Vers un radar pour l'internet *

Matthieu Latapy[1], Fabien Viger[2,1], Benjamin Orgogozo[3,1] et Damien Bobillot[1]

[1] LIAFA, CNRS - Université Paris 7, 2 place jussieu 75005 Paris
[2] LIP6, CNRS - Université Paris 6, 8 rue du capitaine Scott 75015 Paris
[3] CREA, CNRS - École Polytechnique, 1 rue Descartes 75005 Paris

Résumé Nous introduisons dans cet article une idée simple basée sur le principe du radar et appliquée à l'internet. À partir d'un ordinateur connecté, nous sondons périodiquement le réseau vers une ou plusieurs directions voulues avec l'outil `traceroute` et obtenons ainsi une image dynamique de la topologie IP. Nous détaillons le fonctionnement de notre radar et présentons les premiers résultats. En couplant le radar avec des outils de visualisation et d'analyse existants ou en cours de développement, cette mesure nous permet de surveiller en temps réel l'état du réseau, et devrait à terme rendre possible la détection et l'analyse d'anomalies avec des temps de réaction de l'ordre de la minute.

Introduction

La taille et la nature fondamentalement distribuée de l'internet font de sa surveillance (détection d'anomalies comme des pannes ou des attaques, mais aussi observation de sa croissance, par exemple) un défi majeur pour la recherche en informatique. Sa cartographie elle-même pose des problèmes difficiles et est l'objet de nombreux travaux [6,5,15,7,3]. La surveillance du réseau est encore plus délicate, car elle pose des problèmes particuliers. Par exemple, il est essentiel dans ce contexte de réagir en *temps réel*.

Nous proposons ici une approche simple, ne reposant que sur des techniques et des infrastructures disponibles, et consommant peu de ressources, pour répondre (partiellement) à ce défi. Elle consiste en la surveillance de la *topologie* de l'internet au niveau IP telle qu'elle est observée avec l'outil `traceroute`. Nous nous restreignons de plus à la vision, forcément très partielle, qu'une machine de l'internet a de cette topologie.

Cet article se veut très prospectif. Il présente nos résultats préliminaires allant vers la définition de ce que nous appelons un *radar pour l'internet*. Ces premiers travaux sont toutefois suffisants pour valider l'approche puisqu'ils permettent de montrer que toute une gamme de phénomènes non triviaux sont observables de cette façon, et que le temps de réaction du radar peut être très court (inférieur à la minute). Ils permettent également d'apporter un éclairage original sur des problématiques importantes dans l'étude de l'internet, comme l'étude des alias (machines ayant plusieurs adresses IP), les défauts de configuration, la vision arborescente de l'internet vu par une machine, etc.

* Ce travail à été soutenu en partie par l'ACI *Sécurité et Informatique* dans le cadre du projet METROSEC [13].

1 Le principe de fonctionnement

L'outil `traceroute`

`traceroute` [14] est un programme disponible par défaut sur la quasi-totalité des machines connectées à l'internet. Initialement destiné à faire du *monitoring* de réseaux, il permet de connaître le chemin que suivent, au niveau IP, les paquets envoyés par la machine sur laquelle il est lancé vers une machine quelconque. En d'autres termes, il donne la suite des routeurs traversés par les paquets avant d'arriver à destination.

Il repose sur l'utilisation du champ TTL (*Time To Live*) des paquets IP. Celui-ci est décrémenté de 1 à chaque fois qu'il passe par un routeur IP. Lorsqu'un routeur reçoit un paquet de TTL égal à 0, il le supprime et envoie un message d'erreur ICMP à l'expéditeur. Celui-ci connaît ainsi le routeur auquel le paquet s'est arrêté, et peut donc savoir quelle est la suite de routeurs traversés en envoyant des paquets avec TTL initial 1, 2, ... jusqu'à ce que le paquet arrive à destination. Pour une description plus complète, voir [14,12].

Bien sûr, la description rapide que nous venons de donner suppose que tous les routeurs respectent les spécifications, ce qui n'est en pratique pas vrai. Certains routeurs n'envoient qu'un nombre limité de paquets ICMP par unité de temps (un par seconde typiquement), d'autres n'en envoient jamais, d'autres encore ne gèrent pas correctement le champ TTL (ne le décrémentent pas ou routent des paquets de TTL nul), etc. De plus, le routage peut changer *pendant* que `traceroute` est en train de faire sa mesure, ce qui induit des erreurs. Nous verrons par la suite comment mesurer, contourner et/ou traiter certains de ces biais.

Topologie de l'internet

Dans la suite, nous modélisons l'internet par un *graphe* dans lequel les sommets sont des adresses IP et une arête entre deux sommets signifie qu'on peut passer d'une adresse à l'autre en un saut IP. Soulignons le fait que les sommets de ce graphe ne sont pas les routeurs de l'internet : un tel routeur a en effet en général plusieurs adresses IP, et correspond donc à plusieurs sommets. De même, les arêtes ne correspondent pas forcément à des liens physiques : un saut au niveau IP peut traverser un sous-réseau encapsulant les paquets IP dans des paquets d'un autre protocole (ATM ou MPLS typiquement). C'est toutefois cette topologie qui est directement observable à l'aide de `traceroute`.

En effet, dans les limites que nous avons soulignées, l'information fournie par `traceroute` est précisément un *chemin* dans ce graphe, le chemin suivi par les paquets allant de la source à la destination.

À l'aide de multiples `traceroute` vers différentes destinations, on obtient un certain nombre de tels chemins. En fusionnant ces divers chemins, on obtient une vision (partielle et biaisée) du graphe total. C'est essentiellement sur de telles visions que reposent aujourd'hui les études de la topologie de l'internet à grande échelle, surtout aux niveaux IP et routeurs. Nous allons ici utiliser directement ces visions partielles pour surveiller le réseau.

Le radar de l'internet

Le radar repose sur l'utilisation itérée de `traceroute` à partir d'une seule machine connectée à l'internet, appelée *source*, vers un ensemble fixé de destinations. La source mesure *périodiquement*, avec `traceroute`, la route entre elle et chacune des destinations. Chacune de ces mesures est appelé une *passe* du radar.

Le choix de la source et de la destination dépend des objectifs. Si on veut surveiller l'environnement d'une machine en particulier, par exemple, on a tout intérêt à la prendre comme source. On peut aussi vouloir la prendre comme destination d'un radar exécuté ailleurs. Si on veut surveiller l'internet globalement, on prendra des destinations bien réparties. Si on veut surveiller une région géographique, on prendra les destinations dans cette région. Si c'est possible, on fera aussi tourner le radar sur une machine appartenant à la région.

Ces différents scénarios sont actuellement à l'étude. Nous nous focaliserons ici sur le choix d'une source arbitraire (une machine du `liafa`) et de destinations aléatoires (tirage aléatoire d'une adresse IP dans l'espace des adresses possibles[4] puis élimination de celles qui ne répondent pas à un `ping`). En effet, ces décisions n'ont que peu d'impact sur les observations que nous allons faire ici.

Finalement, le radar est paramétré par la donnée de :
- la source s, qui sera ici toujours la même machine du `liafa`,
- l'ensemble de destinations D, qui seront ici toujours choisies aléatoirement comme décrit ci-dessus,
- la période Δ_t entre le début de chaque passe.

Bien sûr, le temps nécessaire à le mesure d'une passe est non négligeable (surtout si le nombre de destinations est grand), et la période Δ_t doit être supérieure à ce temps. On veut toutefois la maintenir aussi petite que possible, car elle conditionne le temps de réaction du radar.

On notera g_i le graphe obtenu lors de la passe i, c'est à dire celle ayant commencé à l'instant $t_0 + i \cdot \Delta_t$, si t_0 est la date de lancement du radar (nous poserons dans la suite $t_0 = 0$). Nous définissons également $G_i = \bigcup_{k \leq i} g_k$, qui est le graphe obtenu par le radar après la i-ème passe et depuis son lancement.

2 La mesure en pratique

Redondance des passes

Nous avons décrit le principe général des passes du radar ci-dessus : à chaque période, on effectue un `traceroute` vers chacune des destinations. La passe est achevée quand chaque destination de l'ensemble D a été sondée.

Cette méthode assure que nous verrons effectivement le chemin de la source vers chacune des destinations. Comme déjà souligné, notamment

[4] L'ensemble des 2^{32} adresses IPv4 possibles privé des adresses spéciales référencées dans [11] et des adresses de broadcast

dans [2], elle est toutefois largement redondante. En effet, la plupart des routes commencent par la même suite de routeurs, et plus généralement elles ont souvent d'importants préfixes communs. Outre le fait que cette redondance va augmenter le temps nécessaire à la mesure, et donc nous obliger à augmenter la période Δ_t, elle va aussi impliquer une surcharge inutile sur certains routeurs et certains liens (voir Figure 1).

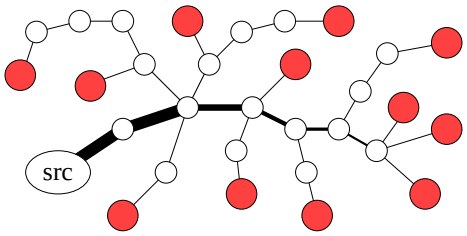


Fig. 1. Représentation de la redondance de l'information obtenue sur les liens lorsqu'on fait un `traceroute` vers chaque destination. Dans cet exemple, on envoie 70 paquets pour découvrir 30 arêtes, soit 2.33 paquets par arête. L'optimal est 1.37 paquets par arête.

Il existe de nombreuses façons d'atténuer cet effet, dépendant de l'objectif. En effet, on peut souhaiter :

- minimiser la charge induite par le radar sur les routeurs. On va sonder les routes en partant des destinations, c'est-à-dire avec un TTL décroissant de la valeur maximale pour chaque destination (sa distance à la source) jusqu'à ce qu'on rencontre une adresse IP déjà vue lors de cette passe. Le nombre de paquets envoyés est alors optimal et est égal au nombre d'arêtes vues (plus le nombre de destinations).
- maximiser la quantité d'information recueillie. On peut par exemple tester la présence de chaque arête $\{i, j\}$ déjà vue en effectuant un `traceroute` à destination de i et de j (en plus du `traceroute` ayant permis de la découvrir). On teste ainsi l'accessibilité de toutes les arêtes de G_{t-1} à chaque période t.
- adapter la mesure à ce qui a été observé, par exemple sonder plus activement une partie de forte variabilité.

Bien sûr, minimiser la charge induite implique une perte de qualité de la mesure obtenue, et réciproquement obtenir une mesure fine implique une charge accrue. Soulignons toutefois que ces questions sont cruciales dans notre contexte, car il y a un lien fort entre la charge induite et la fréquence que nous pourrons nous autoriser. Nous voudrions réduire autant que possible la durée des passes, donc la charge, afin d'augmenter la qualité de notre vision *de la dynamique*, quitte à ce que la vision obtenue à chaque étape soit moins précise.

Notre approche

Nous avons implémenté plusieurs stratégies. Chacune a ses avantages et ses faiblesses. Les comparer est un problème en soi, sur lequel nous ne voulons pas trop nous attarder ici. Le choix d'une stratégie plutôt qu'une autre est de plus fonction des objectifs. Nous nous limitons ici à présenter la stratégie retenue dans l'objectif d'obtenir une vision aussi fine que possible de la dynamique. En plus de limiter la durée des passes, comme déjà discuté, nous allons vérifier les arêtes dans le désordre afin de minimiser le temps de détection d'une modification de la topologie IP.

Nous associons à chaque arête de G_t l'ensemble des couples (d, τ) où d et τ sont une destination et un TTL ayant permis de la voir dans le passé, ce qui revient à dire que l'arête à été vue entre les sommets d'indices τ et $\tau + 1$ sur la route vers d.[5]

Nous considérons de plus un ensemble E d'arêtes de G_t (nous discuterons ce point ci-dessous), puis, pour la passe $t + 1$, nous effectuons pour chacune des arêtes de E *dans un ordre aléatoire* les opérations suivantes :

- si l'arête a déjà été vue, on passe à la suivante,
- sinon :
  - on choisit aléatoirement un couple (d, τ) ayant permis de la voir dans le passé et n'ayant pas encore été utilisé dans cette passe,
  - s'il n'existe aucun tel couple, on passe à l'arête suivante,
  - sinon :
    - on mesure quelle est l'arête à TTL $= \tau$ vers d et on la marque comme vue,
    - on marque le couple (d, τ) comme utilisé pendant cette passe.

Toutes les arêtes qui ne sont pas marquées comme vues à la fin du processus sont supposées absentes.

Cette méthode assure qu'on fera au plus une requête par arête dans E d'une part, et au plus une requête par couple (d, τ) d'autre part. Elle évite donc la redondance des requêtes tout en introduisant du désordre dans leur séquence.

Il faut toutefois faire attention au choix de E : le prendre trop grand, par exemple prendre l'ensemble des arêtes de G_t, augmente la durée de la mesure alors que le prendre trop petit réduit notre visibilité, puisqu'à chaque passe on voit au plus autant d'arêtes qu'il y en a dans E. Ainsi, si on prend pour E l'ensemble des arêtes de g_t, le nombre d'arêtes vues à chaque étape ne pourra que décroître.

On peut envisager par exemple de choisir pour E les arêtes vues pendant les k dernières étapes, pour un k donné. Ceci a peu d'impact sur les résultats que nous présentons ci dessous, donc nous nous placerons dans le cas où on prend pour E l'ensemble des arêtes vues jusqu'ici, en gardant en tête que d'autres méthodes peuvent accélerer la mesure. Nous initialisons E en effectuant un `traceroute` complet vers chaque destination.

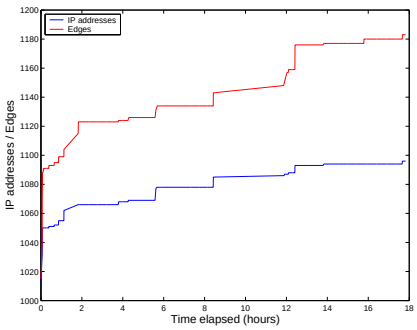
[5] Cette association n'est ni injective ni surjective : une arête peut être présente sur plusieurs routes, et réciproquement, la topologie IP étant dynamique, un même couple (d, τ) peut être associé à plusieurs arêtes différentes.

Convergence de G_t

Le graphe G_t croît au cours du temps, puisqu'on garde l'information accumulée depuis le début. Remarquons que cette croissance peut être sur le très long terme, puisque la mesure s'effectue sur un objet lui-même croissant. Elle est toutefois bornée par le nombre fini d'adresses possibles. Bien sûr, la mesure effectuée par le radar est de plus effectuée sur une période de temps petite devant le temps nécessaire à l'internet pour croître significativement. On peut donc supposer que l'internet est de taille fixe pendant la mesure, et que le radar finira par voir *tout ce qu'il peut voir*.

Cette convergence peut toutefois être très lente, puisque des changements dans le routage peuvent survenir de façon très ponctuelle, suite à un problème quelconque sur le réseau. Elle dépend de plus de l'approche retenue pour la mesure.

Nous montrons en Figure 2 l'évolution du nombre de sommets et d'arêtes dans G_t en fonction du temps pour un ensemble D de 200 destinations aléatoires et une période $\Delta_t = 30s$ (2033 passes). Elle est représentative de ce que nous avons observé dans toutes nos mesures, quelles que soient les optimisations choisies : le nombre de sommets et d'arêtes est stable pendant des périodes de temps importantes (souvent une à deux heures), puis augmentent généralement par palier. Cette croissance n'est toujours pas terminée après un délai relativement long (18 heures, soit plus de 2000 passes), mais elle permet déjà de dire que des modifications du routage, dues à des *événements* à identifier, changent de façon significative la vision obtenue. Nous y reviendrons dans la suite.



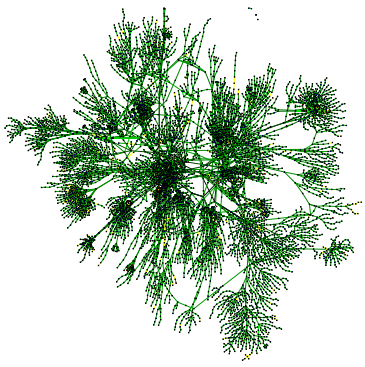


Fig. 2. À gauche : Évolution du nombre total d'adresses IP et d'arêtes découvertes par le radar. À droite : Graphe obtenu par un autre radar vers 2000 adresses IP aléatoires, avec une période $\Delta_t = 500s$ pendant 9h (65 passes).

3 Visualisation

La mesure décrite dans la section précédente fournit une grande quantité de données, sous la forme des graphes g_i et G_i. Analyser ces données, afin d'en tirer de l'information pertinente, suppose de s'en être fait au préalable une intuition. Une façon efficace (mais pas la seule) pour construire une telle intuition est la *visualisation*.

Dans le contexte des grands réseaux, et de l'internet en particulier, la visualisation n'est employée que de façon marginale par la plupart des chercheurs. Ceci est probablement essentiellement dû au fait que visualiser un grand graphe est un défi en soi, voir par exemple [1,9]. De plus, la visualisation n'a en général qu'un apport *qualitatif*, qu'il s'agit de compléter par des analyse statistiques *quantitatives*.

Dans notre contexte toutefois, nous pensons que la visualisation joue un rôle essentiel, notamment :

- pour amener à définir des statistiques réellement porteuses de sens car en relation avec les observations,
- pour identifier visuellement des biais induits par la mesure, notamment les motifs particuliers.

Nous montrons Figure 2 un dessin [6] d'un graphe G_t obtenu par le radar sur 2000 destinations aléatoires, avec une période $\Delta_t = 500s$ pendant 9 heures (65 passes).

Cette première image montre déjà que la visualisation permet effectivement, même sur une donnée de cette taille, de mettre en évidence des propriétés structurelles. Par exemple, le graphe est faiblement connecté, comporte de larges parties arborescentes mais aussi de grands cycles, et on identifie aisément des parties distinctes du réseau. On voit aussi que, contrairement à ce qui est communément supposé, *la vision qu'une machine a de l'internet en effectuant des* `traceroute` *n'est pas un arbre.*

Elle montre toutefois aussi que la visualisation est difficile, les recouvrements d'arêtes étant nombreux et le dessin lui même d'une complexité difficile à appréhender. Au delà, il faut garder à l'esprit le fait qu'il existe de nombreuses façons de dessiner un même graphe. Chaque méthode de dessin a ses avantages et ses inconvénients, et peut induire l'observateur en erreur.

Cette vision purement statique du graphe G_t n'est bien sûr qu'une première étape : de même que la visualisation d'un graphe permet de se faire une intuition pour son analyse, la visualisation de son évolution temporelle permet de se faire une intuition sur sa dynamique, encore plus difficile à appréhender sans ce recours. Nous avons par exemple produit des animations montrant sur des graphes G_t la succession des visions g_i obtenues lors des passes $i < t$. À terme, nous souhaitons effectuer la visualisation de la dynamique en temps réel afin de proposer aux opérateurs un outil de surveillance visuelle de leur réseau.

Une autre direction prometteuse pour la visualisation à grande échelle est de *géolocaliser* les adresses IP (c'est-à-dire déterminer leur longitude

[6] Ce dessin à été réalisé avec le logiciel de visualisation GUESS [10] développé par HP

et leur latitude) et ainsi de dessiner le graphe sur une carte du monde. Des outils permettant de faire de telles géolocalisation efficacement sont actuellement disponibles [4,8] et pourraient être intégrés avec profit au radar.

Nous n'approndissons toutefois pas ces perspectives ici, préférant montrer dans la section suivante comment la visualisation que nous avons obtenue permet d'analyser la mesure effectuée par le radar.

4 Premiers résultats

Bicliques et routeurs multi-interfaces

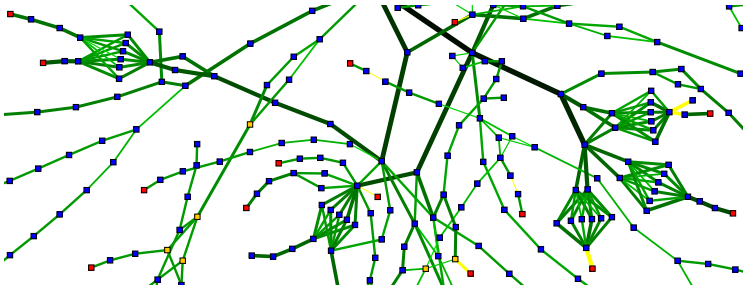


Fig. 3. Agrandissement d'une partie d'un graphe obtenu en scannant 100 adresses aléatoires pendant une demi-heure (180 passes)

Nous montrons Figure 3 un exemple très courant de motif particulier rencontré lors de la visualisation : les *bicliques*. Une biclique est une paire d'ensembles de sommets tels qu'il n'y a aucune arête entre deux sommets d'un même ensemble mais toutes les arêtes possibles entre les sommets des deux ensembles. Si les tailles de ces ensembles sont n et m, on parle alors de (n, m)-biclique. De plus, si E et E' forment une (n, m)-biclique et E' et E'' forment une (m, l)-biclique, alors on dira que E, E' et E'' forment une (n, m, l)-biclique. Sur la Figure 3, on peut notamment voir quatre $(1, 6, 2)$-bicliques et une $(1, 5, 2)$-biclique.

La découverte de ces bicliques est non-triviale : ces structures très particulières n'ont *a priori* aucune raison d'être, et ne correspondent certainement pas à l'existence de liens physiques entre des routeurs différents. Elles ont toutefois une explication simple. En effet, supposons qu'un routeur donné ait plusieurs interfaces, disons n. Lorsqu'un `traceroute` passe par ce routeur, il peut répondre avec n'importe laquelle de ses interfaces. On va donc voir ce routeur comme un ensemble de n sommets. De plus, ces n sommets vont être reliés à l'adresse IP vue juste avant par

traceroute. De même, elle vont être liées à celle d'après, ce qui apparaît pour le radar comme une $(1, n, 1)$-biclique. Si le routeur d'avant a lui-même m interfaces et celui d'après en a l, on verra une (m, n, l)-biclique.

La présence de routeurs multi-interfaces sur l'internet est un fait bien connu, et l'identification de leurs interfaces multiples, appelé *anti aliasing*, est un problème classique dans le domaine de la cartographie de l'internet [3]. Le radar permet une methode originale pour la résolution de ce problème. Cette méthode est de plus simple et sûre en regard des méthodes actuellement employées. Elle constitue donc une alternative, ou du moins un complément, à ces méthodes.

La détection automatique de bicliques est facile sur des données de cette taille. Nous avons donc effectué sur le graphe G_t les fusions de sommets décrites dans la Figure 4 : tant qu'il existe un couple de sommets i, j de degrés supérieurs ou égaux à 2 liés *exactement aux mêmes sommets*, on les fusionne en un unique sommet. Ce processus commence systématiquement par les sommets de plus hauts degrés, pour éviter de fusionner par erreur des feuilles liées au même sommet (à droite dans la Figure 4).

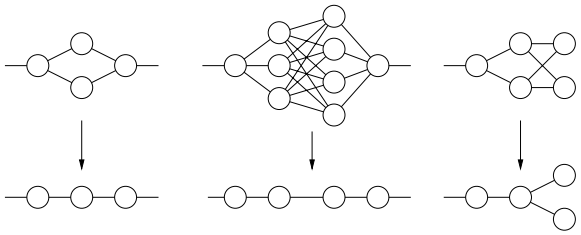


Fig. 4. Exemples de fusions de sommets IP représentant les interfaces d'un même routeur.

Cette méthode reste à affiner, mais nos résultats préliminaires montrent déjà une diminution de 10 à 15% du nombre d'arêtes du graphe (le nombre de sommets diminuant dans une moindre mesure). Le graphe résultant est en général beaucoup plus proche d'un arbre : son degré moyen s'approche de 2. Soulignons de plus qu'une telle simplification du graphe permet d'améliorer sa visualisation.

Boucles, cliques et routeurs déficients

Dans un graphe, une *boucle* est une arête liant un sommet à lui-même. En principe, bien sûr, il n'y a aucune raison que de telles boucles apparaissent sur l'internet. Dans la vision du radar, ceci suppose qu'un même routeur réponde pour deux TTL consécutifs. Ceci est bien sûr possible, par exemple si le routage a changé et que le nombre de sauts nécessaires pour atteindre ce routeur a augmenté (d'une unité) entre l'envoi des deux

paquets le concernant. Un tel phénomène ne peut toutefois arriver que très rarement, et n'est pas reproductible.
Au contraire, nous avons constaté la présence de nombreuses boucles dans les graphes G_i (105 dans le graphe de la Figure 2, par exemple).
Une explication probable de ce phénomène, est proposée Figure 5 : un routeur mal configuré transmet un paquet ayant un TTL égal à zéro au lieu de génerer un paquet d'erreur ICMP. Notre vision en est alterée, puisque d'une part on ne voit pas le routeur fautif, et d'autre part on voit apparaître une boucle sur le routeur suivant. De tels problèmes sont connus pour certains routeurs : voir par exemple [14].
Nous sommes en mesure grâce au radar de détecter de telles anomalies, de les dénombrer et/ou de les corriger en enlevant les boucles et en ajoutant les sommets manquants, par exemple.

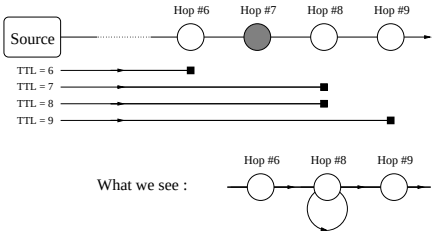


FIG. 5. Explication possible des boucles de la topologie IP : le 7e routeur transmet les paquets quand leur TTL est nul au lieu de générer un ICMP.

Lors de l'observation des données produites par le radar, un phénomène encore plus surprenant est apparu : G_t contient souvent des *cliques*, c'est-à-dire des groupes de sommets tous reliés deux à deux, de tailles non négligeables (de cinq à plus de dix sommets).
Une idée naturelle, et couramment avancée, pour expliquer ce phénomène, est la présence d'un sous-réseau non IP (comme une boucle optique, ou un sous-réseau ATM ou MPLS). En effet, ces structures particulières apparaîtraient comme des cliques dans la topologie de l'internet au niveau IP : le TTL n'étant décrémenté qu'à l'entrée et à la sortie de ces structures, chaque point d'entrée apparaît lié à chaque point de sortie. Si les points d'entrée sont aussi des points de sortie, ce qui est généralement vrai, ils apparaissent donc comme une clique.
Cette explication est crédible dans le cas d'une exploration exhaustive de la topologie IP, mais dans notre cas le radar explore trop peu de routes (et qui plus est, toujours depuis la même source) pour pouvoir découvrir les cliques dues à de telles structures : il faudrait passer par tous les couples (entrée, sortie) possibles.
Les cliques de taille supérieure à 5 vue par le radar nécessitent une autre explication, qui peut en fait provenir des deux phénomènes déjà rencon-

trés. Sur la Figure 5, supposons en effet que le routeur noté `Hop #8` soit un routeur multi-interfaces. Plus généralement, supposons qu'un routeur multi-interfaces se trouve juste derrière un routeur déficient (du même type que `Hop #7` sur la Figure 5). La boucle se transforme alors en lien entre deux interfaces du routeur multi-interfaces, tous ces liens pouvant apparaître au cours du temps. On obtient ainsi, en effectuant un grand nombre de `traceroute` incluant ces deux routeurs (comme nous le faisons dans le radar) une clique entre les différentes interfaces du routeur multi-interfaces.

5 Aller plus loin

Le travail présenté ici se veut largement prospectif. Bien que les premiers résultats montrent la pertinence de l'approche, il existe encore beaucoup de voies à explorer pour arriver à ce but. Nous présentons ici deux voies qui nous semblent parmi les plus prometteuses de celles que nous avons actuellement en friche.

Réductions du graphe

Une réduction bien connue en théorie des graphes est la réduction au k-core, qui consiste à retirer itérativement tous les sommets de degré inférieur ou égal à k. Une telle réduction permet de se focaliser sur la partie $(k+1)$-connexe du graphe, composée dans notre contexte essentiellement de routeurs centraux et/ou pour lequel le routage est complexe. Elle est également un outil précieux pour l'analyse et la visualisation.

La réduction au 1-core correspond à la suppression des parties arborescentes du graphe. Dans le cas du radar,il perd typiquement entre 60% et 80% de ses arêtes et de ses sommets. Ce nombre passe à plus de 90% (jusqu'à 100% dans certaines données) dans le cas du 2-core. Citons également une approche alternative, celle de supprimer itérativement tous les sommets de degré *égal* à 2 en les remplaçant par une arête liant directement leurs deux voisins. Ainsi, une longue branche se transforme en simple arête.

Les applications de ces réductions, outre l'aspect facilitant la visualisation, sont aussi la possible focalisation de la mesure sur une partie plus connectée : on pourra par exemple diminuer la charge sur les routeurs n'appartenant pas à la 1-core et se concentrer sur les autres. En termes d'analyses topologiques, ces réductions ont également du sens : le degré moyen du graphe réduit n'a pas la même signification que celui du graphe complet, la distance moyenne est également changée, ...

Dynamique

La perspective essentielle sur laquelle nous travaillons actuellement est bien sûr l'étude de la dynamique de la vue du radar, notamment les différences entre deux graphes successifs g_i et g_{i+1}. Les premières observations on montré un certain nombre de phénomènes :

- certains routeurs ne répondent pas toujours à cause de leur limitation en paquets ICMP et apparaissent donc comme clignotants,
- les routeurs multi-interfaces (bicliques) apparaissent à chaque passe comme une seule adresse IP sur une route, mais cette adresse varie sans cesse,
- l'équilibrage de charge (*load balancing*), se traduisant par des portions de routes clignotantes.

Nous sommes actuellement en train de collecter des statistiques montrant l'importance de ces phénomènes et leur impact sur la visualisation dynamique du graphe. Il est d'ores et déjà clair que ces phénomènes accaparent toute la dynamique apparente, rendant nécessaire un filtrage capable d'éliminer leur bruit.

Nous avons dans un premier temps contourné ce problème en rassemblant les graphes $g_i, \cdots, g_{i+p}$ pour un p choisi selon l'intervalle de temps voulu. La comparaison de $\bigcup_{0 \leq k < p} g_{i+k}$ et de $\bigcup_{0 \leq k < p} g_{i+k+p}$ permet en effet de faire uniquement ressortir les changements de topologie à plus long terme (celui de p passes plutôt qu'une seule). Dès lors, toute apparition ou disparition d'arête est porteuse d'un sens non trivial (changement de routage, panne, attaque, ...).

Ces changements, bien que plus rares, restent relativement fréquents. Ils ne peuvent plus être interprêtés comme des dynamiques relevant du fonctionnement *normal* du réseau, même si la plupart correspondent certainement à des changement de tables de routages suite à des congestions ou à des accords inter AS. Leur étude reste toutefois à faire.

Conclusion

Nous avons présenté ici une approche et de premiers résultats visant au développement d'un outil de surveillance de l'internet à un niveau global. Il s'agit d'observer périodiquement, avec une période courte, la vision qu'une machine a de la topologie IP en effectuant des `traceroute` vers un ensemble de destinations.

Après avoir dans un premier temps détaillé le mode de fonctionnement du radar, nous avons introduit la visualisation comme élément clé dans l'analyse des résultats. Le rôle de cette dernière est d'autant plus important que nous sommes encore à un stade prospectif.

Nous avons toutefois obtenu de premiers résultats prometteurs, dont certains fournissent même un éclairage nouveau sur certains aspects de la cartographie de l'internet étudiés par ailleurs. Il s'agit maintenant de définir un ensemble de métriques conçues pour les graphes dynamiques et adaptées à la détéction d'anomalies sur l'internet.

Soulignons pour terminer que le radar dispose d'importants atouts : malgré la simplicité des moyens mis en oeuvre, les perspectives qu'offrent le sondage adaptatif et le temps de réaction très court permettent d'espérer à terme la création d'un outil de surveillance polyvalent et temps-réel pour l'internet. Ajoutons que la possibilité de coupler plusieurs radars coopératifs, répartis sur l'internet, est une voie extrêmement prometteuse même si elle soulève des questions difficiles.

Références

1. G. Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for drawing graphs : An annotated bibliography. *Computational Geometry : Theory and Applications*, 5(4) :235–282, 1994.
2. B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Efficient algorithms for large-scale topology discovery. In *ACM Sigmetrics*, 2005.
3. R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. *Proc. of IEEE INFOCOM*, pages 1371–1380, 2000.
4. B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of internet hosts. In *ACM SIGCOMM Internet Measurement Conference*, 2004.
5. J.-L. Guillaume and M. Latapy. Relevance of massively distributed explorations of the internet topology : Simulation results. In *IEEE Infocom*, 2005.
6. R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of the Internet : A Statistical Physics Approach*. Cambridge University Press, 2004.
7. CAIDA organisation, www.caida.org.
8. Geobytes company, www.geobytes.com.
9. Graph drawing symposium graphdrawing.org.
10. GUESS software webpage, www.hpl.hp.com/research/idl/projects/graphs/.
11. RFC 3330 : Special use ipv4 addresses, www.faqs.org/rfcs/rfc3330.html.
12. RFC 792 : Internet control message protocol, www.faqs.org/rfcs/rfc792.html.
13. Projet metrosec : Métrologie pour la sécurité www.laas.fr/METROSEC/.
14. traceroute manual, www.zytec.com/traceroute.man.html.
15. Traceroute@home project, www.tracerouteathome.net.