qui est également muni de son en-tête. Enfin, le paquet Réseau est découpé en un ou plusieurs morceaux qui transitent sous forme de trames de la couche Lien, chacune disposant d'un délimiteur de début et de fin.*

L2. On peut définir plusieurs types de ces entités [‡] :

- les hôtes terminaux, des ordinateurs formant les "utilisateurs finaux" d'Internet, tels que les ordinateurs personnels et les serveurs applicatifs, et qui disposent d'une adresse IP,
- les routeurs, des ordinateurs dont le rôle est d'acheminer du traffic qui ne leur est pas destiné directement à travers le réseau, suivant une logique coopérative, et qui disposent également d'une adresse IP,
- les *switches*, qui n'ont qu'une logique locale et qui aiguillent le traffic entre leurs interfaces selon une configuration pré-établie [§], et qui ne disposent pas d'adresses IP.

Ces entités sont connectées à Internet à travers des *interfaces physiques*. Ces interfaces physiques disposent le plus souvent d'une adresse appelée adresse MAC pour les protocoles de lien respectant cette convention, comme par exemple Ethernet ou Wi-Fi [11].

Parmi ces entités, seules les 2 premières sont également capables d'extraire et d'interpréter des paquets IP au niveau de la couche Réseau à partir des trames de la couche Lien. Ce sous-ensemble composé des hôtes et des routeurs [¶] de L2 forme la *topologie logique d'Internet*, ou *topologie L3* $\subseteq$ L2. Chaque entité (hôte ou routeur de L3) est capable de lire les paquets IP et pour certaines, de les interpréter comme des paquets de plus haut niveau, tels qu'ICMP, TCP ou UDP. Ces entités sont

‡. Certains de ces types portent des noms parfois utilisés dans le commerce. Les descriptions données ici font office de définition des termes tels qu'ils seront utilisés dans le reste de cette thèse.

§. On trouve parfois l'appellation de "switch L3 ", qui correspond selon notre définition à un cas particulier de routeur

¶. On parle ici des routeurs qui opèrent au niveau L3. Les routeurs opérant dans les couches supérieurs peuvent toujours être assimilés à un cas particulier de routeurs opérant au niveau L3. Sauf mention contraire, un routeur sera toujours supposé être un routeur L3.

identifiées au niveau L3 par leur(s) interface(s), qui sont munies d'une *adresse IP*. Les interfaces logiques correspondent le plus souvent à des interfaces physiques.

Nous définissons ainsi les deux topologies que nous allons étudier en détails.

Définition 1 (Nœud de la topologie physique)**.** *Soit V_2 l'ensemble des entités connectées à Internet au niveau* L2*. Chacune entité $\overline{v} \in V_2$ dispose d'un ensemble d'interfaces $\{v_0, v_1, \ldots, v_d\}$ au niveau* L2*. Une entité $\overline{v} \in V_2$ est appelée un* noeud de la topologie physique.

Définition 2 (Arète de la topologie physique)**.** *Soit $v \in \overline{v}$ et $u \in \overline{u}$. S'il existe un lien physique entre u et v permettant à $\overline{u}$ d'envoyer directement des paquets* IP *à $\overline{v}$, alors on dit que $\{\overline{u}, \overline{v}\} \in V_2 \times V_2$ est une* arete de la topologie physique*. On note $E_2 \subset V_2 \times V_2$ l'ensemble des* aretes de la topologie physique.

Définition 3 (Topologie physique et degré dans la topologie physique)**.** *On note $I_2 = (V_2, E_2)$ le graphe comportant l'ensemble des noeuds* L2 *et les liens qui leur permettent de communiquer au niveau* L2*. G_2 est appelé* topologie physique d'Internet*. En particulier, si $\overline{v} = \{v_0, \ldots, v_d\}$, alors $d = d_2(\overline{v})$ est le* degré *de $\overline{v}$ dans la topologie physique.*

Définition 4 (Nœud de la topologie logique)**.** *Soit V_3 l'ensemble des entités connectées à Internet au niveau* L3 *et disposant d'au moins une* adresse IP [‖]*. Chacune de ces entités $\overline{v} \in V_3$ dispose d'un ensemble d'interfaces identifiées par leurs adresses* IP *$\{v_0, \ldots, v_n\}$. Un entité $\overline{v} \in V_3$ est appelée un* noeud de la topologie logique.

Définition 5 (Arète de la topologie logique)**.** *Chaque interface $v \in \overline{v}$ permet à $\overline{v}$ d'envoyer des messages à un ou plusieurs autres noeuds* L3 *directement, sans passer par un autre nœud intermédiaire (ce qu'on appelle un* hop*). Pour chacun de ces noeuds $\overline{u}$, si v est capable d'envoyer des paquets* IP *à $\overline{u}$ à travers l'une de ses interfaces u, on appelle* arete de la topologie logique *la paire $\{\overline{v}, \overline{u}\}$. Notons qu'à un lien $\{\overline{v}, \overline{u}\}$ peuvent correspondre plusieurs couples d'interfaces (v, u) sous-jacents. On note $E_3 \subset V_3 \times V_3$ l'ensemble des* liens de la topologie logique.

Définition 6 (Topologie logique et degré dans la topologie logique)**.** *On note $I_3 = (V_3, E_3)$ le graphe comportant l'ensemble des noeuds* L3 *et les liens qui leur permettent de communiquer au niveau* L3 *en un* hop *(*modulo *leur redondance). G_3 est appelé* topologie logique d'Internet*. En particulier, si $\overline{v}$ est capable de communiquer avec un ensemble $\{\overline{u_1}, \ldots, \overline{u_d}\}$ de noeuds de la topologie logique en un* hop*, alors $d = d_3(\overline{v})$ est le* degré *de $\overline{v}$ dans la topologie logique. Le degré d'un noeud est inférieur ou égal à son nombre d'interfaces ($|\overline{v}| \leq d_3(\overline{v})$).*

Une propriété fondamentale de ces topologies que nous étudierons extensivement dans cette thèse est leur *distribution de degrés* :

‖. En particulier, $V_3 \subset V_2$

Définition 7 (Distribution de degrés des topologies d'Internet)**.** *On appelle* distribution de degrés de la topologie logique *(*resp.de la topologie physique*) la distribution* $d_2(V_2)$ *(*resp. $d_3(V_3)$*) définie par* $d_2(V_2)(n) = |\{\overline{v} \in V_2, |V(\overline{v})| = n\}|$ *(*resp. $d_3(V_3)(n) = |\{\overline{v} \in V_3, |V(\overline{v})| = n\}|$*). On note* $\hat{d}_2(V_2)$ *(resp.* $\hat{d}_3(V_3)$*) cette distribution normalisée (telle que sa somme soit égale à 1).*

Il est particulièrement important ici de distinguer la *liste des interfaces* d'un noeud $\overline{v} \in V_2 \supset V_3$, et la *liste de ses voisins.*(**Figure 1.2**)

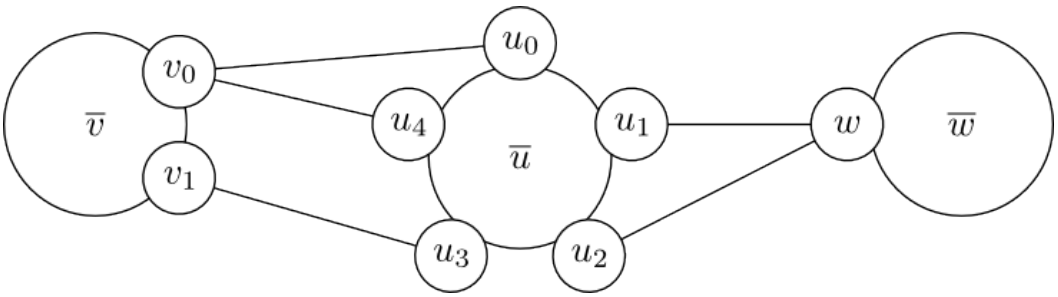


Figure 1.2 – *On distingue soigneusement les* interfaces de $\overline{u}$ *(*$u_0, \ldots, u_4$*), les* interfaces voisines de $\overline{u}$ *(*v_0, v_1, w*) et enfin les voisins de* $\overline{u}$ *(*$\overline{v}, \overline{w}$*). Ces trois notions sont naturellement liées mais leur confusion est à l'origine d'erreurs historiques.*

Cela a une importance considérable dans l'interprétation de la sortie de l'outil TRACEROUTE (**Chapitre 2**). La confusion entre *lien entre interfaces*, et *lien entre noeuds* est la cause d'erreurs historiques [67, 65]. Dans tout notre travail, nous désignerons toujours une *interface* par une lettre romaine (v) et un *noeud* comme une classe d'interfaces appartenant à ce noeud ($\overline{v}$). Cette notation se justifie par le fait que les noeuds sont presque systématiquement désignés implicitement par l'une de leurs interfaces plutôt que par un nom propre. L'ensemble des interfaces d'un noeud $\overline{v}$ est tout simplement identifiée à $\overline{v}$, et l'ensemble des voisins d'un noeud $\overline{v}$ sera noté $V(\overline{v})$. Le problème de déterminer si deux interfaces v_0 et v_1 appartiennent à un même noeud $\overline{v}$ est connu sous le nom d'*anti-aliasing*. Le problème inverse, déterminer la liste de toutes les interfaces d'un noeud donné $\overline{v}$, est un problème ouvert connu sous le nom d'*aliasing*. Ces deux problèmes seront traités en profondeur dans le **Chapitre 3**.

Dans ces deux définitions, nous ignorons les interfaces physiques ou logiques déconnectées d'Internet, par exemple les interfaces connectées à un réseau privé (par exemple `192.168.0.1`), les boucles locales (`127.0.0.1`), ou toutes les autres interfaces dont l'adresse correspond à une adresse invalide sur Internet (RFC 5735 [14]). On note $\mathbb{I}$ l'ensemble des adresses IP valides sur Internet.

Lorsqu'un noeud $\overline{v}$ appartient à la fois à V_2 et V_3, comme c'est le cas des routeurs et des hôtes, et lorsqu'aucune confusion n'est possible, on identifiera une interface physique dans L2 et l'adresse IP de l'interface correspondante au niveau logique.

Lorsqu'un noeud $\overline{v}$ dispose d'une unique interface v, et lorsqu'aucune confusion n'est possible, on identifiera v et $\overline{v}$. C'est fréquemment le cas des hôtes terminaux (*end hosts*) qui ne disposent que d'une seule interface physique et logique les reliant à Internet.

À l'aide du formalisme que nous avons introduit, nous pouvons définir plus formellement certains types de noeuds.

Définition 8 (Switch)**.** *On appelle* switch *tout noeud* $\overline{v} \in V_2$ *qui n'est pas également un noeud de* L3 *(*$\overline{v} \notin V_3$*).*

Définition 9 (Hôte)**.** *On appelle* hôte *(*host*) tout noeud* $\overline{v} \in V_2$ *qui est également un noeud* L3 *(*$\overline{v} \in V_3$*).*

Définition 10 (Hôte terminal (*end-host*))**.** *On appelle* hôte terminal *tout* hôte *qui possède une unique interface.*

Définition 11 (Routeur)**.** *On appelle* routeur *tout* hôte *qui n'est pas un* hôte terminal.

Comme nous l'avons déjà mentionné, même si ces appellations ne correspondent pas toujours à l'appellation commerciale, en termes de topologie physique ou logique, on peut toujours s'y ramener, et c'est pour cette raison que nous avons choisi de conserver un formalisme justifié par des considérations topologiques plutôt que commerciales. Par exemple, ce que l'on appelle commercialement un *hub* est équivalent à un *switch* selon la définition qui précède. Ce que l'on appelle commercialement un *switch* L3 est équivalent à un routeur selon la définition qui précède. De même, les "routeurs" domestiques que l'on peut trouver chez des particuliers sont le plus souvent configurés en mode passerelle (*gateway*) et selon notre définition ils sont équivalents à des hôtes terminaux, puisqu'ils ont une unique interface connectée à Internet. La topologie physique est donc formée exclusivement de *switches*, de *routeurs* et d'*hôtes terminaux* (*end-hosts*), et la topologie logique est formée exclusivement de *routeurs* et d'*hôtes terminaux.*

Nous pouvons alors définir et distinguer les *chemins* et les *routes* sur Internet.

Définition 12 (Chemin sur Internet)**.** *On appelle* chemin *sur* G_2 *(*resp. *sur* G_3*) une suite* $(\overline{v}_0, \ldots, \overline{v}_n)$ *de* noeuds *de* V_2 *(*resp. *de* V_3*) tels que* $(\overline{v}_i, \overline{v}_{i+1}) \in E_2$ *(*resp. $\in E_3$*).*

Définition 13 (Route sur Internet)**.** *On appelle* route *sur* G_2 *(*resp. *sur* G_3*) une suite* $((v'_0, v_1), \ldots, (v'_n, v_{n+1}))$ *telle que* v_i *et* v'_i *sont des interfaces d'un certain noeud* $\overline{v}_i$ *de* V_2 *(*resp. *de* V_3*) , et* v'_i *est une interface capable de communiquer avec l'interface* v_{i+1} *au niveau* L2 *(*resp. L3*). On appelle* chemin sous-jacent *de cette route le chemin* $(\overline{v}_0, \ldots, \overline{v}_{n+1})$*, et* trace *de cette route la suite des* interfaces entrantes $(v_1, \ldots, v_n)$.

La distinction entre *chemin* et *route* est importante, mêmes si les deux notions sont très proches. À toute route correspond un chemin sous-jacent, mais la notion de route est plus précise, car elle décrit la liste des interfaces entrantes et sortantes parcourures par des paquets IP. Un chemin, en revanche, ne décrit qu'une suite de nœuds empruntés et abstrait l'information concernant les interfaces parcourues. Un chemin est donc une information moins précise qu'une route.

Les deux topologies d'Internet les plus couramment étudiées sont très liées, car les noeuds de la topologie physique sont également des noeuds de la topologie logique, et les liens de la topologie logique sont induits par la topologie physique. L'inverse n'est pas vrai : deux noeuds connectés dans la topologie logique ne sont pas nécessairement connectés dans la topologie physique (**Figure 1.3**).

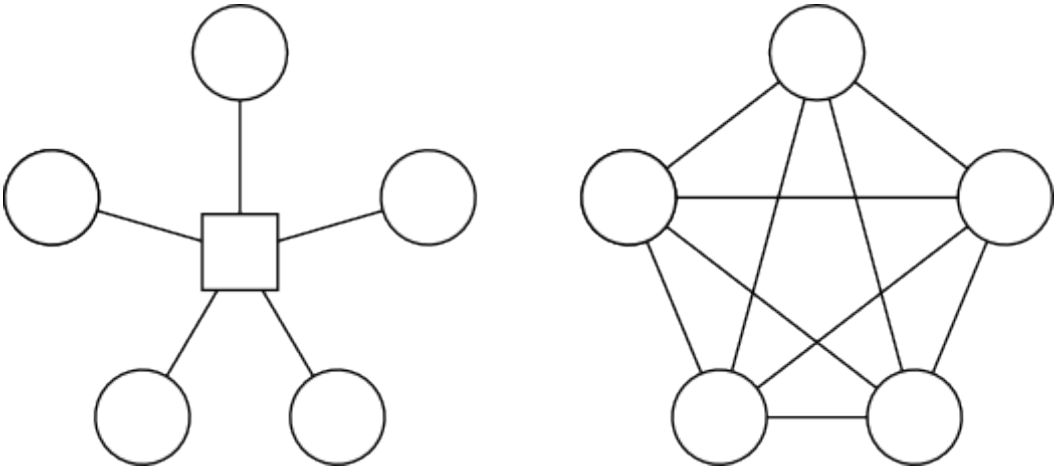


Figure 1.3 – *Un ensemble d'entités connectées à Internet dans la topologie physique (à gauche), et leur projection dans la topologie logique (à droite). Les hôtes (cercles) sont tous connectés à un switch (carré). Ce type de configuration physique se projette en une clique dans la topologie logique.*

La distinction entre ces deux topologies est très importante. Conceptuellement, elles représentent des objets différents et leurs caractéristiques ne sont pas directement équivalentes. On peut par exemple imaginer une topologie logique avec un degré moyen très élevé correspondant à une topologie physique avec un degré moyen très faible. Sur la **Figure 1.3**, on constate par exemple que pour un même réseau, la topologie logique est de degré moyen $N/(N+1)$, alors que la topologie logique est de degré moyen $(N+1)/2$. Du point de vue de la mesure, ce sont également des topologies très différentes puisque les primitives de mesure que nous décrirons, telles que TRACEROUTE ou PING, exploitent des caractéristiques spécifiques des couches L2 ou L3. Enfin, certains phénomènes tels que le *tunneling* [62] n'ont pas de sens au niveau physique puisqu'ils interviennent uniquement au niveau logique. Nous verrons que les méthodes de mesure que nous avons étudié exploitent le plus souvent les relations entre les deux topologies pour déduire des informations sur l'une ou l'autre.

1.2 Approches historiques

Dans cette section, nous positionnerons la problématique historique dans laquelle s'inscrit cette thèse. Nous verrons d'abord à quels questionnements historiques elle s'attache à contribuer (**Section 1.2.1**), l'approche historique de modélisation pour y répondre (**Section 1.2.2**), et les approches historiques de cartographie pour les compléter (**Section 1.2.3**).

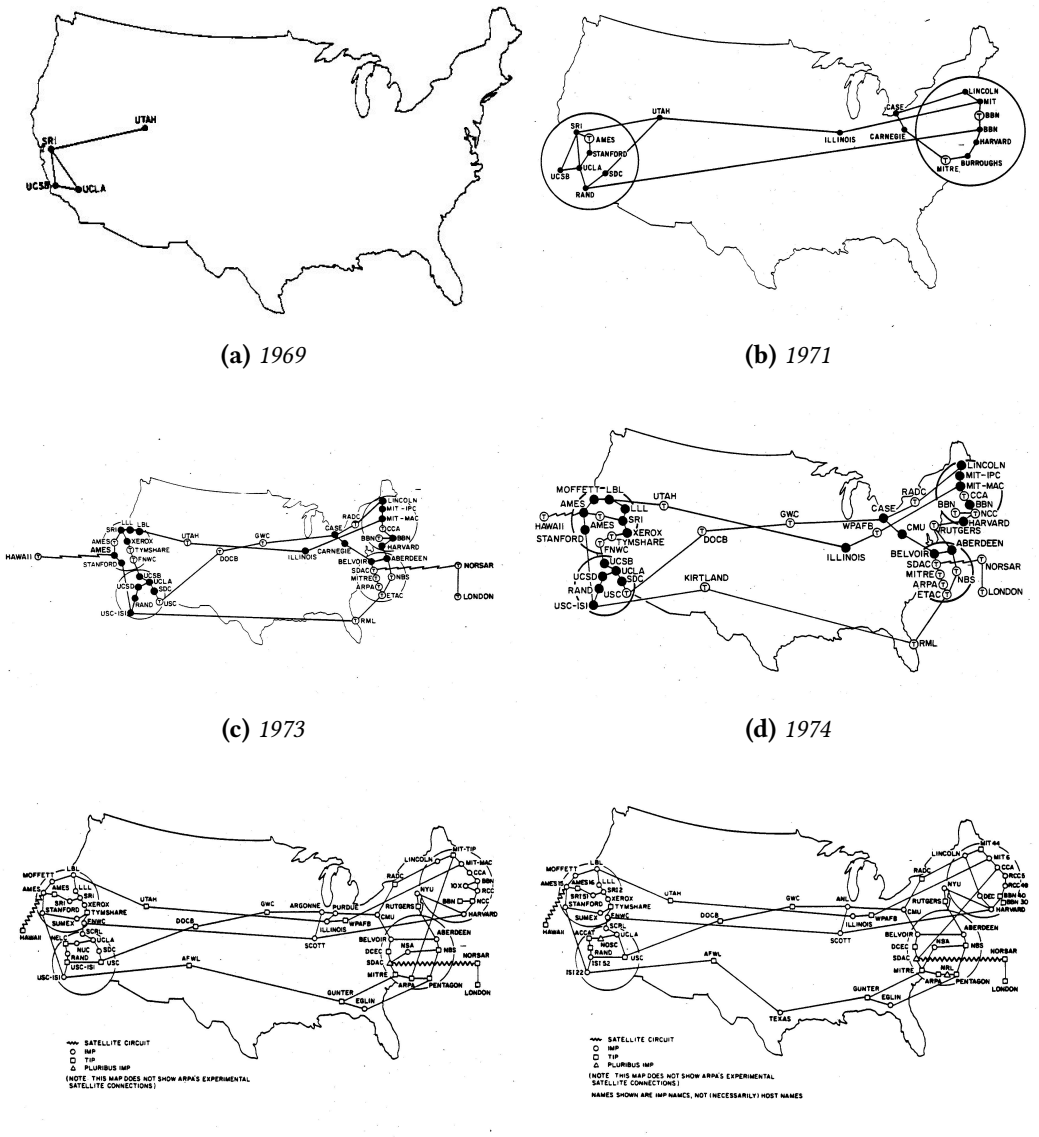
1.2.1 Génèse de la problématique

La lecture la plus communément admise de son histoire fait d'Internet l'héritier d'un certain nombre de réseaux de commutation de paquets, son ancêtre le plus direct d'un point de vue architectural étant *ARPANET*. Le premier lien de ce réseau est bien identifié, et il reliait entre elles les universités de Californie, Los Angeles (UCLA) et l'institut de recherche de Stanford. Ce lien a été établi le 29 octobre 1969. Le 5 décembre de la même année, 2 noeuds supplémentaires furent connectés : l'univeristé de l'Utah et l'université de Californie, Santa Barbara. En 1972, le réseau comportait 23 sites, et cinq ans plus tard, en 1977, plus d'une centaine d'ordinateurs étaient connectés. Jusqu'à cette année-là, la société *BBN Technologies*, travaillant sur le projet *ARPANET*, parvenait à conserver une carte précise et vraissemblablement exacte de tous les noeuds et liens du réseau (**Figure 1.4**, [32]). Mais dès lors, ses rapports deviennent plus précautionneux, et suggèrent déjà que la carte pourrait s'avérer inexacte, ne reflétant que des informations déclaratives (**Figure 1.5**).

Cette incertitude va naturellement augmenter avec l'explosion de la taille du réseau et l'adoption de TCP/IP au début des années 80, permettant de connecter entre eux plusieurs sous-réseaux. La structure de ces sous-réseaux n'est plus gérée de manière centralisée, et *BBN* abandonne l'idée de la connaître ; elle ne se contente déjà plus que d'éditer une carte basée sur des informations déclaratives à l'échelle inter-réseaux (**Figure 1.6**).

À la fin des années 80, l'inter-réseau TCP/IP prend le nom d'*Internet*. Au début des années 90, l'accès est ouvert au grand public, et le nombre d'ordinateurs connectés dépasse les 10 millions. Le seuil des 100 millions est franchi au début des années 2000 ; au début des années 2010, le nombre d'ordinateurs connectés dépasse le milliard. (**Figure 1.7**, [37])

Avec l'importence qu'a prise Internet aujourd'hui, l'utilisation d'un *modèle* d'Internet est extrêmement fréquente. Pour analyser le réseau, prédire son comportement, ou tout simplement le décrire, on a besoin d'une représentation aussi réaliste que possible vis à vis des caractéristiques auxquelles on s'intéresse. C'est particulièrement important lorsqu'on traite du réseau directement, comme par exemple pour traiter du routage ou de la robustesse des télécommunications, mais également lorsqu'on s'intéresse à des abstractions au dessus du réseau, comme par exemple le Web ou l'économie numérique.



(a) *1969* **(b)** *1971*

(c) *1973* **(d)** *1974*

(e) *1976* **(f)** *1977*

Figure 1.4 – *Cartes du réseau* ARPANET, *1969-1977[32]*

1.2.2 Modèles formels d'Internet

À cause de la très grande taille d'Internet et surtout, de sa gérance très décentralisée, par sa nature même d'inter-réseaux, l'espoir de maintenir une carte exacte du réseau au fur et à mesure de sa construction a depuis longtemps été abandonné. Une des premières approches historiques, dite approche *montante*** repose sur une idée simple : le réseau est certes complexe, mais il ne serait qu'une

**. De l'anglais *bottom-up*

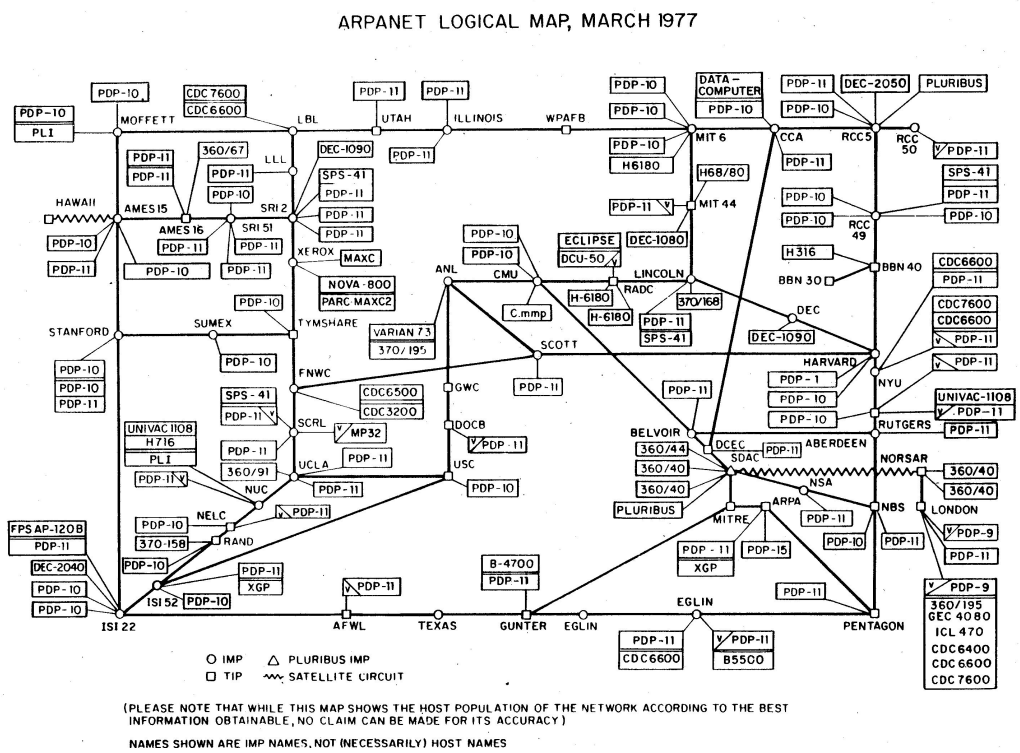


Figure 1.5 – *Carte* best effort *du réseau* ARPANET, *1977[32]*

combinaison logique de composantes simples résultant de l'ingénierie humaine. Il faudrait donc correctement représenter ces composantes simples, et combiner ces représentations, pour obtenir une représentation fidèle du réseau. Cette approche est celle qui correspond à la vision pédagogique d'Internet, et elle est issue de la communauté des réseaux. Elle considère le réseau avant tout comme une création d'ingénierie, et non comme un objet d'observation. Une conséquence directe de cette vision est qu'avec les bonnes prémisses, on devrait pouvoir représenter fidèlement le réseau. En pratique, cette approche conduit à la conception d'*algorithmes de génération de graphes* supposés représenter fidèlement Internet.

Le modèle historique le plus populaire est celui de Waxman *et al.* [66] et repose sur des graphes aléatoires contraints par la distance euclidienne entre les noeuds. Ce modèle est efficace pour représenter de petits réseaux de l'échelle d'*ARPANET*. Des modèles plus détaillés[17, 68] ont été établis par la suite, en combinant plusieurs modèles selon des variétés de sous-graphes locales correspondant à différentes échelles administratives du réseau.

Hélas, en plus de la complexité réelle des composantes et leur variété en dépit d'efforts de standardisation, la combinaison extrêmement complexe de ces composantes repose sur des paramètres topologiques, correspondant précisément à l'implémentation des autorités locales decentralisées. Ces paramètres sont le plus

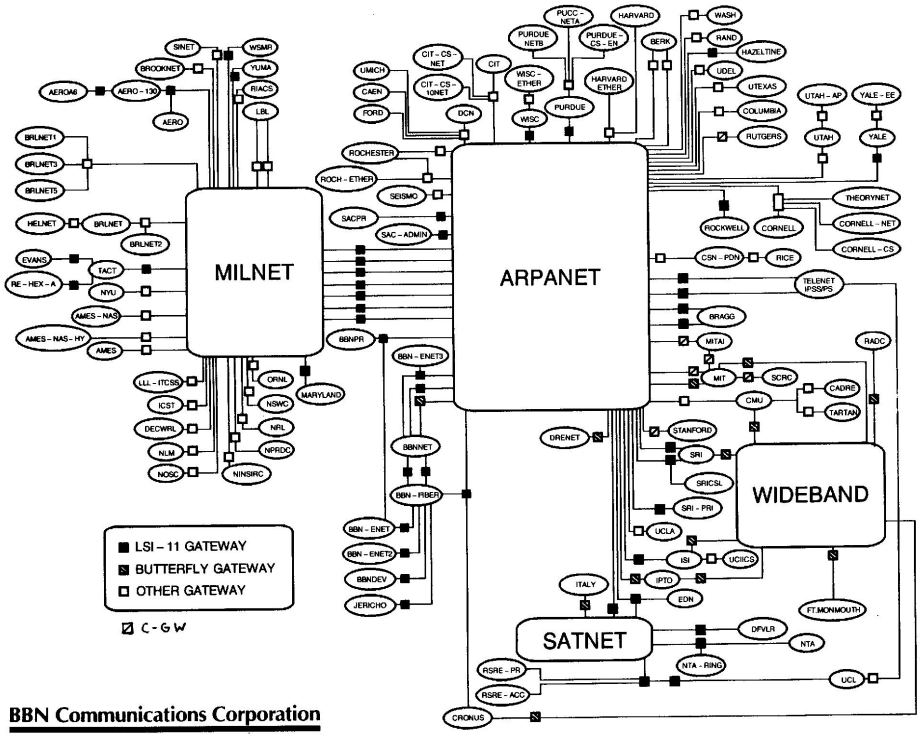


Figure 1.6 – *Carte* best effort *du réseau* TCP/IP, *1985[32]*

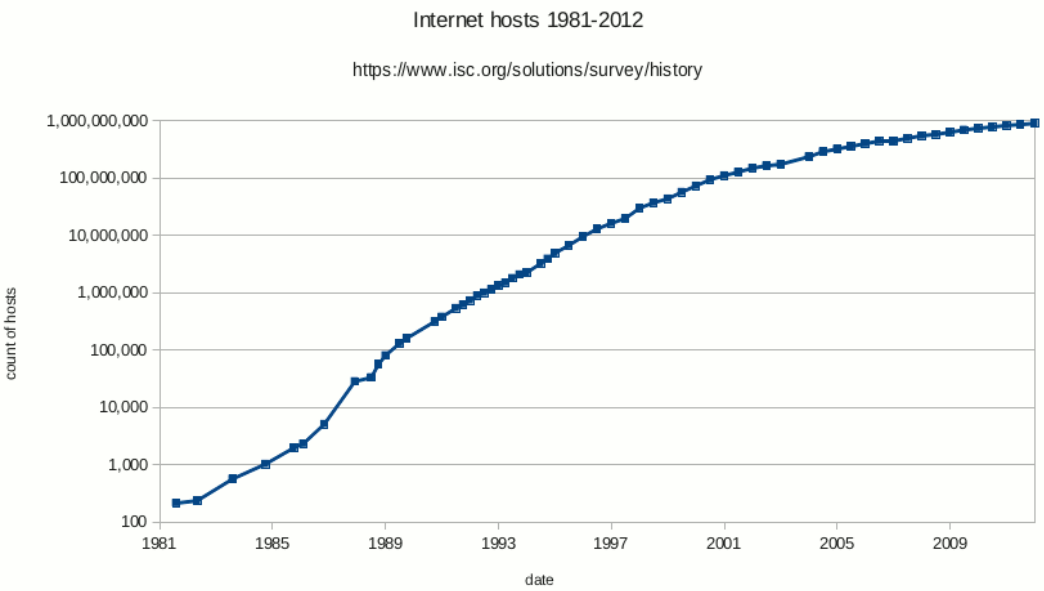


Figure 1.7 – *Croissance du nombre d'hôtes connectés au réseau, 1981-2012[37]*

souvent laissés à l'intuition du modélisateur, ce qui conditionne le réalisme de la représentation.[49]

1.2.3 Cartographie d'Internet

Face à la dépendance des approches *montantes* de modélisation à des paramètres topologiques inconnus, une idée nouvelle a émergé à la fin des années 90 et au début des années 2000. Cette approche, dite *descendante*, consiste à considérer le réseau comme un objet d'observation, au même titre que peuvent l'être des objets non synthétiques tels que la faune ou la flore. Il s'agit alors de *mesurer* le réseau afin d'en établir une *carte*, et d'utiliser cette carte pour déterminer les propriétés topologiques du réseau, qu'on injecterait ensuite dans les modèles formels.

L'étude de Govindan *et al.* [25] (1997) s'attache à la cartographie de la topologie inter-domaines, c'est à dire la topologie au niveau *AS*, et repose sur les traces BGP, qui sont utilisées pour extraire la distribution de degré et le diamètre de cette topologie. Les travaux de Pansiot et Grad[48] (1998) se focalisent pour la première fois sur le réseau au niveau des hôtes L3, plus précisément aux routeurs, en utilisant des sondes TRACEROUTE paramétrées avec l'option LSRR (*Loose Source and Record Route*). Ces derniers travaux ont initié un large ensemble de recherches pour plus d'une décennie sur la cartographie du réseau, qui adoptent tous une stratégie similaire, décomposant les travaux en une partie de cartographie du réseau reposant sur des observables, et une partie sur l'extraction de propriétés topologiques à partir de ces cartes. Le travail ayant eu la plus forte influence à cet égard est vraisemblablement celui de Faloutsos*et al.* [19] (1999), qui estime que le réseau s'organise en distribution de degré en loi de puissance.

Beaucoup de travaux ont suivi reposant sur des mesures TRACEROUTE, TRACERTREE [59, 42], MRINFO [44] et d'autres outils analogues, ainsi que des regards critiques sur la fiabilité de telles mesures. Ces critiques portent d'abord sur la fiabilité technique des outils, par exemple face à la dynamique ou aux mesures d'obfuscation pratiquées par les administrateurs réseaux (blocage de trafic ICMP, non-respect des directives de *source-routing* et *route-recording*...) [30, 65, 39, 9, 58, 18, 31]. Mais au delà des problèmes techniques, peut-être résolubles, de ces méthodes, une critique plus fondamentale apparaît : les cartes construites à partir de ces mesures seraient intrinsèquement biaisées [58, 67, 40, 1, 16, 28, 29, 41, 43], dans la direction de topologies en loi de puissance. En effet, il apparaît qu'observer le réseau en aggrégeant les traces de routes issues d'un nombre limité de nœuds en confère une vision formée d'une réunion d'arbres, ce qui biaise les degrés observés.

Pour tenter de remédier à ce biais, des efforts importants ont été réalisés pour augmenter la taille et la qualité des cartes, mais ces améliorations se sont avérées insuffisantes à éliminer le biais[67, 41, 4]. La nature même des réunions d'arbres depuis un nombre limité de racines biaise la topologie, et augmenter la taille de ces arbres ne résoud pas ce biais.

1.3 Notre approche

Après plus d'une décennie de recherches, le bilan de la cartographie d'Internet est mitigé. L'idée d'utiliser des outils de diagnostic réseau pour *observer* le réseau comme un objet naturel est une contribution indéniable qui a permis d'améliorer sa compréhension, mais les espoirs de déterminer les paramètres de modélisation permettant de représenter fidèlement le réseau au niveau macroscopique n'ont pas été satisfaits. Notre lecture est qu'à l'origine, l'objectif de la cartographie était bien de déterminer ces paramètres, mais qu'elle est progressivement devenue un objectif propre, qui l'a d'une certaine manière cantonnée dans une classe de problèmes qui a été mise en évidence comme intrinsèquement biaisée.

Pour cette raison, nous avons choisi de développer une approche différente. Plutôt que d'utiliser des outils de mesure pour établir des cartes sur lesquelles ont pourrait *lire* les propriétés topologiques qui nous intéressent pour les injecter dans des modèles, **nous utilisons ces outils de mesure pour mesurer *directement* ces propriétés topologiques**, plutôt que de faire appel à une carte intermédiaire. Nous espérons, de cette manière, éviter le biais intrinsèque de la *cartographie*, qui tend à donner une vision du réseau comme une réunion d'arbres correspondant à l'observation du réseau depuis un nombre limité de points de vue.

Notre démarche repose sur une compréhension claire de nos objectifs et une maîtrise de chacune des étapes de notre démarche. Notre objectif est d'obtenir une évaluation très fiable d'une propriété topologique du réseau et pour l'obtenir, nous devons maîtriser les biais potentiels induits par les mesures empiriques. Nous nous proposons de mettre en place la démarche théorique suivante, pour mesurer des propriétés topologiques du réseau.

- Soit une propriété du réseau définie par la distribution d'une certaine fonction $p : V \supseteq V' \to \mathbb{R}$ (propriété de noeuds) ou $p : E \supset E' \to \mathbb{R}$ (propriété d'aretes).
- Soit une primitive de mesure $\tilde{p} : V' \supset V'' \to \mathbb{R}$ ou $\tilde{p} : E' \supset E'' \to \mathbb{R}$, qui estime p sur un sous-ensemble des noeuds ou des aretes concernées par p, c'est à dire telle que $\tilde{p} = p_{|V''}$ ou $\tilde{p} = p_{|E''}$.
- Soit une procédure d'échantillonage de V'' ou de E'' telle que la distribution de p sur un échantillon converge vers la distribution de p sur V' ou que la distribution de p sur un échantillon converge vers la distribution de p sur E'.
- On tire un échantillon $\tilde{V'}$ ou $\tilde{E'}$ et on mesure $\tilde{p}$ sur cet échantillon.
- On estime alors que la distribution de p sur $\tilde{V'}$ ou $\tilde{E'}$ est égale à la distribution de $\tilde{p}$, et donc que la distribution de p sur V' ou E' est égale à la distribution de $\tilde{p}$ sur l'échantillon.

Cette méthode nous permet d'identifier clairement les potentiels biais, et de distinguer les biais d'implémentation (calcul de $\tilde{p}$), les biais topologiques (procédure d'échantillonage), et les biais statistiques (convergence de la distribution).

1.4 Organisation

Cette thèse présente les travaux que nous avons réalisé pour explorer la pertinence et la faisabilité de la mesure orientée propriété de la topologie d'Internet, dans le cas particulier où la propriété d'intérêt est la distribution de degrés, ou plus exactement les distributions de degrés des topologies L2 et L3. Elle suit une organisation générale chronologique, qui correspond à la progression de nos travaux : nous avons d'abord exploré une approche inspirée des mesures basées sur TRACEROUTE pour mesurer la distribution de degré de la topologie logique (**Chapitre 2**), puis nous avons conçu une primitive de mesure très fiable et un meilleur protocole d'échantillonage pour mesurer la distribution de degrés de la topologie physique (**Chapitre 3**), que nous avons exploré en profondeur en mesurant les tables de routages (**Chapitre 4**). Chacune de ces sections suit en revanche un découpage réalisé *a posteriori* et qui nous semble amener de la manière la plus pertinente les conclusions que nous en avons tiré et les perspectives qu'elles ont ouvertes (**Chapitre 5**).

Chapitre 2

Mesure de la topologie logique

Historiquement, la topologie logique est celle qui a la première fait l'objet de mesures massives. Les travaux de Faloutsos *et al.* [19] et Pansiot *et al.* [48] sur la topologie logique ont initié toute une série d'opérations de mesure destinées à mesurer cette topologie en utilisant des outils de diagnostic réseau. Le plus utilisé à cet effet est sans aucun doute traceroute. En principe, traceroute permet d'obtenir le chemin dans la topologie logique parcouru par un paquet envoyé depuis la machine qui exécute traceroute vers une cible donnée de la topologie logique. L'utilisation massive de traceroute a porté l'espoir d'établir des cartes de la topologie logique en aggrégant des collectes de chemins. Mais de nombreux travaux ont montré qu'en plus de problèmes techniques liés à l'outil traceroute [65, 18, 58], la topologie logique cartographiée est intrinsèquement biaisée [40, 1, 67, 43, 16, 29, 41, 58].

Parce que la topologie logique et particulièrement l'utilisation de l'outil traceroute pour la mesurer sont au cœur de l'état de l'art, c'est sur cette base que nous avons décidé de mener nos travaux préliminaires pour une première mesure orientée propriété de la topologie d'Internet.

Notre première contribution a été d'analyser en détails le fonctionnement de traceroute et de mettre en évidence les confusions à l'origine d'erreurs d'interprétation historiques de ses résultats (**Section 2.1**). En surmontant ces confusions, nous avons mis au point une interprétation plus restreinte de traceroute, qui constitue notre primitive de mesure de bas niveau (**Section 2.2**) qui permet d'obtenir une interface d'un voisin d'une cible de la topologie L3. Une utilisation distribuée de cette primitive de mesure de bas niveau nous a permis de concevoir une primitive de mesure de haut niveau (**Section 2.3**) qui permet d'obtenir, en principe, la liste des interfaces extérieures de tous les voisins tournés vers le cœur d'un nœud quelconque d'Internet. Nous avons enfin conçu une procédure d'échantillonnage (**Section 2.5**) et de correction d'erreurs (**Section 2.6**) qui nous donne un premier moyen d'évaluer la distribution de degré des routeurs du cœur dans la topologie logique à l'aide de l'outil traceroute. Le principe de cette méthode a été validé par des simulations (**Section 2.7**). Enfin, pour attester de la faisabilité pratique de notre méthode, nous l'avons expérimentée au cours d'une mesure réelle depuis un ensemble de moniteurs du réseau Planetlab [13] (**Section 2.8**). Cette expérimentation nous a permis d'ajuster la méthode à des contraintes expérimentales pour en déduire un protocole de mesure (**Section 2.9**). Nous avons enfin

pu positionner ces travaux et leur contribution propre mais également déterminer leurs limites (**Section 2.10**). Nous avons pu en tirer des conclusions importantes pour la suite de nos travaux (**Section 2.11**).

2.1 Interprétation rigoureuse de TRACEROUTE

L'outil TRACEROUTE est un utilitaire de diagnostic réseau utilisé habituellement pour tenter d'identifier la liste des nœuds de la topologie logique qui sont parcourus par les paquets IP pour transiter d'un nœud *source* vers un nœud *destination*. Il se base sur les RFC 1192 [6], RFC 792 [52], et RFC 1812 [3].

Les paquets IP disposent d'un champ, *Time-to-Live* ou TTL, qui indique le nombre de *hops* autorisés qu'il reste au paquet. Ce compteur est initialisé par l'expéditeur du paquet, normalement à des valeurs suffisamment élevée pour atteindre sa destination dans des conditions régulières, par exemple 64 ou 128. Chaque fois qu'un paquet traverse un nœud, ce nœud doit décrémenter la valeur du TTL de 1 avant de faire suivre le paquet. Si ce TTL atteint 0, le nœud qu'il traverse doit supprimer le paquet, et renvoyer à l'expéditeur du paquet un message ICMP TIME EXCEEDED contenant un identifiant unique du paquet incriminé, pour signaler un probable problème de routage. Ce mécanisme permet d'éviter que des paquets piégés dans des boucles de routage n'emcombrent le réseau indéfiniment.

TRACEROUTE exploite ce principe en forgeant des paquets avec des TTL faibles croissants, dans le but intentionnel de faire générer des messages ICMP TIME EXCEEDED à des routeurs à une *hop*-distance de leur point de départ donnée, en chemin vers leur destination.

La **Figure 2.1** illustre le fonctionnement de TRACEROUTE. Supposons qu'un moniteur $\overline{m} \in V_3$, (c'est à dire un hôte de la topologie logique, *cf.* **Définition 6**) fabrique un paquet IP d'identifiant p en indiquant dans l'en-tête sa propre adresse m comme adresse d'expédition, une adresse t (pour *target*) d'une cible $\overline{t} \in V_3$ comme adresse de destination, et une certaine valeur n comme TTL. Ce paquet contient un message ICMP, TCP ou UDP qui demande à la cible de répondre qu'elle a bien réceptionné le paquet.

Le moniteur $\overline{m}$ envoie ce paquet p d'identifiant i et écoute ensuite les paquets ICMP TIME EXCEEDED et les messages de réponse de la cible. Si $\overline{m}$ reçoit un paquet de réponse, c'est que la cible a été atteinte et le paquet a parcouru moins de n *hops*. Si $\overline{m}$ ne reçoit aucun paquet après un certain temps d'expiration (*timeout*), alors le paquet d'origine est considéré comme perdu. Si $\overline{m}$ reçoit un paquet ICMP TIME EXCEEDED contenant l'identifiant i d'origine, alors il lit l'adresse de l'expéditeur de ce paquet ICMP, qui est celle d'une interface r_n (adresse IP) d'un certain routeur $\overline{r_n} \in V_3$. On peut alors supposer que le paquet p a effectué n *hops* en chemin vers t et a expiré en traversant $\overline{r_n}$. Dans la topologie logique, cela signifie que le routeur $\overline{r_n}$ se trouve à une distance n sur un chemin entre $\overline{m}$ et $\overline{t}$.

TRACEROUTE (**Algorithme 1**) répète cette opération en partant de $n = 1$ jusqu'à ce que la cible soit atteinte, c'est à dire si pour un n donné, il obtient une réponse appropriée de la part de la cible. TRACEROUTE affiche la liste de tous les nœuds ayant renvoyés des paquets ICMP TIME EXCEEDED, dans l'ordre des TTL n croissants.

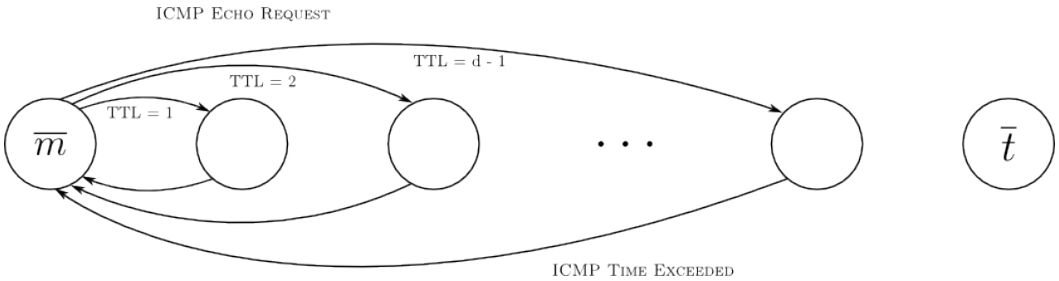


FIGURE 2.1 – *Un moniteur $\overline{m}$ possédant une interface m envoie des paquets* IP *avec des* TTL *croissants depuis 1 vers une cible $\overline{t}$ désignée par une interface t. Les nœuds qui sont traversés pour atteindre t génèrent des paquets* ICMP TIME EXCEEDED *qui permettent de les identifier.*

La nature exacte du paquet d'origine p dépend du protocole de transport (L4) utilisé. S'il s'agit du protocole ICMP, alors le paquet contient un message ICMP ECHO REQUEST. Si la cible répond à ce type de messages, ce qui demande une configuration explicite et spécifique, alors elle répond avec un paquet ICMP ECHO REPLY. Comme beaucoup de routeurs et d'hôtes sont configurés pour *ne pas* répondre aux paquets ICMP ECHO REQUEST, on peut alternativement utiliser des sondes TCP ou UDP. Les sondes TCP utilisent des paquets TCP SYN qui réclament l'ouverture d'une connexion TCP, à laquelle la cible répond avec un paquet TCP ACK. Les sondes UDP sont adressées à un port aléatoire *a priori* inutilisé, pour que la cible réponde avec un paquet ICMP DESTINATION UNREACHABLE indiquant que ce port ne peut être atteint.

Le bon fonctionnement de TRACEROUTE suppose que les sondes peuvent à la fois parvenir à la cible et aux nœuds intermédiaires, et que les paquets ICMP TIME EXCEEDED et les paquets de réponse en provenance de la cible (dépendant du type de transport choisi) soient correctement générés et soient capables de parvenir au moniteur à leur tour. Les messages ICMP ECHO REQUEST sont très souvent ignorés, puisqu'ils n'ont pas d'autre utilité propre que le diagnostic et sont parfois considérés comme des atteintes à la sécurité. En revanche, les sondes TCP et UDP sont indiscernables de paquets correspondant à un fonctionnement "normal" (à l'exception de leur TTL très faible). Certains *firewalls*, cependant, considèrent qu'un paquet avec un TTL trop faible (inférieur à 10, par exemple) est suspect et le rejettent. Enfin, certains routeurs et *firewalls* bloquent complètement le traffic ICMP, ce qui empêche les réponses ICMP TIME EXCEEDED de parvenir au moniteur. Pour des raisons de sécurité analogue, certaines cibles ne répondent pas de paquets ICMP DESTINATION UNREACHABLE aux sondes UDP, ou refusent d'accepter des

Algorithme 1 TRACEROUTE

```
function GénérerDemandeRéponse(transport, port = NULL)
    if transport == ICMP then
        RENVOYER ICMP (ECHO REQUEST)
    else if transport == UDP then
        RENVOYER UDP (port = port, corps = RandomBytes())
    else if transport == TCP then
        RENVOYER TCP (port = port, corps = SYN)
procedure EnvoyerSonde(transport, port = NULL, identifiant, destinataire,
TTL)
    e ← EntêteIP(expediteur = HostAddress, destinataire = destinataire,
identifiant = identifiant, TTL = TTL)
    c ← GénérerDemandeRéponse(transport, port)
    paquet ← PaquetIP(EnTête = e, Corps = c)
    EmettrePaquetIP(paquet)
procedure TRACEROUTE(destinataire, transport, port = NULL, timeout)
    n ← 1
    Print("TRACEROUTE depuis " HostAddress " vers " destinataire " :")
    loop
        identifiant ← RandomBytes()
        envoyer-sonde(transport, identifiant, destinataire, n)
        réponse ← AttendreRéponse(type = TimeExceeded, identifiant =
identifiant, timeout = timeout)
        if Type(réponse) == CibleAtteinte(transport) then
            Print("Cible " destinataire " atteinte après " n " hops")
            Exit()
        else if Type(réponse) == Timeout then
            Print("hop " n " ⇄ " Expéditeur(réponse))
        else
            Print("hop " n " ⇄ *)
        n ← n + 1
```

connexions TCP (donc de répondre aux sondes TCP) qu'elles n'ont pas au préalable autorisées.

Lorsque TRACEROUTE fonctionne correctement, c'est à dire que le traffic n'est pas bloqué, il fournit donc une liste $(r_1, \ldots, r_n, t)$ d'adresses IP où r_k correspond à la réponse pour le k-ième *hop*.

L'*interprétation classique de* TRACEROUTE, à la base de la plupart des travaux de cartographie au niveau IP, est souvent équivalente, de manière implicite, à la formulation suivante :

Hypothèse 1 (Interprétation classique de TRACEROUTE)**.** *La liste* $(r_1, \ldots, r_n, t)$ *produite par* TRACEROUTE *depuis* m *vers* t *correspond à la* trace *d'une certaine* route *parcourue par les sondes (*cf. **Définition 13***), donc par les paquets* IP*, pour atteindre* t *depuis* m *(d'où le nom de l'utilitaire,* TRACEROUTE*), et chaque paire* $\{r_k, r_{k+1}\}$ *est donc un lien entre deux interfaces au niveau* L3*.*

À la lumière du formalisme que nous avons introduit, cette interprétation apparaît très inexacte, sans même parler des effets liés à la dynamique.

Elle repose notamment sur une confusion entre les *nœuds* L3 traversés par les paquets (chaque $\overline{r_k}$), et les *interfaces* utilisées par ces nœuds pour répondre (chaque r_k). Cette confusion pourrait être légitime dans deux cas : (1) dans le cas où chaque $\overline{r_k}$ ne dispose que d'une seule interface connectée à Internet, ou (2) dans le cas où chaque $\overline{r_k}$ utilise toujours la même interface ($\overline{r_k}$) pour répondre à TRACEROUTE, *dans tous les cas.*

Le cas (1) est au mieux très rare. Supposons qu'un routeur $\overline{r_k}$ dispose d'une unique adresse IP r_k. Alors soit $\overline{r_k}$ est de degré 1 dans la topologie L3, soit toutes ses connexions au niveau L3 se font par l'intermédiaire d'une entité de niveau L2 qui n'est pas une entité de niveau L3. Si $\overline{r_k}$ n'est pas de degré 1, alors l'entité de niveau L2 intermédiaire qui n'est pas de niveau L3 est assimilable topologiquement à un *switch*. Or, de très nombreux routeurs ne sont ni de degré 1 au niveau logique, ni connectés exclusivement à travers l'intermédiaire d'un *switch*. L'approximation résultant de (1) est donc fausse dans tous ces cas là.

Examinons à présent le cas (2). Une adresse r_k apparaît dans le résultat de TRACEROUTE depuis m vers t si un certain routeur $\overline{r_k}$ a envoyé un message ICMP TIME EXCEEDED en direction de m en utilisant pour cela l'interface r_k. r_k est donc l'interface *choisie par* $\overline{r_k}$ pour router un message ICMP à destination de m. Il y a ici également deux sous-cas possibles : soit $\overline{r_k}$ est configuré pour utiliser *toujours* sa même interface r_k pour générer des messages ICMP TIME EXCEEDED, soit l'interface r_k choisie dépend d'une manière ou d'une autre de m et éventuellement d'autres paramètres. Certains routeurs sont en effet configurés pour adopter le premier comportement. Mais cela ne peut être le cas général, et la rareté de cette configuration a été démontrée expérimentalement dans plusieurs travaux liés à l'*anti-aliasing* [39, 31]. Le seul cas possible restant dans le cas général est donc le

suivant : l'interface choisie par $\overline{r_k}$ pour envoyer le paquet ICMP TIME EXCEEDED à m dépend de m, et éventuellement d'autres paramètres. Informellement, cela signifie que pour répondre à une sonde TRACEROUTE, $\overline{r_k}$ utilise une interface "tournée vers m", ou du moins une interface spécifiquement choisie pour envoyer un message vers m.

Les conséquences de cette interprétation abusive vont bien au delà d'une simple considération formelle : elles conduisent à supposer l'existence de liens en réalité inexistants, même dans des cas très simples. Supposons (comme illustré en **Figure 2.2**) que deux hôtes terminaux (*end hosts*) m et t soient connectés à travers un routeur $\overline{r}$ possédant deux interfaces $\overline{r_m}$ et $\overline{r_t}$, la première connectée à $\overline{m}$ et l'autre à $\overline{t}$. La seule route depuis $\overline{r}$ vers m passe donc par l'interface r_m. En exécutant TRACEROUTE depuis m vers t, on obtient en principe la sortie suivante :

```
TRACEROUTE depuis m vers t:
hop 1 ⇄ r_m
Cible t atteinte après 2 hops.
```

Alors d'après **Hypothèse 1**, il doit exister un lien (r_m, t), ce qui n'est pas vrai.

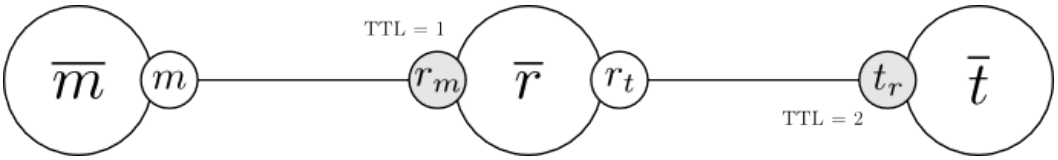


FIGURE 2.2 – *m envoie un* TRACEROUTE *vers t. La première sonde est réceptionnée par $\overline{r}$ qui répond avec son interface r_m. L'interprétation classique suppose l'existence d'un lien inexistant entre r_m et t.*

Ce premier défaut dans l'interprétation classique de TRACEROUTE peut toutefois être corrigé, si on l'exprime correctement en termes de nœuds L3 plutôt qu'en termes d'interfaces. Cette hypothèse corrigée est la suivante :

Hypothèse 2 (Interprétation classique de TRACEROUTE (corrigée))**.** *Soit la liste $(r_1, \ldots, r_n, t)$ produite par* TRACEROUTE *depuis m vers t. Alors pour chaque k, $\{\overline{r_k}, \overline{r_{k+1}}\} \in E_3$, c'est à dire que $\overline{r_k}$ et $\overline{r_{k+1}}$ sont voisins au niveau logique puisqu'une sonde passe de $\overline{r_k}$ à $\overline{r_{k+1}}$ en un* hop.

Mais plusieurs travaux antérieurs [65, 67, 61, 63, 65, 2] ont montré que même cette interprétation est fausse dans de nombreux cas, à cause de la dynamique des routes et notamment celle induite par l'équilibrage de charge. Supposons (comme illustré en **Figure 2.3**) par exemple que $\overline{m}$ soit connecté à un routeur $\overline{r}_1$, lui-même connecté à deux routeurs $\overline{r}_2$ et $\overline{r}_3$. $\overline{r}_2$ est connecté à un routeur $\overline{r}_4$ connecté à $\overline{t}$

mais pas à $\overline{r}_3$, et $\overline{r}_3$ est connecté à un routeur $\overline{r}_5$ connecté à $\overline{t}$ mais pas à $\overline{r}_4$. Si $\overline{r}_2$ pratique l'équilibrage de charge (*load-balancing*), par exemple en envoyant un paquet sur deux à destination de t vers r_2 et r_3 alternativement, alors TRACEROUTE peut fournir la sortie suivante :

```
TRACEROUTE depuis m vers t:
hop 1 ⇄ r1
hop 2 ⇄ r2
hop 1 ⇄ r5
Cible t atteinte après 4 hops.
```

Alors d'après **Hypothèse 2**, $\overline{r}_2$ est un voisin de $\overline{r}_5$ au niveau logique, ce qui n'est pas vrai. (**Figure 2.3**)

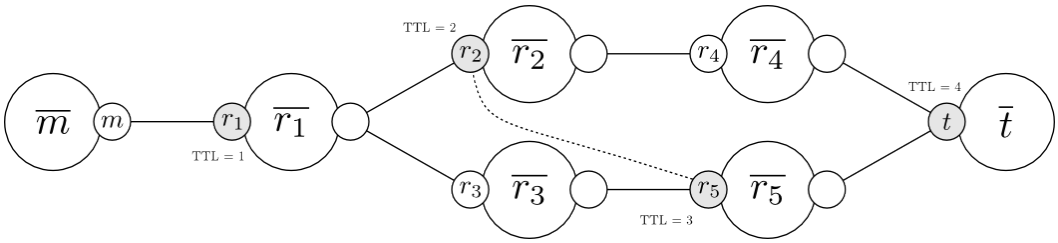


FIGURE 2.3 – *$\overline{m}$ envoie un* TRACEROUTE *vers t en utilisant son interface m, en passant par $\overline{r}_1$ qui effectue de l'équilibrage de charge.* TRACEROUTE *indique successivement r_2 et r_5 dans sa sortie, pourtant $\overline{r}_2$ et $\overline{r}_5$ ne sont pas voisins. Les liens réels sont indiqués en traits pleins, le lien inexistant suggéré par* TRACEROUTE *est en pointillés.*

Si pour ces raisons il semble peu vraissemblable de valider une hypothèse de haut niveau sur les résultats de TRACEROUTE, on peut toutefois se référer à la description rigoureuse que nous avons donné de son fonctionnement pour formuler une hypothèse à un plus bas niveau qui découle directement de cette description :

Hypothèse 3 (Interprétation bas niveau de TRACEROUTE)**.** *Soit $(r_1, \ldots, r_n, t)$ le résultat de* TRACEROUTE *depuis m vers t. Alors pour chaque k, r_k est une interface d'un certain nœud $\overline{r_k}$ de* L3*, qui se trouve sur* une *route depuis m vers t à une* hop*-distance k de m. Chaque r_k dépend de m et éventuellement d'autres paramètres.*

Nous allons maintenant voir comment exploiter cette hypothèse pour mesurer une information fiable et pertinente sur une cible t à partir d'un moniteur m.

2.2 Primitive de mesure de bas niveau basée sur TRACEROUTE

Nous avons exposé (**Section 2.1**) une interprétation réaliste (**Hypothèse 3**) du résultat de TRACEROUTE depuis un moniteur $\overline{m}$ à travers une interface m vers une cible $\overline{t}$ désignée par une adresse t. Nous avons en outre montré qu'il était difficile d'exploiter ce résultat pour opérer des déductions sur d'éventuels liens *entre les nœuds intermédiaires* donnés par TRACEROUTE, en particulier car les sondes successives peuvent emprunter des routes différentes. Pour cette raison, nous avons décidé de nous intéresser uniquement aux informations données par le *dernier résultat donné par* TRACEROUTE *avant d'atteindre la cible.* Dans ce chapitre, nous utiliserons ainsi la notation suivante :

Définition 14 (Observation d'une cible depuis un moniteur)**.** *Soit* $(r_1, \ldots, r_n, t)$ *le résultat de* TRACEROUTE *depuis* m *vers* t*. On note* $m(t) = r_n \in \mathbb{I}$ [*] *et on l'appelle* observation de t depuis m.

Alors d'après **Hypothèse 3**, $m(t)$ est une interface d'un certain routeur $\overline{m(t)}$ qui se trouve à une *hop*-distance n de m en chemin vers t, tandis que t se trouve à une *hop*-distance $n+1$ de m. Plus précisément, il s'agit de l'interface choisie par $\overline{m(t)}$ pour envoyer un paquet ICMP TIME EXCEEDED vers m.

Supposons temporairement que *toutes les sondes envoyées par* TRACEROUTE *depuis* m *vers* t empruntent la même route. Alors $\overline{m(t)}$ est nécessairement un voisin de $\overline{t}$ dans la topologie logique, et plus précisément la dernière sonde TRACEROUTE (celle qui atteint t) passe en un *hop* de $\overline{m(t)}$ à $\overline{t}$ (**Figure 2.4**). Ce cas simple suggère la manière dont nous allons tenter d'interpréter $\overline{m(t)}$ comme un voisin au niveau logique de $\overline{t}$.

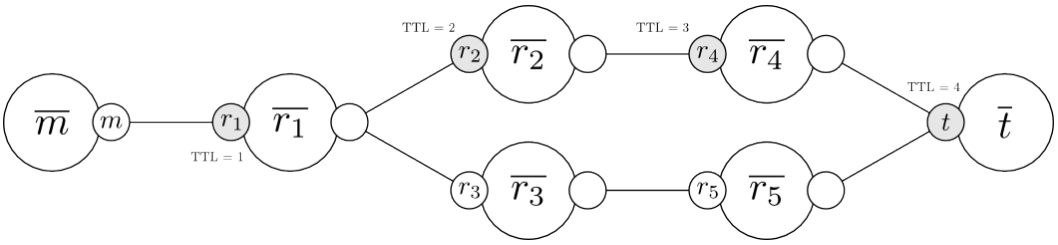


FIGURE 2.4 – *$\overline{m}$ lance un* TRACEROUTE *vers t en utilisant son interface m, et toutes les sondes empruntent la même route. Alors $\overline{r_n} = \overline{m(t)}$ est nécessairement un voisin de $\overline{t}$.*

Mais comme nous l'avons déjà évoqué (**Section 2.1**), toutes les sondes de TRACEROUTE n'empruntent pas nécessairement la même route. En revanche, il suffit que les deux dernières sondes de TRACEROUTE empruntent des routes *de*

[*]. Par conséquent, $\overline{m(t)} = \overline{r_n} \in V_3$

même longueur : même dans le cas d'un changement de route entre ces deux sondes, l'information de la longueur totale de la route empruntée suffit à conclure que $\overline{m(t)}$ est un voisin au niveau logique de $\overline{t}$ (**Figure 2.5**). En particulier :

Proposition 1 (Interprétation de l'observation d'un moniteur vers une cible)**.** *Si toutes les routes depuis m vers t ont la même longueur, alors $\overline{m(t)}$ est un voisin au niveau logique de $\overline{t}$.*

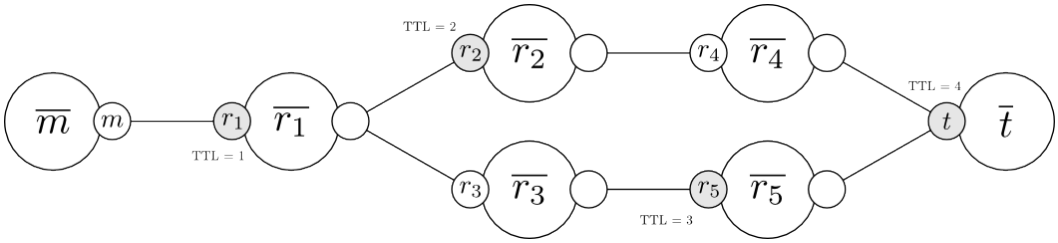


Figure 2.5 – *$\overline{m}$ lance un* TRACEROUTE *vers t en utilisant son interface m. Deux routes différentes sont parcourues alternativement par les sondes, mais ces deux routes font la même longueur. On peut conclure que $\overline{m(t)} = \overline{r_5}$ est bien un voisin de $\overline{t}$.*

Inversement, en revanche, si les deux dernières sondes de TRACEROUTE (celle qui provoque l'observation de $m(t)$, et celle qui atteint $\overline{t}$) empruntent des routes de longueur différentes, $\overline{m(t)}$ peut très bien ne *pas* être voisin de $\overline{t}$. Le cas le plus simple est celui de l'équilibrage de charge entre deux sous-routes de longueur distincte. Soit par exemple (comme illustré en **Figure 2.6**) un moniteur $\overline{m}$ connecté à un certain routeur $\overline{r_1}$. Ce routeur est connecté à deux autres routeurs $\overline{r_2}$ et $\overline{r_3}$. $\overline{r_3}$ est connecté directement à $\overline{t}$, tandis que $\overline{r_2}$ est connecté à un autre routeur $\overline{r_4}$, qui lui est connecté à $\overline{t}$. Appelons $\overline{R}$ la route qui relie $\overline{m}$ à $\overline{t}$ en empruntant $(\overline{r_1}, \overline{r_3}, \overline{t})$. Appelons $\overline{R'}$ la route empruntant $(\overline{r_1}, \overline{r_2}, \overline{r_4}, \overline{t})$. $\overline{R}$ est de longueur 3, et $\overline{R'}$ est de longueur 4. Si $\overline{r_1}$ pratique un équilibrage de charge uniforme entre $\overline{r_2}$ et $\overline{r_3}$, alors TRACEROUTE peut renvoyer la liste $(\overline{r_1}, \overline{r_2}, \overline{t})$. Dans ce cas, $\overline{m(t)} = \overline{r_2}$, alors que $\overline{r_2}$ n'est *pas* un voisin de $\overline{t}$.

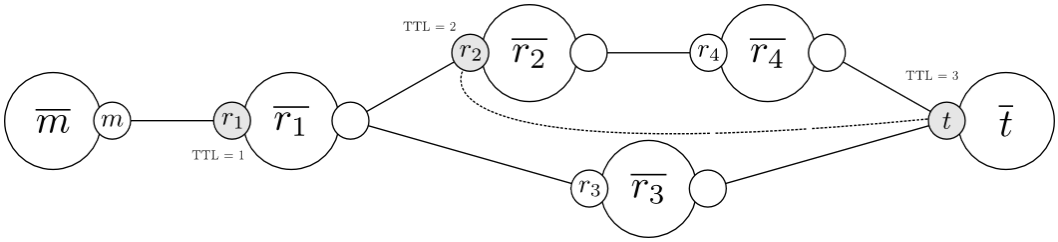


Figure 2.6 – *$\overline{m}$ lance un* TRACEROUTE *vers t en utilisant son interface m. Deux routes différentes sont parcourues alternativement par les sondes, mais ces deux routes n'ont pas la même longueur. Dans ce cas, $\overline{m(t)} = \overline{r_2}$ n'est* pas *un voisin de $\overline{t}$. En pointillés, le lien inexistant entre $\overline{r_2}$ et $\overline{t}$.*

Nous adresserons la problématique de détecter le cas défavorable où les routes entre un moniteur et une cible sont de longueur variable ultérieurement (**Section 2.6**). Pour l'instant, nous noterons :

Définition 15 (Ensemble des moniteurs observant correctement une cible)**.** *On appelle* ensemble des moniteurs observant correctemment une cible t *l'ensemble* $\mathbb{M}(t)$ *des moniteurs depuis lesquels toutes les routes vers* t *sont de même longueur ; en particulier tels que* $\overline{m(t)}$ *est un voisin de* $\overline{t}$*.*

Pour une certaine cible $t \in \mathbb{I}$, nous considérons alors que $m \in \mathbb{M}(t) \mapsto m(t) \in \mathbb{I}$ est notre *primitive de bas niveau basée sur* TRACEROUTE.

2.3 Primitive de mesure de haut niveau basée sur TRACEROUTE

Nous avons vu (**Section 2.2**) que nous pouvons utiliser une interprétation très prudente de TRACEROUTE pour mesurer, à l'aide d'un moniteur $m \in \mathbb{M}(t)$, une interface $m(t)$ d'un voisin au niveau logique d'une certaine cible t. Nous avons également noté que sauf dans le cas d'une configuration explicite imposant à un routeur d'utiliser toujours la même interface pour envoyer des paquets ICMP TIME EXCEEDED, le choix de l'interface $m(t)$ par $\overline{m(t)}$ dépend de m (donc de $\overline{m}$) et éventuellement d'autres paramètres (tels qu'un équilibrage stochastique). Enfin, évidemment, $\overline{m(t)}$ dépend de m (donc de $\overline{m}$) puisqu'il dépend des routes empruntées par les sondes TRACEROUTE depuis m vers t.

Soient alors $\overline{m} \in \mathbb{M}(t), \overline{m'} \in \mathbb{M}(t)$ deux moniteurs **distincts**. Il est possible que $m(t) \neq m'(t)$. Deux cas sont possibles : (1) $\overline{m(t)} = \overline{m'(t)}$ ou (2) $\overline{m(t)} \neq \overline{m'(t)}$. Le cas (1) se présente lorsque m et m' observent deux interfaces distinctes d'un même routeur. Le cas (2) se présente lorsque les routes empruntées par les sondes TRACEROUTE depuis m et m' atteignent $\overline{t}$ par des voisins différents (**Figure 2.7**).

En utilisant TRACEROUTE vers une même cible $\overline{t}$ désignée par l'une de ses adresses t depuis deux moniteurs, on peut donc collecter davantage d'information sur les voisins logiques de $\overline{t}$. Nous généralisons ce principe avec un ensemble arbitraire de moniteurs :

Définition 16 (Observation d'une cible depuis un ensemble de moniteurs)**.** *Soit* $M = \{m_1, \dots m_n\}$ *un ensemble d'interfaces appartenant chacune à des moniteurs* $\overline{M} = \{\overline{m_1}, \dots, \overline{m_n}\} \subset \mathbb{M}(t)$*. On note* $M(t) = \{m_1(t), \dots, m_n(t)\}$ *et* $\overline{M(t)} = \{\overline{m_1(t)}, \dots, \overline{m_n(t)}\}$*. On appelle* $M(t)$ observation de t depuis M*.*

Par construction, puisque $M \subset \mathbb{M}(t)$, alors $\overline{M(t)}$ est un ensemble de voisins au niveau logique de $\overline{t}$ et en particulier, $|\overline{M(t)}| \leq d_3(\overline{t})$. Nous allons examiner les

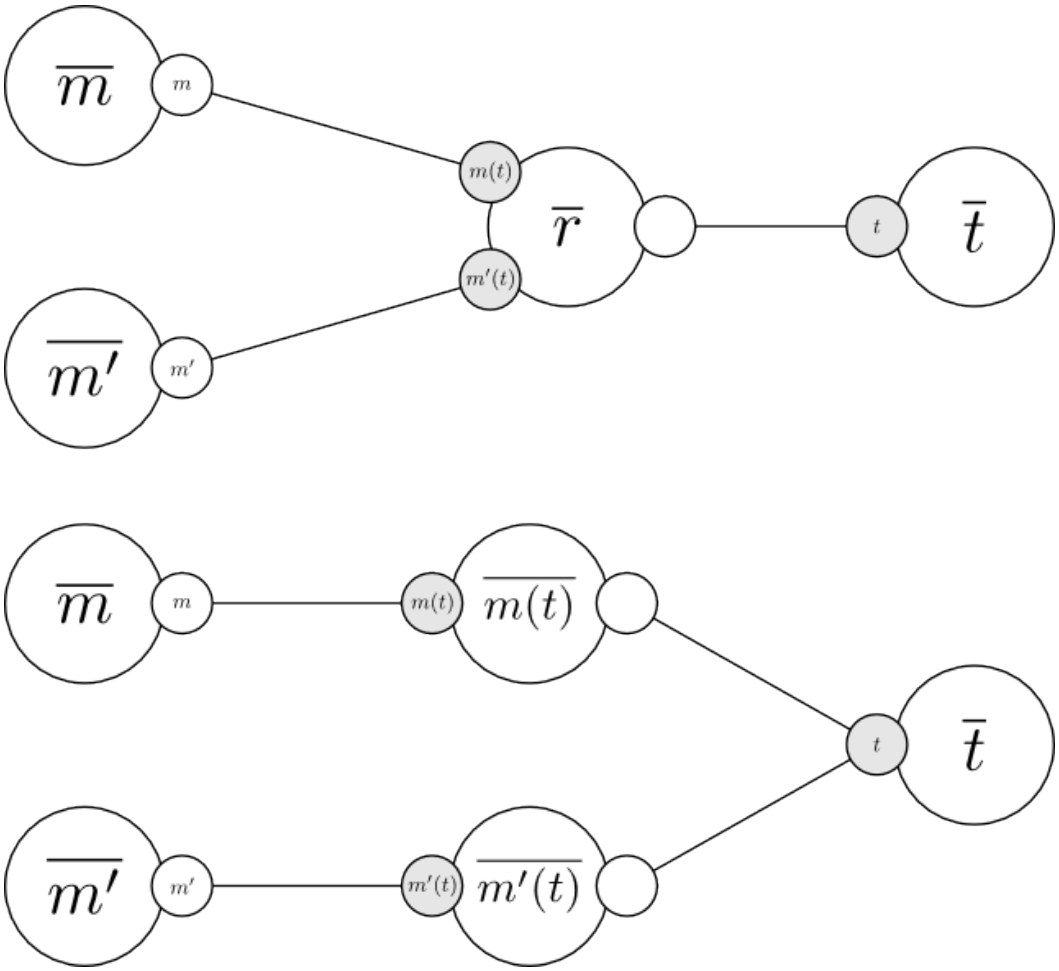


FIGURE 2.7 – *En haut, (1) : les sondes* TRACEROUTE *atteignent* $\overline{t}$ *par le même voisin* $\overline{r} = \overline{m(t)} = \overline{m'(t)}$ *mais* $\overline{r}$ *utilise deux interfaces différentes* $m(t)$ *et* $m'(t)$ *pour y répondre. En bas, (2) : les sondes* TRACEROUTE *atteignent* $\overline{t}$ *par des voisins distincts de* $\overline{t}$ *et donc* $\overline{m(t)} \neq \overline{m'(t)}$.

conditions sur m et t pour que $\overline{M(t)}$ soit le plus grand possible. Plus précisément, nous allons tenter de déterminer sous quelles conditions sur M et t on peut observer *tous* les voisins de $\overline{t}$, c'est à dire que $|\overline{M(t)}| = d_3(\overline{t})$.

Remarquons d'abord que cette condition dépend à la fois de M et de t. Puisque chaque moniteur m observe au plus *une seule* interface d'*un seul* voisin de $\overline{t}$, alors $|\overline{M(t)}| \leq |M|$ (le cas d'égalité survient lorsque chaque moniteur observe une interface d'un voisin différent). En particulier, si $|M| < d_3(\overline{t})$, la condition n'est pas réalisable. On doit donc *au minimum* disposer de davantage de moniteurs que le degré de la cible.

Mais il ne suffit pas d'avoir un grand nombre de moniteurs : il faut également qu'ils soient positionnés correctement par rapport à la cible, et à ses voisins, pour qu'on puisse tous les observer à l'aide des sondes TRACEROUTE. Supposons (comme

illustré en **Figure 2.8**) par exemple qu'un certain voisin de $\overline{t}$ soit un nœud de degré 1, c'est à dire que son seul voisin est $\overline{t}$. Alors quel que soit l'ensemble des moniteurs M, il est impossible d'observer ce voisin à l'aide de sondes TRACEROUTE.

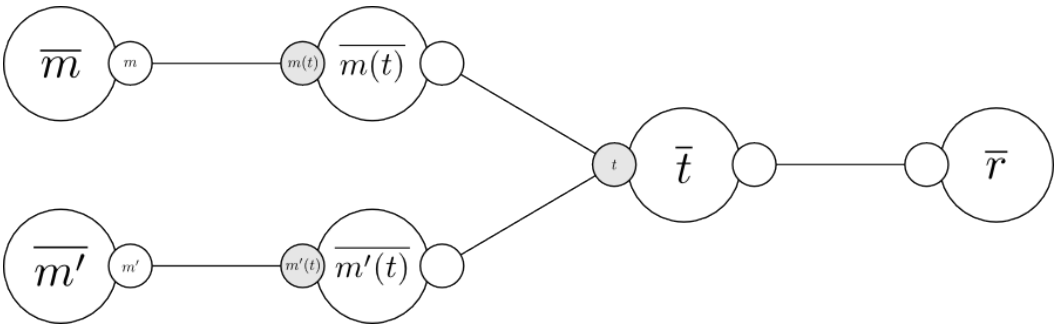


FIGURE 2.8 – *La cible $\overline{t}$ est voisine d'un nœud $\overline{r}$ de degré 1. Même si tous les autres nœuds du graphe sont des moniteurs, alors il est impossible d'observer $\overline{r}$ avec* TRACEROUTE.

Nous allons généraliser ce principe en définissant, pour une cible $t \in \overline{t}$ donnée et l'un de ses voisins $\overline{v}$, l'ensemble des nœuds qui sont capables d'observer $\overline{v}$ en ciblant t.

Définition 17 (Nœuds capables d'observer un voisin donné d'une cible)**.** *Soit $t \in \overline{t}$ une cible et $\overline{v} \in V(\overline{t})$. On note $\mathbb{M}(\overline{t}, \overline{v})$ l'ensemble des nœuds $\overline{u}$ tels qu'il existe un chemin de $\overline{u}$ vers $\overline{t}$ qui passe par $\overline{v}$. On appelle cet ensemble l'*ensemble des nœuds capables d'observer $\overline{v}$.

En particulier, puisque $\overline{u(t)}$ se trouve sur un chemin de $\overline{u}$ à $\overline{t}$ ne passant pas par $\overline{t}$, alors $\overline{u(t)} = \overline{v} \Rightarrow \overline{u} \in \mathbb{M}(\overline{t}, \overline{v})$. Nous allons tenter de caractériser cet ensemble plus en détails.

On partitionne les topologies d'Internet en deux sous-ensemble de nœuds complémentaires, le *cœur* et le *bord* d'Internet (illustré en **Figure 2.9**), ainsi définis :

Définition 18 (Cœur et bord d'Internet)**.** *Soit (U_n) la suite telle que $U_0 = V_3$ et U_{n+1} est l'ensemble des nœuds de degré > 1 dans U_n (c'est à dire dans le sous-graphe de G_3 dont les liens ont leurs deux extrémités dans U_n). On note C_3 et on appelle le* cœur de la topologie logique *la limite (finie) de (U_n). On note B_3 et on appelle le* bord de la topologie logique *l'ensemble $V_3 \setminus C_3$.*

Moins formellement, le cœur de la topologie logique d'Internet C_3 correspond à l'ensemble des nœuds d'Internet dont on a retiré récursivement les nœuds de degré 1, ou de manière équivalente dont on a retiré récursivement les arbres. Le bord correspond aux nœuds ainsi retirés.

Nous pouvons ainsi définir deux notions très importantes pour caractériser les voisins d'une certaine cible $t \in \overline{t}$ par rapport à la mesure :

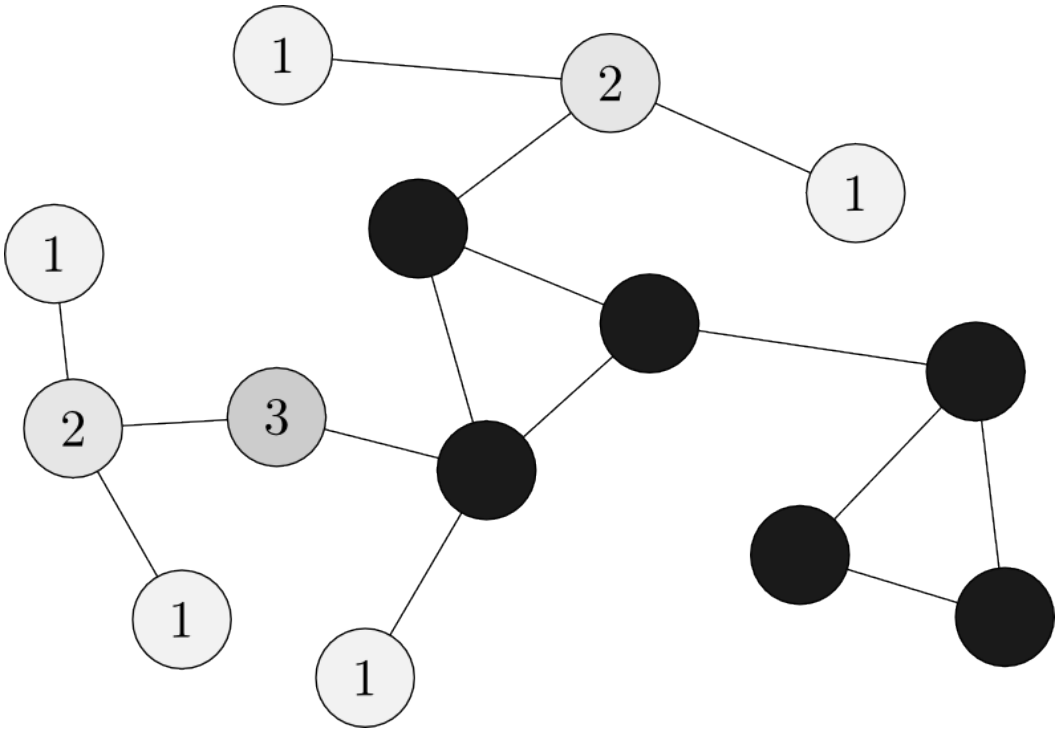


FIGURE 2.9 – *On retire récursivement les nœuds de degré 1 d'un graphe. Chaque nœud du* bord *est marqué par un nombre k, qui correspond à l'itération à laquelle il est supprimé. Les nœuds restants (en noir) sont les nœuds du* cœur.

Définition 19 (Arbre enraciné, fils et parent dans un arbre enraciné)**.** *Soit A un arbre et u un nœud de A. On appelle* arbre enraciné *de racine u le couple (A, u) qu'on note $A(u)$, et on dit alors que u est la racine de cet arbre. Soit v un nœud de $A(u)$, et w un voisin de u. Il y a alors deux possibilités qui s'excluent mutellement : soit (1) v est sur l'unique chemin de u vers w, soit (2) w est sur l'unique chemin de u vers v. Dans le premier cas, on dit que v est le* parent *de w. Dans le deuxième cas, on dit que v est un* fils *de w.*

Définition 20 (Voisinage d'une cible tourné vers le bord, degré dans le bord)**.** *Soit un nœud $\overline{t} \in V_3$. Notons $C_3(\overline{t}) = V(\overline{t}) \cap C_3$ et $B_3(\overline{t}) = V(\overline{t}) \cap B_3$. Soit $A(\overline{t})$ l'arbre enraciné de racine $\overline{t}$. On appelle* voisinage de $\overline{v}$ tourné vers le bord *les nœuds de $B_3(\overline{t})$ qui sont également des fils de $\overline{t}$ dans $A(\overline{t})$ et on note $B_3(V(\overline{t}))$ cet ensemble. On note $d_{B_3}(\overline{t}) = |B_3(V(\overline{t}))|$ et on appelle cette valeur le* degré dans le bord *de $\overline{t}$.*

Définition 21 (Voisinage d'une cible tourné vers le cœur, degré dans le cœur)**.** *Soit un nœud $\overline{t} \in V_3$. On appelle* voisinage de $\overline{t}$ tourné vers le cœur *l'ensemble $C_3(V(\overline{t})) = V(\overline{t}) \setminus B_3(V(\overline{t}))$. Si $\overline{t} \in C_3$, alors $C_3(V(\overline{t})) = C_3(\overline{t})$. On note $d_{C_3}(\overline{t}) = |C_3(V(\overline{t}))|$ et on appelle cette valeur le* degré dans le cœur *de $\overline{t}$.*

Si un certain voisin $\overline{v} \in V(\overline{t})$ est dans le cœur, il ne peut pas être un fils de $\overline{t}$ dans $A(\overline{t})$, sinon il serait lui-même racine d'un sous-arbre de $A(\overline{t})$) et il serait donc

dans le bord. Donc si $\overline{v}$ est dans le cœur, il appartient nécessairement au voisinage tourné vers le cœur de $\overline{t}$. Soit alors un voisin $\overline{v}$ de $\overline{t}$ dans le bord. Il y a deux cas possibles : soit $\overline{t}$ est lui-même dans le bord (1), soit $\overline{t}$ est dans le cœur (2).

Si $\overline{t}$ est dans le bord (1), alors soit $\overline{v} \in C_3(V(\overline{t}))$ (1.a), soit $\overline{t} \in B_3(V(\overline{v}))$ (1.b). On peut alors déduire $\mathbb{M}(t, \overline{v})$. Dans le cas (1.a), tous les chemins depuis $\overline{v}$ vers le cœur passent par $\overline{t}$, donc $\mathbb{M}(t, \overline{v})$ est exactement l'ensemble des nœuds qui sont eux-mêmes des descendants de $\overline{v}$ dans l'arbre enraciné $A(\overline{v})$. Dans le cas (1.b), c'est la situation inverse : tous les chemins depuis $\overline{t}$ vers le cœur passent par $\overline{v}$, et donc $\mathbb{M}(t, \overline{v})$ est l'ensemble des nœuds qui *ne sont pas des descendants de* $\overline{t}$ dans l'arbre enraciné $A(\overline{t})$, c'est à dire $\mathbb{M}(t, \overline{v}) = V_3 \setminus A(\overline{t})$. Dans le cas (1) (illustré en **Figure 2.10**) où $\overline{t}$ est dans le bord, soit la cible $\overline{t}$ est sur les chemins depuis un voisin $\overline{v}$ vers le cœur et donc les moniteurs capables de l'observer sont les descendants de ce voisin $\overline{v}$ (1.b), soit ce voisin $\overline{v}$ est sur les chemins depuis la cible $\overline{t}$ vers le cœur et donc les moniteurs capables de l'observer sont tous les autres nœuds du graphe (1.a) (**Figure 2.10**).

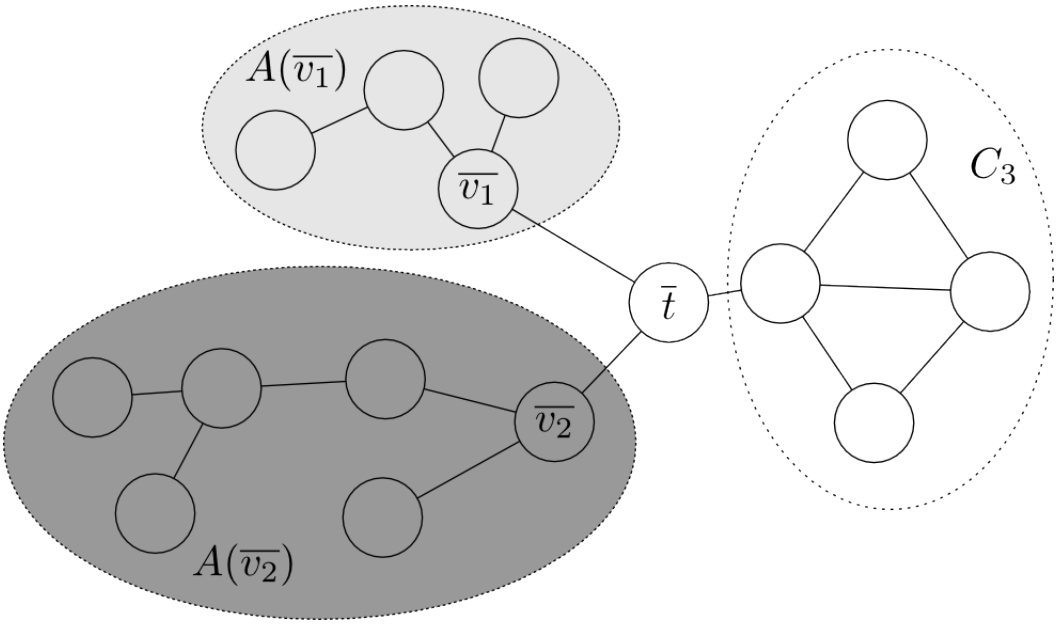


Figure 2.10 – *On souhaite observer les voisins d'une cible* $\overline{t}$ **dans le bord**. *Les voisins de* $\overline{t}$ *tournés vers le bord ne sont observables que depuis les sous-arbres dont ils sont la racine. Les voisins de* $\overline{t}$ *tournés vers le cœur en revanche sont observables depuis n'importe quel nœud qui n'est pas un descendant d'un voisin de* $\overline{t}$ *tourné vers le bord.*

Dans le cas (2) (illustré en **Figure 2.11**) où $\overline{t}$ est dans le cœur, alors $\overline{v}$ est nécessairement la racine d'un arbre enraciné de nœuds du bord. Donc pour passer par $\overline{v}$ en chemin vers $\overline{t}$ il faut et il suffit d'être un descendant de cet arbre. Donc $\mathbb{M}(t, \overline{v}) = A(\overline{v})$. (**Figure 2.11**).

On a donc ici établi les relations entre les moniteurs, les cibles et leurs voisins pour déterminer quels moniteurs peuvent observer quels voisins de quelle cible.

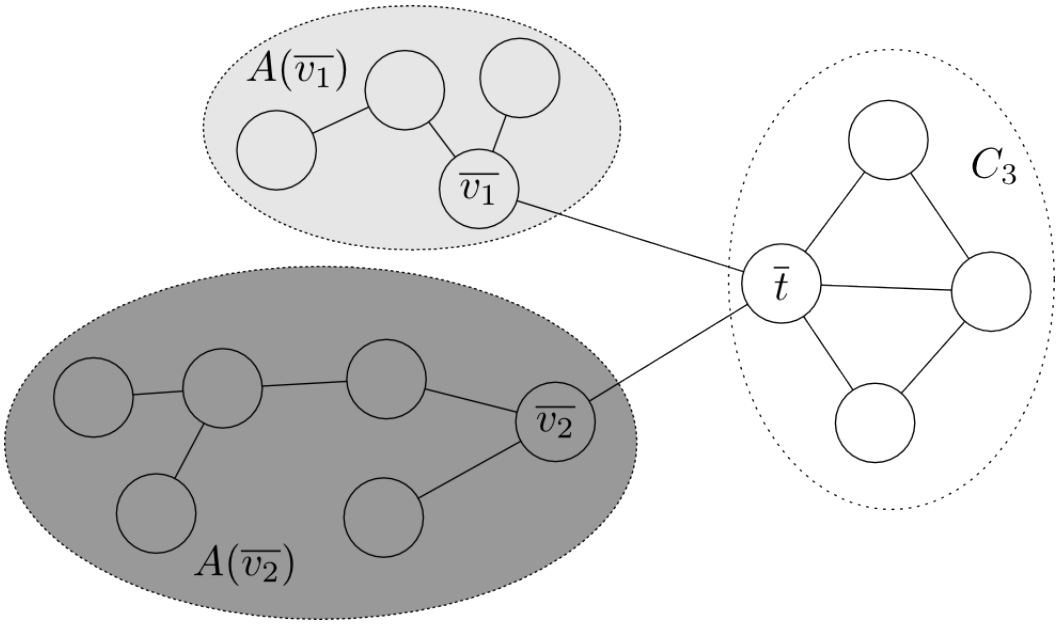


FIGURE 2.11 – *On souhaite observer les voisins d'une cible $\overline{t}$* **dans le cœur** *qui sont dans le bord. Les nœuds qui peuvent observer ces voisins-là avec* TRACEROUTE *sont exactement les nœuds qui sont des descendants de l'arbre dont ils sont la racine.*

Nous avons en particulier démontré que pour une cible donnée, pour observer les voisins tournés vers le bord, il faut disposer de moniteurs situés précisément dans les arbres dont ces voisins sont les racines. Dans la démarche de mesure qui nous intéresse ici, nous disposerons d'un ensemble fixé de moniteurs à notre disposition pour effectuer une mesure. On ne pourra donc pas supposer que nous disposons arbitrairement de moniteurs dans n'importe quel arbre de G_3. Typiquement, notre ensemble de moniteurs sera assimilé à un ensemble de nœuds du bord réparti de manière arbitraire dans le réseau. En conséquence, *a priori*, on ne pourra être garanti d'observer que les voisins tournés vers le cœur d'une cible donnée, sauf si *par hasard* nous disposons d'un moniteur qui se trouve précisément être un descendant de l'arbre dont l'un des voisins du bord de la cible est la racine. Cette explication se formalise ainsi :

Proposition 2 (Mesure des voisins dans le cœur)**.** *Soit une cible $\overline{t}$ donnée dans le cœur d'Internet, et M un ensemble de moniteurs. Supposons qu'aucun moniteur $m \in M$ ne soit un descendant d'un arbre dont la racine est un voisin de $\overline{t}$ tourné vers le bord. Alors $\overline{M(t)}$ est un sous-ensemble des voisins de $\overline{t}$ tournés vers le cœur, c'est à dire $\overline{M(t)} \subset C_3(V(\overline{t})) = C_3(\overline{t})$.*

Nous allons désormais essayer de déterminer des conditions pour obtenir l'inclusion inverse, c'est à dire pour que $C_3(V(\overline{t})) \subset M(t)$, et donc que $C_3(\overline{t}) = M(t)$. Il faut et il suffit que pour chaque voisin $\overline{v}$ de $\overline{t}$ dans le cœur (ou tourné vers le cœur, ce qui est équivalent car $\overline{t}$ est supposée dans le cœur), on dispose d'au

moins un moniteur $m \in M$ tel que les sondes TRACEROUTE depuis m vers t passent par $\overline{v}$, c'est à dire $m \in \mathbb{M}(t, \overline{v})$. Notons $\mathbb{N}(t) = \bigcup_{\overline{v} \in C_3(\overline{t})} \mathbb{M}(t, \overline{v})$. Alors il faut et il suffit que $M \subset \mathbb{N}(t)$. Comme nous l'avons déjà mentionné, plus $\overline{t}$ possède de voisins dans le cœur, plus il est difficile *a priori* d'observer tous ses voisins, c'est à dire plus il faudra de moniteurs positionnés dans G_3 à des endroits différents. L'examen formel des différentes configurations locales et globales permettant de conclure sort du cadre de ce travail, mais cette formulation du problème nous a permis de réaliser des simulations (détaillées en **Section 2.7**) et de conclure :

Hypothèse 4 (Validité de la primitive de haut niveau basées sur TRACEROUTE)**.** *Soit une cible $\overline{t} \in C_3$ et M un ensemble de moniteurs. Si M est assez grand et assez bien réparti, alors $\overline{M(t)} = C_3(\overline{t})$ et en particulier $|\overline{M(t)}| = d_{C_3}(\overline{t})$.*

En particulier, si tous les voisins de $\overline{t}$ sont dans le cœur, alors $d_3(\overline{t}) = d_{C_3}(\overline{t}) = |\overline{M(t)}|$. Pour une certaine cible $t \in \overline{t} \in C_3$ dans le cœur, nous considérons alors $M \subset \mathbb{N}(t) \mapsto M(t) \subset I$ notre *primitive de mesure de haut niveau basée sur* TRACEROUTE. (**Figure 2.12**)

2.4 Estimation de la distribution de degré du cœur d'Internet au niveau logique

Nous avons expliqué dans les sections précédentes comment, à l'aide d'un ensemble de moniteurs M suffisamment grand et suffisamment bien réparti dans le réseau, nous sommes capables d'observer des interfaces de tous les *voisins dans le cœur* d'une cible $\overline{t}$ — en particulier si cette cible est dans le cœur, puisqu'une cible dans le bord possède au plus un unique voisin tourné vers le cœur, *a fortiori* dans le cœur.

Remarquons d'abord que notre primitive de mesure de haut niveau permet d'obtenir une *liste d'interfaces appartenant à des voisins dans le cœur d'une cible désignée par l'une de ses interfaces*. Le problème de quotienter cette liste pour identifier les interfaces appartenant à un même nœud est le problème de l'*anti-aliasing*, déjà évoqué précédemment. Des travaux antérieurs [30, 31, 39] ont déjà proposé des solutions satisfaisantes pour résoudre ce problème et il sort donc du cadre de ce travail. Nous nous bornerons donc à mesurer de telles listes d'interfaces sans chercher ici à réaliser l'*anti-aliasing*, c'est à dire passer de $M(t)$ à $\overline{M(t)}$. Nous supposerons que cette partie du problème est acquise comme résolue, et que si nous disposons de $M(t)$, alors nous pouvons directement déduire $\overline{M(t)}$.

L'application de notre méthode *mesure orientée propriétée* aux *routeurs du cœur de la topologie logique d'Internet* consiste à appliquer notre *primitive de mesure de haut niveau basée sur* TRACEROUTE depuis un ensemble de moniteurs vers un certain ensemble de cibles T, *tiré de manière uniformément aléatoire parmi les*

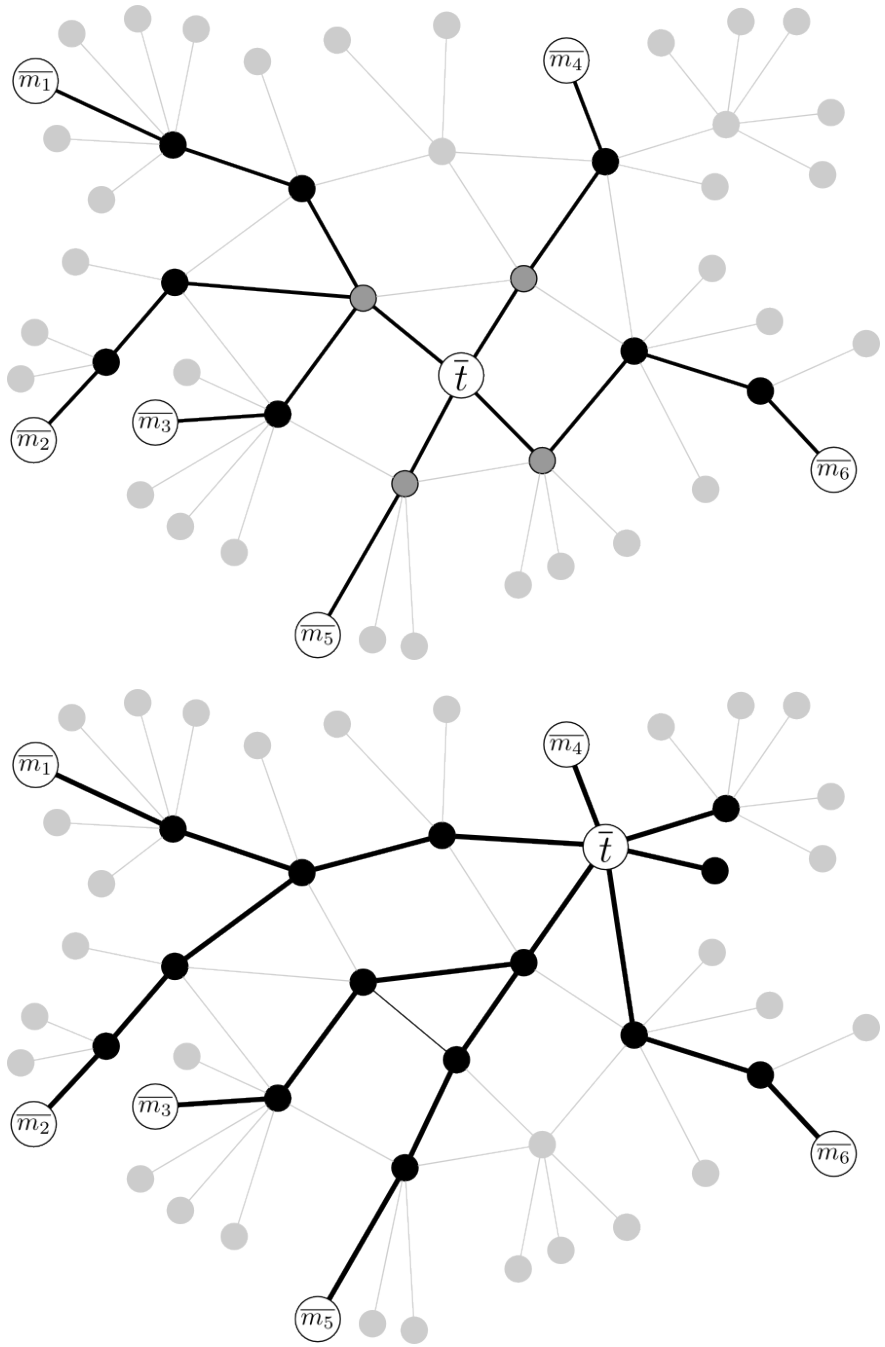


Figure 2.12 – *Les liens en noir sont parcourus par des sondes* TRACEROUTE*, les liens en gris ne le sont pas. En haut (a) : la cible est dans le cœur et nous disposons d'un ensemble de moniteurs qui permet d'observer au moins une fois chacun de ses voisins qui sont tous dans le cœur également. En bas (b) : la cible est dans le cœur mais certains de ses voisins sont dans le bord. Nous ne disposons pas de moniteurs dans les arbres dont ces voisins sont les racines, donc nous n'observons que les voisins de la cible qui sont dans le cœur.*

adresses du cœur de l'Internet (c'est à dire *sans biais de sélection*). Commençons par définir notre méthode pour un certain ensemble de moniteurs vers un ensemble cibles dans le cœur d'Internet :

Définition 22 (Mesure depuis un ensemble de moniteurs d'un ensemble de cibles)**.** *Soit $T = \{t_1, \ldots, t_n\} \subset \mathbb{I}$ un ensemble de cibles désignées par leurs interfaces tel que pour chaque k, $\overline{t_k} \in C_3$. On note $M(T) = \{(t_1, M(t_1)), \ldots, (t_n, M(t_n))\}$, et $\overline{M(T)} = \{(\overline{t_1}, \overline{M(t_1)}, \ldots, (\overline{t_n}, \overline{M(t_n)})\}$.*

On va en particulier s'intéresser à une propriété de $M(T)$, la distribution de degré dans le cœur mesurée de notre ensemble de cibles :

Définition 23 (Mesure depuis un ensemble de moniteurs de la distribution de degré dans le cœur d'un ensemble de cibles)**.** *Soit T un ensemble de cibles et M un ensemble de moniteurs. On note $d(\overline{M(T)})$ la distribution des degrés mesurés, définie par $d(\overline{M(T)})(k) = |\{\overline{t}, |\overline{M(t)}| = k\}|$. On note $\hat{d}(\overline{M(T)})$ cette distribution normalisée (telle que sa somme soit égale à 1).*

Si les informations données par $M(T)$ sont intéressantes dans l'absolu, le principe de *mesure orientée propriétée* repose sur notre capacité à déduire des *propriétés de la topologie logique d'Internet* à partir de mesures ciblées. Plus précisément, nous allons tenter d'estimer la distribution de degré dans le cœur des nœuds du cœur, définie par :

Définition 24 (Distribution de degré (dans le cœur) du cœur d'Internet)**.** *On note $d(C_3)$ et on appelle* distribution de degré (dans le cœur) du cœur d'Internet *la distribution définie par $d(C_3)(n) = |\{\overline{v} \in C_3, |C_3(V(\overline{v}))| = n\}|$. On note $\hat{d}(C_n)$ cette distribution normalisée (telle que sa somme soit égale à 1).*

Notre objectif est, en ayant à notre disposition un ensemble de moniteurs M qu'on suppose assez grand et assez bien distribué pour mesurer correctement n'importe quelle cible t donnée, de choisir un ensemble de cibles T de telle sorte que $\hat{d}(\overline{M(T)}) = \hat{d}(C_3)$, ou de minimiser autant que possible la différence entre ces deux distributions (définie par $||\hat{d}(\overline{M(T)} - \hat{d}(C_3)|| = \sum_n |\hat{d}(\overline{M(t)})(n) - \hat{d}(C_3)(n)|$). De cette manière, nous pouvons effectuer une mesure ciblée (de l'ensemble des cibles T) afin d'estimer $\hat{d}(C_3)$, une propriété topologique fondamentale de G_3. Pour ce faire, nous allons choisir T de telle manière qu'il soit un *échantillon représentatif* de C_3 du point de vue de l'observation de cette propriété. Autrement formulé, supposons que T soit un ensemble de cibles tel que $\overline{T}$ soit un ensemble tiré de manière *uniformément aléatoire* dans le cœur d'Internet. Alors les propriétés topologiques des nœuds de $\overline{T}$ sont représentatives des propriétés topologiques des nœuds de C_3, et en particulier, plus $\overline{T}$ est grand (en restant uniformément aléatoire), plus les propriétés des nœuds de $\overline{T}$ sont proches de celles de la totalité du cœur d'Internet. Pour le cas de la distribution de degré, on peut le formaliser de la manière suivante :

Proposition 3 (Estimation de la distribution de degré du cœur logique d'Internet)**.** *Soit M un ensemble de moniteurs suffisamment grand et suffisamment bien réparti. Soit $(T_k)_k$ une suite d'ensembles de cibles telle que la suite $(|\overline{T_k}|)_k$ soit strictement croissante et chaque $\overline{T_k}$ est un ensemble de cibles tiré de manière uniformément aléatoire dans C_3. Alors la suite $(\varepsilon_k)_k = (||\hat{d}(\overline{M(T_k)}) - \hat{d}(C_3)||)_k$ tend vers 0.*

La qualité de l'estimation (ε_k), si l'on suppose que chaque (T_k) est un échantillon uniforme de taille $|T_k|$, n'est pas un problème topologique, mais un problème statistique. On peut interpréter $\hat{d}(C_3)$ comme un ensemble de variables aléatoires $X_n = \hat{d}(C_3)(n)$ et chaque $\hat{d}(\overline{M(T_k)})(n)$ comme un estimateur de X_n. Dans ce cas, on peut tester l'adéquation empirique de $\hat{d}(\overline{M(T_k)})$ avec une distribution donnée, par exemple une loi de Poisson ou une loi de puissance, en réalisant des tests d'hypothèses classiques. Ces tests sortent du strict cadre de notre travail, mais on peut mentionner le *test du* χ^2, ou le *test de Kolmogorov-Smirnov* [64].

Cela signifie que pour obtenir une estimation aussi précise que l'on souhaite de la propriété qui nous intéresse, la distribution de degré du cœur logique d'Internet (et la comparer à une distribution théorique donnée), il suffit de réaliser les conditions suivantes :

- (1) Être capable de tirer un ensemble de cibles T d'interfaces de nœuds du cœur tel que $\overline{T}$ soit uniformément aléatoire dans C_3 et suffisamment grand,
- (2) Disposer d'un ensemble de moniteurs M suffisamment grand et suffisamment bien réparti,
- (3) Effectuer une mesure correcte de $M(T)$, ce qui suppose en particulier d'éliminer les $m(t)$ incorrects (les couples (m, t) tels que les routes de m vers t ne sont pas toutes de même longueur, ou tels que m soit un descendant de l'arbre enraciné de racine t),
- (4) Calculer $\overline{M(T)}$ (pour déduire $\hat{d}(\overline{M(T)})$).

Nous examinerons la condition (1) dans la **Section 2.5**. La condition (2) sera supposée vraie, mais le réalisme de cette hypothèse sera examiné à travers des simulations en **Section 2.7**. La condition (3) repose sur un filtrage rigoureux des données collectées par TRACEROUTE *a posteriori* que nous allons décrire en **Section 2.6**. Enfin, la condition (4) repose sur des méthodes d'*anti-aliasing* décrites dans des travaux antérieurs.

2.5 Échantillonage des cibles dans le cœur

Dans la **Section 2.4**, nous avons montré que notre méthode orientée propriété pour estimer la distribution de degré du cœur logique d'Internet repose sur notre capacité à sélectionner un ensemble de cibles T tel que $\overline{T}$ soit uniformément aléatoire dans le cœur d'Internet. Dans le strict cadre de notre travail sur la topologie

logique, nous n'avions pas réussi à aller jusqu'au bout de cette démarche †. Nous avons cependant mis au point une méthode de sélection qui permet de sélectionner T tel que T soit *uniformément aléatoire dans l'ensemble des interfaces appartenant à des nœuds du cœur*. Nous procédons en plusieurs étapes, en supposant que l'on dispose d'un ensemble M de moniteurs suffisamment grand et suffisamment bien réparti :

- (1) On tire **de manière uniformément aléatoire** un ensemble T_0 d'entiers 32 bits, qui représentent des *adresses* IP (qui peuvent être valides ou invalides),
- (2) On extrait l'ensemble $T_1 = T_0 \cap \mathbb{I}$ de ces adresses qui sont valides,
- (3) On utilise notre primitive de mesure de haut niveau pour calculer $M_1 = M(T_1)$,
- (4) On extrait de cette mesure M_1 le sous-ensemble de données M_2 dont on a supprimé les données incorrectes, c'est à dire issues de couples (m, t) tels qu'il existe des routes de longueurs différentes de m vers t ou tels que m soit un descendant de t dans $A(\bar{t})$, (*cf.* **Section 2.6**)
- (5) On extrait de T_1 le sous-ensemble T_2 des cibles t telles que $M_2(t) = M(t) \cap M_2$ observe $\bar{t}$ avec un degré tourné vers le cœur **strictement supérieur** à 1.

Notons d'abord qu'aucune de ces étapes ne nuit à l'uniformité aléatoire de l'ensemble des cibles, et que par conséquent l'ensemble final T_2 est également uniformément aléatoire. Il nous reste à vérifier que T_2 est bien un sous-ensemble de C_3. Les étapes (2), (3) et (4) garantissent, comme démontré lors des sections précédentes, que $M_2(t)$ correspond bien à un ensemble d'interfaces des voisins tournés vers le cœur de $\bar{t}$. Comme M est supposé suffisamment grand et suffisamment bien réparti, alors $M_2(t)$ contient au moins une interface de chacun des voisins tournés vers le cœur de $\bar{t}$. En appliquant une méthode d'*anti-aliasing* à $M_2(t)$, on obtient $\overline{M_2(t)}$ et donc $|\overline{M_2(t)}|$, le *degré dans le cœur* de $\bar{t}$, c'est à dire le nombre de voisins *tournés vers le cœur* de $\bar{t}$. L'étape (5) revient donc à conserver uniquement les cibles qui ont *au moins 2* voisins *tournés vers le cœur*. Or, par construction, les nœuds du bord sont **exactement** les nœuds qui ont 0 ou 1 seul voisin *tourné vers le cœur*, puisqu'ils sont situés sur des arbres ; réciproquement, si un nœud est sur un arbre, alors il est sur le bord (**Figure 2.13**). On peut en conclure que $\overline{T_2} \subset C_3$, ce que nous cherchions à démontrer.

2.6 Filtrage des résultats

La méthode que nous avons décrite en **Section 2.4**, et en particulier l'échantillonage des cibles dans le cœur détaillé en **Section 2.5**, repose sur deux hypothèses.

†. Même si nous y sommes parvenus dans des travaux ultérieurs (**Chapitre 3**)

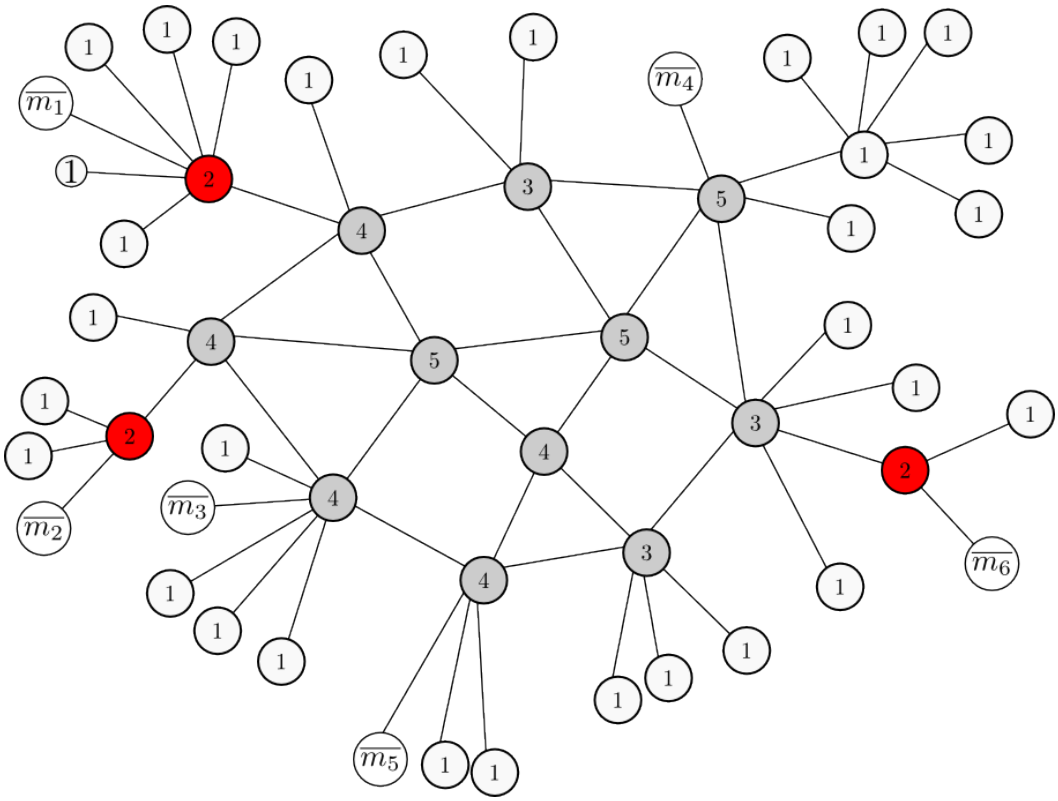


Figure 2.13 – *On étiquette les nœuds de G_3 par leur degré mesuré depuis l'ensemble M. Les nœuds de C_3 (en gris) sont les nœuds dont le degré mesuré est strictement supérieur à 1 qui ne sont pas des ancêtres des moniteurs dans le sous-arbre qui les relie au cœur (en rouge).*

Pour une certaine mesure $M(T)$, composée de mesures ponctuelles $m(t)$, nous supposons que (1) les chemins empruntés par TRACEROUTE depuis m vers t sont de longueur constante, et que (2) m n'est pas dans $A(\bar{t})$. Pour garantir ces deux hypothèses, nous partons d'une mesure $M(T)$ et nous détectons les couples (m, t) qui ne satisfont pas soit (1) soit (2) ; nous extrayons un sous-ensemble des données privé de ces couples. Il nous suffit donc de disposer d'un moyen de tester (1) et (2) pour chaque couple $(m, t) \in M \times T$.

Le cas (1) peut être détecté en exécutant plusieurs fois TRACEROUTE depuis m vers t successivement. En effet, TRACEROUTE, dans l'interprétation rigoureuse que nous en avant donné en **Section 2.1**, fournit une information sur la longueur de la route empruntée par la dernière sonde, celle qui atteint la cible $\bar{t}$. En répétant plusieurs fois TRACEROUTE, on peut en principe détecter, en comparant les longueurs obtenues, la propriété désirée. Si lors des mesures successives, TRACEROUTE renvoie au moins deux longueurs distinctes, alors on peut supprimer le couple (m, t) de notre jeu de données en le considérant comme suspect et potentiellement tel que $\overline{m(t)}$ ne soit pas un voisin de $\bar{t}$. Idéalement, il nous faudrait être capable de majorer le risque, c'est à dire d'obtenir une majoration de la probabilité qu'il existe

bien des routes de m vers t de longueur différentes, en supposant que la réalisation de TRACEROUTE depuis m vers t au cours de n itérations ont toutes fourni une longueur égale.

Nous n'avons pas pu obtenir de réponse dans le cas général, mais nous avons étudié un cas simple pour comprendre le phénomène. Supposons qu'il y ait exactement deux routes de m vers t de deux longueurs distinctes $l_1 < l_2$, qui sont équiprobables, c'est à dire que pour chaque TRACEROUTE de m vers t, ces deux routes soient empruntées par la dernière sonde (celle qui atteint $\bar{t}$) avec une probabilité de $\frac{1}{2}$. Quand une sonde TRACEROUTE est envoyée depuis m vers t avec un TTL égal à l_1, elle emprunte l'une de ces deux routes. Si elle emprunte la route la plus courte (de longueur l_1), alors cette sonde atteint $\bar{t}$ et TRACEROUTE se termine et renvoie cette longueur l_1. Si au contraire elle emprunte la route la plus longue (de longueur l_2), alors TRACEROUTE se terminera ultérieurement et renverra une longueur entre $l_1 + 1$ et l_2. Il s'ensuit que, si l'on effectue TRACEROUTE depuis m vers t, la probabilité que l'on observe *toujours* la longueur l_1 est majorée par $\frac{1}{2^{n-1}}$. Au moins dans ce cas simplifié, le risque de ne pas rejeter (m, t) alors qu'il est possible que $\overline{m(t)} \notin V(\bar{t})$ décroît exponentiellement avec le nombre de fois qu'on exécute TRACEROUTE sans constater de longueurs distinctes. On peut donc raisonnablement supposer qu'il suffit de seulement quelques exécutions successives de TRACEROUTE depuis m vers t (de l'ordre d'une dizaine) pour garantir avec une très grande confiance que $\overline{m(t)} \in V(\bar{t})$. On peut augmenter arbitrairement la confiance dans cette garantie en augmentant le nombre d'itérations de TRACEROUTE. Une autre méthode, la *méthode des routes longues*, est une autre perspective qui sera discutée plus loin (**Chapitre 5**).

Le cas (2) peut également être résolu en utilisant des mesures TRACEROUTE complémentaires. Par définition, il ne peut se produire que dans le cas où $\overline{m}$ est un nœud du bord, et un descendant de $A(\bar{t})$. Donc $\bar{t}$ sépare V_3 en deux parties $A = A(\bar{t})$ et $\complement_A = V_3 \setminus A(\bar{t})$, telles que tous les chemins depuis un nœud de l'une de ces parties vers un nœud dans l'autre de ces parties passe nécessairement par $\bar{t}$ (**Figure 2.14**). On veut être capable de détecter le cas où $m \in \complement_A$. On effectue alors, pour chaque couple de moniteurs (m_1, m_2), TRACEROUTE depuis m_1 vers m_2. Comme $\bar{t}$ répond aux sondes TRACEROUTE (sinon il ne serait pas dans T), alors si m_1 et m_2 ne sont pas dans la même partie du découpage entre A et $\complement_A$, alors une interface de $\bar{t}$ (détectée comme telle par *anti-aliasing*) est nécessairement présente dans la sortie de TRACEROUTE de m_1 vers m_2, qu'on notera $\overline{[m_1, m_2]}$. Par contraposée, si $\bar{t}$ n'*apparaît pas* dans TRACEROUTE depuis m_1 vers m_2, c'est à dire si $\bar{t} \notin \overline{[m_1, m_2]}$, alors nécessairement soit m_1 et m_2 sont tous les deux dans A, soit ils sont tous les deux dans $\complement_A$.

Fixons à présent un moniteur m pour déterminer s'il se trouve dans le cas (2). Supposons que c'est le cas et que $m \in A$. Alors tous les moniteurs dans $M \cap \complement_A$ sont tels que $\bar{t}$ apparaît dans TRACEROUTE vers m. Comme par hypothèse M est suffisamment grand et suffisamment bien réparti pour observer correctement le

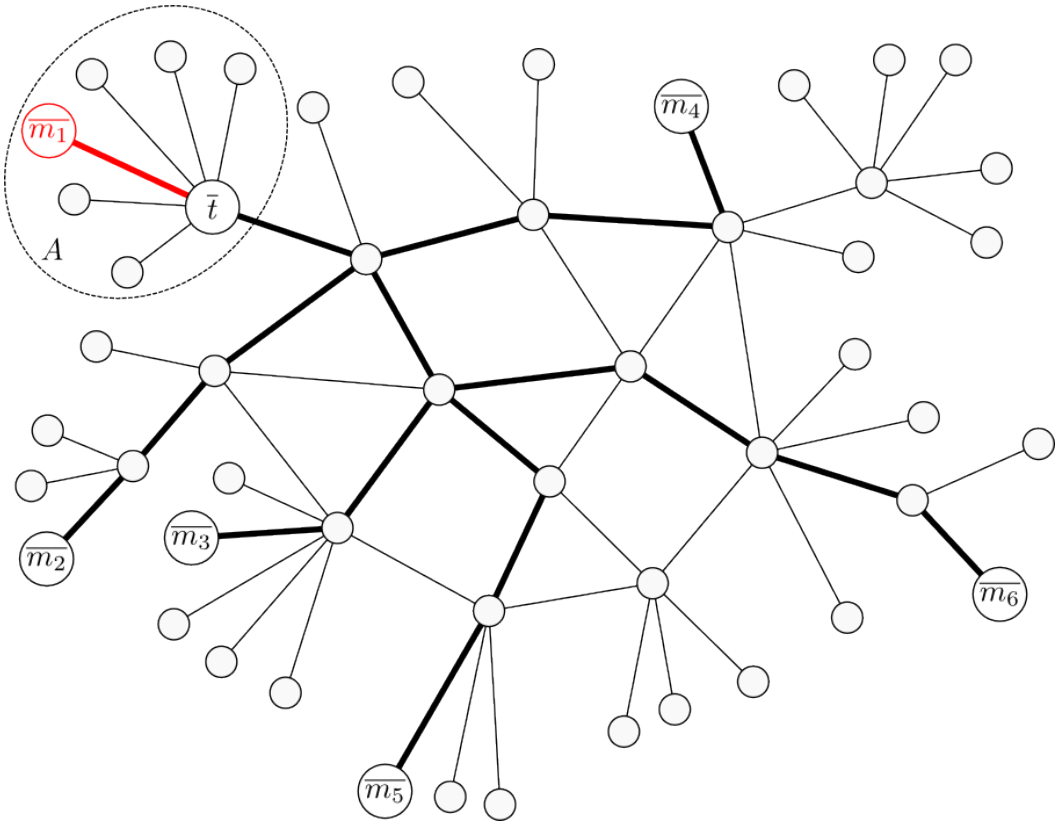


Figure 2.14 – *$\bar{t}$ est un nœud du bord, et donc racine d'un arbre A. On découpe G_3 en deux parties A et $\complement_A$. Tous les chemins qui passent de A à $\complement_A$ passent par $\bar{t}$ et réciproquement. On peut donc déduire que si deux moniteurs sont chacun dans une partie différente de G_3, alors $\bar{t}$ apparaît dans* TRACEROUTE *depuis l'un des moniteurs vers l'autre.*

cœur d'Internet, il n'est pas vraissemblable que presque tous les moniteurs se trouvent dans A. Donc $M \cap \complement_A$ représente nécessairement une forte proportion des moniteurs, qu'on peut minorer par exemple à $p = \frac{1}{2}$. Alors nécessairement au moins une proportion p des sorties de TRACEROUTE depuis M vers m contiennent $\bar{t}$. Par contraposée, si moins d'une proportion inférieure à p des sorties de TRACEROUTE depuis M vers m contiennent $\bar{t}$, alors on peut **déduire** que $m \notin A$. Pour résoudre le cas (2), on compte donc le nombre d'occurences parmis les sorties de TRACEROUTE depuis M vers m et on supprime (m, t) de notre jeu de données si $\bar{t}$ apparaît dans au moins une proportion p des résultats. On ne peut pas conclure la réciproque, mais nous nous satisfaisons du risque de faux-positifs (éliminer (m, t) alors que $m \notin A(t)$), notre considération essentielle étant d'avoir le plus faible taux de faux-négatif (conserver (m, t) alors que $m \in A(t)$). Notons que l'estimation $p = \frac{1}{2}$ est *très* prudente, en pratique nous avons grossièrement estimé que p est au moins de $\frac{9}{10}$ si l'ensemble M est bien choisi.

En résumé, pour filtrer les résultats et garantir que $\overline{M(T)}$ correspond bien à

l'observations de voisins des cibles $\overline{T}$ qui sont tournées vers le cœur, nous procédons de la manière suivante :

- Pour chaque couple (m, t) on effectue TRACEROUTE n fois depuis m vers t et on supprime (m, t) de notre jeu de données si on obtient au moins 2 longueurs distinctes. (Traitement du cas (1))
- Pour chaque couple (m_1, m_2) on effectue TRACEROUTE depuis m_1 vers m_2 et pour chaque $m \in M$ et chaque $t \in \bar{t}$ on compte $N(m, \bar{t})$ le nombre de moniteurs m' tels que $\bar{t}$ apparaît dans TRACEROUTE depuis m' vers m. Si $\frac{N(m,\bar{t})}{|M|} > p = \frac{1}{2}$ alors on supprime (m, t) de notre jeu de données. (Traitement du cas (2))

2.7 Simulations

La qualité de notre estimation de la distribution de degré dans le cœur des routeurs du cœur d'Internet repose sur une hypothèse importante : que l'on dispose d'un ensemble de moniteurs M qui soit suffisamment grand et suffisamment bien réparti. Dans cette section, nous ne cherchons pas à déterminer si un ensemble M *donné* souscrit à ces hypothèses (nous reviendrons sur cette question dans le **Chapitre 3** et le **Chapitre 4**), mais plutôt à déterminer s'il est *réaliste* de supposer qu'avec un ensemble limité de moniteurs, on peut correctement estimer le degré dans le cœur d'une cible, et a fortiori la distribution de degré dans le cœur du cœur d'Internet. Pour ce faire, nous avons réalisé des simulations de notre méthode de mesure sur des modèles simplifiés d'Internet inspirés des approches historiques.

Notre première hypothèse simplificatrice, correspondant aux approches historiques de modélisation [40, 29] et d'observation [29], est que les sondes empruntées par TRACEROUTE suivent des plus courts chemins (choisis aléatoirement, lorsque plusieurs sont disponibles) pour atteindre leur destination. Il est à noter que cette hypothèse n'est pas celle que nous avons refuté en **Section 2.1**, puisqu'elle porte sur chacune des sondes de TRACEROUTE, pas sur la sortie de TRACEROUTE. L'ensemble des moniteurs M est un ensemble uniformément aléatoire dans le graphe et l'ensemble des cibles T est la totalité du graphe.

Notre deuxième hypothèse simplificatrice concerne le type de topologie sur lequel nous réalisons nos simulations. Nous avons utilisé dans le cadre de ce travail des graphes aléatoires à distribution de degré suivant une loi de Poisson. Nous montrerons que nous avons ultérieurement réalisé des simulations sur d'autres topologies dans le cadre de notre travail sur la topologie physique (**Chapitre 3**). Un graphe aléatoire en loi de Poisson possède deux paramètres principaux : le nombre de nœuds et son degré moyen. Il est difficile en pratique de simuler des graphes dont la taille totale est comparable à celle d'Internet (qui se compte en milliards de nœuds), nous avons donc choisi d'utiliser des graphes de taille 10^6 et 10^7. Pour le

degré moyen, nous avons choisi un degré moyen de 20, ce qui est élevé par rapport aux hypothèses des approches historiques, et *a priori* défavorable à notre méthode (qui par nature est plus efficace en principe lorsque le degré des cibles est faible et requiert donc moins de moniteurs). Nous avons constaté que les résultats obtenus pour des degrés moyens inférieurs étaient qualitivement identiques au cas où le degré moyen est 20.

Des simulations comparables ont été réalisées par Lakhina *et al.* [40] pour évaluer les résultats de l'évaluation de la distribution de degré d'un graphe en utilisant la méthode classique consistant à aggréger les plus courts chemins entre un ensemble de 10 moniteurs et un ensemble de 10^3 cibles. Dans ces simulations, les graphes étaient également des graphes aléatoires en loi de Poisson, de taille 10^5 et de degré moyen 15. L'un des résultats les plus édifiants de ces simulations est que la distribution alors mesurée n'est pas une loi de Poisson, mais ressemble davantage à une loi de puissance (*powerlaw*). Ce résultat a même été démontré formellement dans le cas d'un seul moniteur [1]. À défaut d'invalider totalement la méthode classique, ces simulations montrent la pertinence de la démarche : même sur des graphes simulés, l'approche historique est intrinsèquement biaisée. Si notre méthode donne de meilleurs résultats dans des conditions similaires, on peut raisonnablement espérer qu'elle donnera également des meilleurs résultats lors d'une mesure réelle.

Nous avons donc calculé les distributions mesurées par notre méthode sur ces graphes artificiels (**Figure 2.15**). Comme on pouvait s'y attendre, lorsque le nombre de moniteurs est très faible ($|M| = 25$), le degré moyen observé est 15. Ce n'est pas surprenant puisque le nombre de moniteurs est très proche du degré moyen réel, et que donc les nœuds de fort degré (> 25) ne peuvent pas être correctement observés. Il est toutefois intéressant de remarquer que même dans ce cas très défavorable, la *nature* de la distribution est correctement observée, puisqu'on observe une distribution en loi de Poisson, et seul son paramètre (le degré moyen) est sous-évalué, là où la méthode classique observe une distribution de nature différente (loi de puissance) [‡].

En augmentant le nombre de moniteurs, la qualité de l'estimation s'améliore très rapidement. À partir de $|M| = 200$, la distribution observée est indiscernable de la distribution réelle, et il n'y a plus d'amélioration significative en augmentant ce nombre. On remarque en particulier que même pour les cibles de très fort degré (> 30), l'estimation est excellente. Pour les deux graphes (de taille 10^6 et de taille 10^7), le degré maximal observé n'est inférieur que de deux unités au degré maximal réel (43 au lieu de 45 et 46 au lieu de 48). Les nœuds de degré réel > 30 sont observés pour 97% d'entre eux comme ayant un degré > 30 (89% pour les nœuds de degré > 35), ce qui signifie que les nœuds de fort degré ne sont pas observés avec un degré relativement faible. Nous pouvons donc conclure qu'au moins sur de

‡. Voir [40].

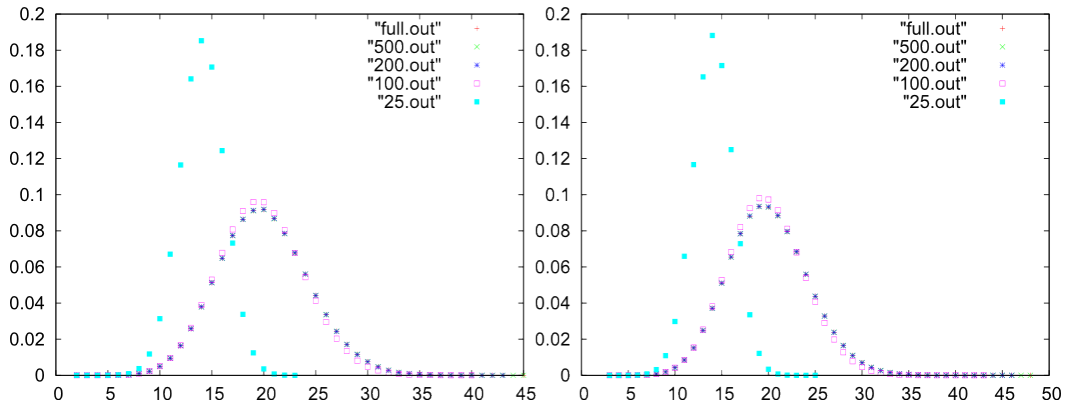


Figure 2.15 – *Pour chacun des graphes (de taille 10^6 (gauche) et 10^7 (droite)) on compare la distribution de degré réelle (*full.out*) avec la distribution de degré mesurée avec différents ensemble de moniteurs, de taile 25, 100, 200, et 500. La nature de la distribution est toujours bien observée comme une loi de Poisson. À partir de 200 moniteurs, et indépendamment de la taille du graphe, la distribution réelle et la distribution mesurée sont presque indiscernables, même pour les forts degrés.*

tels graphes, avoir un ensemble moniteurs répartis aléatoirement de taille modeste ($|M| = 200$, ce qui est comparable à la taille de l'ensemble des moniteurs *Planetlab*) suffit à très bien estimer la distribution de degré.

Cependant, V_3 est *a priori* beaucoup plus vaste, puisque le nombre d'hôtes connectés est plus proche de 10^9 en ordre de grandeur [12]. Mais dans nos simulations, on ne constate aucune différence notable entre les résultats obtenus pour le graphe de taille 10^6 et pour le graphe de taile 10^7. En particulier, le nombre de moniteurs requis pour réaliser une observation de qualité, à part pour certains nœuds de degré extrêmement élevé, apparaît indépendant de la taille du graphe. Ceci suggère que les résultats devraient être de la même qualité pour des graphes arbitrairement grand, sans qu'il soit nécessaire d'augmenter le nombre de moniteurs en conséquence. Nous n'avons pas pu réaliser la preuve formelle de cette hypothèse, mais nous l'avons ultérieurement éprouvée pour d'autres topologies (**Chapitre 3**).

2.8 Mesure avec *Planetlab*

Afin d'attester la faisabilité pratique de la mesure réalisée par $M(T)$ pour un certain ensemble M de moniteurs et un échantillon T de cibles, nous avons implémenté une mesure réelle. Cette mesure n'avait pas pour objectif d'estimer directement la distribution $\hat{d}(V_3)$ mais plutôt d'explorer les problèmes pratiques rencontrés lors d'une mesure et d'y chercher des solutions.

Pour cette mesure, nous avons utilisé un ensemble de moniteurs dans l'ensemble des machines mises à notre disposition par *Planetlab* [13]. L'évaluation de

la qualité de cet ensemble de moniteurs a fait l'objet d'un travail plus approfondi au **Chapitre 3**, en particulier pour déterminer si cet ensemble est bien "suffisamment grand et suffisamment réparti", tel que nous l'avons défini dans ce chapitre. Certains de ces moniteurs sont sujets à des problèmes techniques réguliers (coupures de courant, déconnexions intempestives, filtrage d'ICMP, ...). Comme il est difficile de détecter ces problèmes *a priori*, nous avons décidé d'utiliser *tous* les moniteurs à notre disposition et de procéder à un nettoyage des données *a posteriori*, une étape de toutes manières imposée par la méthode théorique que nous avons décrite en **Section 2.5** et **Section 2.6**.

De manière analogue, nous avons tiré un échantillon aléatoire d'entiers 32 bits, que nous avons expurgé de ceux correspondant à des adresses IP invalides pour obtenir une liste d'adresses IP valides. Nous avons envoyé une sonde ICMP Echo Request à chacun de ces adresses et enregistré les 10^4 premières à répondre.

Munis de cette liste, nous avons effectué une collecte de données sur environ 30h de la manière suivante. Pour chaque moniteur à notre disposition, nous avons généré une permutation aléatoire de la liste des cibles, et lancé traceroute depuis ce moniteur vers chacune des cibles dans l'ordre de cette permutation de la liste. Nous utilisons ici traceroute ICMP, ce qui signifie que les sondes envoyées contiennent le message ICMP Echo Request, auquel les cibles devraient en principe répondre puisqu'elles ont été sélectionnées au départ avec ce message. Cette opération a été répétée 10 fois depuis chaque moniteur, et toutes les sorties de traceroute ont été archivées pour analyse ultrérieure. Les moniteurs n'ayant toujours pas terminé leur mesure à l'issue de ces 30h (moins de 10% des moniteurs qui avaient démarré), pour des raisons diverses (coupures, problèmes de réseau, ...), ont été supprimés du jeu de données.

La première étape d'analyse a consisté à supprimer de notre jeu de données les moniteurs et les cibles considérées comme peu fiables. Un des points méthodologiquement important est que les critères utilisés pour supprimer une cible de notre jeu de données doivent être indépendants *a priori* du degré de cette cible ; dans le cas contraire, ce filtrage pourrait biaiser la distribution issue de la mesure. En n'utilisant uniquement que des critères de filtrages *a priori* indépendants du degré, nous nous assurons que nous n'induisons pas un biais méthodologique. Notre premier filtrage permet d'éliminer de notre jeu de données les moniteurs et les cibles qui ont été temporairement indisponibles pendant notre mesure. Pour ce faire, on compte pour chaque moniteur et pour chaque cible le nombre de sondes traceroute qui ont atteint leur destination (depuis un moniteur ou vers une cible).

Au total, nous avons collecté 36,928,702 sorties de traceroute depuis 544 moniteurs vers 10,000 cibles, et examiné leur répartition parmi les moniteurs (**Figure 2.16**) dont elles étaient à l'origine et les cibles (**Figure 2.17**) dont elles étaient la destination. Nous avons compté pour chaque moniteur le nombre de traceroute ayant atteint leur cible. Nous avons constaté que seuls 68 moniteurs sur 544 (environ 12%) comptaient moins de 71783 traceroute ayant atteint leur

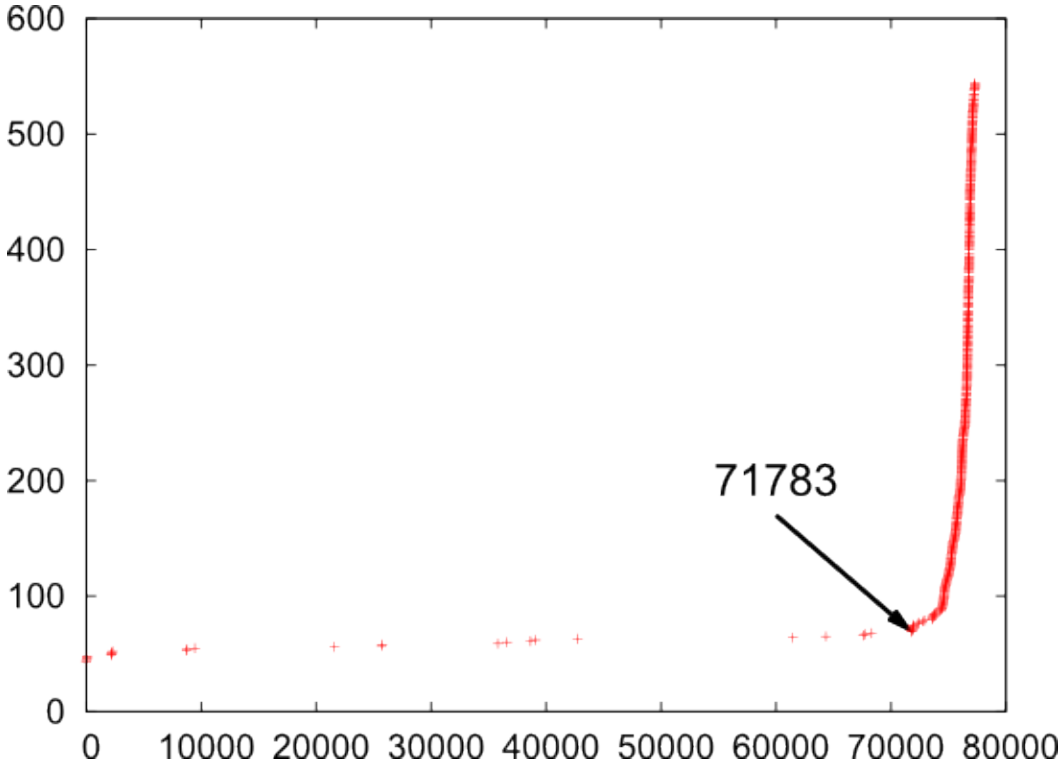


Figure 2.16 – *Pour chaque moniteur, on compte le nombre de* traceroute *qui ont atteint leur cible. Un point* (x, y) *de la distribution cumulative montrée ici se lit :* x moniteurs ont moins de y traceroute ayant atteint leur cible. *En particulier, le point* $(68, 71783)$ *indique qu'au plus 68 moniteurs ont moins de 71783* traceroute *ayant atteint leur cible.*

cible, tandis que le moniteur ayant le *plus* de traceroute ayant atteint leur cible en compte 77,316 (environ 92%). Nous avons donc conservé 88% de notre ensemble initial de moniteurs, qui observent *tous* au moins 92% autant de traceroute que le *meilleur* moniteur du point de vue de ce critère. De manière analogue, nous avons compté pour chaque cible le nombre de traceroute l'ayant atteinte. Il est important de rappeler qu'*a priori*, ce nombre n'a aucune corrélation avec le degré du nœud L3 sous-jacent. Nous avons constaté que 2,314 cibles sur les 9,395 qui ont été atteintes par au moins un traceroute (environ 25%) ont été observées par moins de 3,502 traceroute, à comparer aux 4,760 traceroute ayant atteint la cible qui l'a été le plus (environ 74%). Nous avons donc conservé 7,081 cibles ayant chacune été observée par au moins 3,502 traceroute ayant atteint leur cible. Remarquons que comme chaque moniteur génère au plus 10 sorties de traceroute différentes, chacune de ces cibles a donc été observée par au moins 350 moniteurs.

Avec cette expérimentation, nous avons aussi pu attester de la pertinence de notre méthode de détection des cas où certains moniteurs ont des routes de différentes longueurs vers certaines cibles. Plus précisément, rappelons qu'un couple (m, t) est considéré comme impropre et éliminé de notre jeu de données si, au

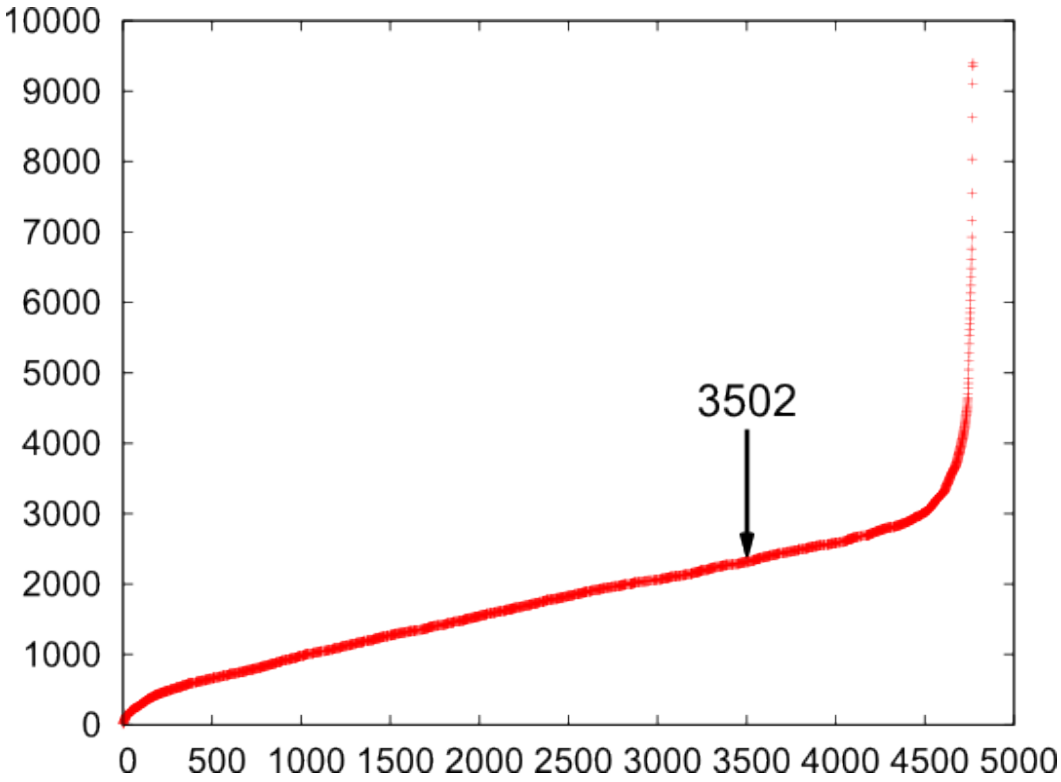


Figure 2.17 – *Pour chaque cible, on compte le nombre de* traceroute *qui ont atteint leur cible. Un point* (x, y) *de la distribution cumulative montrée ici se lit :* x cibles ont moins de y traceroute les ayant atteint. *En particulier, le point (2314, 3502) indique qu'au plus 2314 cibles ont moins de 3502* traceroute *les ayant atteint.*

cours des 10 traceroute depuis m vers t, au moins 2 longueurs différentes ont été observées. Environ 75% de ces couples ont produit des résultats de traceroute de longueurs constantes. Seulement 980 cibles sur les 7,081 restantes à ce stade (environ 14%) sont observées par moins de 350 moniteurs avec des traceroute de longueur constante (**Figure 2.18**). Inversement, toutes les cibles restantes ont été observées par *au moins* 350 moniteurs avec tous leurs traceroute à longueur constante, suggérant que pour les couples (m, t) restants, $m(t)$ est bien une interface d'un voisin de $\bar{t}$. Après suppression de ces couples impropres, il restait 6,101 cibles.

Au cours de cette mesure d'expérimentation nous n'avons pas obtenu d'estimation directe de la distribution de degré. Plusieurs étapes intermédiaires auraient encore été nécessaires, telles que nous les décrirons en **Section 2.9**. Nous avons toutefois, comme illustration du principe, calculé la distribution du *nombre d'interfaces tournées vers le cœur de voisins tournés vers le cœur* de notre ensemble de cibles (dont nous n'avons pas vérifié à ce stade qu'elles sont elles-mêmes dans le cœur) (**Figure 2.19**).

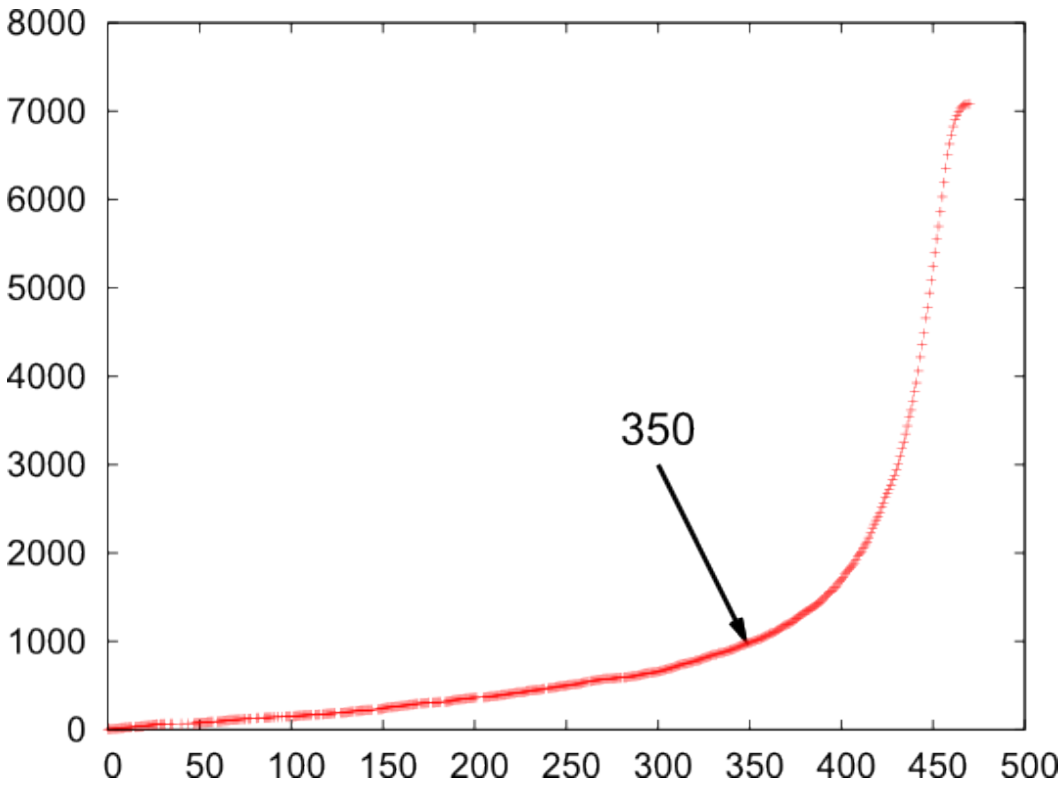


Figure 2.18 – *Pour chaque cible, on compte le nombre de moniteurs dont* tous *les résultats de* traceroute *vers cette cible ont donné la même longueur. Un point* (x, y) *de la distribution cumulative montrée ici se lit : au plus* y *cibles ont été observées par moins de* x *moniteurs avec uniquement des résultats de même longueur. En particulier, le point* $(350, 980)$ *indique qu'au plus 980 cibles ont été observées par moins de 350 moniteurs avec des résultats de même longueur.*

2.9 Protocole complet

Après avoir étudié les aspects à la fois théoriques et pratiques liés à notre méthode de mesure, nous avons pu établir un protocole complet qui permet d'évaluer la distribution de degré des routeurs du cœur au niveau logique.

Soit les paramètres suivants :

- M_0, un ensemble de moniteurs.
- $N \in \mathbb{N}$, un nombre initial de cibles.
- $n \in \mathbb{N}$, un nombre d'itération de traceroute depuis chaque moniteur vers chaque cible.
- $p \in [0, 1]$, un seuil de rejet des moniteurs (tel que décrit en **Section 2.6**).

1. Pour chaque couple (m, m') de moniteurs on effectue traceroute depuis m vers m' et on note $[m, m']$ la liste des adresses apparaissant dans la sortie.

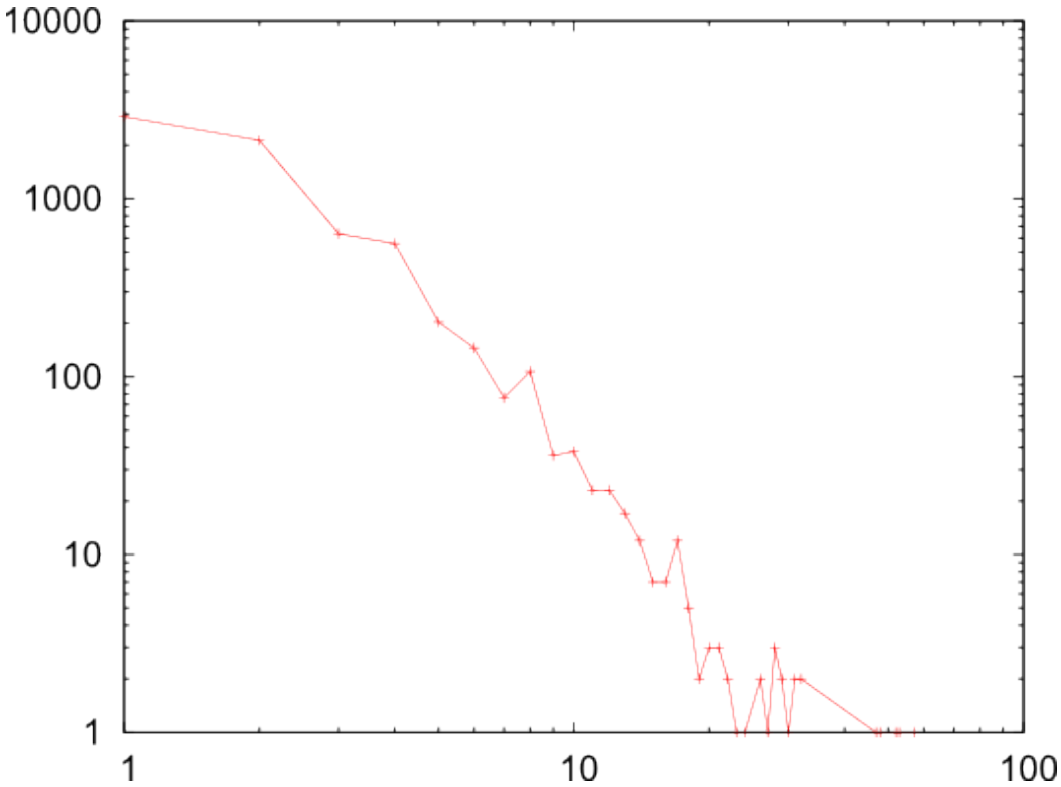


Figure 2.19 – *Distribution du nombre d'interfaces tournées vers le cœur des voisins tournés vers le cœur de notre ensemble de cibles. Un point (x, y) dans cette distribution signifie que y cibles ont été observées avec x interfaces tournées vers le cœur de voisins tournés vers le cœur.*

2. Pour chaque couple (m, m') de moniteurs on effectue un *anti-aliasing* sur chaque adresse dans $[m, m']$ et on obtient $\overline{[m, m']} = \{\overline{v}, v \in [m, m']\}$.
3. On tire des entiers avec un générateur aléatoire uniforme dans $\mathbb{I}$. On envoie un paquet ICMP Echo Request aux adresses correspondantes, et on accepte une adresse si on obtient une réponse ICMP Echo Reply adéquate. On s'arrête lorsque N réponses ont été acceptées, et on note T_0 l'ensemble des adresses correspondantes ($|T_0| = N$).
4. Chaque moniteur $m \in M_0$ répète n fois l'opération :
   - (4.a) Mélanger l'ensemble T_0 pour produire la liste $T_{m,n}$
   - (4.b) Pour chaque $t \in T_{m,n}$, effectuer traceroute depuis m vers t ; on note la réponse de l'avant-dernière sonde $m_n(t)$ et $l_n(m, t)$ la longueur du résultat, lorsque ces informations sont disponibles.

   On note $\hat{m}(t) = \cup_n \{m_n(t)\}$ pour chaque n où $m_n(t)$ est défini (la n-ième itération de traceroute depuis m vers t a obtenu une réponse), et $\hat{m}(t) = \emptyset$ si aucun $m_n(t)$ n'est défini.
5. On supprime des données restantes tous les couples (m, t) tel qu'il existe i, k tels que $l_i(m, t) \neq l_j(m, t)$.

6. On supprime des données collectées les moniteurs ayant un nombre anormalement bas de réponses (au sens de la distribution cumulative) à leurs sondes TRACEROUTE.
7. On supprime des données restantes les cibles ayant un nombre anormalement bas de réponses (au sens de la distribution cumulative) à leurs sondes TRACEROUTE.
8. On supprime des données restantes les cibles t telles qu'il existe au moins un moniteur m tel qu'au moins une proportion supérieure à p des autres moniteurs m' sont tels que $\bar{t} \in \overline{[m, m']}$ (indiquant que m est un descendant de $\bar{t}$ et donc que $\bar{t}$ est possiblement dans le cœur).
9. On note M et T les ensembles de moniteurs et de cibles après filtrage. Pour chaque couple (m, t) dans les données restantes, on effectue un anti-aliasing de $\hat{m}(t)$ pour obtenir $\overline{\hat{m}(t)} = \{\bar{v}, v \in \hat{m}(t)\}$.
10. On estime alors que pour chaque $t \in T$, $V_{C_3}(\bar{t}) = \cup_{m \in M} \overline{\hat{m}(t)}$. En particulier, on estime que $d_{C_3}(\bar{t}) = |\cup_{m \in M} \overline{\hat{m}(t)}|$.
11. On calcule la distribution $d_{M,T}$ définie par $d_{M,T}(k) = |\{\bar{t}, t \in T \; et \; d_{C_3}(\bar{t}) = k\}|$.
12. On en déduit la distribution normalisée $\hat{d}_{M,T}$, une estimation de la distribution de degré des routeurs du cœur.

On peut exprimer la "complexité" du protocole en termes de nombre de TRACEROUTE effectués au total et en nombre de requêtes d'anti-aliasing (qu'on considère ici comme une opération atomique). On supposera que la longueur moyenne des routes empruntées par TRACEROUTE d'un moniteur à un autre est λ, et on note $\tilde{d}$ le degré moyen de T.

- **Nombre de TRACEROUTE effectués :** $O(|M_0|^2)$ pour l'étape (1), $O(n)$ pour l'étape (4.b) répétée N fois donc $O(nN)$ pour l'étape (4), soit un total de $\boldsymbol{O(|M_0|^2 + nN)}$.
- **Nombre d'*anti-aliasing* effectués :** $O(\lambda|M_0|^2)$ pour l'étape (2), $O(\tilde{d}|M_0|)$ pour l'étape (5), soit un total de $\boldsymbol{O(\lambda|M_0|^2 + \tilde{d}|M_0|)}$.

On peut considérer que λ peut être majoré par 20 (une estimation plus fine de λ peut être réalisée pour M_0 donné, lors de l'étape (1)). Le degré moyen $\tilde{d}$ peut être majoré par 20. Cette approche de la complexité du protocole nous permet d'évaluer la charge réseau que l'on déploie lors de la mesure. C'est une considération importante dans la mesure où certains réseaux sont hostiles au traffic ICMP utilisé par TRACEROUTE, et limitent parfois durement son utilisation.

Le paramètre sur lequel il est le plus difficile d'agir est en général M_0, mais la **Section 2.7** suggère qu'un nombre limité de moniteurs bien distribués ($|M_0|$ de l'ordre de 10^2) permet d'obtenir une très bonne estimation. La question de la bonne distribution sera explorée plus en profondeur dans le **Chapitre 3**.

N sera bien entendu choisi le plus grand possible, la limite pratique étant la vitesse de sélection des cibles et le taux de réponse aux sondes ICMP Echo Reply. Pour notre expérimentation pratique **Section 2.8**, nous avons réalisé la sélection de cette liste de cibles depuis un seul hôte, mais distribuer ce processus sur l'ensemble des moniteurs M_0 permet d'accélérer l'efficacité et donc d'obtenir un ensemble T_0 d'autant plus grand, pour peu que l'on soit capable d'effectuer les mesures ultérieures dans un temps raisonnable.

n est un paramètre de la mesure qui dépend de l'estimation qu'on fait en **Section 2.6** pour la probabilité de détecter sans faux négatif[§] les couples (m, t) tels que traceroute est susceptible de fournir des résultats de longueur variable. Plus n est élevé, plus cette probabilité est proche de 1, mais n a un impact linéaire sur le nombre de traceroute à effectuer et ne peut donc être trop élevé. Une valeur de $n = 10$ nous a semblé un compromis satisfaisant comme expliqué en **Section 2.6**, mais il pourrait être plus faible avec une estimation plus précise de cette probabilité.

p est un paramètre de la mesure qui dépend de l'estimation qu'on fait en **Section 2.5** pour la probabilité de détecter sans faux négatifs[¶] qu'une cible est dans le bord. Plus p est élevé, plus le risque est faible, mais il augmente alors le risque de faux positif[‖] et diminue la taille de l'ensemble final T et donc la qualité de l'estimation $\hat{d}$ finale. Il faut donc déterminer p suffisamment élevé pour écarter le risque de faux négatif tout en conservant T suffisamment grand pour que la qualité de l'estimation $\hat{d}$ soit satisfaisante. En pratique, $p = \frac{1}{2}$ nous a semblé un bon compromis, comme expliqué en **Section 2.5**.

2.10 Limites de l'approche

Nous avons mis au point un protocole clair permettant de mesurer rigoureusement la distribution de degré dans le cœur des routeurs du cœur d'Internet. La fiabilité de ce résultat est sans commune mesure avec les approches historiques d'aggrégation de résultats de traceroute en cartes. Cependant, il possède plusieurs limites à la fois théoriques et pratiques.

Tout d'abord, il repose sur un usage intensif de sondes ICMP et sur la bonne implémentation par les cibles et leurs voisins des standards. Cette hypothèse est également réalisée par les approches historiques, mais elle fait l'impasse sur une réalité objective : de très nombreux hôtes bloquent totalement ou partiellement les paquets ICMP ou ne sont pas coopératifs à leur égard. Il suffit d'un seul hôte sur une route entre un moniteur et une cible, dans un sens ou dans l'autre, qui bloque le traffic ICMP, pour obscurcir totalement les mesures entre ce moniteur et cette cible.

§. C'est à dire supposer à tort que (m, t) fournit toujours des résultats de longueur constante.

¶. C'est à dire que supposer à tort qu'aucun moniteur $\overline{m}$ n'est un descendant de $A(\overline{t})$ et donc d'écarter le risque que $\overline{t}$ soit dans le bord.

‖. C'est à dire de supposer à tort que $\overline{t}$ est peut-être dans le bord et donc de l'écarter.

En conséquence, un grand nombre de résultats de TRACEROUTE sont incomplets et certains hôtes sont totalement indiscernables entre eux par les sondes ICMP. Pire encore, certains hôtes pratiquent des configurations très singulières qui leur font renvoyer des informations probablement fausses.

Notre protocole repose aussi à plusieurs endroits clé sur une primitive de mesure fiable pour réaliser l'*anti-aliasing* de plusieurs adresses IP. Ces primitives existent [30, 39] mais, comme TRACEROUTE, elles sont souvent limitées par le filtrage du traffic de diagnostic (comme ICMP), ou par des configurations singulières [31]. S'il est raisonnable de supposer qu'une cible qui a répondu à ICMP ECHO REQUEST sera également coopérative à l'égard de l'*anti-aliasing*, il est moins clair que ses voisins le seront également. Si ils ne le sont pas, alors on pourra soit sous-évaluer le degré d'une cible (en n'observant pas l'un de ses voisins), soit le sur-évaluer (en comptant plusieurs voisins pour plusieurs interfaces d'un même voisin). En revanche, ce n'est pas un problème pour les adresses apparaissant dans les $[m, m']$ puisqu'on cherche juste à identifier si les cibles (réceptives à l'*anti-aliasing*) s'y trouvent. Le fait que les autres adresses n'y répondent pas n'a pas d'impact négatif sur notre protocole.

D'une manière générale, la charge de réseau opérée sur une cible et son entourage proche peut être assez élevé. Si l'ensemble de moniteurs est grand, ce que l'on souhaite, alors il faut prendre le soin d'étaler suffisamment les mesures dans le temps pour ne pas risquer de surcharger les cibles et leur voisinage, mais suffisamment peu pour éviter les phénomènes de dynamique des routes. On peut en particulier craindre un biais spécifique : que la capacité d'une cible à encaisser un traffic ICMP intensif soit indirectement relié à son degré. En effet, on peut imaginer qu'une cible de fort degré corresponde à une entité physique plus puissante en termes de capacité de calcul et de stockage, et donc soit moins susceptible de limiter le taux de paquets ICMP sur une courte durée. Ceci biaiserait la distribution finale $\hat{d}$ en faveur des nœuds de fort degré.

Remarquons que bien que celà ne soit pas une limite spécifique à notre approche, la primitive de mesure de bas niveau **Section 2.2** repose sur le décrément du TTL des paquets IP porteurs des sondes ICMP. Or, certains réseaux, en particulier ceux qui pratiquent le *tunneling* ou le *multi-homing*, forment des aberrations topologiques au niveau de G_3, qu'il est très difficile de détecter à l'aide de sondes ICMP. Ce problème n'est pas directement lié à notre méthode, mais plutôt à l'interprétation du graphe G_3, qui ne considère *que* le réseau Internet, et pas les réseaux locaux qui y sont connectés à travers des passerelles mais qui reposent également sur le protocole IP. Dans le cas du *tunneling* [62, 50], il s'agit même d'utiliser des paquets IP locaux pour porter des paquets IP Internet, ce qui complexifie l'interprétation du *hop* et du décrément TTL.

2.11 Conclusion

Le travail présenté ici retrace notre démarche pour appliquer le principe de mesure orientée propriété à la topologie d'Internet historiquement la plus étudiée : celle de la topologie logique, observée à travers l'outil de mesure TRACEROUTE. En détaillant précisément le fonctionnement de TRACEROUTE et en explicitant les hypothèses sur lesquelles il repose, nous avons pu déduire une interprétation rigoureuse qui nous a servi à établir une primitive de mesure fiable et dans les résultats de laquelle nous pouvons avoir une grande confiance. Cette primitive de mesure nous a permis d'établir un protocole de mesure complet dont la portée est clairement établie : estimer la distribution de degré dans le cœur de la topologie logique. Nous en avons validé le principe grâce à des simulations, et nous avons réalisé une mesure réelle pour en éprouver la pertinence opérationelle.

En réalisant cette mesure réelle, nous avons pu déceler des faiblesses pratiques importantes qui en limitent l'exploitation. La lourde charge réseau exigée par l'*anti-aliasing*, en particulier, a représenté un obstacle pratique important. L'omniprésence du filtrage du trafic ICMP est l'autre obstacle majeur de notre méthode de mesure. Même si à l'égard de ces deux limites, notre méthode reste beaucoup plus fiable que d'autres approches historiques, elle ne nous a pas permis de tirer des conclusions claires sur la distribution de degré réelle dans le cœur de la topologie logique.

En revanche, notre démarche nous a permis de mieux comprendre les mécanismes aux niveaux L2 et L3 qui étaient pertinents dans la réalisation d'une mesure orientée propriété. C'est en cherchant à améliorer le procédé d'*anti-aliasing* pour mesurer la topologie logique que notre intérêt s'est porté sur une interprétation plus rigoureuse des méthodes historiques, et nous a motivé à porter notre attention sur la mesure de la topologie physique, étudiée dans les chapitres suivants.

Chapitre 3

Mesure de la topologie physique

La topologie physique d'Internet correspond peut-être davantage à l'intuition qu'on se fait du réseau lorsque l'on n'est pas familier du protocole IP. Elle représente la capacité physique des entités connectées à échanger des informations. Cette capacité physique est ensuite exploitée au niveau logique, et les deux topologies sont intrinsèquement reliées, mais la topologie logique abstrait totalement l'existence d'entités capables de manipuler des paquets sans être capable d'en émettre ou d'en recevoir par elles-mêmes, ainsi que le détail des interfaces physiques mobilisées par des entités logiques. Si on peut assez directement déduire une topologique logique d'une topologie physique, la réciproque est fausse. Ceci est d'autant plus important que la configuration physique d'un hôte logique peut avoir un grand impact sur sa performance. Dans le cas d'un routeur, par exemple, son efficacité dépend souvent grandement de la performance individuelle de ses interfaces physiques, et non pas uniquement de la performance de sa couche logique.

La topologie physique a historiquement suscité moins d'attention que la topologie logique, principalement car les approches historiques reposent sur l'exploitation de traceroute qui opère au niveau logique. Pourtant, nos travaux préliminaires ont mis en évidence les limites d'une approche au niveau logique. À l'inverse, nos travaux pour tenter de dépasser ces limites, et en particulier sur l'anti-aliasing, nous ont mis sur la voie d'une approche différente. Au lieu de corriger l'approche logique en mobilisant des caractéristiques de la topologie physique, nous avons décidé d'adapter notre méthode pour qu'elle opère directement au niveau physique. En plus d'éviter beaucoup des problèmes liés à l'approche logique, elle permet de réaliser une observation d'un objet plus précis.

Nous présentons ici notre méthode de mesure de la topologique physique. Elle repose sur une primitive de mesure de bas niveau UDP Ping inspirée des méthodes d'anti-aliasing qui sonde une cible depuis un moniteur (**Section 3.1**). Cette primitive de mesure de bas niveau est mobilisée de manière distribuée (depuis un ensemble de moniteurs vers une cible) pour constituer une primitive de haut niveau (**Section 3.2**), pour évaluer le degré au niveau physique d'une cible. Grâce à une méthode d'échantillonage sans biais nous en déduisons une évaluation de la distribution de degré physique des routeurs du cœur (**Section 3.3**). Le principe est validé par de nouvelles simulations (**Section 3.4**). La qualité de cette estimation repose sur des caractéristiques de l'ensemble de moniteurs utilisé, dont nous avons

approfondi l'étude (**Section 3.5**). Nous avons attesté de la faisabilité expérimentale de notre méthode en réalisant une nouvelle mesure réelle (**Section 3.6**). Avec les retours de cette expérimentation, nous avons pu établir un protocole complet (**Section 3.7**), et effectuer une validation, à la fois de nos heuristiques d'évaluation de la qualité d'un ensemble de moniteurs, et réinjecter les résultats dans notre modèle de simulations pour attester de leur pertinence (**Section 3.8**). À l'aide de ces résultats, nous avons pu établir les points forts et les limites de notre approche et conclure (**Section 3.9**).

3.1 Primitive de mesure de bas niveau basée sur UDP Ping

Un hôte du réseau est un nœud appartenant à la fois à L3 et L2, qui est capable de recevoir et de fabriquer des nouveaux paquets. Un hôte se distingue des autres nœuds (non-hôtes, tels que les switchs) de L3 et L2 par son comportement quand il reçoit un paquet IP [53]. Si l'en-tête *Destination Address* appartient à la liste des adresses que cet hôte identifie comme les siennes, alors il remonte ce paquet à la couche d'abstraction supérieure. Sinon, il examine l'en-tête *Time-to-Live* (ou TTL) du paquet. Si sa valeur est zéro, l'hôte émet un nouveau paquet ICMP *Time Exceeded* en direction de l'adresse indiquée dans l'en-tête *Source Address*. Si sa valeur est supérieure à zéro, l'hôte émet un nouveau paquet IP avec le même contenu que le paquet initial, mais dont l'en-tête TTL est décrémentée de 1 (et l'en-tête *Checksum* mise à jour).

Soit $\overline{m}$ un hôte que l'on appelle un *moniteur* muni d'une unique interface désignée par son adresse IP m, depuis lequel nous sommes capable d'exécuter un programme avec un accès privilégié. Soit $\overline{t}$ un hôte que l'on appelle une *cible* et dont on connaît l'une des adresses IP t. Supposons que l'on envoie un paquet UDP depuis $\overline{m}$ vers t, contenant un message arbitraire, et à destination d'un port non utilisé p. Lorsque $\overline{t}$ réceptionne ce paquet, sa couche L3 reconnaît le paquet comme lui étant destiné et le transmet à la couche supérieure — en l'occurence, UDP. À ce niveau, l'hôte identifie que le paquet est à destination d'un port non utilisé, et émet un paquet ICMP*Destination Unreachable (Code 3/Port Unreachable)* dont la destination est m[6, 5]. Le paquet ICMP émis est encapsulé dans un paquet IP, dont l'en-tête *Source Address* est en principe l'interface d'adresse IP i choisie par l'hôte pour envoyer un paquet à destination de m. Notons en particulier que cette interface i *peut* être différente de t, bien qu'elle appartienne également à l'ensemble $\overline{t}$ des interfaces de la cible (**Figure 3.1**). Cette technique est inspirée de la technique d'*anti-aliasing* utilisée par exemple par *iffinder* [9, 34, 48, 26].

Nous nommons UDP Ping l'outil qui permet d'envoyer des paquets UDP vers un port non utilisé d'une adresse cible t et de réceptionner les paquets d'erreur

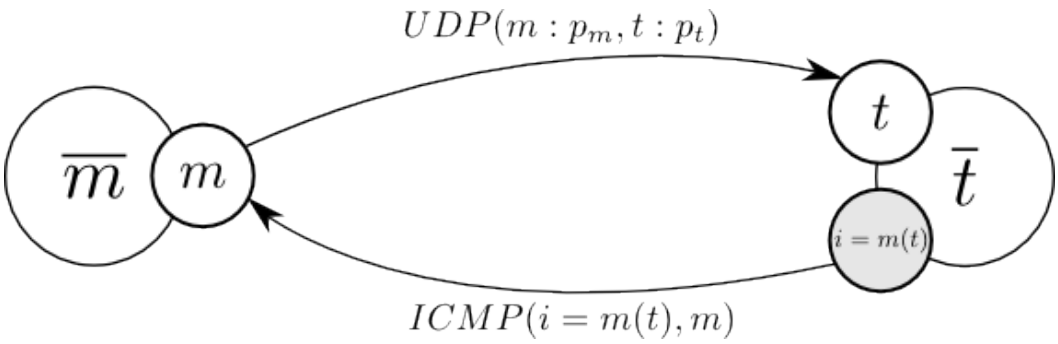


FIGURE 3.1 – *Un hôte moniteur $\overline{m}$ utilise son unique interface m pour émettre un paquet* UDP *avec un port source quelconque p_m et un port destination arbitraire non utilisé p_t vers l'adresse* IP *t (en haut). À la réception, l'hôte cible $\overline{t}$ qui possède l'adresse* IP *t reconnaît un port non utilisé et émet un paquet* ICMP *d'erreur en direction de m. Pour ce faire, elle utilise une certaine interface $i = m(t)$.*

ICMP et de reconnaître les réponses i. Il peut être exécuté depuis n'importe quel hôte avec des privilèges permettant d'envoyer des paquets UDP et de réceptionner les paquets ICMP (ce dernier privilège étant un privilège *root* sur la plupart des systèmes). Dans ce chapitre, nous utiliserions ainsi la notation suivante :

Définition 25 (Observation d'une cible depuis un moniteur)**.** *Soit i le résultat de* UDP PING *depuis m vers t. Si ce résultat est une adresse (c'est à dire qu'une réponse a été obtenue après l'envoi de la sonde), on note $m(t) = i$ et on l'appelle* observation de t depuis m.

Par construction, $m(t) \in \overline{t}$, et plus précisément, $m(t)$ est l'*interface choisie par l'hôte $\overline{t}$ pour émettre à destination de m*. Ainsi, $m(t)$ dépend de m (d'où cette notation). Si $\overline{t}$ est un routeur, en particulier, alors $m(t)$ sera choisie en fonction de la politique de routage de $\overline{t}$. Par exemple, $m(t)$ pourrait être l'interface de $\overline{t}$ qui la relie au niveau *physique* au prochain *hop* dans le plus court chemin au niveau *logique* depuis $\overline{t}$ vers $\overline{m}$ (**Figure 3.2**).

Remarquons que contrairement à notre primitive de mesure basée sur TRACEROUTE (2.2), la qualité de la réponse est binaire : soit le moniteur reçoit une réponse de la cible, et on note cette réponse $m(t)$, soit le moniteur ne reçoit pas de réponse (si la sonde n'a pas atteint la cible, ou que la cible n'a pas répondu, ou que la réponse n'a pas atteint le moniteur). En particulier, l'ensemble $\mathbb{M}(\overline{t}) = \cup_{t' \in \overline{t}} \mathbb{M}(t)$ des moniteurs observant une cible est exactement l'ensemble des moniteurs ayant obtenu une réponse à leur sonde.

Pour une interface $t \in \overline{t}$ donnée d'une cible donnée, on considère l'ensemble des hôtes tels que, si on exécutait UDP PING depuis cet hôte vers $\overline{t}$, on obtiendrait t.

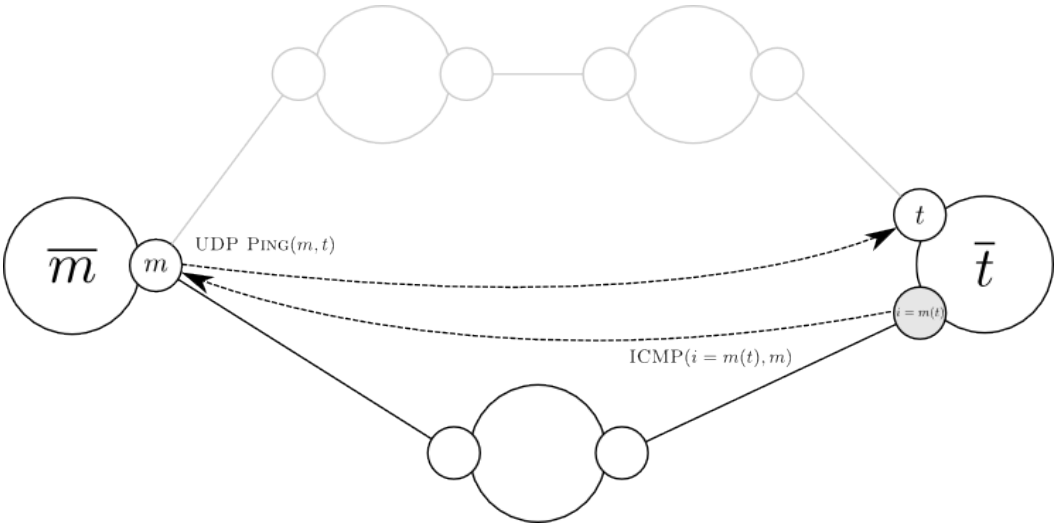


Figure 3.2 – *Un moniteur m envoie une sonde* UDP Ping *vers t. La cible $\overline{t}$ répond en utilisant une interface $m(t)$ qui se trouve être différente de t, et qui correspond à ici à l'interface vers le premier* hop *du plus court chemin de retour vers m.*

Définition 26 (Nœuds capables d'observer une interface donnée d'une cible)**.** *Soit $\overline{t}$ une cible et $t' \in \overline{t}$ une interface de cette cible (égale ou non à t). On note $\mathbb{M}(t')$ l'ensemble $\{m, m(t) = t'\}$ et on l'appelle* ensemble des nœuds capables d'observer l'interface t'.

Un certain type d'interfaces est particulièrement difficile à observer. Soit une cible $\overline{t}$ munie d'une certaine interface t' telle que t' relie $\overline{t}$ à un ensemble de nœuds qui sont tous dans le bord. Alors pour observer t' en utilisant des sondes UDP, il faut nécessairement disposer d'un moniteur qui est précisément dans cet ensemble ; en particulier, un tel moniteur est nécessairement dans le bord.

On définit donc :

Définition 27 (Interface tournée vers le bord (resp. vers le cœur))**.** *Soit $t' \in \overline{t}$ une interface telle que t' est présente sur tous les chemins entre n'importe quel noeud de (t') et le cœur. On dit alors que t' est* tournée vers le bord. *Inversement, si t' n'est pas* tournée vers le bord, *on dit qu'elle est* tournée vers le cœur, *et dans ce cas il existe au moins un moniteur dans le cœur capable d'observer t'.*

Autrement dit, si t' est une interface tournée de $\overline{t}$ vers le bord, alors un moniteur m ne peut observer t' avec UDP Ping que si m dans l'arbre relié au cœur en passant par t'.

Cette section nous a permis de définir et d'interpréter la primitive de mesure $m \in \mathbb{M}(\overline{t}) \mapsto m(t) \in \mathbb{I}$. Pour la suite, nous appelerons cette primitive notre *primitive de mesure de bas niveau basée sur* UDP Ping.

3.2 Primitive de mesure de haut niveau basée sur UDP Ping

Nous avons vu que UDP Ping permet d'obtenir, depuis un moniteur m, une interface d'une cible $\overline{t}$ désignée par l'une de ses adresses IP t, et que cette interface $m(t)$ dépend du moniteur m. Des sondes UDP de cette nature ont déjà été utilisées pour résoudre le problème de l'*anti-aliasing* : étant données deux adresses t_1 et t_2 d'une cible, il s'agit de déterminer si t_1 et t_2 appartiennent au même hôte, c'est à dire si $\overline{t_1} = \overline{t_2}$. Si l'on exécute UDP Ping (ou *iffinder*) depuis un moniteur m vers t_1 et t_2 et que l'on obtient le même résultat t (qui peut être ou non égal à t_1 ou t_2), alors on conclut que t_1, t_2 et t sont des *alias* d'un même hôte $\overline{t} = \overline{t_1} = \overline{t_2}$. Nous nous intéressons ici au problème inverse : étant donnée une cible $\overline{t}$ désignée par l'une de ses adresses IP t, déterminer la liste de toutes ses interfaces.

Considérons, comme illustré en **Figure 3.3**, le cas simplifié d'une cible $\overline{t}$ désignée par une adresse IP t, ayant exactement deux interfaces t et t', et deux moniteurs $m \in D(\overline{t}, t)$ et $m' \in M(\overline{t}, t')$. Alors $m(t) = t$ et $m'(t) = t'$. En particulier, puisque chaque interface de $\overline{t}$ est observée par l'un des deux moniteurs, alors $\{m(t), m'(t)\} = \overline{t}$. Autrement dit, si pour chaque interface de $\overline{t}$ on dispose d'un moniteur qui permet d'observer cette interface, on obtient exactement la liste des interfaces de $\overline{t}$ (**Figure 3.3**).

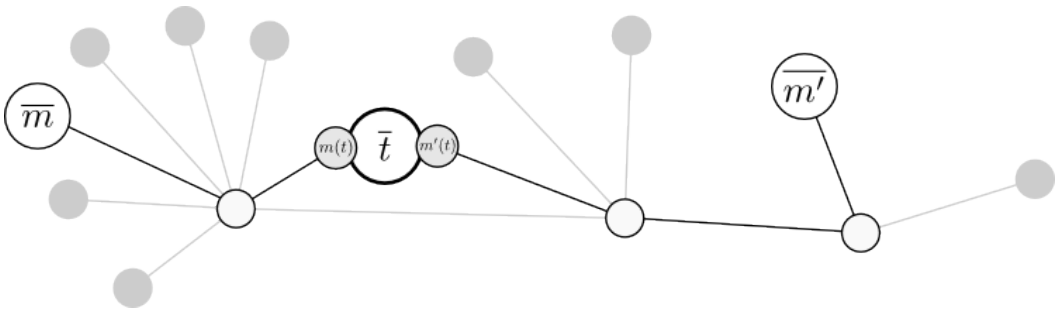


Figure 3.3 – *La cible $\overline{t}$ dispose d'exactement deux interfaces qui sont utilisées chacune pour adresser deux parties disjointes du réseau. Si l'on dispose d'un moniteur dans chacune de ces parties disjointes, alors on peut observer les deux interfaces de $\overline{t}$.*

On définit, plus généralement :

Définition 28 (Interfaces d'une cible observées par un ensemble de moniteurs)**.** *Soit M un ensemble de moniteurs. On note $M(t)$ et on appelle* ensemble des interfaces observées par M pour la cible t *l'ensemble $M(t) = \{m(t), m \in M\} \subset \overline{t}$.*

L'objectif est alors d'obtenir $M(t) \supset \overline{t}$ pour obtenir l'égalité. Soit une cible $\overline{t} = \{t_0, \ldots, t_k\}$ désignée par une adresse IP t, et un ensemble de moniteurs M. Alors si pour chaque $j \in [\![0, k]\!]$, il existe au moins un certain $m_j \in M$ tel que

$m_j \in \mathbb{M}(\bar{t}, t_j)$, alors $M(t) = \bar{t}$. Autrement dit, si l'ensemble M est correctement réparti par rapport aux interfaces de $\bar{t}$, alors en exécutant UDP Ping depuis chacun des moniteurs de M, on obtient la liste complète des interfaces $\bar{t}$.

La difficulté repose donc sur l'obtention d'un ensemble de moniteurs M adéquat, capable d'observer toutes les interfaces d'une cible. Cependant, cette approche se révèle peu exploitable en pratique dans le cas d'une cible quelconque, à cause des interfaces tournées vers le bord. En effet, nous avons vu précédemment que pour être capable d'observer une interface tournée vers le bord avec UDP Ping, il faut disposer d'un moniteur qui se trouve précisément dans un arbre du bord qui est relié au cœur par cette interface. Si nous souhaitons réaliser cette condition pour n'importe quelle cible, il faudrait avoir un ensemble de moniteurs dans chaque sous-arbre du bord, c'est à dire tous les nœuds du bord…

On va donc s'intéresser uniquement aux interfaces dans le cœur d'une cible. Si $\bar{t}$ est dans le bord, alors elle ne possède qu'une seule interface dans le cœur. Inversement, si une cible est dans le cœur, alors elle possède au moins deux interfaces dans le cœur (**Figure 3.4**). Il existe un cas particulier notable, où un nœud du cœur possède au moins une interface dans le bord. On appelle ce type de nœuds un nœud de *branchement*, et tout arbre du bord est relié au cœur par un unique nœud de branchement. Cette notion sera mobilisée pour identifier certaines configurations (**Section 3.5**).

Dans le cas où nous nous intéressons à des routeurs dans le cœur, nous pouvons espérer que si nous disposons d'un ensemble suffisamment grand et bien réparti de moniteurs, alors nous pouvons bien observer toutes ses interfaces dans le cœur. En effet, si des interfaces ne sont pas observées par un tel ensemble, c'est que ces interfaces ne sont probablemment pas sollicitées dans le fonctionnement normal du routeur, et peuvent être négligées du point de vue de la topologie. L'hypothèse selon laquelle un ensemble de moniteurs de taille raisonnable suffit à observer toutes les interfaces dans le cœur d'une cible donnée est validée plus précisément à l'aide de simulations (**Section 3.4**).

Pour une certaine cible $t \in \mathbb{I}$, nous considérons alors $M \subset \mathbb{M}(t) \mapsto M(t) \subset \mathbb{I}$ notre *primitive de mesure de haut niveau basée sur* UDP Ping.

3.3 Echantillonage rigoureux dans le cœur

Nous avons vu que sous réserve de disposer d'un ensemble de moniteurs M suffisamment grand et suffisamment bien réparti, nous étions capables d'obtenir la liste des interfaces d'une cible t appartenant à un routeur $\bar{t}$ dans le cœur. En particulier, $|M(t)| = |\bar{t}|$, le degré physique de $\bar{t}$. Si nous étions capables d'obtenir un ensemble de cibles T uniformément réparties dans le cœur, en calculant la distribution des degrés observés depuis M vers T, alors nous obtiendrions une

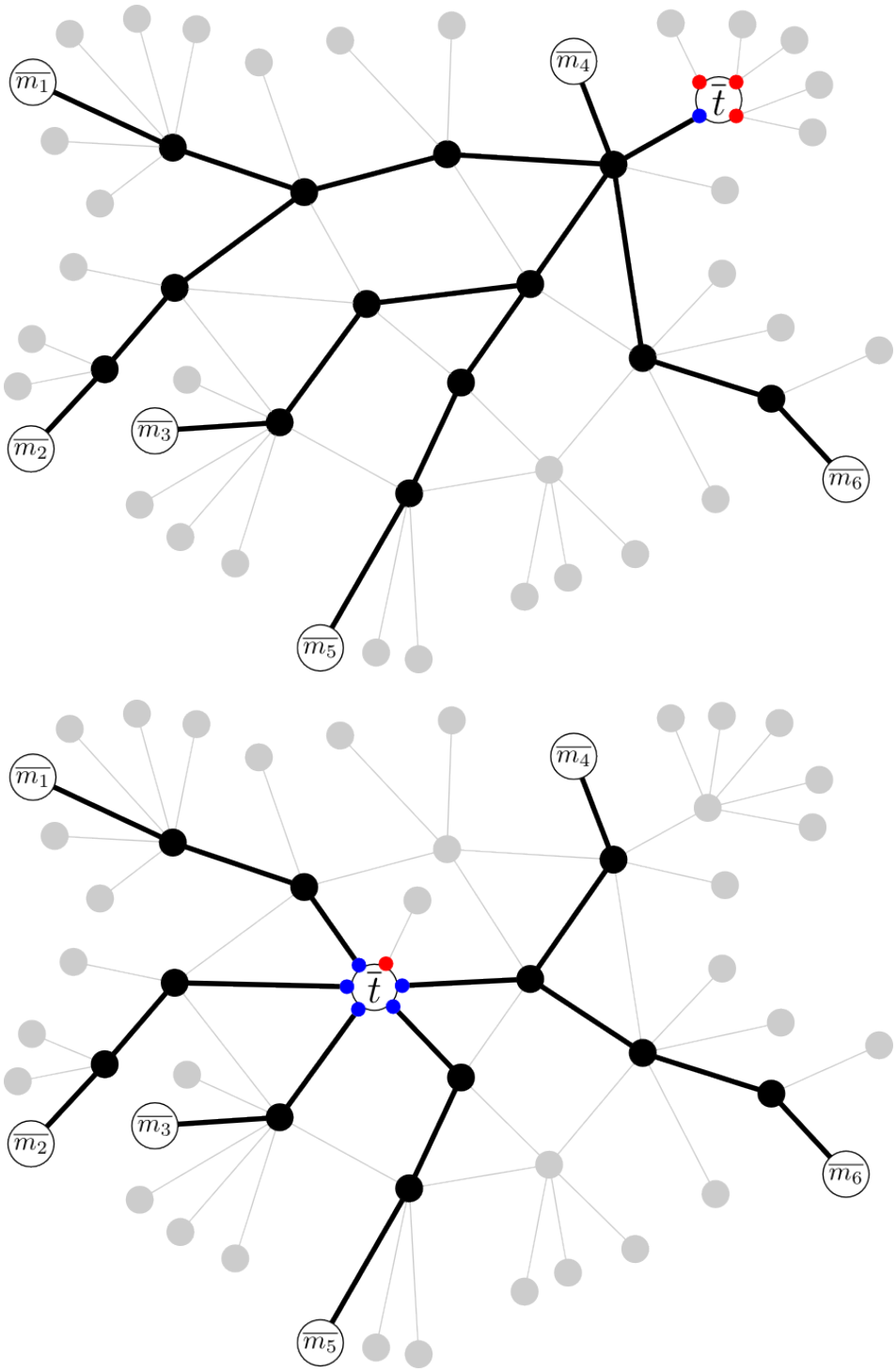


Figure 3.4 – *En haut (a) : la cible $\overline{t}$ est dans le bord, et possède une seule interface tournée vers le cœur (en bleu), qu'on observe avec un ensemble de moniteurs bien réparti. En bas (b) : la cible $\overline{t}$ est dans le cœur, et on observe toutes ses interfaces tournées vers le cœur (en bleu).*

estimation de la distribution des degrés physiques du cœur, et il suffirait d'avoir $|T|$ assez grand pour garantir la représentativité de cet échantillon.

Malheureusement, nous ne savons pas tirer un ensemble uniformément aléatoire d'hôtes, et encore moins cibler spécifiquement les routeurs du cœur tout en conservant l'uniformité. En revanche, nous sommes capables de tirer uniformément un ensemble d'*adresses* IP, qui ne sont autres que des nombres entiers particuliers. À partir d'un tel ensemble, nous pouvons réaliser des transformations (filtrages) qui, chacune, conservent l'uniformité de l'échantillon en ce qu'elles sont indépendantes du degré des hôtes. Nous pouvons trouver une série de transformations uniformes qui réduise notre ensemble à un ensemble uniforme d'adresses IP d'interfaces dans le cœur de routeurs du cœur, et qui répondent correctement aux sondes UDP Ping (**Section 3.3.1**). Il ne s'agit pas encore d'un ensemble uniforme de *routeurs*, mais d'un ensemble d'*adresses*. Cependant, nous sommes capables de déduire la distribution de degrés des routeurs du cœur à partir de la distribution de degrés de cet échantillon, par une méthode de correction du biais (**Section 3.3.2**).

3.3.1 Echantillonage d'adresses IP de routeurs du cœur

On part initialement d'un ensemble T_0 d'entiers de 32 bits choisis uniformément aléatoirement, et on applique une série de transformations uniformes (φ_k), indépendantes du degré des hôtes sous-jacents, pour construire des ensembles filtré définis par $T_{n+1} = \varphi_n(T_n) \subset T_n$. Chacune des transformations réalisées est indépendante des autres et peut, théoriquement, être opérée dans n'importe quel ordre. Cependant, pour des considérations opérationelles, certaines sont plus commodes à exécuter le plus tôt possible, par exemple parce qu'elles réduisent le nombre de calculs à effectuer pour les filtrages les plus lourds qu'on réalise ultérieurement.

La première transformation φ_1 écarte les adresses qui (a) ne correspondent pas à des adresses IP routables [33] ou qui (b) sont routables mais pour lesquelles les moniteurs de notre ensemble n'obtiennent pas de réponses valides. Nous supposons également qu'il n'y a pas de raison intrinsèque pour que les hôtes de faible degré se comportent différemment des hôtes de fort degré à ce titre, et donc que cette transformation est indépendante du degré.

La deuxième transformation φ_2 vise à éliminer les hôtes qui ne sont pas des routeurs du cœur, c'est à dire les routeurs du bord. Les routeurs du bord sont situés dans un arbre, et par conséquent, ils n'ont qu'une seule interface tournée vers le cœur. Soit $\bar{t}$ un routeur du bord. Pour chaque moniteur m, il y a trois cas possibles : soit m n'obtient pas de réponse, soit m est dans le même arbre que $\bar{t}$ et il observe alors une interface tournée vers le bord de $\bar{t}$, soit m n'est pas dans le même arbre et il observe l'unique interface de $\bar{t}$ qui est tournée vers le cœur. Nous verrons en **Section 3.5.1** que nous sommes capables de caractériser ces cas, c'est à dire

détecter si un moniteur se trouve dans le même arbre qu'une cible (**Figure 3.5**, haut). Réciproquement, si au moins deux interfaces distinctes t et t' sont observées par des moniteurs qui ne sont pas dans le même arbre que la cible, alors $\bar{t}$ ne peut être dans le bord (**Figure 3.5**, bas). Pour détecter un hôte qui n'est pas un routeur du cœur, il suffit donc d'écarter les adresses t telles que $M^*(t)$, qui est égal à $M(t)$ privé des éventuelles interfaces observées par des moniteurs se trouvant dans le même arbre, soit réduit à 0 (aucune observation) ou 1 interface (l'unique interface dans le cœur de $\bar{t}$). Dans tous les cas, si l'on supprime toutes les adresses $\bar{t}$ telles que $|M^*(t)| \leq 1$, alors toutes les adresses restantes appartiennent nécessairement à des routeurs du cœur. Comme cette configuration n'est pas reliée au degré, cette transformation est également indépendante du degré.

La dernière transformation φ_3 vise à éliminer les adresses qui correspondent à des interfaces de routeur du cœur mais qui ne sont pas elles-mêmes des interfaces dans le cœur. Sous l'hypothèse que M soit suffisamment grand et suffisamment bien distribué, alors $M(t)$ est exactement l'ensemble des interfaces de $\bar{t}$ dans le cœur. Il suffit donc de tester si $t \in M(t)$ et supprimer t si $t \notin M(t)$, opération qui est toujours indépendante du degré de $\bar{t}$.

On note $\varphi = \varphi_3 \circ \varphi_2 \circ \varphi_1$ et $T = \varphi(T_0)$.

À ce stade, T est un ensemble uniformément aléatoire (par rapport à la distribution de degrés) d'adresses d'interfaces dans le cœur de routeurs du cœur.

3.3.2 Correction du biais

Nous suppons que nous disposons d'un ensemble T uniformément aléatoire d'adresses d'interfaces dans le cœur de routeurs du cœur, et nous cherchons à obtenir une estimation de la distribution des degrés physiques des routeurs du cœur.

Remarquons tout d'abord que nous n'avons pas ici échantillonné des *routeurs* (hôtes), mais des *adresses* IP. Or, chaque routeur du cœur possède autant de chances d'être sélectionné par adresse qu'il possède d'adresses, c'est à dire que plus son degré est élevé, plus il a de chances, proportionellement à son degré (son nombre d'interfaces dans le cœur), d'être tiré par son adresse. Notons p_k la fraction réelle de routeurs de degrés k dans la distribution de degrés, et p'_k la fraction des adresses appartenant à des routeurs du cœur de degré k. Alors, d'après notre remarque, $p'_k \sim k \cdot p_k$. Le facteur linéaire est simplement un facteur de normalisation assurant que $\sum_k p_k = \sum_k p'_k = 1$, et on obtient donc :

$$p_k = \frac{p'_k}{k} \cdot \frac{1}{\sum_i \frac{p'_i}{i}}$$

Pour obtenir une estimation de p_k à partir d'un échantillon T, il suffit donc de mesurer $p'_k(T)$, la fraction de routeurs de degré k parmi les routeurs sous-jacents

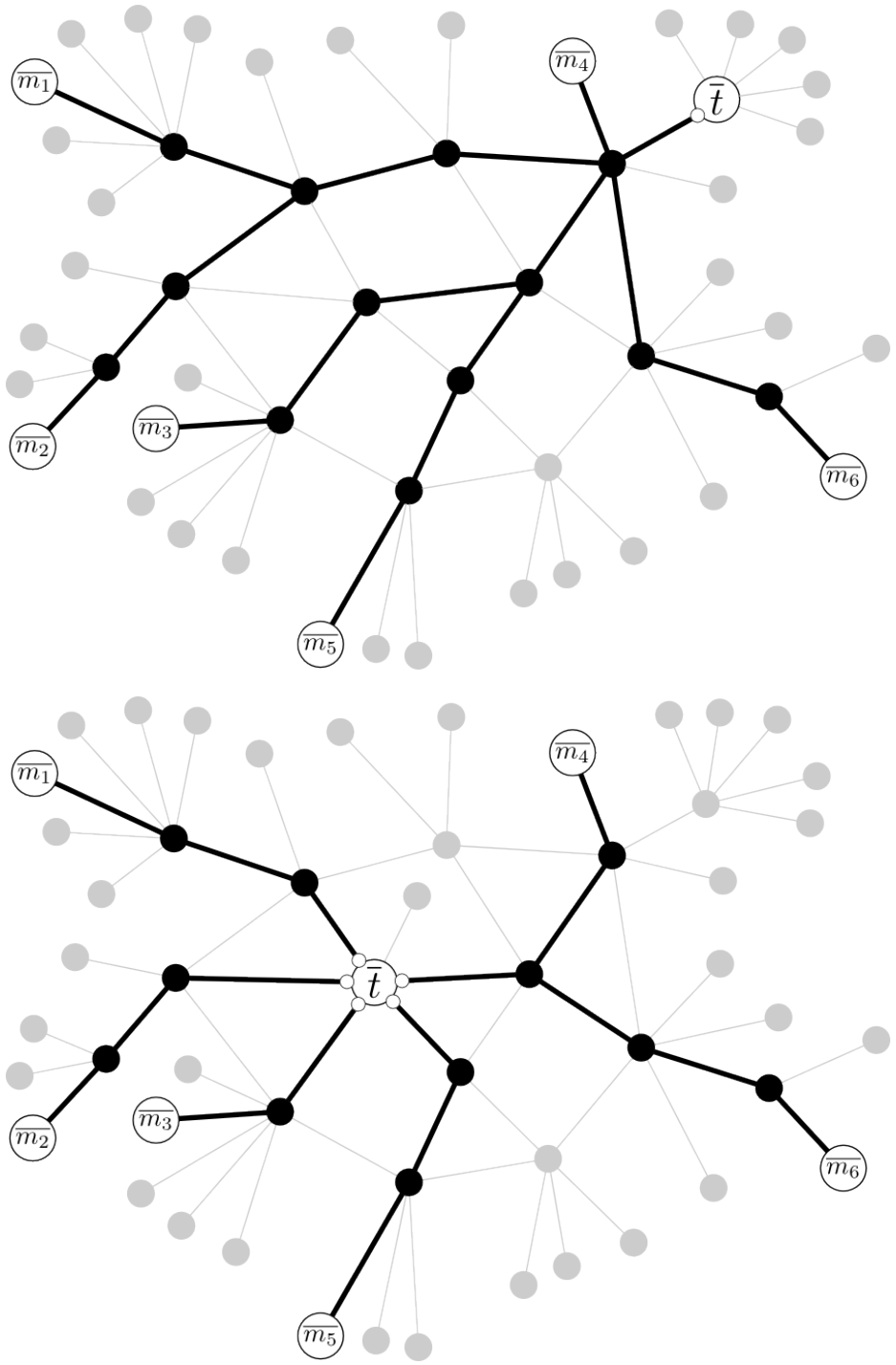


Figure 3.5 – *En haut (a) : Une cible dans le bord $\overline{t}$ possède une unique interface tournée vers le cœur, t, qui est observée par tous les moniteurs qui ne sont pas dans le même arbre (en haut). En bas (b) : Une cible dans le cœur, en revanche, a au moins 2 interfaces observables par des moniteurs qui ne sont pas dans le même arbre.*

aux adresses de T, et de corriger cette estimation en utilisant la formule ci-dessus. On obtient finalement :

$$p_k(T) = \frac{p'_k(T)}{k} \cdot \frac{1}{\sum_i \frac{p_i(T)}{i}}$$

La qualité de l'estimation $p_k \simeq p_k(T)$ dépend alors uniquement de la nature de la distribution (p_k) et de $|T|$. Cette transformation de distribution est illustrée en **Figure 3.6**.

Remarquons que le biais de sélection que nous venons de décrire possède un atout important. En effet, on s'attend à ce qu'il y ait relativement peu de nœuds de degré élevé (ce qui sera confirmé en **Section 3.6**), ce qui pourrait présenter le risque de ne pas en échantillonner. Si nous échantillonions les routeurs de manière strictement uniforme, un routeur de degré k serait échantilloné avec une probabilité p_k. Or, notre biais de sélection nous fait échantilloner un routeur de degré k avec une probabilité $k \cdot p_k$. Ceci nous permet d'avoir une bonne confiance dans la qualité de l'estimation à la fois pour les faibles degrés (qui sont très nombreux) et pour les forts degrés (dont l'échantillonage est assuré par notre méthode).

3.4 Validation du principe

Notre approche repose sur l'hypothèse que nous pouvons disposer d'un ensemble de moniteurs M suffisamment grand et suffisamment bien réparti pour pouvoir observer toutes les interfaces dans le cœur d'un routeur du cœur quelconque. La question à laquelle nous voulons répondre est la suivante : quel est le risque que notre estimation du degré d'un nœud soit différente de son degré réel, et de combien de moniteur devons nous disposer pour obtenir une estimation fiable de la distribution de degrés ?

Pour répondre à cette question, nous avons entrepris une démarche de simulations sur des graphes synthétiques. Nous avons utilisé comme moniteurs des nœuds de degré 1 (représentant des hôtes terminaux). Les cibles étaient tous les nœuds du cœur, afin de découpler la problématique qui nous intéresse de la problématique d'échantillonage qui est un problème purement statistique et qui n'est pas directement lié à notre méthode. Nous avons supposé que chaque cible répondait à un moniteur en empruntant le plus court chemin (ou l'un des plus courts chemins choisi aléatoirement en cas de choix multiples). Nous nous sommes intéressés à deux modèles de topologies : les topologies en loi de Poisson, typiques des distributions de degrés homogènes, et les topologies en loi de puissance, typiques des distributions de degrés hétérogènes. Nous ne nous attendons pas à ce que la distribution réelle d'Internet corresponde exactement à l'un de ces deux modèles, mais ils représentent des cas extrêmes de ce qu'elle pourrait être. Nous

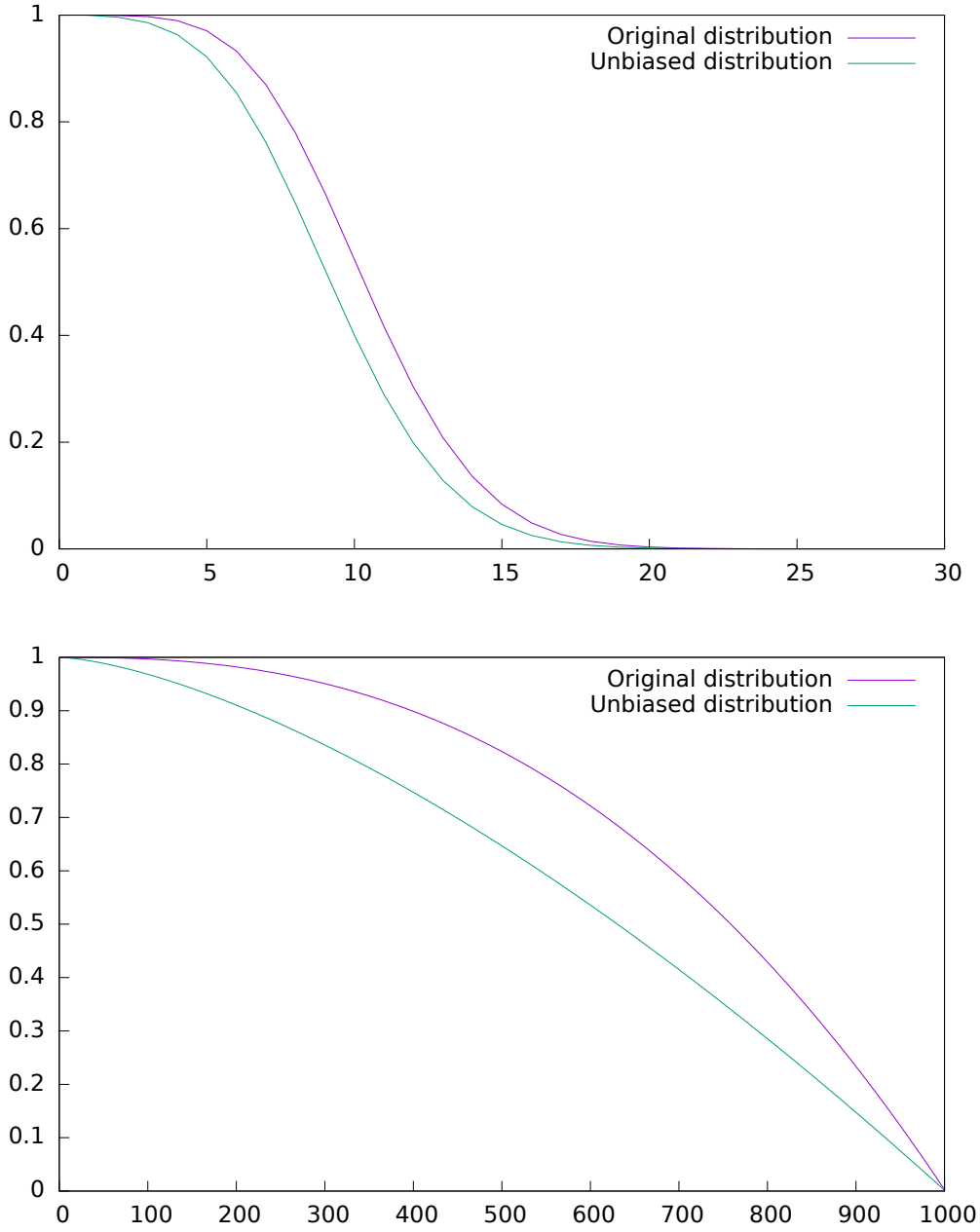


Figure 3.6 – *Fonction de distribution cumulative inversée (*ICDF*)) d'une loi de Poisson de moyenne $\lambda = 10$ (en haut), et d'une loi de puissance d'exposant $\alpha = 2.5$ (en bas) avant et après la transformation de correction du biais.*

avons simulé l'observation de la distribution de degrés pour différentes tailles de l'ensemble des moniteurs : 12, 25, 50, 100, 200, 400 et 800 moniteurs.

Le résultat de ces simulations est exposé en **Figure 3.7** et **Figure 3.8**. Comme on peut s'y attendre, la distribution de degrés est assez mal observée dans le cas

de 12 moniteurs. En particulier, le degré maximum observé est celui du nombre total de moniteurs, 12, et les nœuds de fort degré ont leur degré très sous-évalué. Cependant, on peut relever que même avec 12 moniteurs, la nature de la distribution (homogène ou hétérogène) apparaît très clairement. Lorsqu'on augmente le nombre de moniteurs, la qualité de l'observation s'accroît rapidement. Dès 200 moniteurs, la distribution réelle est indiscernable de la distribution observée dans le cas homogène. Dans le cas hétérogène, l'observation est très fidèle jusqu'à un très haut degré, au delà duquel la qualité de l'estimation s'appauvrit. Ceci n'est pas surprenant, dans la mesure où le degré mesuré ne peut excéder le nombre total de moniteurs, et où les chances de manquer certaines interfaces grandit lorsque le degré croît vers ce nombre. Toutefois, pour les nœuds de degré relativement faible, par exemple inférieur à 20, la distribution observée est indiscernable de la distribution réelle.

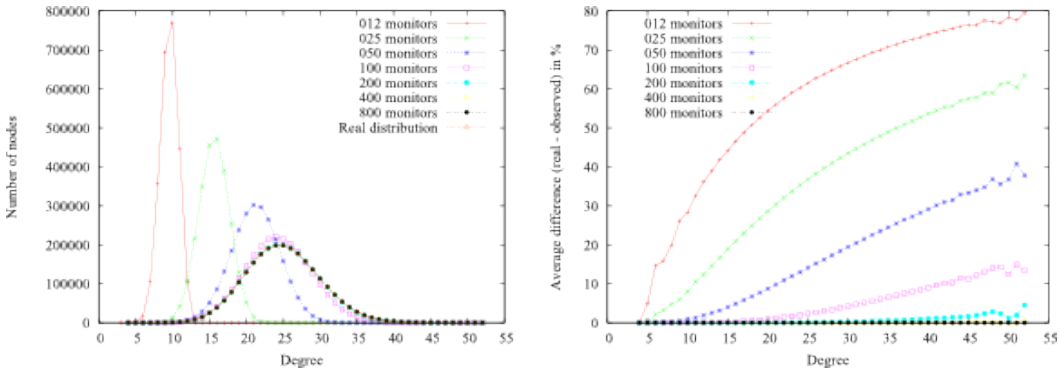


Figure 3.7 – *Observations réalisées par la simulation dans le cas d'une topologie en loi de Poisson de degré moyen* 25 *de* $2.5 * 10^6$ *nœuds. À gauche : la distribution réelle, en orange, et les distributions observées pour différentes tailles de l'ensemble des moniteurs ; en abscisse, le degré* k *et en ordonnée, la fraction* p_k *des nœuds du cœur ayant ce degré. À droite : Comparaison entre les fractions de chaque degré dans la distribution réelle et dans les distribution mesuréespour différents nombres de moniteurs ; en abscisse, le degré* k *et en ordonnée, la différence (en pourcentage) entre la fraction de nœuds de degré* k *dans la distribution réelle et dans la distribution mesurée.*

Ces assertions sont illustrées par la **Figure 3.7** (à droite) et la **Figure 3.8** (en haut à droite). On constate que pour 200 moniteurs, l'estimation de *chacun* des nœuds, et pas seulement de la distribution complète, est très bonne dans le cas de la topologie homogène, ce qui démontre que notre méthode est très performante pour ce type de graphes. Concernant la topologie en loi de Puissance, l'estimation est excellente pour les nœuds de degré faible. Les nœuds de degré 2, par exemple, sont observé pour 95% d'entre eux avec leur degré réel. Si l'on étend jusqu'aux nœuds de degré ≤ 10, alors cette proportion est de 85%. Pour les nœuds de degré relativement faible, notre méthode est donc très satisfaisante.

Le cas problématique pour notre méthode semble être l'estimation des nœuds de fort degré des distributions hétérogènes, et en particulier les nœuds dont le

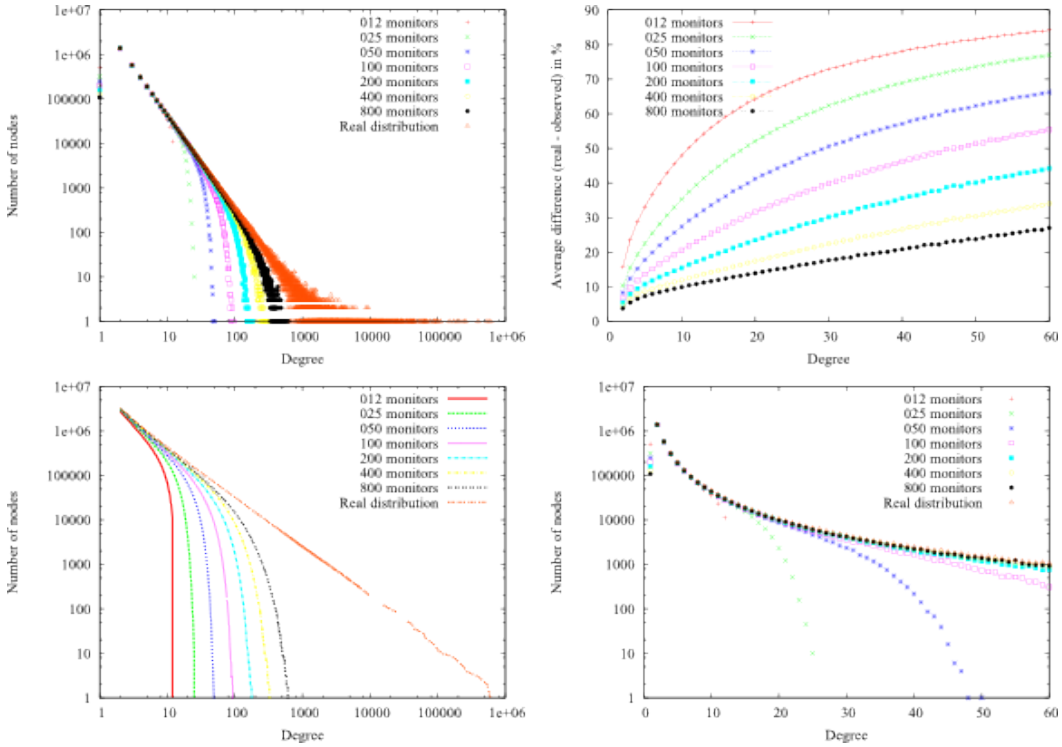


Figure 3.8 – *Observations réalisées par la simulation dans le cas d'une topologie en loi de puissance d'exposant* 2.1 *de* $1.0 * 10^7$ *nœuds. En haut à gauche : la distribution réelle, en orange, et les distributions observées pour différentes tailles de l'ensemble des moniteurs ; en abscisse, le degré* k *et en ordonnée, la fraction* p_k *des nœuds du cœur ayant ce degré. En haut à droite : comparaison entre les fractions de chaque degré dans la distribution réelle et dans les distributions mesurées pour différents nombres de moniteurs ; en abscisse, le degré* k *et en ordonnée, la différence (en pourcentage) entre la fraction de nœuds de degré* k *dans la distribution réelle et dans la distribution mesurée. En bas à gauche : la distribution cumulative inverse réelle, en orange, et les distributions cumulatives inverses observées pour différentes tailles de l'ensemble des moniteurs. En bas à droite : zoom sur la distribution de degrés pour les faibles degrés (inférieurs à 60) en échelle* lin-log.

degré excède ou se rapproche du nombre total de moniteurs, $|M|$. Cependant, il s'agit bien d'un effet de seuil, et non d'une mauvaise observation en soi. Le nuage de points montre que les nœuds de très fort degré sont sous-évalués, mais qu'ils ne sont pas confondus avec des nœuds de faible degré. Quel que soit le nombre de moniteurs, la pire estimation du degré d'un nœud degré donné (le point le plus bas sur l'axe des ordonnées) augmente avec le degré. Pour 200 moniteurs, par exemple, la pire estimation d'un nœud de degré supérieur à 1000 l'estime comme ayant un degré de 136.

En conclusion, dans le cas de la topologie en loi de Poisson comme dans le cas de la topologie en loi de puissance, la nature et la forme de la distribution de degré sont correctement observés, même avec un nombre très faible de moniteurs. Par

ailleurs, la distribution observée converge rapidement vers la distribution réelle quand le nombre de moniteurs augmente. Le degré réel de chacun des nœuds de faible degré (et de haut degré dans le cas d'une distribution homogène) est correctement observé, et les nœuds de fort degré peuvent être sous-évalués, mais jamais observés comme des nœuds de faible degré.

En dehors du cadre de cette thèse, C. Crespelle et F. Tarissan ont mené des travaux de validation approfondis [15] pour explorer un grand nombre de cas, notamment des topologies en loi de Poisson avec d'autres degrés moyens et des topologies en loi de puissance avec d'autres exposants. Ils ont montré que la taille totale du graphe avait très peu d'impact sur la qualité de l'observation pour un nombre donné de moniteurs. Ainsi, les résultats obtenus sur des graphes synthétiques de quelques millions de nœuds restent légitimes dans le cas d'Internet dont la taille totale est plus grande de plusieurs ordres de grandeur.

3.5 Evaluation d'un ensemble de moniteurs

La qualité de l'observation du degré de chaque cible, et par conséquent de notre estimation de la distribution de degrés, repose sur l'hypothèse que nous disposons d'un ensemble de moniteurs adapté. Des simulations (**Section 3.4**) ont montré qu'un nombre relativement restreint de moniteurs permet en principe de réaliser une estimation très précise, mais sous l'hypothèse que ces moniteurs sont des nœuds choisis aléatoirement parmi les nœuds de degré 1. La simple taille de l'ensemble de moniteurs ne suffit évidemment pas. Par exemple, avoir plusieurs moniteurs dans le même arbre du bord peut présenter un intérêt limité, dans la mesure où leurs observations seront souvent redondantes, puisque les chemins depuis une cible vers tous ces moniteurs seront vraissemblablement initiés par la même interface de cette cible. En pratique, le problème se pose plutôt dans l'autre sens : étant donné un ensemble de moniteurs que l'on a à disposition, il faudrait pouvoir évaluer s'il est suffisamment bien réparti pour obtenir une estimation satisfaisante, et, éventuellement, extraire de cet ensemble de moniteurs un sous-ensemble de moniteurs non-redondants pour éviter de réaliser des mesures inutiles. Pour répondre à cette question, nous avons mis au point trois approches différentes et complémentaires : la colocalisation des moniteurs (**Section 3.5.1**), la diversité des observations (**Section 3.5.2**), et la convergence des résultats (**Section 3.5.3**).

3.5.1 Colocalisation des moniteurs

Nous nous intéressons au cas des moniteurs situés dans le même arbre du bord, qu'on appelle *moniteurs colocalisés*. Remarquons d'abord qu'un moniteur m donné peut identifier quel est le nœud du cœur à la racine de cet arbre : il s'agit d'un nœud de *branchement* (voir **Section 3.1**), qui est un nœud du cœur dont l'une des

interfaces le relie à un nœud du bord dont m est un descendant dans cet arbre enraciné. Pour identifier ce nœud depuis m, nous avons conçu un outil très proche d'UDP Ping, nommé UDP Explore, et qui procède de la manière suivante, illustrée en **Figure 3.9**. Pour chaque distance d en démarrant à 1, m envoie un certain nombre K de paquets UDP Ping (paquet UDP arbitraire vers un port non utilisé), avec un TTL égal à d, chacun à destination d'une adresse routable aléatoire, et collecte les paquets ICMP *Time Exceeded* qui sont générés à l'expiration du TTL. L'ensemble des adresses ayant répondu à de tels paquets est noté $d(m)$. Puisque m est une feuille d'un arbre du bord, alors *tous* les paquets vers une destination aléatoire empruntent le seul chemin possible vers le reste du graphe, c'est à dire qu'ils remontent le long de cet arbre. Soit $\overline{d}(m)$ l'ensemble des *hôtes* sous-jacent à $d(m)$, calculé par *anti-aliasing*. Il y a alors deux cas. Soit $\overline{d}(m)$ est réduit à un seul hôte, auquel cas cet hôte est nécessairement l'ancêtre de rang d de $\overline{m}$ dans l'arbre du bord dont il est une feuille, soit $\overline{d}(m)$ contient au moins deux hôtes, et alors d majore la hauteur totale de l'arbre dont il est une feuille. Le cas limite ($d_{\max}$, le dernier d tel que $|\overline{d}(m)| \leq 1$) expose l'ancêtre de rang h de m où h est la hauteur de l'arbre dont il est une feuille, c'est à dire que $\overline{d}(m)$ est réduit à la racine de cet arbre, ou encore que $\overline{d}(m)$ est le nœud de *branchement* de cet arbre. Plus précisément, on sait alors que $d(m)$ est l'unique interface de $\overline{d}(m)$ tournées vers m. Dans ce cas, on note $\beta(m) = d(m)$ et $\overline{\beta}(m) = \overline{d}(m)$.* L'ensemble de *toutes* les interfaces observées successivement par UDP Ping depuis m est noté $\mathbb{B}(m) = \{d(m), d \leq d_{\max}\}$.

Notons qu'il se peut qu'UDP Explore n'obtienne aucun résultat pour une distance d donnée, pour les mêmes raisons que UDP Ping peut ne pas obtenir de réponse, par exemple à cause du filtrage ICMP ou d'un *firewall* sur l'un des hôtes traversés. Ceci n'est pas tellement problématique, sauf dans le cas où aucune réponse n'est obtenue pour le dernier d tel que $|\overline{d}(m)| \leq 1$. Dans ce cas, on ne connaît pas $\beta(m)$.

Soit alors m et m' deux moniteurs quelconques. Alors m et m' sont dans le même arbre du bord si et seulement si $\overline{\beta}(m) = \overline{\beta}(m')$. Donc pour décider si m et m' sont dans le même arbre, il suffit d'exécuter UDP Explore depuis m et depuis m' et d'effectuer un test d'*anti-aliasing* entre leurs résultats.

Cette notion de moniteur colocalisé nous permet d'exprimer la notion intuitive correspondant à deux routeurs *proches* et donc susceptibles de réaliser des observations identiques, c'est à dire telles que les cibles observées utilisent la même interface pour leur répondre (**Figure 3.10**). Notre méthode pour les détecter est inclusive, c'est à dire que deux moniteurs non colocalisés ne seront jamais identifiés comme tels. En revanche, deux moniteurs colocalisés peuvent éventuellement ne pas être détectés comme tels, par exemple si les hôtes sur les chemins vers le cœur d'un moniteur ne répondent pas aux sondes UDP Ping, et ne permettent pas de conclure.

*. β pour "branchement".

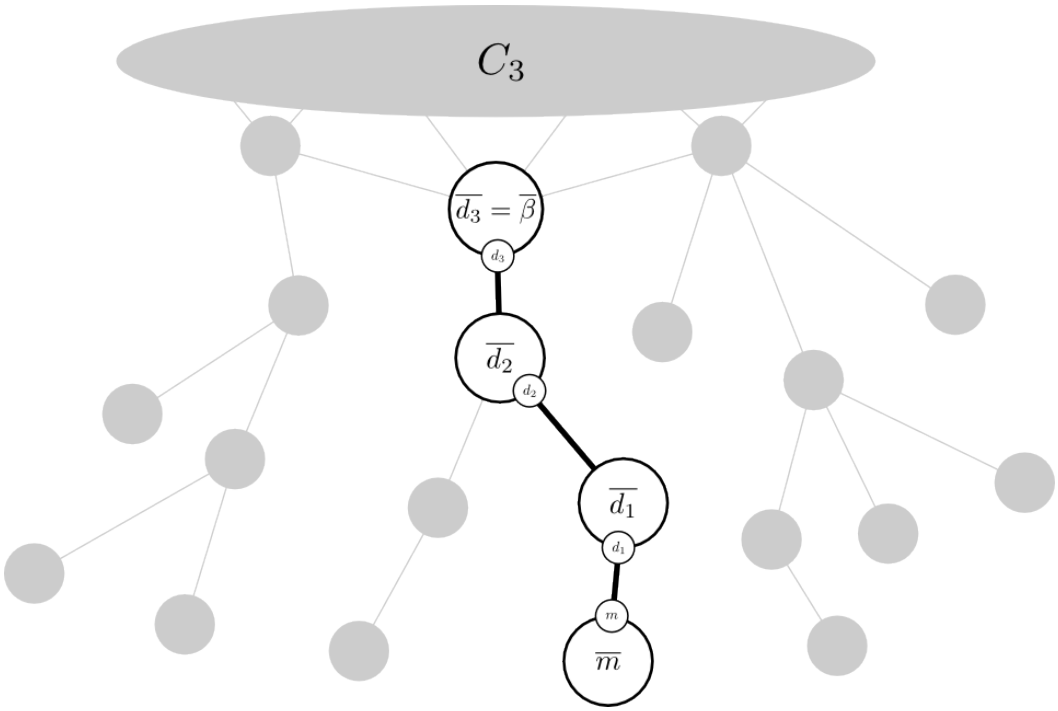


Figure 3.9 – *Le moniteur m est une feuille d'un arbre enraciné dont la racine est un nœud du cœur. Les sondes* UDP Explore *(parcourant les liens et les nœuds en noir) permettent d'identifier tous les ancêtres de $\overline{m}$ dans cet arbre, et en particulier de trouver la racine $\overline{\beta}(m)$ en les énumérant.*

UDP Explore est également utilisé pour effectuer la transformation φ_2 décrite en **Section 3.3**. Pour chaque moniteur m, m est susceptible d'observer une interface t de $\overline{t}$ qui n'est pas dans le cœur si et seulement si, $t \in \mathbb{B}(m)$. Cette assertion est valable même si UDP Explore n'obtient pas de réponses. En effet, dans un tel cas, $\overline{t}$ filtre le traffic ICMP ou UDP sur l'interface t, ce qui empêche UDP Explore de fournir un résultat, mais également UDP Ping de réaliser une observation invalide d'une interface qui ne serait pas dans le cœur. Plus généralement, pour un ensemble de moniteurs M, une interface t d'une cible qui ne serait pas une interface du cœur ne peut être observée que si $t \in \mathbb{B}(M)$ avec $\mathbb{B}(M) = \cup_{m \in M} \mathbb{B}(m)$ †.

Grâce aux résultats d'UDP Explore, nous pouvons définir une notion de qualité *intrinsèque* d'un ensemble de moniteurs, correspondant à l'ensemble des emplacements topologiquement distincts représentés dans l'ensemble, et pouvant potentiellement mener à des observations d'interfaces différentes au moins pour certaines cibles. Cette qualité intrinsèque est définie par rapport à l'ensemble total de ces emplacements, et s'exprime en termes d'ensembles-quotients de la relation d'équivalence induite par β. Un *emplacement dans le réseau* est une classe d'équi-

†. Ce qui justifie notre notation $\mathbb{B}$ pour *border blacklist*, c'est à dire une *liste noire* d'interfaces potentiellement observables alors qu'elles ne sont pas dans le cœur mais dans le bord.

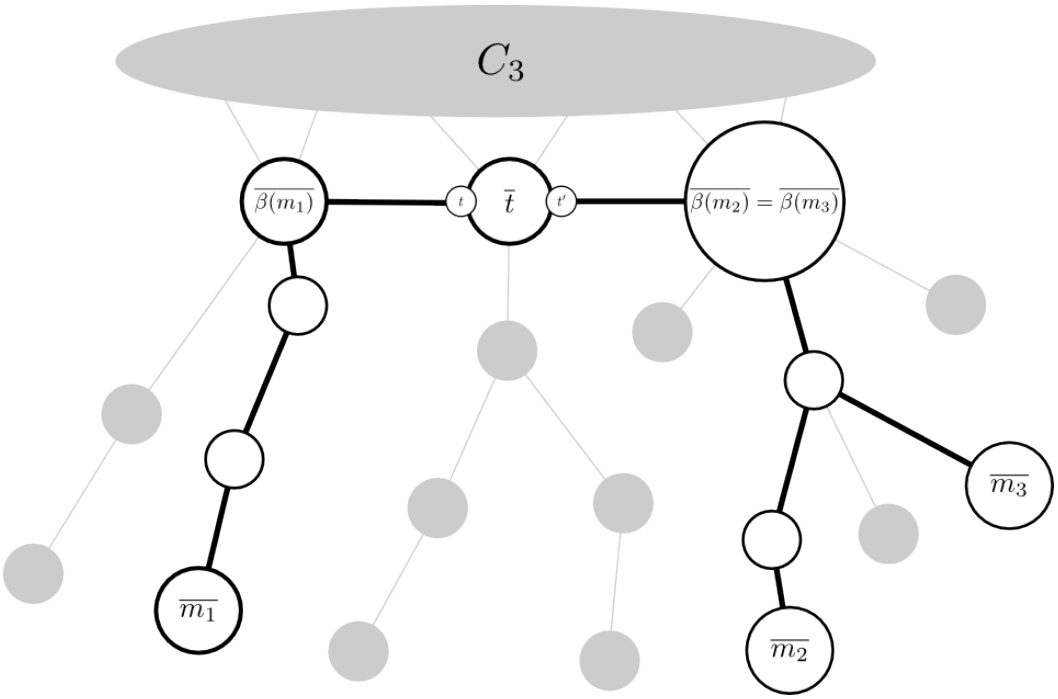


Figure 3.10 – *m_2 et m_3 sont dans le même arbre du bord, et tous les chemins depuis le cœur vers n'importe lequel d'entre deux passe par leur nœud de* branchement $\overline{\beta(m_2)} = \overline{\beta(m_3)}$. *Par exemple, le nœud $\bar{t}$ utilise la même interface t' pour leur répondre, alors qu'il utilise son interface t pour répondre à un autre moniteur m_3 situé dans un autre arbre du bord, de racine $\overline{\beta(m_1)}$.*

valence sur l'ensemble L_3 pour la relation d'égalité sur le critère β, c'est à dire pour la relation $v \equiv u \Leftrightarrow \beta(v) = \beta(u)$. On considère l'ensemble-quotient $L_3/\equiv$. La *qualité intrinsèque* d'un ensemble de moniteurs M est alors définie par $M/\equiv$ (plongé dans $L_3/\equiv$).

Toutes les classes n'ont pas la même valeur pour ce qui est de l'apport à l'observation, mais en première approximation, on peut s'intéresser à la *taille* du quotient $|M/\equiv|$ pour évaluer la qualité intrisèque de M.

3.5.2 Diversité des observations

La notion de moniteurs colocalisés nous permet d'exprimer une qualité intrinsèque d'un ensemble de moniteurs, comme une sorte de "degré de liberté" *a priori* de notre ensemble correspondant au nombre d'emplacements topologiquement distincts dans notre ensemble de moniteurs (les classes d'équivalences induites par β). Un point de vue plus pragmatique et complémentaire consiste à s'intéresser *a posteriori* à la *diversité des observations* obtenues pour une mesure d'un ensemble de cibles T par un ensemble de moniteurs M. Pour ce faire, nous nous fixons une

fonction de qualité $Q : (T, M) \mapsto Q(T, M) \in \mathbb{R}$, qui définit une évaluation de la diversité des observations, et doit respecter quelques critères qui correspondent à l'intuition :

- Un ensemble de cibles ou de moniteurs vide a une qualité nulle :

$$Q(\emptyset, M) = Q(T, \emptyset) = 0 \tag{3.1}$$

- La qualité est croissante par ajout de moniteurs :

$$Q(T, M \cup \{m\}) \geq Q(T, M) \tag{3.2}$$

- La qualité est au minimum conservée par fusion d'ensemble de cibles pour un ensemble de moniteurs fixé :

$$Q(T \cup T', M) \geq \min(Q(T, M), Q(T', M)) \tag{3.3}$$

- La qualité réduite à une cible unique est croissante par ajout d'interfaces observées :

$$|\overline{M(t)}| \geq |\overline{M'(t)}| \Rightarrow Q(\{t\}, M) \geq Q(\{t\}, M') \tag{3.4}$$

Par exemple, on peut se contenter de compter le nombre d'interfaces distinctes observées au total, c'est-à-dire $Q_0(T, M) = \sum_{t \in T} |\overline{M(t)}|$. Alors, dans ce cas, pour deux ensembles de moniteurs M et M', et si $Q_0(T, M') > Q_0(T, M)$, on peut considérer que M' est *meilleur* que M en ce sens qu'il observe davantage d'interfaces des cibles.

On peut utiliser une fonction de qualité plus subtile, par exemple pour prendre en compte le fait que chaque interface d'un nœud de faible degré est vraissemblablement plus facile à observer que les interfaces d'un nœud de fort degré. En effet, supposons pour simplifier que pour un nœud de degré d, chacune de ses interfaces est observée avec une probabilité $\frac{1}{d}$. Alors en moyenne pour N moniteurs indépendants [‡] tirés aléatoirement, une interface d'un nœud de degré d sera observé par $\frac{N}{d}$ moniteurs. Pour contrebalancer ce facteur, on peut utiliser la fonction de qualité Q_1 définie par $Q_1(T, M) = \sum_{t \in T} |\overline{M(t)}| d(t)$ où $d(t)$ est le degré de t, qui est ici approximé par $|\overline{M(t)}|$, ce qui donne $Q_1(T, M) \simeq \sum_{t \in T} |\overline{M(t)}|^2$.

Une fonction de qualité est également majorée, pour un ensemble de cibles T donné, par le cas idéal où *tous les nœuds du réseau sont des moniteurs*, auquel cas *toutes les interfaces de toutes les cibles sont observées*. On peut éventuellement utiliser cette notion pour normaliser théoriquement la qualité, même si ce facteur de normalisation n'est pas vraiment calculable en pratique :

$$Q^* : (T, M) \mapsto \frac{Q(T, M)}{Q(T, L_3)} \in [0, 1]$$

‡. Ce qui est bien sûr également une simplification puisque c'est cette indépendance que nous cherchons à approcher.

Pour une fonction de qualité Q telle que décrite ci-dessus, on peut calculer l'impact de l'*ajout* d'un nouveau moniteur m à un ensemble de moniteurs M, en calculant $\Delta Q(T, M, \{m\}) = Q(M \cup \{m\}) - Q(M)$ (qui est une valeur positive d'après l'hypothèse 3.2). Idéalement, on souhaite maximiser Δ à chaque ajout de moniteur, pour augmenter la qualité de M tout en minimisant sa taille de manière gloutonne, pour limiter la charge imposée au réseau par la mesure.

D'après les simulations effectuées en **Section 3.4**, on s'attend à ce que Q se rapproche rapidement de sa valeur maximale, c'est à dire à ce que Q^* se rapproche rapidement de 1 à mesure que M grandit, sauf bien sûr si l'on rajoute des moniteurs qui n'augmentent pas la fonction de qualité — c'est à dire s'ils sont colocalisés, et c'est pour cette raison que cette approche est *complémentaire* à l'approche de la colocalisation. En particulier, on peut combiner les deux approches en réduisant un ensemble de moniteurs M à son quotient $M/\equiv$ et en sélectionnant un représentant de chacune des classes pour étendre une fonction de qualité à un ensemble d'*emplacements du réseau* au lieu d'un ensemble de moniteurs. Cette approche combinée possède plusieurs avantages : tout d'abord, elle permet de répondre à la question soulevée auparavant des valeurs relatives des emplacements du réseau ; et elle permet d'utiliser la colocalisation pour avoir une approche plus fine que l'approche gloutonne pour optimiser la qualité d'un ensemble de moniteurs. Enfin, elle permet de vérifier la pertinence de chacune des approches, en s'assurant qu'augmenter le nombre d'emplacements augmente bien la qualité d'un ensemble de moniteurs, et qu'ajouter des moniteurs déjà colocalisés ne l'augmente pas (ou peu).

Cependant, il n'est pas courant qu'on puisse arbitrairement étendre l'ensemble des moniteurs, *a fortiori* avoir le loisir de choisir un moniteur supplémentaire optimal (en maximisant sa contribution à la qualité). En pratique, on utilisera cette notion non pas pour améliorer la qualité d'un ensemble donné, mais pour décider si un ensemble de moniteurs donné est de *suffisamment bonne qualité*. Pour ce faire, nous exploitons le fait que lorsque l'ensemble des moniteurs atteint une qualité suffisante, sa qualité Q est proche de la valeur maximale, c'est à dire que sa qualité normalisée Q^* est proche de 1. Après une mesure réelle, on injecte les résultats de la mesure, filtrés par des ensembles croissants de moniteurs ajoutés dans un ordre aléatoire, $\emptyset = M_0 \subset M_1 \subset \ldots M_{n-1} \subset M_n \subset M_{n+1} \ldots \subset M_{|M|-1} \subset M_{|M|} = M$, depuis l'ensemble vide $\emptyset$ jusqu'à l'ensemble de tous les moniteurs M, et on calcule la qualité de chaque $Q(M_n)$.

Si M est suffisamment grand et suffisamment bien réparti, alors on s'attend à ce que la courbe $n \mapsto Q(M_n)$ atteigne un régime stationnaire, à partir duquel $Q(M_n)$ est très proche de $Q(M)$. Notons que malheureusement la réciproque n'est pas vraie. Si $n \mapsto Q(M)$ atteint un régime stationnaire, cela ne signifie pas nécessairement que M est assez grand. En effet, si par exemple M est composé d'un très petit nombre de moniteurs à des emplacements distincts et d'énormément de moniteurs colocalisés, alors on s'attend à ce que $Q(M_n)$ atteigne rapidement un régime

stationnaire, sans pour autant être assez grand et bien réparti. Cette approche permet donc de valider (ou d'invalider) qu'un ensemble de moniteurs est optimal *localement*, c'est à dire qu'en rajoutant des moniteurs qui sont informellement *de même nature*, par exemple "appartenant à l'ensemble des hôtes de *Planetlab* ", on n'améliore plus la qualité de l'ensemble des moniteurs.

3.5.3 Convergence des résultats

Une dernière approche, semblable à celle des fonctions de qualité, consiste à s'intéresser non plus aux détails de notre mesure, mais à son résultat final, c'est à dire les fractions de nœuds ayant un degré observé donné p_k. Cette interprétation peut être vue comme un cas particulier d'une fonction de qualité normalisée particulière, définie de la manière suivante pour $M' \subset M$:

$$Q_{p_k}(T, M') = |\{t \in T, |\overline{M'(t)}| = k\}|$$

Soit, en normalisant :

$$Q^*_{p_k}(T, M') = \frac{p_k(M')}{p_k}$$

où $p_k(M')$ est la fraction de moniteurs observés avec un degré k en utilisant les moniteurs de M'.

Là encore, on s'attend à ce que cette fraction converge rapidement si l'ensemble M est assez grand et assez bien réparti. L'avantage de cette approche particulière est surtout conceptuel, dans la mesure où le critère de validation est directement lié au résultat final auquel on s'intéresse et s'abstrait des détails de l'implémentation.

Tout comme pour la fonction de qualité, ce calcul peut être raffiné en s'intéressant non pas à tous les moniteurs successifs, mais en se restreignant à un représentant par classe d'équivalence pour la relation de colocalisation $\equiv$.

3.6 Mesure réelle

Nous avons réalisé une mesure réelle pour mettre à l'épreuve notre méthode de mesure. Les objectifs étaient multiples. Le premier est bien sûr d'attester de la faisabilité pratique de notre méthode. L'un des enjeux majeurs à ce niveau concerne le filtrage des résultats, qui présente *a priori* le risque de supprimer tellement de données que la quantité fiable restante serait trop faible pour effectuer une estimation du résultat. Cette mesure devait également nous permettre d'éprouver nos méthodes théoriques de validation des résultats, qui reposent sur des données concrètes et s'effectuent *a posteriori*. Enfin, elle devait nous donner un aperçu

du résultat, fût-il sur un ensemble de moniteurs très particulier (en l'occurence *Planetlab*).

Nous allons présenter dans cette section les conditions du déroulement de notre mesure réelle (**Section 3.6.1**), et les résultats que nous avons obtenus (**Section 3.6.2**). Les données issues de cette expérimentation seront également mobilisées en **Section 3.8** pour mettre à l'épreuve nos méthodes de validation.

3.6.1 Déroulement

Notre mesure s'est déroulée en plusieurs étapes distinctes : la constitution d'une liste initiale de cibles, l'exécution d'UDP Ping depuis chaque moniteur vers chaque cible, l'exécution d'UDP Explore depuis chaque moniteur, et l'analyse et le filtrage des données collectées. Afin de limiter l'impact de la dynamique du réseau, nous avons enchaîné les 3 premières étapes.

La constitution de la liste correspond en partie à la méthode décrite en **Section 3.3.1**. Cependant, les filtrages φ_2 et φ_3 ne peuvent être complètement réalisés au fur et à mesure de la constitution de la liste et nécessitent d'avoir déjà réalisé la mesure UDP Ping et la mesure UDP Explore. Nous pouvons donc choisir de constituer la liste de telle sorte qu'après φ_1, on ait un nombre arbitraire de cibles restantes, mais il faut la constituer assez grande pour qu'après les filtrages ultérieurs, $T = \varphi(T_0)$ soit assez suffisamment grand.

Nous avons donc tiré des adresses IP (entiers 32 bits) de manière uniformément aléatoire successivement jusqu'à obtenir $N = 3 \times 10^6$ adresses qui étaient des adresses *valides et routables*, et surtout, qui *répondent aux sondes* UDP Ping, en envoyant une sonde isolée depuis un unique moniteur. L'ensemble obtenu correspond directement à $T_1 = \varphi_0(T_0)$, et nous avons arrêté le procédé lorsque nous avions atteint $|T_1| = N$. Cette opération, réalisée depuis un unique moniteur de notre ensemble, a duré environ 10 heures. Notons que par simplicité, nous avons réalisé ce tirage depuis un moniteur unique, mais que nous pouvons énormément accélérer ce processus en distribuant le tirage (et surtout l'exécution d'UDP Ping associée) sur plusieurs moniteurs.

Notre ensemble de moniteurs initial M_0 était composé d'environ $K = 700$ machines de la plateforme *Planetlab* [13]. Certains de ces moniteurs n'étaient pas exploitables, à cause de problèmes de connectivité par exemple, ou étaient colocalisés entre eux, mais ces considérations ont été gérées *a posteriori* afin de réaliser une collecte aussi large que possible.

Nous avons envoyé notre outil UDP Ping pré-compilé sur chacun de ces moniteurs, ainsi qu'une liste des cibles T_1, mélangée dans un ordre aléatoirement choisi pour chacun des moniteurs. Ce mélange est destiné à limiter dans une certaine mesure l'envoi simultané de très nombreuses sondes vers une unique cible, même si cette méthode est assez rudimentaire. Une fois la mise en place terminée, la

mesure UDP Ping à proprement parler a pu avoir lieu, et a duré environ 4 heures. Durant ces 4h, chaque cible a en principe reçu au maximum $K = 700$ sondes UDP Ping, et chaque moniteur a émis au maximum $N = 3 \times 10^6$ sondes UDP Ping. Afin de nous permettre d'étudier la stabilité de la mesure, la mesure UDP Ping a été répétée 3 fois à la suite. L'exécution d'UDP Explore depuis chaque moniteur s'est déroulée en quelques minutes. Au total, avec la création de la liste de cibles (l'opération la plus longue), les 3 prélèvements UDP Ping et le prélèvement UDP Explore, l'opération s'est déroulée sur moins de 24h, et n'a fait supporter aux cibles et à leur voisinage qu'une charge très modérée de quelques centaines de paquets UDP. À ce stade, nous avons rappatrié toutes les données mesurées (UDP Ping et UDP Explore) sur une machine locale pour procéder à l'analyse *a posteriori*.

	Itération 1	Itération 2	Itération 3
$\lvert M_0 \rvert$	619	625	622
$\lvert M \rvert$	421	442	442
$\lvert \mathbb{B}(M) \rvert$	1040	1107	1097
$\lvert T_1 \rvert$	2849740	2734548	2699642
$\lvert T_1 \rvert - \lvert \varphi'_1(T_1) \rvert$	10150	9842	11048
$\lvert T_1 \rvert - \lvert \varphi''_1(T_1) \rvert$	590605	527346	544252
$\lvert T_1 \rvert - \lvert \varphi^*_1(T_1) \rvert$	600755	537188	555300
$\lvert T_1 \rvert - \lvert \varphi_2(T_1) \rvert$	2842281	2727422	2692135
$\lvert T_1 \rvert - \lvert \varphi_3(T_1) \rvert$	2634226	2519320	2488483
$\lvert T \rvert$	5593	5623	5619

Tableau 3.1 – *Filtrage des données après la mesure.*

L'analyse a démarré par l'application pragmatique des filtres décrits précédemment. Leur effet quantitatif est récapitulé dans le **Tableau 3.1**. Pour UDP Ping, le résultat d'un prélèvement prend la forme, pour chaque moniteur m, d'une liste de couples $(t, m(t))$, où $m(t)$ est l'adresse de l'interface *Source* du paquet ICMP *Destination Unreachable (Code 3/Port Unreachable)* (voir **Section 3.1**), pour chaque paquet de ce type reçu par un moniteur. Quelques aberrations ont été constatées, comme par exemple des réponses multiples de la part de certaines cibles pour une sonde unique. D'autres cibles n'ont répondu qu'à un nombre très limité de moniteurs, probablement à cause d'une disponibilité limitée pendant la durée de la mesure, ou d'une limitation forte du trafic ICMP (*rate limiting*). De même, certains moniteurs n'ont obtenu que très peu de réponses, soit à cause d'une connectivité locale très faible, une disponibilité limitée, ou une surcharge au niveau de l'hôte *Planetlab* (puisque les hôtes *Planetlab* sont partagés entre de nombreux utilisateurs). Pour éviter toute anomalie liée à ces problèmes techniques, nous avons d'abord supprimé de notre jeu de données les cibles donnant au moins une réponse multiple, filtrage que nous notons φ'_1. Puis, nous avons compté pour chaque moniteur, le

nombre de cibles ayant fourni une réponse, et pour chaque cible, le nombre de moniteurs les ayant observé, et nous nous sommes intéressés à la distribution de ces valeurs (**Figure 3.11**). Comme attendu, *la plupart* des moniteurs observent *la plupart* des cibles, et *la plupart* des cibles sont observées par *la plupart* des moniteurs. Pour purifier notre jeu de données, nous avons supprimé tous les moniteurs observant moins de 80% des cibles (les moniteurs restant définissent l'ensemble $M = M_1$), et toutes les cibles observées par moins de 80% des moniteurs (filtrage φ_1''). Le filtrage correspondant à la suppression de cibles pour leur manque de réponses est une extension de φ_1, que l'on note $\varphi_1^* = \varphi_1'' \circ \varphi_1' \circ \varphi_1$ de telle sorte que $T_1^* = \varphi_1^*(T_0) = \varphi_1'' \circ \varphi_1'(T_1)$ puisque $T_1 = \varphi_1(T_0)$.

Pour procéder à la suite du filtrage et exécuter φ_2, nous avons d'abord dû calculer $\mathbb{B}(M)$, qui est une simple reformulation des résultats fournis par UDP Ping. Une fois $\mathbb{B}(M)$ calculé, nous avons supprimé chacune des interfaces apparaissant dans cette liste de nos données, puis compté le nombre d'interfaces observées *restantes* pour chaque cible. L'application de φ_2 correspond à la supression des cibles tel que ce nombre est égal à 0 ou 1. φ_3 est une transformation plus directe, puisqu'elle consiste simplement à éliminer toutes les cibles t telles que $t \notin M(t)$.

En pratique, nous avons profité de la commutativité [§] des transformations (φ_k) et appliqué ces transformations séparément à T_1 et réalisé l'intersection à la fin de la démarche, de telle sorte que $T = \varphi_1'(T_1) \cap \varphi_1''(T_1) \cap \varphi_2(T_1) \cap \varphi_3(T_1) = \varphi_3 \circ \varphi_2 \circ \varphi_1^*(T_0)$.

À l'issue de tous ces filtrages, pour chacune de nos itérations, nous observons environ 5600 cibles. Ces cibles sont garanties, d'après notre méthode, d'être un ensemble d'adresses d'interfaces choisies uniformément aléatoirement dans le cœur de routeurs du cœur, dont nous disposons de la liste de toutes les interfaces dans le cœur observable par l'ensemble des moniteurs de *Planetlab*.

3.6.2 Résultats

Le résultat de notre mesure prend la forme d'une liste de triplets $(m, t, m(t))$ où m décrit M et t décrit T de telle sorte que $m(t)$ est l'interface observée par m pour la cible t losqu'une réponse a été obtenue pour le couple (m, t) (ce qui est garanti d'arriver dans au moins 80% des cas pour chaque moniteur et pour chaque cible, à cause du filtrage que nous avons effectué précédemment). Pour chaque cible t, on calcule l'ensemble $M(t) = \{m(t), m \in M\}$, et en particulier $|M(t)|$, qui correspond au degré dans le cœur observé de la cible $\bar{t}$. Pour chaque entier k, on calcule ensuite la fraction p_k' de cibles ayant un degré observé égal à k. Cette fraction est biaisée linéairement par le degré, comme décrit en **Section 3.3**, et on corrige ce biais en appliquant la transformation $p_k' \mapsto \frac{p_k'}{k}$ puis en normalisant cette fraction entre 0 et 1. Le **Tableau 3.2** indique les valeurs des (p_k) pour nos 3

§. Cette commutativé est simplement due à la commutativité de l'intersection d'ensembles.

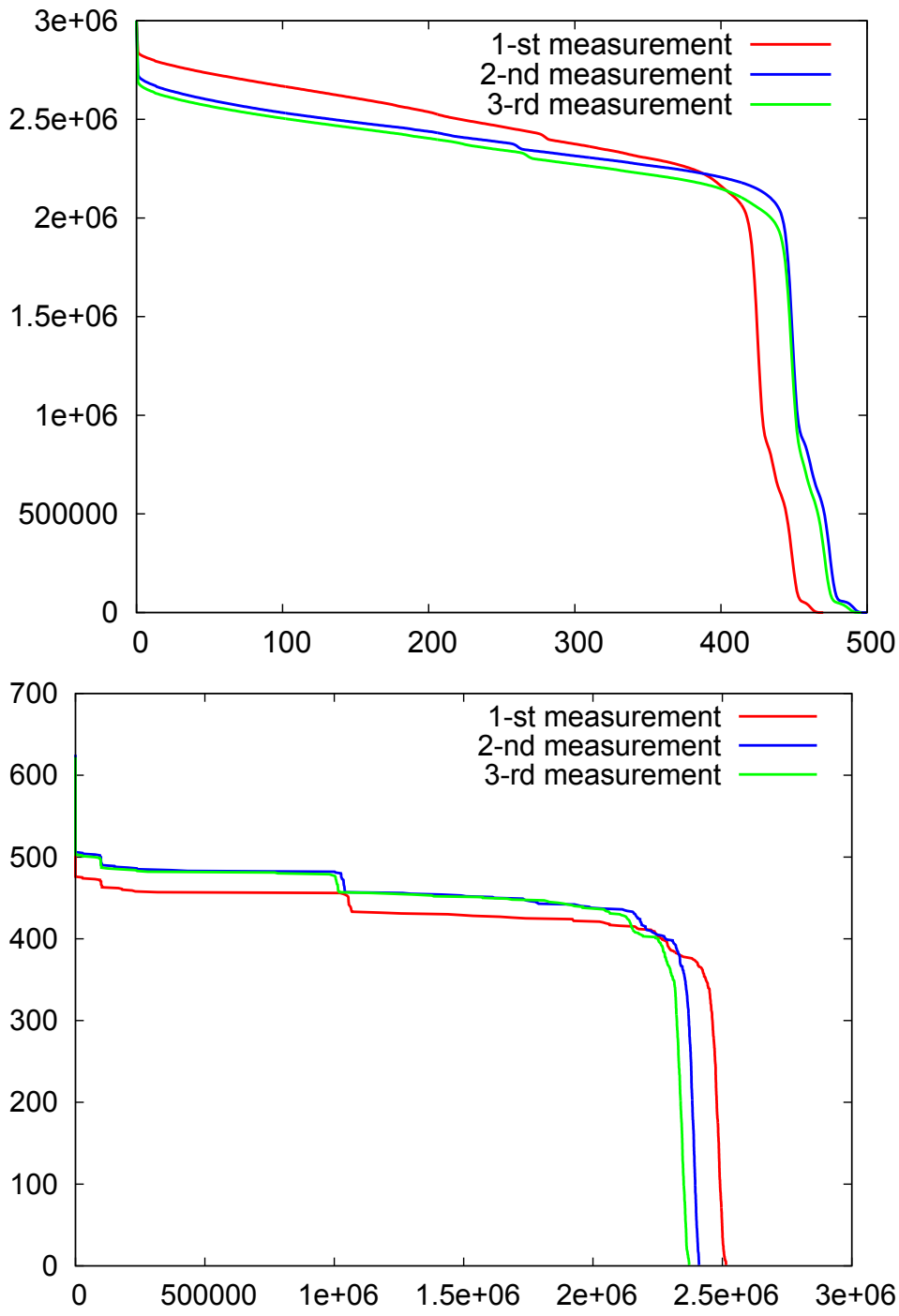


Figure 3.11 – *En haut (resp. en bas) : pour chaque x sur l'axe des abscisses, nous traçons le nombre de cibles (resp. de moniteurs) ayant envoyé (resp. reçu) au moins x réponses* ICMP *lors de notre mesure, pour chacune des 3 itérations.*

mesures, qui sont combinées sous la forme d'une distribution cumulative inverse en **Figure 3.12**.

Degré k	Fraction p_k (Itération 1)	Fraction p_k (Itération 2)	Fraction p_k (Itération 3)
2	0.74770	0.74371	0.75214
3	0.19434	0.19838	0.19258
4	0.02727	0.02727	0.02585
5	0.01551	0.01588	0.01486
6	0.00708	0.00640	0.00644
7	0.00206	0.00224	0.00230
8	0.00175	0.00196	0.00147
9	0.00127	0.00131	0.00145
10	0.00057	0.00044	0.00052
11	0.00056	0.00052	0.00047
12	0.00040	0.00044	0.00047
13	0.00020	0.00023	0.00017
14	0.00025	0.00031	0.00031
15	0.00032	0.00009	0.00017
16	0.00014	0.00025	0.00024
17	0.00023	0.00018	0.00015
18	0.00007	0.00007	0.00007
19	0.00007	0.00009	0.00009
20	0.00002	0.00000	0.00002
21	0.00008	0.00015	0.00008
22	0.00006	0.00000	0.00004
23	0.00000	0.00000	0.00002
24	0.00002	0.00000	0.00002
25	0.00000	0.00005	0.00002
26	0.00000	0.00002	0.00002
27	0.00002	0.00000	0.00002
28	0.00000	0.00002	0.00000
29	0.00002	0.00000	0.00001

Tableau 3.2 – *Fractions des routeurs du cœur de degré k, après correction du biais de sélection.*

Notre première observation porte sur la stabilité des résultats, c'est à dire que les 3 itérations de notre mesure donnent des résultats très similaires, ce qui tend à confirmer que l'échelle de temps (quelques heures) sur laquelle nous avons effectué notre mesure n'est pas excessive.

Les distributions observées montrent clairement que les nœuds de faible degré ont une énorme prévalence, puisqu'environ 75% des nœuds sont de degré 2, et environ 95% ont un degré égal à 2 ou 3. Ce n'est pas choquant, puisque nous nous intéressons ici uniquement aux interfaces dans le cœur, c'est à dire les interfaces utilisées par les routeurs pour router vers des destinations qui ne sont pas dans un sous-réseau du bord qui leur serait rattaché, mais bien pour effectuer un véritable *routage*.

Pourtant, nous observons quelques routeurs de fort degré, et le degré maximal que nous observons est 29, pour une cible. Il est possible que nous n'observions pas

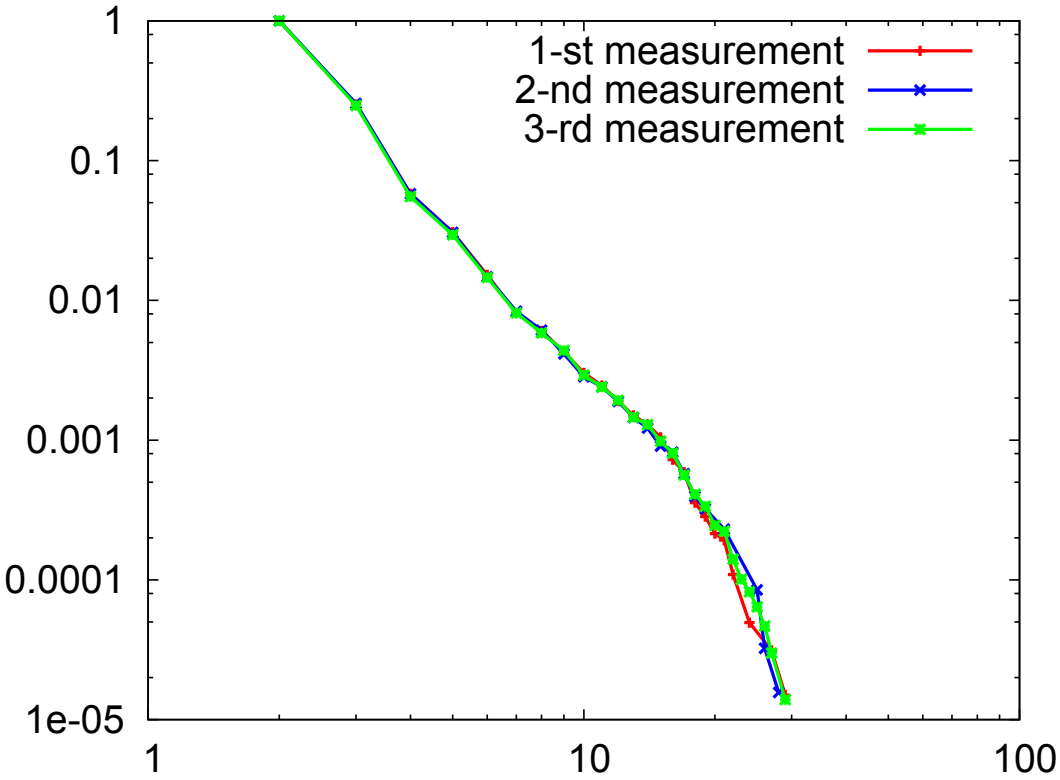


Figure 3.12 – *Pour chaque valeur x en abscisse, on trace la fraction des routeurs du cœur ayant un degré supérieur ou égal à x, en échelle log-log.*

certaines de ses interfaces, mais il est peu probable que son degré réel soit beaucoup plus élevé, puisque le nombre de moniteurs que nous utilisons ($|M| > 400$) dépasse amplement ce nombre. Cette remarque sera approfondie en **Section 3.8**. Il est bien sûr également possible, et probable, que des routeurs de degré supérieur à 29 existent, mais il n'y en a aucun dans l'ensemble que nous avons mesuré, et on s'attend donc à ce qu'ils soient extrêmement rares sur Internet — une hypothèse renforcée par le biais de sélection en faveur des nœuds de fort degré.

3.7 Protocole complet

Cette section fait office de synthèse de notre protocole complet pour évaluer la distribution de degrés dans le cœur des routeurs du cœur, étant donné un ensemble de moniteurs, et elle intègre à la fois la méthode théorique telle que nous l'avons déjà décrite, et les ajustements opérationnels que nous avons tirés de notre mesure réelle.

- Soit les paramètres suivants :
  - M_0, un ensemble de moniteurs.

  - $N \in \mathbb{N}$, un nombre initial de cibles.
- Transférer UDP Ping et UDP Explore pré-compilés vers chaque moniteur de M_0.
- Echantilloner aléatoirement uniformément un ensemble T_1 de N adresses de 32 bits correspondant à des adresses IP valides qui répondent à UDP Ping.
- Exécuter UDP Explore depuis chaque moniteur de M_0.
- Pour chaque moniteur m de M_0, tirer une liste mélangée $T_1(m) =$shuffle (T_1) extraite de T_1 et exécuter UDP Ping vers cette liste.
- Une fois la mesure terminée, rappatrier les résultats vers une machine locale.
- Supprimer les triplets $(m, m(t), t)$ tels qu'il existe au moins un cas de réponse multiple de la part de t.
- Supprimer les triplets $(m, m(t), t)$ tels que m observe moins de 80% des cibles ou t est observée par moins de 80% des moniteurs (on obtient alors M en considérant l'ensemble des m restants).
- Calculer $\mathbb{B}(M)$ à partir des résultats d'UDP Explore.
- Supprimer les triplets $(m, m(t), t)$ tels que $m(t) \in \mathbb{B}(M)$.
- Supprimer les triplets $(m, m(t), t)$ tels qu'il n'existe pas de triplet (m', t, t) pour au moins un certain m'.
- Calculer alors $M(t)$ pour chaque t en utilisant les triplets restants.
- Supprimer les triplets $(m, m(t), t)$ tels que $t \notin M(t)$.
- Supprimer les triplets $(m, m(t), t)$ tels que $|M(t)| \leq 1$.
- Calculer sur $M(t)$ (l'ensemble des triplets restants) chaque fraction $p'_k = \frac{|\{t, |M(t)|=k\}|}{|T|}$.
- Calculer chaque $\frac{p'_k}{k}$ et normaliser pour obtenir chaque p_k.

3.8 Validation

Dans cette section, nous tentons de valider les résultats obtenus lors de notre mesure réelle. Nous validons d'abord la qualité de notre ensemble de moniteurs en utilisant les indicateurs décrits en **Section 3.5** (**Section 3.8.1**). Puis nous réinjectons la distribution de degrés obtenue par la mesure dans notre modèle de simulations pour montrer la cohérence de notre résultat (**Section 3.8.2**).

3.8.1 Qualité de l'ensemble de moniteurs

Pour évaluer la qualité de notre ensemble de moniteurs, nous avons commencé par calculer les classes de colocalisation des moniteurs, comme décrit en **Section 3.5.1**. Une fois ces classes obtenues, nous avons calculé la convergence de la qualité Q_0 avec l'ajout de classes, et la convergence des fractions p_k avec l'ajout de classes.

Nous avons obtenu $n_{\max} = 203$ classes de colocalisation, chaque classe comprenant en moyenne 2.11 moniteurs. Cet ordre de grandeur est cohérent avec la nature de l'ensemble des hôtes de *Planetlab* : chaque institution qui participe à *Planetlab* fournit en général quelques moniteurs au sein du réseau de cette institution, qui sont donc la plupart du temps colocalisés. En examinant les noms DNS des moniteurs appartenant à chaque classe, nous avons pu renforcer cette hypothèse, puisque les moniteurs d'une même classe suivaient très souvent un motif commun du type `*.domain.tld`, comme par exemple `onelab1.info.ucl.ac.be`, `onelab2.info.ucl.ac.be`, et `onelab3.info.ucl.ac.be`, indiquant leur appartenance à une même institution.

Une fois les classes de moniteurs identifiées, nous avons employé la méthode décrite en **Section 3.5.2** pour évaluer la variation de la diversité des observations avec l'ajout de classes de moniteurs. Pour chaque entier k entre 1 et $n_{\max} = 203$, nous avons tiré un grand nombre de combinaisons de k classes de colocalisations. Pour chaque tirage, nous avons considéré l'ensemble M' de moniteurs appartenant à la réunion de ces n classes, et calculé $Q_0(M')$, puis calculé la moyenne notée $Q_0(n)$ de ces valeurs pour tous les tirages d'une taille n donnée. $Q_0(k)$ correspond donc à la moyenne de la qualité lorsqu'on utilise n classes de moniteurs. On procède de même pour la fonction de qualité Q_1 (**Figure 3.13**).

Comme espéré, pour les deux fonctions, la qualité augmente très rapidement avec les premières classes de moniteurs, et se stabilise rapidement. Ceci suggère que rajouter des moniteurs supplémentaires n'augmenterait pas beaucoup la qualité de l'observation, et que par conséquent notre ensemble de moniteurs est d'une qualité raisonnable.

Pour approfondir cette piste, nous avons employé la dernière méthode décrite en **Section 3.5.3**. Pour chaque nombre de classes de moniteurs n, on calcule la valeur moyenne de chaque fraction $p_k(n)$ des cibles observées avec un degré k si l'on se restreint à n classes de moniteurs. Puisqu'on s'intéresse à la convergence de cette valeur, on normalise le résultat en calculant $\frac{p_k(n)}{p_k(n_{\max})}$, où $p_k(n_{\max})$ représente la fraction obtenue en utilisant *toutes* les classes de moniteurs (**Figure 3.14**).

Il est intéressant de remarquer que la vitesse de la convergence change avec le degré k auquel on s'intéresse. Pour les faibles degrés, la fraction p_k converge très rapidement. Pour $k < 5$, une dizaine de classes de colocalisations (ou, formulé d'une autre manière, une dizaine de moniteurs non colocalisés) suffit à estimer p_k avec une précision supérieure à 80%. Ceci s'explique car non seulement il suffit de peu de moniteurs pour correctement observer un nœud d'un faible degré, mais il suffit également de peu de moniteurs pour détecter qu'un nœud de fort degré *n'est pas* un nœud de faible degré. En d'autres termes, même si nous n'estimons pas correctement la fraction des forts degrés, nous estimons tout de même très précisément la fraction des faibles degrés. Pour les nœuds de fort degré, la convergence est beaucoup moins rapide, mais on constate tout de même une stabilisation lorsqu'on s'approche de $n_{\max}$.

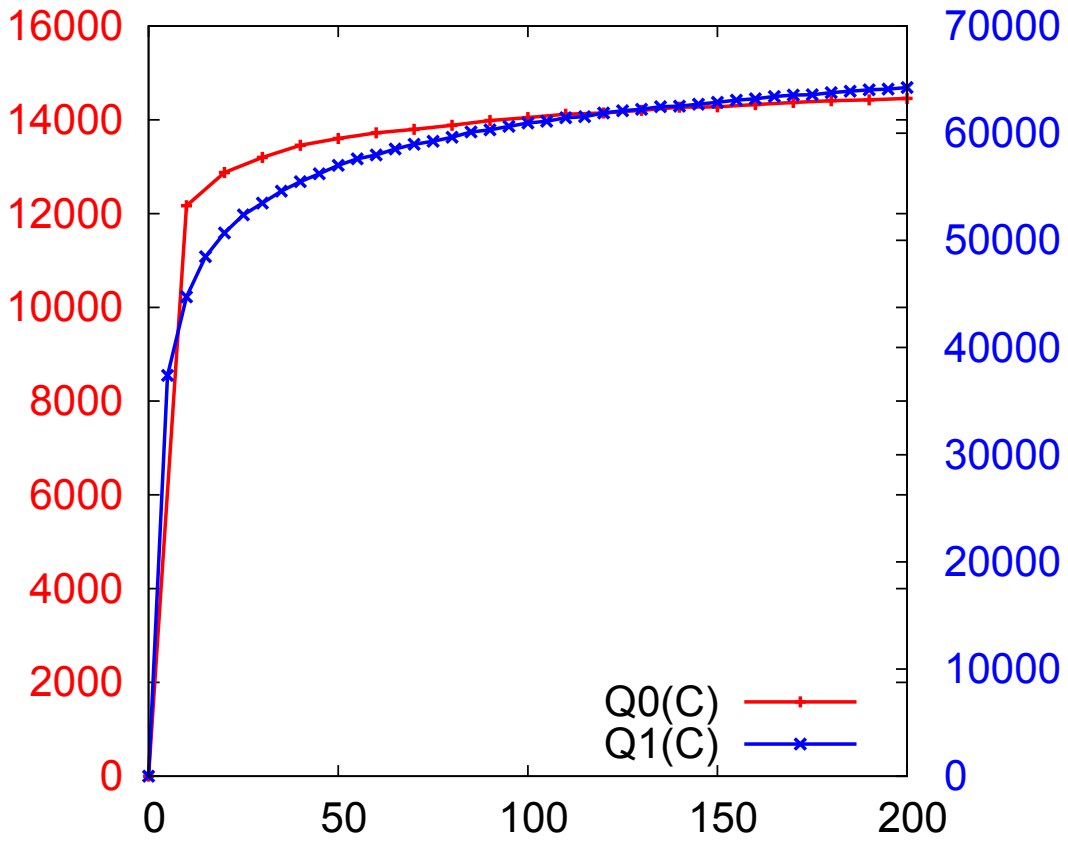


Figure 3.13 – *Pour chaque valeur x en abscisse, on trace la qualité $Q_0(x)$ moyenne et la qualité $Q_1(x)$ moyenne si l'on se restreint à x classes de colocalisation.*

La synthèse de notre travail de validation sur l'ensemble des moniteurs de *Planetlab* suggère qu'il est d'une qualité suffisante pour obtenir des résultats très fiables pour les nœuds de faible degré, et qu'il majore avec des ordres de grandeur raisonnables les fractions de nœuds de fort degré. Cependant, il est clair qu'augmenter le nombre de moniteurs et de classes de colocalisation améliorerait à la fois la précision et la fiabilité des résultats, surtout pour les nœuds de fort degré.

3.8.2 Réinjection dans les simulations

Nous avons justifié la pertinence de notre approche à l'aide de simulations en **Section 3.4**. Nous avions alors utilisé deux modèles simples de graphes aléatoires à distribution de degrés fixée, d'une part en loi de Poisson pour représenter une distribution homogène, d'autre part en loi de puissance pour représenter une distribution hétérogène. Notre conclusion était qualitativement positive dans les deux cas, mais relevait une certaine importance de la nature précise de la distribution réelle que l'on souhaitait mesurer, et qui n'était probablement pas exactement l'un de ces deux cas extrêmes.

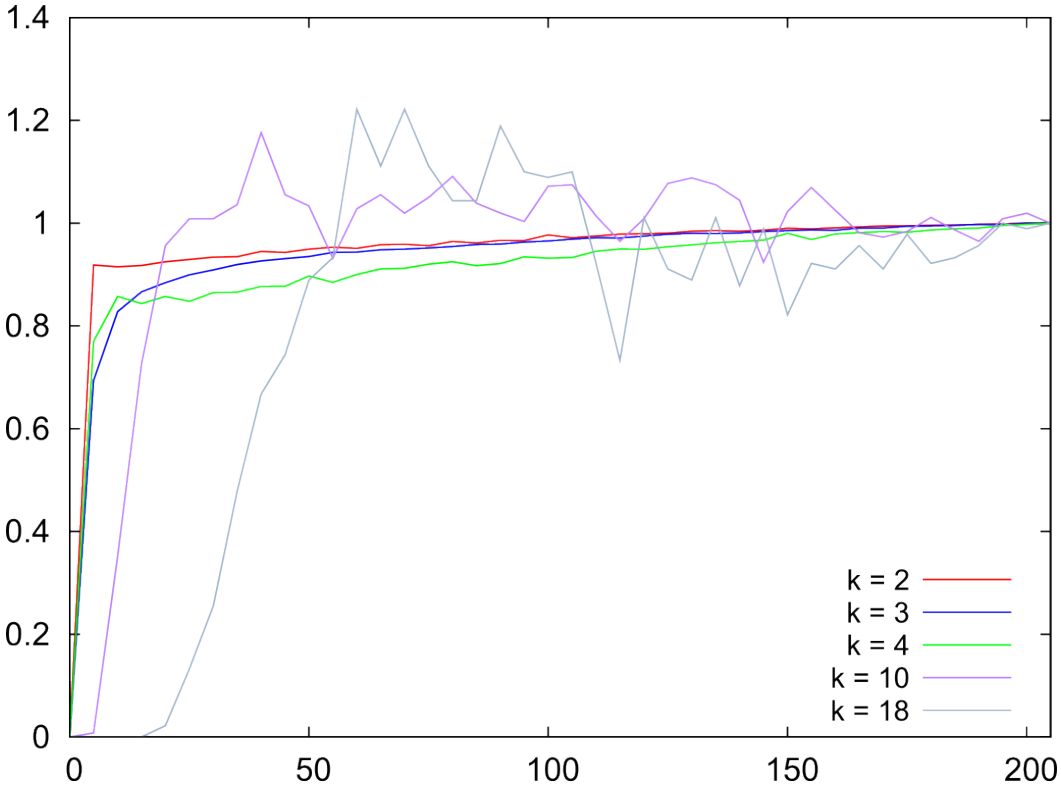


Figure 3.14 – *Pour chaque valeur x en abscisse, on trace pour différents degrés k le rapport entre la fraction de cibles observées avec un degré égal à k, et la fraction finale obtenue en utilisant* toutes *les classes de moniteurs.*

Pour vérifier la cohérence de nos résultats, nous avons donc décidé de réaliser des simulations non plus sur des distributions formelles extrêmes en loi de Poisson ou en loi de puissance, mais tout simplement de reprendre la distribution que nous avons *mesurée* en **Section 3.6** et de vérifier que nous obtenons également des résultats pertinents.

Puisque nous avons déjà montré que la taille totale du réseau est d'une importance très limitée pour notre méthode, nous nous sommes contentés de simuler des graphes comprenant 1 million de nœuds. Nous avons donc généré 5 graphes aléatoires de taille 1 million représentant le cœur d'Internet, respectant chacune des 3 distributions de degrés observée par notre mesure. Pour chaque graphe, nous avons échantilloné 5 ensembles de nœuds, que nous avons utilisé comme moniteurs. Au total, nous avons donc réalisé 75 simulations, pour lesquelles nous avons testé des ensembles de 12, 25, 50, 100, 200, 400 et 800 moniteurs. En réalité, puisque les nœuds de ces graphes représentent des routeurs du cœur, les "moniteurs" de la simulation sont à rapprocher des classes de colocalisation de moniteurs, et les nœuds concernés peuvent être vus comme les racines d'hypothétiques arbres du bord qui leur seraient reliés et dans lesquelles se trouveraient les véritables moniteurs. Ainsi, le nombre de "moniteurs" dans la simulation correspond plutôt

au nombre de moniteurs dans des classes de colocalisation distinctes, de telle sorte que le nombre de 200 moniteurs est celui qui est le plus proche de notre mesure réelle, effectuée avec 203 classes de colocalisations.

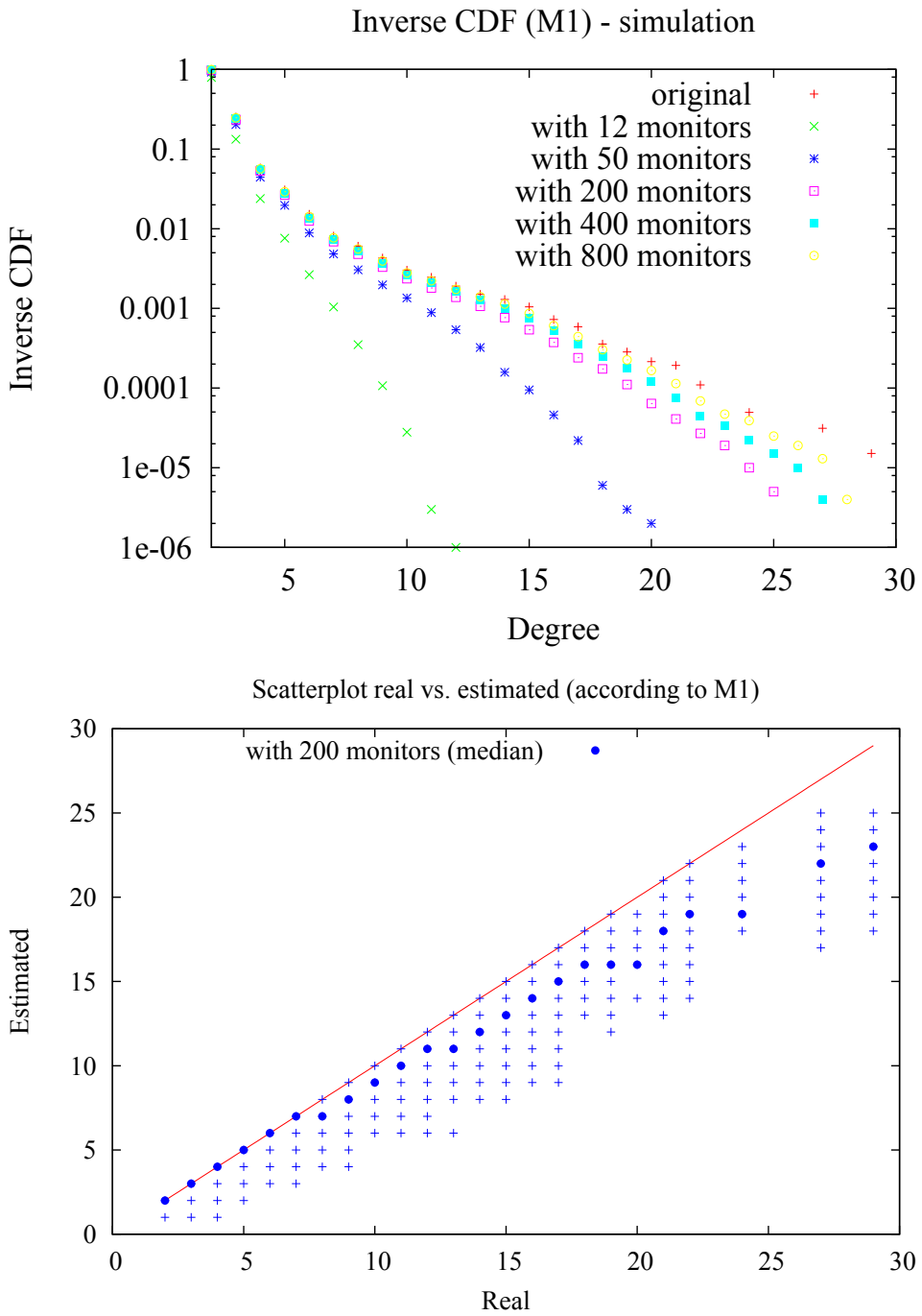


Figure 3.15 – *En haut, la distribution de degrés observée pour différents nombre de moniteurs (qui représentent ici des classes de colocalisation). En bas, la correlation entre le degré observé par 200 moniteurs et le degré réel (un point par nœud et médiane).*

Les résultats de ces simulations sont illustrés en **Figure 3.15** (en haut), et sont constitués des différentes distributions de degrés observées par les différents ensembles de moniteurs. On constate que 200 moniteurs suffisent pour très bien observer la distribution de degrés réelle, même si les fractions observée des nœuds des degrés élevés sont moins bien estimées. La courbe met en évidence à quel point la proportion des nœuds de faible degré est bien estimée : environ 95% des nœuds de degré inférieur ou égal à 10 sont bien observés avec leur degré réel avec 200 moniteurs.

Nous approfondissons en étudiant la différence absolue entre le degré observé de chaque nœud et son degré réel (**Figure 3.15**, en bas). Cette approche confirme que notre méthode parvient très bien à mesurer le degré d'un nœud en particulier — ce qui est plus précis que de correctement estimer la *distribution* des degrés. En particulier, la valeur médiane est très proche du degré réel, même pour les nœuds de fort degré. Et même pour les nœuds de degré élevé, la pire estimation, c'est à dire celle dont le degré observé est le plus bas (donc le plus éloigné du degré réel) n'est jamais trop éloignée du degré réelle. Par exemple, les nœuds de degré 29 ont dans le pire cas été estimés comme ayant un degré de 18 ; les nœuds de degré 27 comme ayant un degré de 17 ; et les nœuds de degré 24 comme ayant un degré de 18. Dans tous les cas, comme nous l'avons déjà mentionné, un nœud de fort degré n'est *jamais* estimé comme ayant un *faible* degré ; même dans le pire cas, il est sous-évalué avec un facteur qui nous paraît raisonnable.

Nous pouvons conclure que les simulations sont en accord avec notre mesure empirique. La distribution de degrés globale est très bien estimée (c'est à dire que la distribution observée est très proche de la distribution réelle) ; en particulier, la fraction des nœuds de faible degré est très précisément connue ; et en particulier, le degré de *chaque* nœud est très bien estimé, pour peu que son degré ne soit pas trop élevé. Augmenter le nombre de moniteurs (non colocalisés) améliorerait la qualité de notre estimation, mais ne changerait pas fondamentalement son aspect général.

3.9 Discussion et conclusion

Notre travail de mesure orientée propriété de la topologie physique d'Internet nous a permis de mettre au point une méthode qui permet d'estimer la distribution des degrés dans le cœur de cette topologie. Cette estimation peut être conduite depuis un ensemble relativement limité de moniteurs, et elle est très fiable, à la fois dans sa forme générale, et dans la valeur précise des fractions des nœuds de faible degré, sans commune mesure avec les estimations tirées des cartes. Pour atteindre cet objectif, nous avons choisi de définir très précisément notre objet d'étude et la propriété à laquelle nous nous intéressions. En analysant les configurations qui pouvaient présenter des problèmes, et en opérant un filtrage très

prudent, nous assurons que les résultats obtenus sont très fiables. Notre mesure réelle montre qu'en dépit de ces filtrages très radicaux, nous arrivons à conserver suffisamment de données, dont la fiabilité est garantie, pour réaliser une estimation statistiquement pertinente. De plus, la précision de cette estimation peut encore être améliorée en raffinant les conditions de la mesure, par exemple en itérant les mesures, en augmentant le nombre de cibles, ou en augmentant le nombre de sites (classes de colocalisation) depuis lesquels nous l'opérons. Nous avons utilisé les sites de *Planetlab*, mais d'autres infrastructures, qui pourraient être utilisées en complément de celle-ci, existent. *DIMES* [60] et *RIPE Atlas* [47], par exemple, mettent à disposition des milliers d'hôtes pour des travaux de ce type. Depuis un tel ensemble de moniteurs et en prenant les précautions nécessaires pour éviter d'encombrer le réseau, il n'est pas exclu d'effectuer une mesure de très grande ampleur qui ciblerait *tout* l'espace IPv4. En effet, même si notre approche distribuée peut sembler massive, elle limite en réalité la charge induite sur les cibles au maximum à une sonde envoyée par moniteur et par cible, en utilisant le protocole UDP qui est rarement filtré sur le chemin puisqu'il est semblable à du trafic applicatif normal. Cette faible charge se démarque des approches traditionelles basées sur des outils comme TRACEROUTE qui utilisent un très grand nombre de sondes. Cette caractéristique de notre mesure nous permet d'ailleurs d'envisager une utilisation de notre protocole sur la durée, en répétant notre mesure de manière régulière au fil du temps, par exemple une fois par jour, afin de capturer la dynamique de la topologie du cœur.

Plutôt que de se hasarder à des hypothèses optimistes sur la bonne implémentation des protocoles ou sur les bonnes pratiques du réseau, nous avons volontairement limité la portée de nos résultats aux objets sur lesquels nous pouvions présenter des garanties de fiabilité. Une conséquence de ce choix est que la distribution de degrés que nous obtenons n'est pas la distribution de degré de l'ensemble du la topologie physique, mais seulement du sous-graphe constitué du cœur d'Internet et de ses arêtes internes (les interfaces dans le cœur). Ce n'est pas forcément un problème, puisqu'il s'agit précisément de la partie de la topologie dans laquelle opère le routage non trivial. Hors du cœur, le routage est le plus souvent limité à de l'acheminement descendant ou ascendant dans des arbres ou des structures arborescentes. Pour les travaux d'analyse du routage d'Internet, la connaissance de la topologie du cœur est donc bien souvent l'objet d'intérêt. Pour autant, pour obtenir une connaissance complète de la distribution de degrés de l'ensemble de la topologie physique, il faudrait certainement utiliser une approche complémentaire, orthogonale à notre approche, pour obtenir une connaissance spécifique du bord du réseau.

Au delà de la qualité de l'ensemble des moniteurs, il subsiste toutefois des problèmes pratiques et conceptuels dans notre méthode. Le premier est, comme nous l'avons déjà mentionné, sa précision limitée pour déterminer la fraction des nœuds d'un degré élevé donné. En effet, même si nous pouvons avec précision

majorer la fraction des nœuds d'un degré *supérieur* à un degré donné, notre méthode est intrinsèquement moins précise sur les degrés élevés, dans la mesure où la probabilité d'observer une interface donnée d'un routeur est *a priori* plus faible si ce routeur est de degré élevé et qu'il faudra donc un ensemble de moniteurs de meilleure qualité (en taille et en répartition) pour la détecter. C'est d'autant plus vrai si l'utilisation de ses interfaces par un routeur de fort degré n'est pas uniforme : si certaines interfaces du cœur d'un routeur sont très peu utilisées par rapport à d'autres, alors il sera plus difficile de disposer d'un moniteur qui sollicite cette interface. La question du choix de l'interface de réponse d'une cible en fonction du moniteur qui a envoyé la sonde est d'ailleurs l'une de nos pistes de travail qui nous a semblé les plus prometteuses et les plus pertinentes à l'issue de cette analyse, et elle est le sujet d'intérêt du **Chapitre 4**.

Chapitre 4

Mesure des tables de routage

La motivation sous-jacente aux travaux de mesure de la topologie d'Internet est diverse, et sa portée est souvent à la fois théorique et pratique. Bien souvent, cependant, son objectif principal est de fournir ou de préciser des paramètres qui sont ensuite injectés dans des modèles formels ou simulatoires qui visent à mieux comprendre, et éventuellement à optimiser, la performance du réseau. Cette performance peut s'exprimer en termes de fiabilité, de résilence aux pannes ou aux attaques, ou de capacité globale à acheminer du trafic. Dans tous les cas, on s'intéresse à la manière dont *les paquets sont acheminés sur le réseau* et plus précisément, à la manière dont les paquets sont *routés* sur le réseau. De la même manière dont nous avons conclu qu'il pouvait être plus pertinent de mesurer directement la distribution de degrés plutôt que de mesurer des cartes sur lesquelles on lirait la distribution de degrés, nous en sommes arrivés à l'objectif de mesurer directement le *comportement des routeurs* plutôt que de l'extrapoler à partir de modèles de graphes à distribution de degré fixée.

En utilisant une méthode de mesure dérivée d'UDP Ping et une analyse fine des résultats de cette méthode, nous avons donc tenté de mesurer directement le comportement spécifique de certains routeurs, qui répondent à UDP Ping, pour effectuer leur activité de routage. Cette dernière peut la plupart du temps s'exprimer sous une forme très synthétique, qui correspond en outre à une implémentation matérielle ou logicielle réelle, la table de transmission (*forwarding table*), qui est une version compilée et optimisée de la table de routage (*routing table*) d'un routeur. C'est cette table de transmission que nous allons chercher à estimer. Nous commencerons par poser le contexte du problème en détaillant nos objets d'intérêts (**Section 4.1**). Puis nous montrerons de quelle manière nous pouvons utiliser les résultats d'UDP Ping pour préciser notre connaissance, exprimée sous forme de contraintes formelles, de la table de transmission d'un routeur ciblé (**Section 4.2**). Nous combinerons des hypothèses de travail réalistes avec ces contraintes pour réaliser une inférence de la table de transmission (**Section 4.3**). Cette méthode sera éprouvée par une mesure réelle (**Section 4.4**). Enfin, nous présenterons nos conclusions sur ce travail (**Section 4.5**).

4.1 Structure des tables de transmission

Un *routeur* est un hôte L3 particulier, qui peut occuper diverses fonctions annexes, mais son rôle principal est assez simplement résumé : lorsqu'il reçoit

un paquet dont il n'est pas la destination, il doit être capable de le transmettre (*forward*) sur l'une de ses interfaces, avec l'objectif ultime que ce paquet finisse par atteindre sa destination. Les routeurs implémentent donc localement des *politiques de routage*, qui sont définies par divers algorithmes de routages, et dont l'objectif global est d'offrir au réseau la capacité d'acheminer des paquets depuis n'importe quelle source vers n'importe quelle destination — si possible rapidement.

En règle générale, chaque routeur dispose d'une *table de routage*, qui prend la forme d'une liste de règles abstraites portant sur le comportement que le routeur doit adopter lorsqu'il reçoit des paquets à transmettre. Les règles dans cette table de routage sont d'origines variées. Elles peuvent provenir d'algorithmes de routages, tels que BGP [56] ou OSPF [46], d'accords de routage ou de *peering* au sein de l'*AS* dont elles font partie, d'algorithmes de qualité de service (*QoS*) ou d'équilibrage de charge (*load-balancing*), ou encore de configuration statique *ad hoc*. La forme précise d'une table de routage est assez variable et dépend beaucoup des spécificités matérielles et logicelles du routeur, mais elle peut toujours être interprété comme un ensemble de règles abstraites qui, combinées entre elles, forment une fonction de routage $R : p \mapsto i(p)$ qui à chaque paquet p associent une interface $i(p)$ de ce routeur sur lequel il doit transmettre p. Remarquons que sous cette forme, R n'est pas déterministe, puisqu'elle peut par exemple réaliser de l'équilibrage de charge ou une reconfiguration dynamique.

En pratique, cependant, les problématiques liées à l'entretien de la table de routage, hétéroclite et complexe, dynamique, la rendent peu utilisables directement pour résoudre à une fréquence très soutenue les décisions de transmission, allant de quelques dizaines à plusieurs millions par seconde. Pour cette raison, la fonction de routage R d'un routeur est généralement compilée sous une forme optimisée, qu'on appelle la *table de transmission* (*forwarding table*). La table de transmission est une version compilée de la table de routage, qui expose un comportement équivalent, mais qui est adaptée aux contraintes très fortes de performance auxquelles sont soumises les routeurs et en particulier les routeurs du cœur. Ses implémentations reposent souvent sur du matériel spécifique qui implémente de gigantesques tables associatives très performantes nommées *Content-adressable memory* (*CAM*), mais qui peuvent être émulées sur des routeurs plus modestes par des tables de hachage en *RAM*. Dans tous les cas, il s'agit de réduire un ensemble de règles sémantiquement assez complexes à des règles dont la logique se réduit à la consultation d'une table pré-définie. Plus précisément, il s'agit de réduire la décision que le routeur doit réaliser quand il reçoit un paquet, à décider en fonction de l'adresse de destination de ce paquet, sur quel sous-ensemble de ses interfaces il peut la transmettre.

Théoriquement, une table de transmission pourrait être *complète*, c'est à dire disposer d'autant d'entrées qu'il existe de destinations *possibles*, c'est à dire environ 2^{32} correspondant aux 32 bits d'une adresse IP. En pratique, il faudrait disposer d'une mémoire adressable de plusieurs milliards d'entrées. Pour diminuer cette

taille et augmenter leur efficacité, les routeurs regroupent les entrées par *préfixes*. Si la fonction de routage R d'un routeur $\overline{r}$ est telle que, pour un certain préfixe d'adresse ρ/n de longueur n, tous les paquets à destination d'une adresse qui est préfixée par ρ/n doivent être transmis sur l'une des interfaces d'un ensemble fixé qu'on note $R(\rho/n) \subset \overline{r}$, alors on peut compiler toutes ces règles en une unique règle $\rho/n \rightsquigarrow R(\rho/n)$. Cette manière de procéder est d'autant plus efficace qu'en organisant les entrées d'une table par préfixe, on peut facilement trouver le plus long préfixe auquel la destination d'un paquet donné correspond, et donc trouver la règle *la plus spécifique*, celle que le routeur doit utiliser, pour transmettre ce paquet. De plus, elle permet d'exploiter l'organisation historique d'un grand nombre de réseaux en préfixes correspondant à l'attribution par l'*IANA* et ses mandataires régionaux de blocs d'adresses, formalisée par la methode *CIDR* (*Classless Inter-Domain Routing*) [27, 55, 23, 22]. Conçue précisément avec l'objectif de limiter la croissance de la taille des tables de transmission organisées par préfixes, cette méthode préconise l'attribution de blocs d'adresses contigus et correspondant à des préfixes. Cette méthode reste une recommandation qui peut très bien ne pas être respectée, particulièrement par les utilisateurs finaux des adresses, mais elle offre un cadre efficace au moins dans la mesure où elle est correctement implémentée par les autorités administratives des domaines de haut niveau, par exemple au niveau *AS*. Ces autorités étant concernées en premier lieu par le problème de la *mise à jour des tables* et de la *diffusion des routes*, au cœur du fonctionnement d'algorithmes de routages tels que BGP, elles sont enclines à utiliser la méthode *CIDR* pour limiter la lourdeur des échanges.

Sur un plan très pragmatique, la méthode *CIDR* est commode puisqu'elle introduit une notation permettant de désigner un ensemble de 2^{32-n} adresses sous la forme ρ/n où ρ est la représentation décimale de la première adresse du bloc (dont l'adresse binaire se termine donc par n fois le bit 0). Le **Tableau 4.1** donne un exemple d'un fragment d'une table de transmission par préfixes présentée avec cette notation. Chaque ligne présente un préfixe ρ/n et l'ensemble $R(\rho/n)$ des interfaces de sorties du routeur $\overline{r}$ correspondantes. Une telle ligne signifie que si $\overline{r}$ doit transmettre un certain paquet dont l'adresse de destination a pour plus long préfixe dans la table ρ/n, alors le routeur utilisera l'une des interfaces de $R(\rho/n)$ pour la transmettre.

Dans la suite de ce chapitre, nous supposerons donc qu'une *table de transmission* d'un hôte $\overline{r}$ est définie par un ensemble $D = \{D_0, \ldots, D_n\}$ de *règles*, chaque règle étant formée par un couple $D_k = (\rho_k/n_k, R(\rho_k/n_k))$ avec $R(\rho_k/n_k) \subset \overline{r}$.

4.2 Contraintes obtenues avec UDP PING

Nous avons présenté la structure formelle des tables de transmission en **Section 4.1**. Notre objectif dans cette section va être d'énoncer des *contraintes* sur la

Préfixe de destination ρ/n	Interfaces de sortie $R(\rho/n) \subset \overline{r}$
...	...
128.32.0.0/13	$\{83.238.96.26\}$
128.10.0.0/13	$\{195.114.175.54\}$
128.112.193.64/26	$\{83.238.96.26\}$
128.112.139.0/26	$\{83.238.96.26, 195.114.175.54\}$
128.114.63.0/26	$\{83.238.96.26, 195.114.175.54\}$
...	...

Tableau 4.1 – *Fragment d'une table de transmission par préfixe CIDR d'un routeur $\overline{r}$. Chaque règle associe un prefixe de l'adresse de destination avec un ensemble d'interfaces de sortie du routeur.*

table de transmission d'un routeur donné qui répond aux sondes UDP Ping qui seront ultérieurement combinées avec des hypothèses d'inférence.

Comme nous l'avons déjà expliqué précédemment (**Section 3.1**), lorsqu'un hôte choisit l'interface de sortie qu'il va utiliser pour émettre un paquet ICMP, il utilise la même logique de routage que lorsqu'il effectue une transmission de paquets ; au niveau L3, le fait que ce paquet soit à l'origine de l'hôte lui-même ou un paquet à transmettre est indifférent. Pour une certaine destination m, si une cible $\overline{t}$ utilise son interface t pour émettre un paquet ICMP vers m, alors on peut en déduire que ce choix découle de sa logique de routage à l'égard de m. Cette logique de routage peut s'exprimer en termes de tables de transmission. Puisque t a été choisie pour émettre en direction de m, alors il existe une certaine règle $D_m = (\rho_m/n_m, t)$ dans la table de transmission de $\overline{t}$ telle que ρ_m/n_m est un préfixe de m, et en plus ce préfixe est le plus long parmi tous les préfixes de règles de la table de transmission de $\overline{t}$. En utilisant précisément UDP Ping pour faire générer un paquet ICMP à t en direction d'un moniteur m, nous pouvons ainsi formaliser :

Proposition 4 (Contrainte sur la table de transmission issue d'UDP Ping)**.** *Soit m un moniteur et t une cible qui répond à* UDP Ping *($m(t) \in \overline{t}$ est bien définie). Alors il existe une règle $(\rho_m/n_m, R)$ dans la table de transmission de $\overline{t}$ telle que :*
- *ρ_m/n_m est un préfixe de m*
- *$m(t) \in R$*
- *ρ_m/n_m est le plus long préfixe de m dans la table de transmission de $\overline{t}$*

Comme il existe nécessairement au moins une règle dont le préfixe est un préfixe de m (quitte à ce que préfixe soit $0/0$ dans le cas trivial), cette contrainte peut se reformuler d'une manière plus inductive :

Proposition 5 (Contrainte sur la table de transmission issue d'UDP Ping (forme inductive))**.** *Soit m un moniteur et t une cible qui répond à* UDP Ping *($m(t) \in \overline{t}$ est bien définie). Alors, pour toute règle $D = (\rho/n, R)$ dans la table de transmission*

de $\bar{t}$, si ρ/n est le plus long préfixe de m dans la table de transmission de $\bar{t}$, on a nécessairement $m(t) \in R$.

Si l'on organise l'espace IPv4 en fonction des préfixes binaires (comme on pourrait le faire avec tout intervalle contigu d'entiers), alors l'espace des adresses peut être interprété sous la forme d'un arbre $\mathbb{A}$ binaire équilibré de préfixes binaires. Chaque nœud représente un préfixe ρ/n. La racine est le préfixe trivial $0/0$. Les fils du nœud ρ/n sont les deux préfixes $\rho.0/(n+1)$ et $\rho.1/(n+1)$ *. Pour un routeur $\bar{t}$ donné, on définit alors $\mathbb{A}(\bar{t})$ qui est un étiquetage de $\mathbb{A}$, tel que l'étiquette $R(\rho/n)$ du nœud ρ/n est le plus petit sous-ensemble de $\bar{t}$ qui contient toutes les interfaces utilisées par $\bar{t}$ pour transmettre à des adresses préfixées par ρ/n. Autrement dit, on définit l'étiquette $R(\rho/n)$ d'un nœud de manière inductive, de telle sorte que pour toute adresse d, $R(d/32) = R(d)$, et $R(\rho/n) = R(\rho.0/(n+1)) \cup R(\rho.1/(n+1))$. Réciproquement, à partir de $\mathbb{A}(t)$, il est aisé de calculer une table de transmission optimale qui soit compatible avec les règles exprimées par les étiquettes des feuilles. Il suffit de trouver tous les préfixes de l'arbre ρ/n tels que $R(\rho.0/(n+1)) \neq R(\rho.1/(n+1))$ et d'exprimer les deux règles $D_{\rho.0/(n+1)} = (\rho.0/(n+1), R(\rho.0/(n+1))$ et $D_{\rho.1/(n+1)} = (\rho.1/(n+1), R(\rho.1/(n+1))$. Dans ce cadre formel, une contrainte fournie pas UDP Ping s'exprime par $m(t) \in R(m/32)$ et, par induction, tous les ancêtres de $m/32$, c'est à dire tous les nœuds correspondants à des préfixes de m, doivent contenir $R(m)$ et donc $m(t)$.

L'information obtenue avec UDP Ping depuis un seul moniteur m reste très limitée. En revanche, nous pouvons utiliser UDP Ping de manière distribuée depuis un ensemble de moniteurs M pour obtenir davantage d'information. Chaque moniteur de notre ensemble M qui reçoit une réponse de t fournit une contrainte sur $\mathbb{A}(t)$ et donc sur la table optimale qui en résulte. Remarquons que dans ce cas, la portée de l'ensemble des contraintes dépend beaucoup de M et, notamment, de la répartition des bits de poids fort dans les adresses qui composent M. Par exemple, si tous les moniteurs appartiennent à un même préfixe de taille n, alors la portée des contrainte sera limitée aux 2^{32-n} adresses de destination qui correspondent à ce préfixe. À l'inverse, si notre ensemble de moniteurs est nombreux et réparti de manière homogène dans l'espace IP, alors il fournira une information riche sur l'aspect de la table de transmission de la cible.

4.3 Inférence

Nous avons expliqué dans la **Section 4.2** comment nous pouvons utiliser UDP Ping depuis un ensemble de moniteurs vers une certaine cible pour obtenir de l'information sur la table de transmission de cette cible en organisant les résultats

*. Ici et dans tout ce qui suit, $\rho.0$ (resp. $\rho.1$) représente le préfixe ρ concaténé avec 0 (resp. avec 1).

en fonction des préfixes des moniteurs. Ces informations imposent des *conditions nécessaires* sur cette table de transmission. Mais beaucoup de tables de tranmissions sont compatibles avec ces conditions nécessaires. Nous pouvons formuler des hypothèses, fondées par d'autres travaux ou une connaissance *a priori* de la forme des tables de transmission, pour choisir une table de tranmission en particulier parmi celles qui respectent ces contraintes. Le travail d'*inférence* consiste donc à formuler un choix parmi les tables de transmissions qui sont compatibles avec les contraintes que nous avons énoncées. Nous présenterons 3 schémas d'inférence, dont nous discuterons la pertinence : le *schéma le plus spécifique* (**Section 4.3.1**), le *schéma le moins spécifique* (**Section 4.3.2**), et le *schéma AS* (**Section 4.3.3**).

4.3.1 Schéma le plus spécifique

Le schéma d'inférence le plus simple consiste à choisir, parmi les tables compatibles avec les contraintes engendrées par UDP Ping, celle qui ne réalise *aucune* hypothèse sur les destinations qui ne sont pas dans notre ensemble de moniteurs. Formellement, elle consiste à assigner à chaque destination d (et donc au préfixe $d/32$) qui n'est pas dans M, un ensemble d'interfaces de sorties vide ($R(d) = \emptyset$). Le calcul de la table de transmission associé devient trivial, puisqu'il suffit d'énoncer toutes les règles $(m/32, m(t))$, en fusionnant éventuellement les règles qui peuvent l'être de manière récursive.

Ce schéma d'inférence est extrêmement simple tant sur le plan conceptuel que sur le plan pratique, mais il est peu réaliste, et surtout il est d'une portée très limitée, puisqu'il ne fournit aucune information sur le comportement de la cible pour des destinations qui ne sont pas dans l'ensemble des moniteurs.

4.3.2 Schéma le moins spécifique

Le schéma d'inférence le plus spécifique est celui qui réalise le *moins* d'hypothèses sur le comportement de la cible. À l'extrême opposé, nous pouvons chercher le schéma résultant des hypothèses les plus généralisatrices possibles, tout en restant bien sûr dans le cadre des hypothèses formulées par la mesure UDP Ping. Pour cela, au lieu d'assigner aux feuilles inconnues de $\mathbb{A}(t)$ un ensemble d'interfaces vide, on complète l'arbre pour minimiser le nombre de règles total nécessaire à exprimer les contraintes de la mesure. Plus directement, pour chaque moniteur m, on calcule le plus grand sous-arbre qui contient m et aucun autre moniteur, et on assigne à chaque feuille de cet arbre le même ensemble d'interfaces de sorties $R(m)$. De cette sorte, la règle issue de ce sous-arbre une fois projeté sous la forme d'une table de transmission est $D_m = (\rho/n, R(m))$, où ρ/n est exactement la racine du sous-arbre en question.

Ce schéma correspond tout simplement à supposer que chaque moniteur est un représentant d'un certain préfixe *CIDR* qui est utilisé par la cible pour arbitrer sa

logique de routage. La méthode d'inférence correspond alors à trouver le plus court préfixe (le moins spécifique) qui corresponde à cette hypothèse. Bien entendu, cette hypothèse est très optimiste, et sa pertinence dépend beaucoup de la qualité de l'ensemble des moniteurs M. Si M est suffisamment grand, et suffisamment bien réparti, on peut espérer qu'il parcoure bien l'ensemble des règles de la cible. Si ce n'est pas le cas, alors ce schéma fournit tout de même une approximation des règles de la cible qui sera d'autant plus pertinent que M est grand et bien réparti.

Comme la représentation complète de $\mathbb{A}(t)$ est lourde à manipuler, nous proposons un algorithme simple et efficace pour calculer le schéma le moins spécifique directement à partir du tableau associatif qui représente l'observation de t depuis M (**Algorithme 2**). Cet algorithme prend en argument l'ensemble des moniteurs M organisé sous la forme d'un arbre binaire de recherche et un tableau associatif R qui contient les contraintes fournies par UDP Ping associées à chaque moniteur, sous la forme d'un ensemble d'interfaces $R[m]$. La notation $M[a, b]$ désigne l'ensemble des adresses de M qui sont comprises entre a et b, mais cet ensemble est simplement représenté en mémoire par les deux entiers a et b et une référence à M. La représentation de M sous la forme d'un arbre binaire de recherche permet de trouver rapidement ($O(\lg |M|)$ †) la plus petite et la plus grande adresse de M dans InférerTable(M, R). Dans TrouverPivot(M, R, ρ/n, a, b), elle permet également de trouver rapidement ($O(\lg |M[a, b]|) < O(\lg |M|)$) la *plus grande adresse de* $M[a, b]$ *préfixée par* $p.0/(n + 1)$ et la *plus petite adresse de* $M[a, b]$ *préfixée par* $p.1/(n + 1)$. Dans TousIdentiques(M, R, a, b), chaque comparaison $R(m) = R(a)$ consiste à comparer deux listes de taille majorée par $|\bar{t}|$; la complexité totale d'un appel de cette fonction est donc $O(|M[a, b]| \times |\bar{t}|)$. En dehors de ces points techniques, cet algorithme est simplement une résolution de type diviser-pour-régner qui découpe les sous-problèmes de M selon un pivot dont le calcul s'effectue en $O(\lg |M|)$ et avec une sous-routine dont le calcul s'effectue en $O(|M[a, b]| \times |\bar{t}|)$. Sa complexité totale est donc de $O(|M| \lg |M|)$ en amortissant la complexité de la sous-routine. La terminaison est évidente car si $n = 32$, alors ρ/n contient au plus un élément, *a fortiori* $M \cap \rho/n$ contient au plus un élément et donc TousIdentiques(M, R, a, b) renvoie vrai. La preuve de la correction est également assez directe, puisqu'il suffit de montrer les invariants suivants :

- Si $(m_0, m_1) =$ TrouverPivot(R, ρ/n, a, b), alors $M[a, b] = M[a, m_0] \cup M[m_1, b]$, et tous les moniteurs de $M[a, m_0]$ admettent $\rho.0/n + 1$ comme préfixe, et tous les moniteurs de $M[m_1, b]$ admettent $\rho.1/n + 1$ comme préfixe.
- TousIdentiques(M, R, a, b) renvoie vrai si et seulement si tous les moniteurs de $M[a, b]$ admettent la même contrainte (égale à $R(a)$ en particulier).
- Si $D =$ InférerSousTable(M, R, ρ/n, a, b), alors D est la plus petite (en termes de taille de la liste D) table de transmission compatible avec R.

Les deux premiers invariants sont immédiats, et le troisième découle des deux premiers.

†. On note lg la fonction logarithme binaire, proportionnelle à la fonction ln.

Algorithme 2 Inférence du schéma le moins spécifique

function TrouverPivot(R, ρ/n, a, b)
  renvoyer $(\max\{m \in M[a,b], m \in \rho.0/(n+1)\}, \min\{m \in M[a,b], m \in \rho.1/(n+1)\})$
function TousIdentiques(M, R, a, b)
  renvoyer $\wedge_{m \in M[a,b]}(R(m) = R(a))$
function InférerSousTable(M, R, ρ/n, a, b)
  if TousIdentiques(M, R, a, b) **then**
    renvoyer $\{(\rho/n, R(a))\}$
  else
    $(m_0, m_1) \leftarrow$TrouverPivot(R, ρ/n, a, b)
    $D_0 \leftarrow$InférerSousTable(M, R, $\rho.0/(n+1)$, a, m_0)
    $D_1 \leftarrow$InférerSousTable(M, R, $\rho.1/(n+1)$, m_1, b)
    renvoyer $D_0 \cup D_1$
function InférerTable(M, R)
  renvoyer InférerSousTable(M, R, $0/0$, $\min M$, $\max M$)

4.3.3 Schéma AS

Nous avons vu que le schéma le plus spécifique et le schéma le moins spécifique réalisent tous les deux des hypothèses extrêmes quant à la représentativité de l'ensemble de moniteurs. Le premier ne s'autorise aucune hypothèse et sa portée est donc très limitée. Inversement, le second fait des hypothèses extrêmement optimistes et sa fiabilité dépend énormément de la qualité de l'ensemble des moniteurs. En particulier, si l'ensemble des moniteurs est réduit à quelques moniteurs mais dont les premiers bits sont très bien répartis, alors on va inférer des règles avec des préfixes très courts, donc très généraux. Pour limiter le risque d'inférer des préfixes trop courts pour être réalistes, et plus généralement pour se limiter à des préfixes qu'il semble réaliste de trouver dans une table de tranmission réelle, nous adaptons le schéma le moins spécifique en ajoutant une restriction : tous les préfixes conservés doivent correspondre à un préfixe revendiqué par un *AS*.

En pratique, la méthode d'inférence est sensiblement la même que celle décrite dans l'**Algorithme 2**. Pour l'adapter, au lieu de renvoyer le préfixe ρ/n dans InférerSousTable, on consulte un arbre binaire de recherche V représentant l'ensemble des préfixes revendiqués par des *AS* (cette opération a une complexité négligeable si l'arbre de recherche équilibré est précalculé). Si ρ/n est un tel préfixe, on le renvoie effectivement ; sinon, cela signifie que le préfixe n'est pas assez spécifique (trop court) pour correspondre à un préfixe d'*AS*, et on poursuit la division. Le résultat de cette adaptation est l'**Algorithme 3**. La liste des préfixes revendiqués par les *AS* (principalement dans le cadre de l'implémentation de BGP) est disponible auprès de sources publiques, comme par exemple Routeviews [7]

et Caida [10]. Elle peut être de taille conséquente (plusieurs centaines de millions d'entrées), mais la représentation sous forme d'un arbre binaire de recherche modélisant un ensemble peut-être calculée une seule fois pour un ensemble de cibles quelconque.

Algorithme 3 Inférence du schéma AS

function TrouverPivot(R, ρ/n, a, b)
  renvoyer $(\max\{m \in M[a,b], m \in \rho.0/(n+1)\}, \min\{m \in M[a,b], m \in \rho.1/(n+1)\})$
function TousIdentiques(M, R, a, b)
  renvoyer $\wedge_{m \in M[a,b]}(R(m) = R(a))$
function InférerSousTable(M, R, V, ρ/n, a, b)
  if (TousIdentiques$(M, R, a, b)) \wedge (\rho \in V)$ **then**
    renvoyer $\{(\rho/n, R(a))\}$
  else
    $(m_0, m_1) \leftarrow$TrouverPivot(R, ρ/n, a, b)
    $D_0 \leftarrow$InférerSousTable(M, R, $\rho.0/(n+1)$, a, m_0)
    $D_1 \leftarrow$InférerSousTable(M, R, $\rho.1/(n+1)$, m_1, b)
    renvoyer $D_0 \cup D_1$
function InférerTable(M, R)
  renvoyer InférerSousTable(M, R, $0/0$, $\min M$, $\max M$)

4.4 Mesure réelle avec *Planetlab*

Afin de vérifier la faisabilité de notre méthode, nous avons réalisé une mesure UDP Ping distribuée avec pour objectif d'appliquer notre méthode d'inférence des tables de transmission à un ensemble de cibles réelles. Nous détaillerons d'abord les conditions de cette mesure (**Section 4.4.1**), puis les résultats que nous en avons tiré à l'aide de notre méthode d'inférence (**Section 4.4.2**).

4.4.1 Conditions de la mesure

La faisabilité pratique d'une mesure UDP Ping distribuée a déjà été constatée lors d'une expérience réelle décrite au **Chapitre 3** ‡. Cependant, nous avons choisi de réaliser une nouvelle mesure pour expérimenter notre méthode. La première raison est que pour l'inférence du schéma *AS* décrit en **Section 4.3.3**, nous devons disposer d'une liste des préfixes revendiqués par les *AS* sur la même période de temps que la mesure. La seconde raison est que nous pratiquons une mesure

‡. **Section 3.6**

beaucoup plus focalisée. Là où nous visions à collecter un maximum de cibles correspondant à un tirage uniformément aléatoire dans le but de réaliser une estimation de la distribution de degré au **Chapitre 3**, l'objectif est cette fois-ci d'obtenir des tables de transmission aussi exactes que possible. Nous avons donc limité notre ensemble de cibles à un ensemble que nous avions préalablement construit comme répondant très bien aux sondes UDP Ping, et au lieu d'envoyer une unique sonde UDP Ping depuis chaque moniteur vers chaque cible, nous avons répété l'envoi de sondes plusieurs fois pour capturer l'équilibrage de charge réalisé par chaque cible.

Pour réaliser notre mesure, nous avons à nouveau sollicité l'infrastructure *Planetlab*. Au moment de notre prélèvement, notre ensemble de moniteurs M était composé de 548 moniteurs répartis dans 193 *AS*. Notre ensemble de cibles T était constitué de 2276 cibles extraites d'une mesure antérieure et retenues pour avoir une bonne responsivité à UDP Ping. Notre mesure a consisté en une série de 30 répétitions d'UDP Ping depuis chaque moniteur vers chaque cible, concentrées sur un total d'environ 10 minutes.

4.4.2 Résultats de la mesure

L'aggrégation des résultats nous a fourni une liste L de triplets de la forme $(m, m(t), t)$. Nous avons calculé pour chaque moniteur m et pour chaque cible t l'ensemble $R_t(m) = \{m(t), (m, m(t), t) \in L\}$ des interfaces utilisées par t pour répondre à m. Par ailleurs, nous avons calculé l'arbre binaire de recherche de M, utilisé par nos méthode d'inférence, et l'arbre binaire de recherche V basé sur les données *Routeviews* les plus récentes au même moment. À l'aide de ces données, nous avons appliqué les 3 méthodes d'inférence décrites en **Section 4.3**.

Notre première observation porte tout simplement sur la *taille* des tables inférées, c'est à dire sur le *nombre de règles*, en fonction de la méthode d'inférence utilisée (**Figure 4.1**). Comme attendu, plus le schéma est spécifique, plus la table de transmission inférée possède d'entrées. En effet, les hypothèse généralisatrices permettent de fusionner des sous-arbres possédant les mêmes étiquettes dans l'algorithme d'inférence. Plus il y a d'hypothèses généralisatrices, moins il y a donc de sous-arbres nécessitant l'ajout d'une règle spécifique. Ceci est bien entendu cohérent avec l'utilisation pratique des préfixes *CIDR* qui sont historiquement conçus précisément pour limiter la taille des tables de tranmission.

Notre seconde observation porte sur l'impact du nombre de moniteurs. Comme nous l'avons suggéré précédemment, la répartition de l'ensemble des moniteurs dans l'espace IP, en particulier la répartition des bits de poids fort, affecte considérablement la table de transmission inférée pour une cible donnée. Pour témoigner de l'ampleur du phénomène, nous avons émulé des mesures sur divers nombre de moniteurs en restreignant notre analyse à des jeux de données partiels limités à

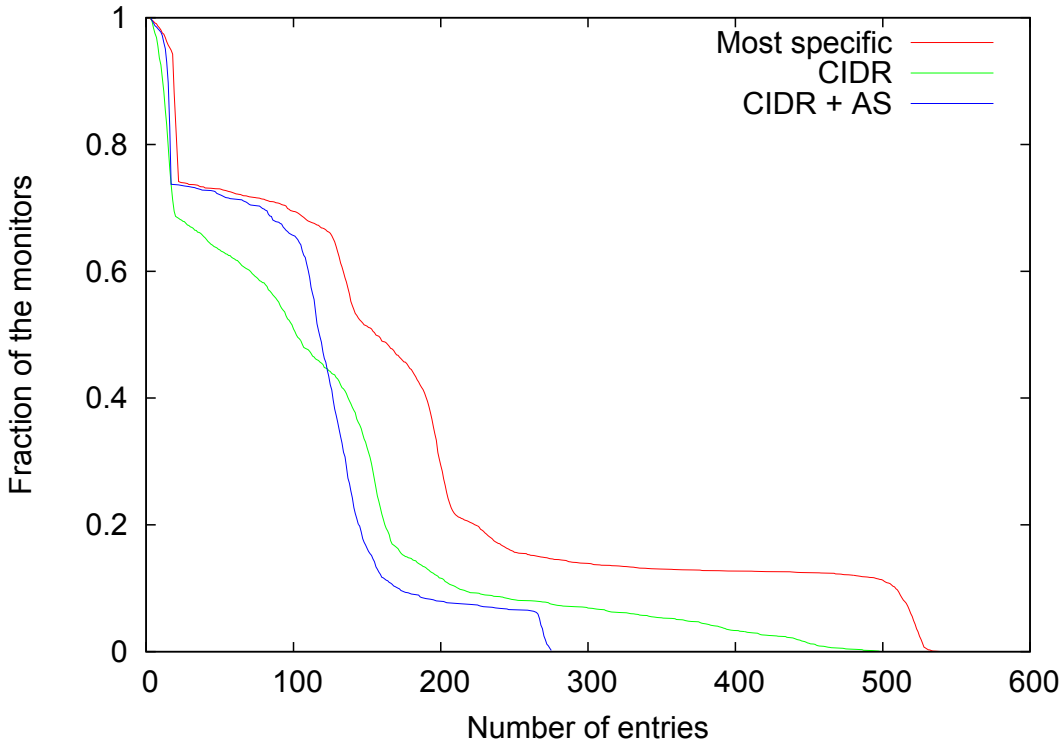


Figure 4.1 – *Pour chacune des 3 méthodes d'inférence, on calcule la distribution cumulative inverse du nombre d'entrées dans les tables de transmission. Pour chaque point x en ordonnée, on trace le nombre de cibles dont la table inférée comporte un nombre d'entrées supérieur ou égal à x.*

une fraction des moniteurs dont nous disposions réellement. Nous avons tracé la distribution cumulative inverse de la taille des tables inférées pour chaque schéma d'inférence, pour plusieurs fractions de notre ensemble de moniteurs totalisant $|M| = 548$ hôtes (**Figure 4.2**, **Figure 4.3**, **Figure 4.4**). Ces figures montrent que le nombre de moniteurs utilisés est trop limité pour obtenir une information aussi riche que celle que nous recherchons, car on observe une sensible différence entre la distribution pour $p = 0.9$ et $p = 1.0$ pour les trois schémas. Cependant, la forte cohérence entre les aspects des distributions pour des valeurs de p distinctes suggèrent que leurs formes générales correspondent au moins à une caractéristique locale.

4.5 Conclusion

Notre travail sur UDP Ping (**Chapitre 3**) nous a permis de mesurer une propriété du cœur de la topologie physique d'Internet, sa distribution de degrés. Notre travail dans ce chapitre a démontré qu'UDP Ping nous permet d'obtenir une information plus complète et plus précise, puisqu'elle permet, en plus de les *quantifier*,

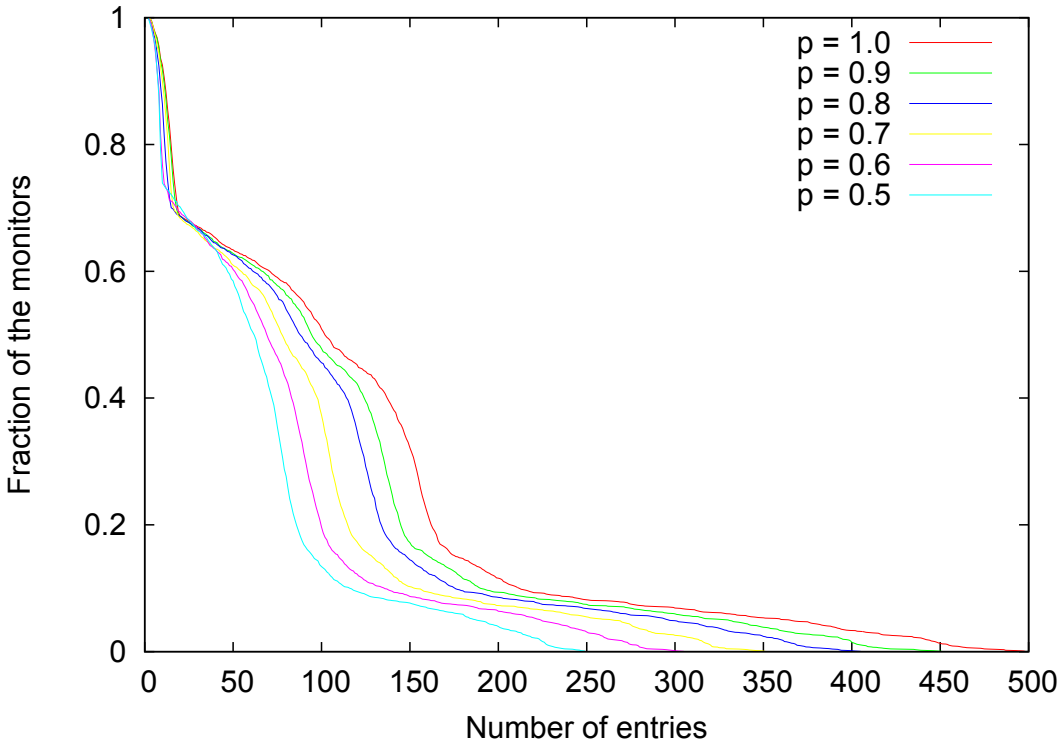


Figure 4.2 – *Pour chaque fraction p, on calcule la taille des tables inférées avec le schéma le moins spécifique en se restreignant à $p|M|$ moniteurs. Pour chaque point x en ordonnée, on trace le nombre de cibles dont la table inférée comporte un nombre d'entrées supérieur ou égal à x.*

de *qualifier* les interfaces d'un routeur du cœur. En effet, nous avons montré qu'une utilisation judicieuse d'un ensemble de moniteurs permet de déterminer, au moins dans certains cas, *comment* ces interfaces sont utilisées. Cette information est plus riche et peut être injectée dans un modèle, formel ou simulatoire, plus représentatif du réseau, particulièrement s'il vise à modéliser la manière dont les paquets circulent à travers les routeurs du cœur.

Cette approche possède des limites, dans la mesure où, selon la confiance (ou la connaissance *a priori*) que l'on a de l'ensemble des moniteurs dont on dispose, on peut estimer le comportement d'un routeur de manière plus ou moins complète. Dans son interprétation la plus prudente, cette approche ne permet de caractériser le comportement d'une cible que vis à vis des moniteurs uniquement. Si l'on s'autorise à postuler que chaque moniteur est représentatif de l'unité administrative de routage dont il fait partie, caractérisée par ou plusieurs préfixes *CIDR*, cependant, on peut sensiblement augmenter la portée de l'approche. Dès lors, notre méthode offre une information pertinente sur la réunion des unités dans lesquelles on dispose d'au moins un moniteur.

À l'inverse, la réflexion sous-jacente à ces conclusions nous permet d'envisager une nouvelle manière de caractériser la qualité d'un ensemble de moniteurs, en

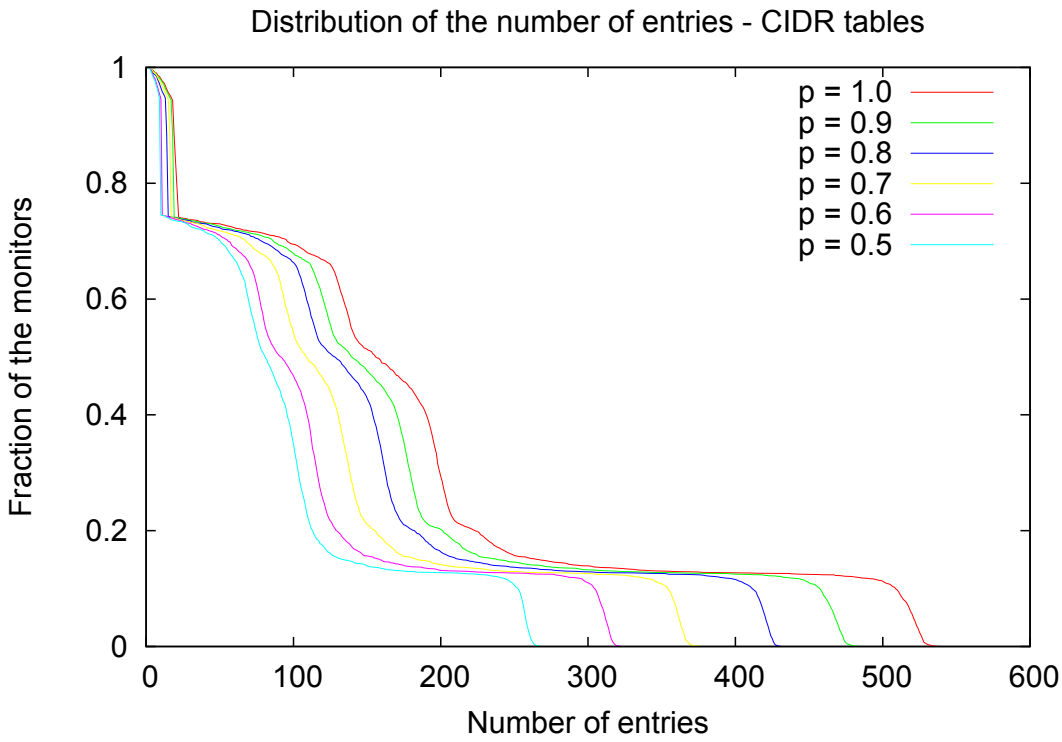


Figure 4.3 – *Pour chaque fraction p, on calcule la taille des tables inférées avec le schéma le plus spécifique en se restreignant à $p|M|$ moniteurs. Pour chaque point x en ordonnée, on trace le nombre de cibles dont la table inférée comporte un nombre d'entrées supérieur ou égal à x.*

considérant les classes de colocalisation induites par ces unités administratives. Si un moniteur est interprété comme un représentant d'une unité administrative, alors deux moniteurs représentant la même unité administrative sont redondants vis à vis d'UDP Ping. Réciproquement, si deux moniteurs ne *font pas* partie de la même unité administrative, alors *a priori* ils ne sont pas redondants, et s'ils *apparaissent* redondants *a posteriori* pour une cible donnée, c'est que cette cible se comporte de manière identique vis à vis de leurs deux unités.

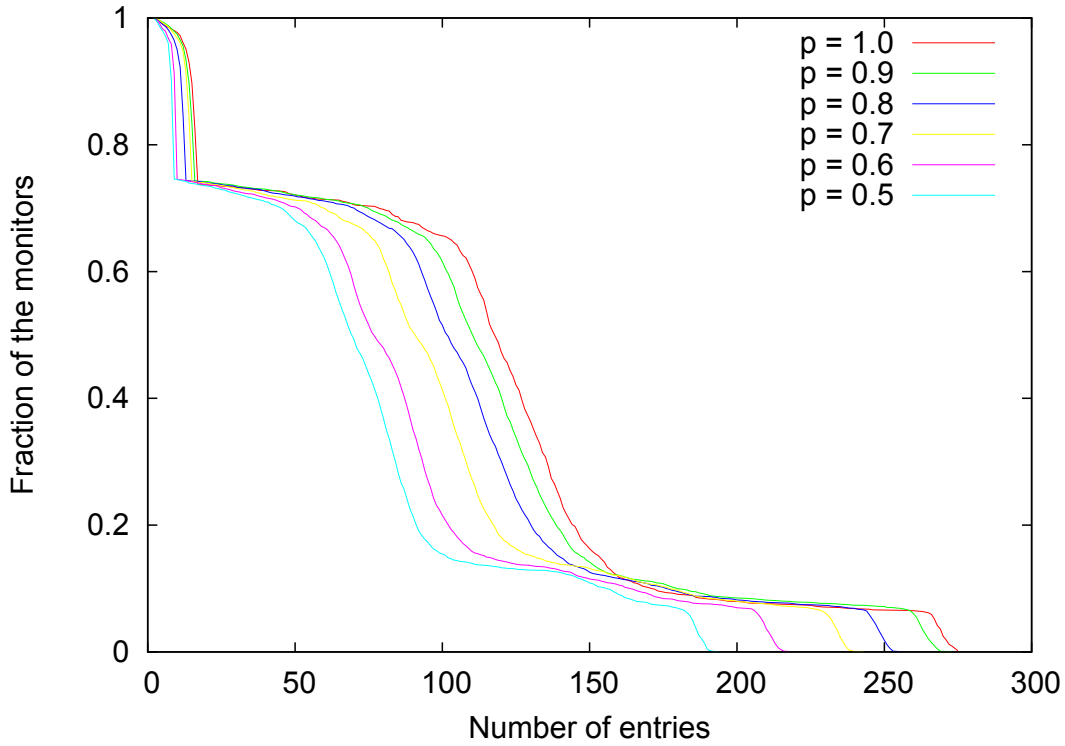


Figure 4.4 – *Pour chaque fraction p, on calcule la taille des tables inférées avec le schéma* AS *en se restreignant à $p|M|$ moniteurs. Pour chaque point x en ordonnée, on trace le nombre de cibles dont la table inférée comporte un nombre d'entrées supérieur ou égal à x.*

Chapitre 5

Conclusions et perspectives

Dans cette dernière partie, nous ferons le bilan de nos contributions puis nous détaillerons les perspectives que nous avons identifiées à partir des travaux que nous avons menés.

5.1 Contributions

L'une des préoccupations principales tout au long de ce travail de thèse a été de tenter de s'abstraire d'un grand nombre d'hypothèses implicites du domaine, pour les réexaminer à l'aune d'une approche qui se voulait à la fois rigoureuse et pragmatique. Notre première contribution a donc été d'adopter une posture volontairement agnostique et de **reprendre l'interprétation et la formalisation de nombreux objets**. L'interprétation rigoureuse de traceroute (**Chapitre 2**), le fonctionnement d'UDP Ping (**Chapitre 3**), et l'interprétation des tables de transmission (**Chapitre 4**), résultent de cette approche sciemment naïve. Notre contribution se situe notamment dans une description pragmatique qui tient compte de l'interprétation assez libérale des RFCs et autres documents normatifs par les acteurs réels. Notre interprétation nous a permis de mieux comprendre les limites de certaines approches historiques, mais également d'ouvrir la voie pour de nouvelles approches.

C'est ainsi que nous avons décidé d'utiliser traceroute non pas pour collecter des cartes, mais pour tenter de **mesurer directement une propriété topologique qui nous intéressait** : la distribution de degré. Dans ce contexte, nous avons dû remettre en question la définition même de distribution de degré, et distinguer différentes modélisation du même objet, en l'occurence le réseau L2 et le réseau L3. La méthode de mesure que nous avons établi avec traceroute nous permet d'**obtenir la liste d'au moins une adresse de chacun des voisins au niveau** L3 **d'un routeur cible**, pour un coût total (en terme de charge réseau) très limité, une approche validée par la simulation. Toutefois, cette approche suppose que le voisinnage de la cible ne bloque pas les paquets ICMP, une hypothèse qui tend à être remise en cause.

Cette limitation nous a incité à réaliser une analyse de l'échantillonage des degrés dans un graphe, et nous a permis de conclure qu'il suffisait pour estimer

correctement la distribution de degré d'un échantillon de taille relativement limitée (quelques milliers de nœuds), pour peu qu'on dispose d'une estimation très précise du degré de chaque nœud. C'est ainsi que nous avons conçu la primitive UDP Ping (**Chapitre 3**) : une **primitive de mesure très exigeante sur la cible, mais qui estime très précisément son degré**. Afin d'en tirer l'estimation de la propriété qui nous intéressait, nous avons également réalisé un **protocole de mesure**, capable non seulement d'obtenir la propriété recherchée (le nombre d'interfaces de la cible), mais également de détecter les cas pouvant biaiser la distribution finale. Nous avons notamment implémenté un outil dérivé d'UDP Ping, UDP Explore, qui permet d'identifier la liste des hôtes séparant un moniteur du cœur de la topologie physique. Un des aspects les plus importants de notre contribution porte précisément sur une analyse rigoureuse des effets de certaines configurations topologiques sur notre primitive de mesure, comment les détecter, et comment les corriger. Cette analyse est validée par des simulations sur plusieurs modèles de topologies, en particulier des topologies homogènes et hétérogènes. Une de nos conclusions est que nous sommes capables d'**estimer avec une très grande confiance les fractions de nœuds de faible degré (<10)** dans le cœur de la topologie physique, et qu'à l'inverse, nous sommes capables de **majorer avec une grande confiance la fraction des nœuds de degré supérieur à un certain degré** d, en extrapolant leur fréquence d'apparition théorique à partir de leur fréquence d'apparition dans notre échantillon. Cette dernière contribution est une contribution importante, notamment puisqu'elle repose sur un raisonnement solide et des hypothèses très clairement établies.

En complément de l'objet que nous souhaitions initialement mesurer, la distribution de degré, notre primitive UDP Ping nous a permis d'obtenir une information plus précise que le nombre de leurs interfaces pour les cibles qui y répondent. En effet, dans une certaine mesure, nous avons établi une manière d'**estimer la table de transmission** (**Chapitre 4**) d'un routeur qui répond aux sondes UDP Ping. Si cette information n'est pas nécessairement simple à exploiter avec des modèles de graphes très généralistes, elle donne indiscutablement une information beaucoup plus riche que le simple degré sur la manière dont l'information *circule* à travers le réseau.

Outre les résultats que nous avons obtenu sur la distribution de degré aux niveaux L2 et L3, et les tables de transmission, cette thèse a été l'occasion d'aborder la problématique plus générale de la métrologie des grands réseaux. Nous avons esquissé, à travers la mesure d'Internet, la problématique de **l'estimation d'une propriété topologique d'un grand réseau réel à l'aide d'une primitive de mesure**. La méthodologie que nous avons employée ici puise largement dans une connaissance du domaine d'application spécifique du réseau Internet, mais elle peut être conceptuellement étendue à beaucoup d'autres cadres ; c'est d'ailleurs l'une de nos principales perspectives (**Section 5.2.2**). Plus généralement, nous avons effectué un travail exploratoire, qui, au-delà des résultats direct, a levé le

voile sur un certain nombre de perspectives. La **description et l'organisation de ces perspectives** présentée ci-dessous constitue une contribution propre.

5.2 Perspectives

Nous avons organisé les perspectives suscitées par nos travaux en trois catégories : celles qui relèvent de l'approfondissement de l'utilisation d'UDP Ping (**Section 5.2.1**), celles qui généralisent l'approche d'échantillonage orienté propriété (**Section 5.2.2**) à d'autres propriétés ou d'autres réseaux, et enfin, des nouveaux objets de la topologie d'Internet (**Section 5.2.3**) qui nous semblent être d'un intérêt particulier.

5.2.1 Approfondissement d'UDP Ping

L'outil UDP Ping que nous avons mis au point et dont nous avons analysé le fonctionnement représente une primitive de mesure précise et fiable, que nous avons conçu pour identifier les interfaces des routeurs du cœur (**Chapitre 3**) et qualifier leur utilisation (**Chapitre 4**). Cependant, un certain nombre de questions restent ouvertes et pourraient faire l'objet d'approfondissements.

5.2.1.1 Validation de la mesure

L'une des perspectives les plus importantes pour la validation de notre travail, outre les garanties théoriques et simulatoires que nous avons déjà présenté, serait de confronter les interfaces identifiées par UDP Ping avec la liste des interfaces réelles d'un routeur du cœur.

La mesure réalisée par UDP Ping est d'autant plus difficile à valider qu'il n'existe bien sûr pas d'annuaire public des interfaces des routeurs du cœur (sans quoi la mesure serait d'un intérêt très limité). Pour valider la liste des interfaces obtenues avec UDP Ping distribué, il faudrait disposer d'un accès à la liste des interfaces d'un routeur du cœur qui répond à UDP Ping. Or, les routeurs du cœur appartiennent le plus souvent à des institutions de grande ampleur, qui ne sont pas nécessairement enclines à confier un accès de cette nature. Les routeurs du cœur sont en effet souvent des machines extrêmement onéreuses (les prix se chiffrent au minimum en dizaines de milliers de dollars, allant jusqu'à plusieurs centaines de milliers) et leur configuration peut être considérée comme une information stratégique. Valider le fonctionnement d'UDP Ping distribué de cette manière suppose donc un travail délicat, afin de convaincre l'autorité responsable d'un tel routeur de fournir la liste de ses interfaces et, idéalement, sa table de transmission. Cette perspective nous semble toutefois importante dans la mesure où elle permettrait de valider ou d'invalider clairement les résultats de notre mesure et notre méthode.

5.2.1.2 Dynamique et mesure longue

Notre travail avec UDP Ping avait pour objectif de capturer une estimation de la distribution de degré des routeurs du cœur de la topologie physique à un instant donné, et à cet effet, nous avons paramétré notre mesure afin qu'elle soit la plus brève possible (sans surcharger les cibles) pour minimiser l'effet de la dynamique du réseau.

Pourtant, puisque nous avons justifié la validité de notre protocole de mesure "instantané", on peut précisément envisager de répéter cette mesure régulièrement au cours du temps, pour capturer la dynamique à moyen et long terme. Une hypothèse fréquente est que le cœur d'Internet est relativement statique, mais cette hypothèse n'est pas étayée par des mesures fiables. Nous proposons deux protocoles complémentaires afin d'explorer cette piste. Le premier consiste à se fixer un ensemble de cibles répondant à UDP Ping et de répéter la mesure UDP Ping distribué à intervalle régulier pendant une longue durée, par exemple toutes les 24h pendant 6 mois, et d'observer l'évolution de la liste des interfaces de chaque cible pendant cette période. Cette mesure permettrait de mesurer l'évolution de la liste des interfaces de *chaque* routeur de notre ensemble de cibles. Le second protocole, complémentaire du premier, consiste à s'intéresser non pas à un ensemble de cibles fixé, mais à évaluer à la même fréquence la distribution de degré des routeur du cœur de la topologie physique, c'est à dire à réaliser un nouvel échantillonage uniforme à chaque itération. De cette manière, on bénéficie du travail d'analyse que nous avons décrit au **Chapitre 3**, et on peut s'intéresser à l'évolution de la *distribution* plutôt qu'à l'évolution des listes d'interfaces, plus exposées aux artefacts de mesure. Ce second protocole est cependant plus coûteux pour le réseau, puisque la construction d'une liste uniforme de cibles requiert l'envoi d'un très grand nombre de sondes UDP Ping à la recherche de cibles y répondant.

5.2.1.3 Autres ensembles de moniteurs

L'une des problématiques mises en évidence par tous nos travaux sur la mesure de la topologie d'Internet est l'importance de l'ensemble des moniteurs, ou points d'observation, depuis lesquels on mesure la topologie. C'est l'une des principales faiblesses de la méthode historique, et c'est ce constat qui en premier lieu nous a servi de piste pour mettre au point nos méthodes de mesure distribuées.

Pour réaliser nos mesures, nous avons utilisé à plusieurs reprises l'infrastructure PlanetLab. C'est une infrastructure riche et très libérale dans ses conditions d'utilisation, qui nous a permis d'établir la faisabilité et la pertinence de notre approche. Cependant, elle reste limitée, à la fois en termes de nombre de moniteurs (quelques centaines), et surtout en termes de diversité des sous-réseaux. En effet, ils n'appartiennent qu'à une collection relativement limitée d'AS (environ 200 au moment de nos mesures) et pour beaucoup sont hébergés dans des réseaux

académiques. En conséquence, on peut craindre qu'ils ne fournissent qu'une vision partielle du réseau. S'il est peu vraissemblable que cela impacte l'estimation de la fraction des nœuds de degrés faibles, en revanche, il est clair que cela peut avoir un effet sur l'estimation de la fraction des nœuds de degré fort. Nos travaux sur les tables de tranmission suggèrent également que dans un tel cadre, l'ensemble des moniteurs de PlanetLab est insuffisant pour obtenir une vision fiable de l'objet mesuré.

Les prérequis pour effectuer une mesure UDP Ping depuis une machine hôte sont assez simples, mais exigeants : il faut pouvoir émettre des paquets UDP arbitraires, et surtout disposer d'un accès non filtré au traffic ICMP entrant pour pouvoir écouter les réponses. Le premier critère est assez facile à obtenir, mais le second exige souvent des privilèges particuliers sur un réseau qui ne filtre pas le traffic ICMP. Nous avons envisagé plusieurs options, dont certaines sont plus ou moins faciles à mettre en oeuvre :

- *DIMES* [60] : Le projet DIMES rassemble des participants volontaires exécutant sur leur machine, souvent personelle, un programme agent qui permet d'exécuter des mesures. Il est aujourd'hui surtout utilisé pour collecter des résultats de traceroute, mais pourrait être adapté pour réaliser des mesures UDP Ping.
- *Looking glasses*, *routeviews* [7] : Mis à la disposition du public principalement afin de partager des informations relatives aux protocoles de routages par les fournisseurs d'accès à Internet ou les points d'échange Internet, les serveurs de *looking glasses* fournissent en général la possibilité de consulter un annuaire BGP, ou d'exécuter ICMP ping, TCP ping ou traceroute. Déployer UDP Ping sur ces serveurs requiert l'accord de parties privées ou industrielles, mais offrirait un point de vue très riche, au sein du cœur d'Internet.
- *RIPE Atlas* [47] : *RIPE Atlas* déploie gratuitement, auprès d'institutions volontaires, un serveur de mesures embarqué, constitué d'une puce alimentée par USB et connectée à Internet via un port RJ45 traditionnel. Sa puce contient un système d'exploitation simplifié dédié à effectuer des mesures telles que ping ou traceroute. À mesure de son déploiement, le réseau RIPE Atlas pourrait devenir un complément intéressant à PlanetLab.
- *CAIDA Ark* [10] : L'infrastructure *Ark* utilisée par *CAIDA* pour réaliser ses prélèvements traceroute distribués est semblable à celle de *RIPE Atlas* et de *PlanetLab* : des institutions volontaires hébergent des machines dédiées à réaliser des prélèvements distribués et comprend plusieurs dizaines de moniteurs répartis dans le monde. Une version embarquée basée sur le *Raspberry Pi* est très proche des sondes *RIPE Atlas*. Certains moniteurs *Ark* ont déjà été utilisés pour des mesures tierces, comme par exemple *MERLIN* [44]. Cette infrastructure est déjà coutumière des mesures distribuées à très grande échelle, et serait donc un complément naturel à PlanetLab, bien que ses moniteurs soient vraissemblablement au moins en partie redondants.

- Hébergeurs commerciaux : Afin de proposer des CDNs * efficaces, de nombreux hébergeurs commerciaux, tels qu'Amazon Web Services [35] ou Microsoft Azure [45], ont déployé des serveurs localisés dans le monde entier, et offrent des machines virtuelles ou dédiées à leurs clients. Bien que le nombre de sites soit très limité (quelques dizaines), ils sont choisis par leurs propriétaires précisément pour leur bonne répartition dans le réseau, et pourraient donc offrir une vision de la topologie complémentaire à celle fournie par les hôtes académiques.
- Extension de navigateur ou applet : Les navigateurs Internet ne fournissent pas d'API exposant les fonctionalités de réseau bas niveau (particulièrement pas l'écoute d'ICMP), mais en revanche, les extensions de navigateur ou les applets (Java, Flash, Silverlight…) peuvent, avec l'autorisation du client, exécuter du code relativement arbitraire. Développer un agent de mesure sous cette forme, pour UDP Ping comme pour d'autres primitives, pourrait être un complément crédible à un projet tel que *DIMES*. Son intérêt résiderait dans la simplicité pour l'utilisateur de participer à des mesures, mais il pose la question de la faisabilité de son déploiement à grande échelle.
- Application mobile : Les appareils connectés mobiles, tels que les téléphones ou les tablettes, exposent le plus souvent aux applications un accès, même limité, à la couche réseau. Ces terminaux présentent un intérêt en termes de diversité des localisations et des routes, mais également un certain nombre de difficultés. Sur les systèmes d'exploitations les plus utilisés du marché, l'accès aux *sockets* privilégiés n'est pas disponible par défaut. Certaines technologies de connectivité sans fil passent par des réseaux virtuels (opaques aux niveau L2 et L3), ce qui complexifie l'interprétation des mesures. Enfin, ces appareils ont souvent des capacités en énergie limitées, et une activité réseau prolongée peut induire une forte consommation.
- Sondes embarquées : À la lisière entre l'application mobile et les sondes de *RIPE Atlas*, les *dongles* ARM † semblent être une opportunité très prometteuse. Fabriqués à très bas coût (de l'ordre entre 5€ et 20€ à l'unité), ces terminaux de quelques centimètres carrés alimentés en USB embarquent des architectures ARM largement assez performantes pour réaliser des mesures réseau, et même effectuer une partie de leur traitement localement. Plusieurs systèmes d'exploitations très documentés supportent ce type de terminaux et rendent trivial le portage des primitives de mesure. Tout comme pour les sondes *RIPE Atlas*, il se pose la question du déploiement et de la maintenance occasionelle.

Pour toutes ces solutions, un problème commun demeure : la difficulté pour un réseau ciblé (ou simplement traversé) par les sondes UDP Ping à différencier une activité de mesure d'une activité hostile de type déni de service. En effet, les administrateurs de réseau sont souvent peu enclins à autoriser les hôtes à

*. *Content Delivery Networks*, réseaux de diffusion de contenu

†. À titre d'exemple, la clé multimedia *Google Chromecast* [36] repose sur un *dongle* ARM de ce type.

émettre des paquets UDP vers des destinations aléatoires, car ils s'exposent alors au risque du *blacklisting*. Cependant, de notre expérience pratique, seuls des réseaux mal configurés (principalement au niveau des tables UDP des pare-feux) sont susceptibles de tomber en panne à cause d'une mesure UDP Ping distribuée.

5.2.1.4 Approche complémentaire pour la bordure

La méthode de mesure du degré par des moniteurs distribués vers des cibles aléatoires est conçue pour mesurer le degré des nœuds du cœur d'Internet. En effet, comme détaillé au **Chapitre 2** et au **Chapitre 3**, les nœuds de la bordure d'Internet ont des interfaces intrinsèquement difficiles à observer à l'aide d'une mesure distribuée telle que celle que nous emloyons.

Nous pensons que cette limitation n'a que peu d'impact conceptuel, dans la mesure où la complexité du routage et la plupart des problématiques associées se situent dans le cœur d'Internet, mais afin d'établir la distribution de degré de la topologie complète et pas simplement du cœur, il faudrait être capable de mesurer la distribution de degré de la bordure d'Internet. Une approche complémentaire doit donc être menée, attachée spécifiquement à mesurer cette propriété. Comme par définition la bordure d'Internet est constituée d'arbres, une piste pourrait être de réaliser certaines hypothèses sur ces arbres (par exemple sur leur régularité), et de tenter de mesurer les paramètres typiques de ces modèles dans la réalité (par exemple la profondeur et le degré moyen de ces arbres).

5.2.2 Echantillonage orienté propriété

Notre approche est conceptuellement novatrice dans ce qu'elle s'attaque à mesurer directement une propriété topologique sur un réseau réel plutôt qu'à tenter d'établir une carte sur laquelle on lirait ensuite cette propriété. Plus précisément, l'approche est de bâtir une primitive de mesure qui permet d'estimer une certaine propriété $p(u)$ d'un nœud (ou d'une arete) u d'un sous-ensemble V de nœuds (ou d'aretes) du réseau G, et une méthode qui permet d'échantilloner des nœuds (ou des aretes) dans ce sous-ensemble, d'une manière qui soit uniforme vis à vis de la propriété mesurée, ou au moins dont la non-uniformité est maîtrisée (c'est à dire qu'on peut la corriger *a posteriori*). Dans le cas d'UDP Ping distribué, par exemple, la propriété estimée est le degré (nombre d'interfaces), le réseau G est le cœur physique d'Internet, et V est le sous-ensemble des nœuds répondant à UDP Ping.

Nous pensons que cette approche simple peut être déclinée, à la fois à d'autres propriétés ou d'autres réseaux, et avec d'autres méthodes d'échantillonage.

5.2.2.1 Application à d'autres réseaux

De nombreux réseaux, à l'instar d'Internet, reposent sur une indexation à 32 bits ou moins, et à ce titre, sont aisés à échantilloner de manière uniforme. C'était par exemple le cas des utilisateurs Facebook jusqu'à 2007, des commentaires sur le site Slashdot, ou encore des utilisateurs du jeu League of Legends sur une région donnée. La plupart de ces réseaux offrent une API publique HTTP, mais dont l'accès est souvent strictement limité en nombre de requêtes par unité de temps. Leur cartographie complète est donc peu vraissemblable, mais un échantillonage adapté peut permettre d'estimer des propriétés topologiques des réseaux sous-jacents, pour peu que l'on démontre l'absence de biais indirect lié à l'attribution d'un identifiant (par exemple, lié à l'ancienneté d'un compte).

5.2.2.2 Marche aléatoire orientée propriété

Hormis l'échantillonage uniforme lié à l'indexation numérique, une autre méthode d'échantillonage intéressante et plus spécifique aux graphes est la marche aléatoire. La difficulté repose alors sur l'uniformité de l'échantillonage relativement à la propriété mesurée, mais une étude rigoureuse peut souvent permettre de s'en abstraire. Cette approche a par exemple été mise en oeuvre dans le cas de l'échantillonage sans biais relatif au degré dans le cas de Facebook [24]. Des techniques similaires peuvent également s'appliquer à des réseaux qu'il est naturel de parcourir avec des marches aléatoires, comme par exemple les réseaux de pages web ou les réseaux P2P [‡].

5.2.3 Nouveaux objets d'intérêt

5.2.3.1 Réseau de routage pondéré

Au moment de valider notre première mesure TRACEROUTE distribuée, et à nouveau pour valider la mesure UDP PING distribuée, nous avons été confrontés à la même interrogation : que penser d'une interface d'une cible qui n'est observée que par un seul ou un très petit nombre de moniteurs ? Cette interface est-elle un faux positif, c'est à dire une interface n'appartenant pas réellement à la cible mais observée par erreur par un petit nombre de moniteurs, ou une interface "rare", difficile à observer ? Dans ce dernier cas, comment justifier le peu de robustesse de la notion de degré dans un cas où manifestement, on aurait aisément pu "rater" cette interface en supprimant un seul moniteur ?

La question plus générale qui se pose est celle de l'importance relative des interfaces d'un routeur (ou des arêtes d'un nœud) dans leur activité de routage.

‡. *Peer to peer*, pair-à-pair.

On peut concevoir cette idée soit *a priori*, comme une *probabilité d'utilisation* ou une table de routage, soit *a posteriori*, comme un *taux d'utilisation*. Dans les deux cas il s'agit de considérer la matrice d'adjacence de la topologie physique, pondérée par des réels qui représentent l'importance d'une interface dans son activité de routage. Cette notion, qui nous semble être le réel objet d'importance derrière la distribution de degré pour l'étude du routage, qu'il soit en modélisation ou en analyse, a inspiré nos travaux sur les tables de tranmission, mais il reste à compléter, notamment dans son approche formelle et son lien avec les algorithmes de routage. Une estimation de cette matrice sur le réseau réel d'une part, sur des graphes synthétiques d'autre part, nous permettrait de valider ou d'invalider des modèles de topologies.

5.2.3.2 Topologie égo-centrée avec UDP Explore

Bien que les sondes d'UDP Ping soient à la base conçue afin de lister toutes les interfaces dans le cœur d'une cible donnée à partir d'un ensemble distribué de moniteurs, nous avons montré que nous pouvions utiliser des sondes similaires sous la forme de l'outil UDP Explore. UDP Explore est utilisé dans notre travail comme un simple outil de validation pour détecter le cas où deux moniteurs sont dans le même sous-arbre de la bordure, mais il permet de lister depuis un moniteur d'une part toutes les interfaces tournées vers le cœur qui appartiennent à des nœuds soit dans le cœur, soit dans d'autres sous-arbres, d'autre part toutes les interfaces tournées vers le moniteur de nœuds qui sont plus haut dans le même sous-arbre.

Plus généralement, l'objet auquel donne accès les sondes UDP Ping depuis un moniteur est l'ensemble des interfaces à une certaine distance d en termes de *hops* au niveau IP d'un nœud et qui sont soit des interfaces tournées vers le moniteur de nœuds dans le même sous arbre que lui tournées vers le cœur, soit des interfaces tournées vers le cœur de nœuds qui ne sont pas dans le même sous-arbre. En plus d'un intérêt propre à cet objet (qui donne en quelque sorte la vision projetée d'un nœud sur la topologie du cœur), c'est un critère de plus pour valider ou invalider un modèle de topologie synthétique, puisqu'il s'agit d'une propriété qu'on peut également calculer sur un réseau formel ou un réseau simulé et la confronter aux observations. Cette approche peut compléter celle qui a déjà été menée par Latapy *et al.* [42].

5.2.3.3 Routes longues

Lors de nos travaux préliminaires sur traceroute, nous avons été amenés à réaliser des études sur la longueur des routes sous-jacentes à nos observations. Plus précisément, nous avons constaté que si nous répétions traceroute depuis

un moniteur donné vers une cible donnée, alors le TTL nécessaire avant d'atteindre la cible pouvait montrer une variabilité.

Or, TRACEROUTE s'arrête généralement d'envoyer des sondes dès qu'une sonde ICMP répond. Des travaux ultérieurs ont montré que des sondes ICMP successives pouvaient emprunter des routes différentes, et qu'une sonde de TRACEROUTE peut être interprété comme un certain tirage aléatoire d'une des routes parmi les routes possibles. Considérons le cas simplifié où seules deux routes a et b existent entre le moniteur m et la cible t, de longueurs respectives $|a| = 10$ et $|b| = 20$, d'une probabilité uniforme de $\frac{1}{2}$. Dès la sonde de TTL 10, il suffit qu'une seule sonde emprunte la route a pour que TRACEROUTE s'arrête ; or pour observer la route b dans son intégralité, il faut au moins atteindre le TTL 20. En particulier, la probabilité d'observer le saut $|b| - 1$ avec TRACEROUTE est majorée par $\frac{1}{2^{|b|-1-|a|}} = \frac{1}{512}$. La probabilité d'observer le voisin de t induit par b est donc très faible.

TRACEROUTE n'est probablement pas l'outil adapté pour observer ces routes, mais dans le cas d'UDP PING, l'idée se tranpose en répétant de très nombreuses fois l'envoi de sondes depuis un même moniteur vers une cible donnée, dans l'espoir d'observer ces interfaces difficiles à observer. Les considérations de charge portée par le voisinage de la cible sont à étudier, mais l'étude des routes longues, qui sont occultées par la plupart des travaux basés sur TRACEROUTE, constitue selon nous une opportunité d'observer davantage d'informations, même depuis un seul moniteur.

Bibliographie

[1] Dimitris Achlioptas, Aaron Clauset, David Kempe, and Cristopher Moore. On the bias of traceroute sampling : or, power-law degree distributions in regular graphs. *J. ACM*, 56(4), 2009. (Cité en pages 23, 27 et 51.)

[2] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Internet Measurement Conference*, pages 153–158, 2006. (Cité en page 32.)

[3] F. Baker. Requirements for IP Version 4 Routers. RFC 1812 (Proposed Standard), June 1995. Updated by RFCs 2644, 6633. (Cité en page 28.)

[4] Paul Barford, Azer Bestavros, John W. Byers, and Mark Crovella. On the marginal utility of network topology measurements. In *IMW*, 2001. (Cité en page 23.)

[5] R. Braden. Requirements for Internet Hosts - Application and Support. RFC 1123 (INTERNET STANDARD), October 1989. Updated by RFCs 1349, 2181, 5321, 5966. (Cité en pages 13 et 64.)

[6] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (INTERNET STANDARD), October 1989. Updated by RFCs 1349, 4379, 5884, 6093, 6298, 6633, 6864. (Cité en pages 13, 28 et 64.)

[7] A. Broido and k. claffy. Analysis of RouteViews BGP data : policy atoms. In *NRDM*, 2001. (Cité en pages 106 et 117.)

[8] R. Bush and D. Meyer. Some Internet Architectural Guidelines and Philosophy. RFC 3439 (Informational), December 2002. (Cité en page 13.)

[9] CAIDA. "iffinder", a tool for mapping interfaces to routers. `http://www.caida.org/tools/measurement/iffinder`. (Cité en pages 23 et 64.)

[10] CAIDA. Macroscopic topology measurement projects, 2010. (Cité en pages 107 et 117.)

[11] IEEE 802 LAN/MAN Standards Committee. Ieee 802® : Local and metropolitan area network standards. (Cité en page 14.)

[12] Internet Systems Consortium. The isc domain survey, 2010. (Cité en pages 11 et 52.)

[13] PlanetLab Consortium. Planetlab : An open platform for developing, deploying and accessing planetary-scale services. `http://www.planet-lab.org/`, 2009. (Cité en pages 27, 52 et 84.)

[14] M. Cotton and L. Vegoda. Special Use IPv4 Addresses. RFC 5735 (Best Current Practice), January 2010. Obsoleted by RFC 6890, updated by RFC 6598. (Cité en page 16.)

[15] Christophe Crespelle and Fabien Tarissan. Evaluation of a new method for measuring the internet degree distribution : Simulation results. *Computer Communications*, 34(5), 2011. (Cité en page 77.)

[16] Luca Dall'Asta, J. Ignacio Alvarez-Hamelin, Alain Barrat, Alexei Vázquez, and Alessandro Vespignani. Exploring networks with traceroute-like probes : Theory and simulations. *Theor. Comput. Sci.*, 355(1), 2006. (Cité en pages 23 et 27.)

[17] Matthew B Doar. A better model for generating test networks. In *Global Telecommunications Conference, 1996. GLOBECOM'96.'Communications : The Key to Global Prosperity*, pages 86–93. IEEE, 1996. (Cité en page 21.)

[18] B. Donnet, M. Luckie, P. Mérindol, and J. Pansiot. Revealing MPLS tunnels obscured from traceroute. *ACM SIGCOMM Computer Communication Review (CCR)*, 42(2), 2012. (Cité en pages 23 et 27.)

[19] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999. (Cité en page 23.)

[20] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, 1999. (Cité en page 27.)

[21] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785, 6266, 6585. (Cité en page 13.)

[22] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR) : The Internet Address Assignment and Aggregation Plan. RFC 4632 (Best Current Practice), August 2006. (Cité en page 101.)

[23] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR) : an Address Assignment and Aggregation Strategy. RFC 1519 (Proposed Standard), September 1993. Obsoleted by RFC 4632. (Cité en page 101.)

[24] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in facebook : A case study of unbiased sampling of osns. In *INFOCOM*. IEEE, 2010. (Cité en page 120.)

[25] Ramesh Govindan and Anoop Reddy. An analysis of internet inter-domain topology and route stability. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 2, pages 850–857. IEEE, 1997. (Cité en page 23.)

[26] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *INFOCOM*, 2000. (Cité en page 64.)

[27] Internet Engineering Steering Group and R. Hinden. Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR). RFC 1517 (Historic), September 1993. (Cité en page 101.)

[28] Jean-Loup Guillaume and Matthieu Latapy. Relevance of massively distributed explorations of the internet topology : simulation results. In *INFOCOM*, 2005. (Cité en page 23.)

[29] Jean-Loup Guillaume, Matthieu Latapy, and Damien Magoni. Relevance of massively distributed explorations of the internet topology : Qualitative results. *Computer Networks*, 50(16), 2006. (Cité en pages 23, 27 et 50.)

[30] Mehmet Hadi Gunes and Kamil Saraç. Resolving ip aliases in building traceroute-based internet maps. *IEEE/ACM Trans. Netw.*, 17(6), 2009. (Cité en pages 23, 42 et 60.)

[31] M.H. Gunes and K. Sarac. Importance of ip alias resolution in sampling internet topologies. In *IEEE Global Internet Symposium*, 2007. (Cité en pages 23, 31, 42 et 60.)

[32] Frank Heart, Alex McKenzie, JM McQuillan, and DC Walden. Arpanet completion report. *BBN Report. Bolt, Beranek and Newman Inc.(BBN). Also published in an edited version as BBN Report*, 4799 :58–63, January 1978. (Cité en pages 8, 19, 20, 21 et 22.)

[33] R. Housley, J. Curran, G. Huston, and D. Conrad. The Internet Numbers Registry System. RFC 7020 (Informational), August 2013. (Cité en page 70.)

[34] B. Huffaker, D. Plummer, D. Moore, and K. Claffy. Topology discovery by active probing. In *SAINT*, 2002. (Cité en page 64.)

[35] Amazon Web Services Inc. Amazon web services. `http://aws.amazon.com/`. (Cité en page 118.)

[36] Google Inc. Google chromecast. `https://www.google.com/intl/fr/chrome/devices/chromecast/`. (Cité en page 118.)

[37] Internet Systems Consortium Inc. Internet systems consortium survey. (Cité en pages 8, 19 et 22.)

[38] CIA U.N. Department of Economic ITU, World Bank Group and Social Affairs. Internet live stats. `http://www.internetlivestats.com/internet-users/`. (Cité en page 11.)

[39] Ken Keys. Internet-scale ip alias resolution techniques. *ACM SIGCOMM Computer Communication Review*, 40(1), 2010. (Cité en pages 23, 31, 42 et 60.)

[40] Anukool Lakhina, John W. Byers, Mark Crovella, and Peng Xie. Sampling biases in ip topology measurements. In *INFOCOM*, 2003. (Cité en pages 23, 27, 50 et 51.)

[41] Matthieu Latapy and Clémence Magnien. Complex network measurements : Estimating the relevance of observed properties. In *INFOCOM*, 2008. (Cité en pages 23 et 27.)

[42] Matthieu Latapy, Clémence Magnien, and Frédéric Ouédraogo. A radar for the internet. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 901–908. IEEE, 2008. (Cité en pages 23 et 121.)

[43] Pascal Merindol, Benoit Donnet, Olivier Bonaventure, and Jean-Jacques Pansiot. On the impact of layer-2 on node degree distribution. In *IMC*, 2010. (Cité en pages 23 et 27.)

[44] Pascal Mérindol, Benoit Donnet, J-J Pansiot, Matthew Luckie, and Young Hyun. Merlin : Measure the router level of the internet. In *Next Generation Internet (NGI), 2011 7th EURO-NGI Conference on*, pages 1–8. IEEE, 2011. (Cité en pages 23 et 117.)

[45] Microsoft. Microsoft azure. `http://azure.microsoft.com/`. (Cité en page 118.)

[46] J. Moy. OSPF Version 2. RFC 2328 (INTERNET STANDARD), April 1998. Updated by RFCs 5709, 6549, 6845, 6860. (Cité en page 100.)

[47] RIPE NCC. Ripe atlas. `https://atlas.ripe.net/`. (Cité en pages 96 et 117.)

[48] Jean-Jacques Pansiot and Dominique Grad. On routes and multicast trees in the internet. *SIGCOMM Comput. Commun. Rev.*, 28(1), 1998. (Cité en pages 23, 27 et 64.)

[49] Vern Paxson and Sally Floyd. Why we don't know how to simulate the internet. In *Proceedings of the 29th conference on Winter simulation*, pages 1037–1044. IEEE Computer Society, 1997. (Cité en page 23.)

[50] C. Perkins. IP Encapsulation within IP. RFC 2003 (Proposed Standard), October 1996. Updated by RFCs 3168, 6864. (Cité en page 60.)

[51] J. Postel. User Datagram Protocol. RFC 768 (INTERNET STANDARD), August 1980. (Cité en page 13.)

[52] J. Postel. Internet Control Message Protocol. RFC 792 (INTERNET STANDARD), September 1981. Updated by RFCs 950, 4884, 6633, 6918. (Cité en page 28.)

[53] J. Postel. Internet Protocol. RFC 791 (INTERNET STANDARD), September 1981. Updated by RFCs 1349, 2474, 6864. (Cité en pages 13 et 64.)

[54] J. Postel. Transmission Control Protocol. RFC 793 (INTERNET STANDARD), September 1981. Updated by RFCs 1122, 3168, 6093, 6528. (Cité en page 13.)

[55] Y. Rekhter and T. Li. An Architecture for IP Address Allocation with CIDR. RFC 1518 (Historic), September 1993. (Cité en page 101.)

[56] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006. Updated by RFCs 6286, 6608, 6793. (Cité en page 100.)

[57] E. Rescorla. HTTP Over TLS. RFC 2818 (Informational), May 2000. Updated by RFC 5785. (Cité en page 13.)

[58] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *JSAC*, 29(9), 2011. (Cité en pages 23 et 27.)

[59] Kamil Sarac and Kevin C Almeroth. Tracetree : A scalable mechanism to discover multicast tree topologies in the internet. *Networking, IEEE/ACM Transactions on*, 12(5) :795–808, 2004. (Cité en page 23.)

[60] Yuval Shavitt and Eran Shir. Dimes : let the internet measure itself. *Computer Communication Review*, 35(5), 2005. (Cité en pages 96 et 117.)

[61] Rob Sherwood, Adam Bender, and Neil Spring. Discarte : a disjunctive internet cartographer. In *SIGCOMM*, pages 303–314, 2008. (Cité en page 32.)

[62] W. Simpson. IP in IP Tunneling. RFC 1853 (Informational), October 1995. (Cité en pages 18 et 60.)

[63] Neil T. Spring, Ratul Mahajan, David Wetherall, and Thomas E. Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1) :2–16, 2004. (Cité en page 32.)

[64] MA Stephens. Tests of fit for the logistic distribution based on the empirical distribution function. *Biometrika*, 66(3) :591–595, 1979. (Cité en page 45.)

[65] Fabien Viger, Brice Augustin, Xavier Cuvellier, Clémence Magnien, Matthieu Latapy, Timur Friedman, and Renata Teixeira. Detection, understanding, and prevention of traceroute measurement artifacts. *Computer Networks*, 52(5), 2008. (Cité en pages 16, 23, 27 et 32.)

[66] Bernard M Waxman. Routing of multipoint connections. *Selected Areas in Communications, IEEE Journal on*, 6(9) :1617–1622, 1988. (Cité en page 21.)

[67] Walter Willinger, David Alderson, and John C. Doyle. Mathematics and the internet : A source of enormous confusion and great potential. *Notices of the AMS*, 56(5), May 2009. (Cité en pages 13, 16, 23, 27 et 32.)

[68] Ellen W Zegura, Kenneth L Calvert, and Michael J Donahoo. A quantitative comparison of graph-based models for internet topology. *IEEE/ACM Transactions on Networking (TON)*, 5(6) :770–783, 1997. (Cité en page 21.)

[69] H. Zimmermann. OSI reference model-the ISO model of architecture for open systems interconnection. *IEEE Trans. Communication (USA)*, COM-28(4) :425–432, April 1980. IRIA/Lab., Rocquencourt, France. (Cité en page 13.)