Évaluation en moyenne d'un algorithme pour la mise à jour d'un arbre de connexion

François Delbot, Christian Laforest, Nicolas Thibault

Évaluation en moyenne d'un algorithme pour la mise à jour d'un arbre de connexion

François Delbot, Christian Laforest et Nicolas Thibault

IBISC /CNRS, Université d'Evry, 523 place des terrasses de l'agora, 91000 Evry, France
Email:{francois.delbot,christian.laforest,nicolas.thibault}@ibisc.fr

Nous proposons un algorithme simple pour la mise à jour d'un arbre couvrant un groupe dont les membres peuvent arriver et/ou partir* à chaque étape. L'objectif est alors de maintenir un arbre dont le diamètre est proche du diamètre minimum. La dynamicité des membres et le respect de la contrainte sur le diamètre peuvent nécessiter la reconstruction de l'arbre. Une *étape de reconstruction* étant coûteuse, nous nous intéressons à la minimisation de ce nombre d'étapes. Si cette démarche a déjà été étudiée, les précédents travaux se sont restreints à l'étude du pire cas (algorithmes d'approximation ou étude du rapport de compétitivité dans le contexte online). Dans cet article, nous analysons le comportement de notre algorithme *en moyenne*. Pour cela, nous proposons quelques encadrements analytiques particuliers et un moyen algorithmique efficace pour calculer de manière exacte l'*espérance* du nombre de reconstructions (sous des hypothèses d'équiprobabilité). Nous donnons alors les grandes tendances de cette espérance.

Mots clefs : Mise à jour d'arbre de connexion, évaluation en moyenne, optimisation, espérance, équiprobabilité

Nous nous intéressons dans cet article aux communications entre les *membres* d'un *groupe* de machines. Nous souhaitons pour cela les interconnecter par un *arbre*, bâti sur le réseau sous-jacent. Une caractéristique importante d'un groupe au sein d'un réseau est qu'il est en constante évolution. On peut en effet évoquer toute forme de forum ou de réunion via un réseau. Dans ce type de rassemblements virtuels, les participants peuvent arriver et/ou partir*, dans un ordre et à des moments inconnus à l'avance. Par exemple, dans les systèmes pair à pair (*peer-to-peer*) il est inconcevable de prédire qui va échanger des données avec qui. Les échanges se font "au fil de l'eau", en fonction de paramètres qu'il est difficile de quantifier à l'avance. Par conséquent, connaître à l'avance la totalité des membres du groupe à couvrir/connecter n'est pas toujours possible en pratique. Nous avons donc choisi d'étudier le cas *dynamique* qui consiste à incrémenter et/ou décrémenter un *arbre de connexion*† pour prendre en compte les arrivées et départs de membres.

Cependant, il est naturel d'exiger que cet arbre maintienne, à chaque étape, un niveau de qualité satisfaisant. Dans cet article nous nous focalisons sur le temps de communication maximum induit entre les membres. Pour définir cela de manière plus précise, nous introduisons les notations suivantes. Les ajouts et/ou retraits successifs font évoluer le groupe courant ; nous noterons $M_0, M_1, \ldots, M_i, \ldots$ la suite des groupes successifs ainsi obtenue (M_0 étant le groupe initial, avant tout changement).

On note $D_G(M_i) = max\{d_G(u,v) : u,v \in M_i\}$ le *diamètre* du groupe $M_i \subseteq V$ dans le graphe $G = (V,E)$ qui représente le réseau (fixe), avec $d_G(u,v)$ la distance minimum entre u et v dans G. $D_G(M_i)$ représente donc le diamètre minimum possible pour M_i. Cependant, comme on interconnecte M_i par un arbre T_i, le diamètre de M_i dans T_i (noté $D_{T_i}(M_i)$) doit être le plus "proche" possible de $D_G(M_i)$. Plus précisément, nous fixons la contrainte suivante : à chaque requête i (ajout ou retrait), le diamètre de M_i dans T_i doit être *au plus* deux fois le diamètre minimum ; à chaque étape, on doit donc garantir : $D_{T_i}(M_i) \leq 2D_G(M_i)$.

Pour garantir cela, nous allons permettre la *reconstruction* totale de l'arbre à certaines étapes. Une telle reconstruction nécessite, d'un point de vue réseau, la mise à jour des tables de routage de chaque membre du groupe, perturbant ainsi les communications en cours et induisant un fort trafic de contrôle. L'objectif est alors de minimiser le nombre d'étapes où ont lieu de telles reconstructions.

* lorsqu'un membre est retiré, il est retiré du groupe, pas du réseau qui reste fixe.

† le choix d'un arbre est motivé par la facilité du routage qu'il induit entre les membres et que c'est une structure dans laquelle il est simple de faire des diffusions par inondations.

État de l'art. Il est simple de voir que s'il n'y a que des ajouts, un algorithme simple permet de ne faire aucune reconstruction (voir par exemple [Thi06]).

Le cas où il n'y a que des retraits a été spécifiquement étudié dans [TL06] ; dans cet article il est montré qu'il existe des situations dans lesquelles, dans le pire cas, $\Omega(\log\ i)$ reconstructions sont nécessaires. Il est aussi proposé un algorithme permettant de mettre à jour un arbre initial induisant, dans le pire cas, $O(\log i)$ reconstructions. Ce type d'étude a été menée pour d'autres paramètres, comme la distance moyenne (voir [TL04a, TL04b, TL06, TL, Thi06]) et le poids (arbre de steiner dynamique, voir [IW91, TL07]).

Cependant, tous ces articles contiennent des études en *pire cas*, avec l'obtention d'algorithmes d'approximation ou d'algorithmes online compétitifs. Nous avons constaté que les situations atteignant les pires cas sont souvent rares et/ou conçus de manière "artificielle". Nous souhaitons aller plus loin dans cet article en étudiant la moyenne du nombre de reconstructions.

L'algorithme étudié dans cet article. L'obtention de résultats en moyenne est un objectif complexe. Si l'algorithme que nous analysons ici est plus simple que celui proposé dans [TL06], il en reprend néanmoins les grandes lignes.

Le groupe initial M_0 est couvert par un arbre T_0, de plus courts chemins enraciné en un membre $r_0 \in M_0$. Notons T_i l'arbre courant et $r \in M_i$ sa racine courante ; T_i est un arbre couvrant M_i, de plus courts chemins à partir de r. Si la requête suivante est l'*ajout* d'un (nouveau) membre u, incrémenter T_i en ajoutant un plus court chemin entre r et u sans créer de cycle ; on obtient alors l'arbre T_{i+1}. Si la prochaine requête $i+1$ est le *retrait* d'un membre $u \in M_i$, deux cas peuvent se produire : si $u \neq r$ alors supprimer u de M_i et mettre à jour l'arbre en élaguant T_i si nécessaire (ce qui donne T_{i+1}). Si le membre retiré est r (la racine de l'arbre courant T_i) alors choisir un nouveau membre $r_{i+1} \in M_{i+1}$ et construire un nouvel arbre de plus courts chemins, T_{i+1} enraciné en r_{i+1}, couvrant M_{i+1}. Dans tous les cas, si $M_i = \emptyset$ le groupe est considéré comme terminé et il ne reçoit alors plus aucune requête.

Il est assez simple de voir qu'à chaque étape, T_i est bien un arbre couvrant le groupe courant M_i et qu'à chaque étape i, on a bien $D_{T_i}(M_i) \leq 2D_G(M_i)$. On note que dans cet algorithme une reconstruction n'a lieu *que* lorsque la racine de l'arbre courant se retire du groupe.

Analyse en moyenne de l'algorithme Pour évaluer le comportement moyen de l'algorithme, on note m_0 la taille du groupe initial et h le nombre maximum de requêtes considérées. Un *historique* est alors une suite de requêtes d'ajouts et/ou de retraits conduisant soit au bout de $h' \leq h$ requêtes à un groupe vide, soit à un groupe de taille $m_h \geq 1$ au bout de h requêtes. Rappelons que lorsque l'on atteint un groupe vide, l'historique s'arrête et le groupe ne "renaît" pas.

Il est facile d'étudier le comportement de l'algorithme dans les situations "extrêmes" ; même si la séquence de requêtes est très longue on peut n'avoir aucune reconstruction (meilleur cas) : par exemple s'il n'y a que des ajouts ou si les retraits ne concernent jamais la racine. A l'autre extrémité, sur une séquence de h requêtes, on peut avoir jusqu'à $\frac{1}{2}(m_0+h)$ reconstructions. Ces deux situations extrêmes sont très différentes.

Pour affiner l'analyse nous allons calculer $E(m_0,h)$, *l'espérance* du nombre de reconstructions, c'est à dire la somme sur tous les historiques (d'au plus h requêtes à partir d'un groupe de taille m_0) possibles du nombre de reconstructions induit par chaque historique multiplié par la probabilité de cet historique. Pour cela, on suppose qu'une requête peut être, de façon *équiprobable* soit l'ajout d'un nouveau membre, soit le retrait d'un membre du groupe courant. Une requête de retrait s'effectue de façon *équiprobable* sur l'ensemble des membres du groupe courant. La probabilité de retirer la racine de l'arbre de connexion est donc $\frac{1}{2m}$, avec m la taille du groupe courant.

Pour obtenir les valeurs de $E(m_0,h)$ nous avons modélisé l'évolution du groupe initial de taille m_0 par un arbre binaire de hauteur h. Chaque sommet de cet arbre est étiqueté par la taille m du groupe qu'il représente. Le fils gauche d'un sommet représente le groupe après un retrait (taille $m-1$) et le fils droit le groupe après un ajout (taille $m+1$). Avec cette représentation, une feuille représente un groupe en fin d'historique (soit parce qu'il est vide, soit parce que l'on a effectué h requêtes).

Nous avons montré que cette représentation permet de calculer $E(m_0,h)$ grâce à la formule $E(m,h) = \sum_{u \in F_g} \frac{R(u)}{2^{h(u)}}$, avec F_g l'ensemble des sommets de l'arbre qui sont *fils gauche* de leur père (qui sont donc des sommets représentant un groupe issu d'un retrait), $h(u)$ est la hauteur de u dans l'arbre et $R(u) = 1/m$ avec m la taille du groupe représenté par le père de u.

Quelques résultats analytiques

Théorème 0.1 *Pour tout $m_0 \geq 1$, pour tout $h \leq m_0$, on a $E(m_0,h) \leq \frac{h}{m_0} \leq 1$.*

Ce résultat met en évidence le fait que pour tout groupe de départ de taille m_0 et pour toute séquence de $h \leq m$ requêtes, le nombre de reconstructions moyen est inférieur à 1. De plus, des résultats expérimentaux nous indiquent que, dans ce cas, plus de 60% des solutions n'effectuent aucune reconstruction.

Théorème 0.2 *Dans le cas où tous les événements sont des retraits (donc si la probabilité d'ajout est nulle), pour tout $h \leq m_0$, on a $E(m_0,h) = \sum_{i=0}^{h-1} \frac{1}{m_0-i}$.*

Dans le cas particulier du théorème 0.2, l'espérance $E(m_0,m_0)$ tend vers l'infini lorsque m_0 tend vers l'infini. Ajoutons que dans le cas où tous les événements sont des ajouts, on a $E(m_0,h) = 0$ pour tout $h \geq 1$.

La suite de l'article traite des autres cas non étudiés ici : ajouts et retraits mêlés, avec un nombre quelconque d'événements.

Calculs numériques de l'espérance. La représentation par arbres binaires permet d'obtenir un algorithme efficace (rapide et peu coûteux en mémoire) pour calculer toute valeur $E(m_0,h)$. Cet algorithme calcule l'espérance en calculant la contribution de l'étage i de l'arbre binaire grâce à la contribution de l'étage $i-1$, grâce à une formule récursive. Cet algorithme nous a permis d'observer l'évolution de l'espérance du nombre de reconstruction. Nous insistons ici sur le fait que les résultats des figures 1 et 2 correspondent au calcul exact de l'espérance (il ne s'agit pas de simulations).

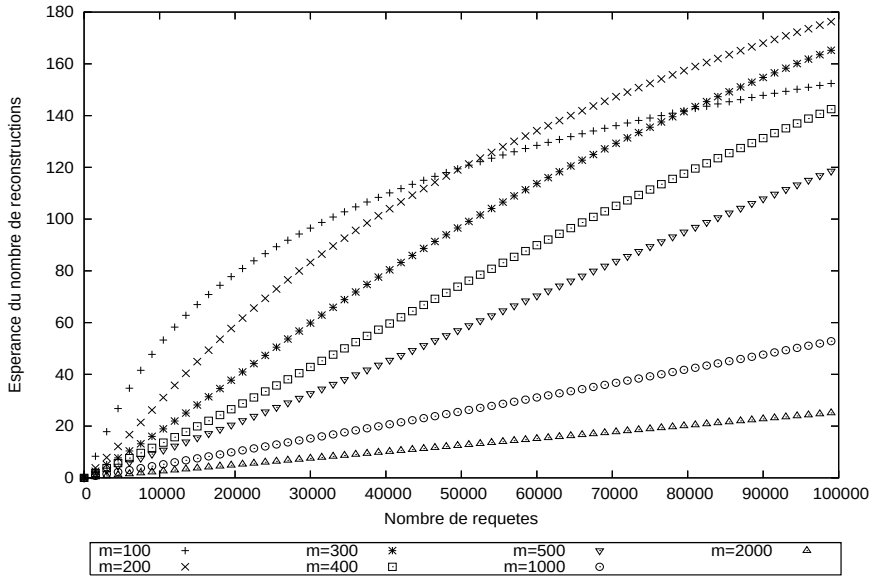

FIG. 1: $E(m_0,h)$ en fonction de h (pour m_0 fixé).

La première constatation que nous pouvons faire concerne le très faible nombre de reconstructions moyen induit par notre algorithme. En effet, on peut lire sur la figure 1 que pour des groupes de départ dont la taille varie de 100 à 2000, et pour un nombre de requêtes allant jusqu'à 100000, l'espérance maximum du nombre de reconstruction (atteinte ici pour $m_0 = 200$ et $h = 100000$) est toujours inférieure à 180 (alors que pour les mêmes valeurs de m_0 et h, le pire des cas induit $\frac{1}{2}(m_0+h) = \frac{1}{2}(200+100000) = 50100$ reconstructions).

Les résultats de la figure 2 permettent quant à eux de montrer l'évolution de l'espérance du nombre de reconstructions en fonction de la taille du groupe de départ lorsque le nombre de requêtes est fixé. Les courbes alors obtenues montrent que l'espérance n'est pas monotone en fonction de la taille du groupe de départ (comme on aurait pu le penser intuitivement *a priori*).

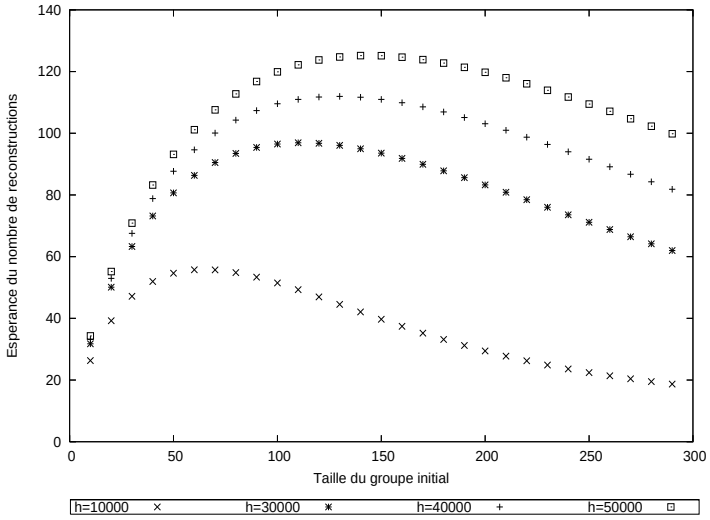

FIG. 2: $E(m,h)$ en fonction de m (pour h fixé).

Représentativité de l'espérance Nous nous sommes également intéressés, de manière expérimentale, à la "représentativité" de l'espérance. En effet, nous avons observé qu'en moyenne, sur 100000 expériences avec un groupe de départ de taille $m_0 = 100$ et un nombre de requêtes variant de 100 à 60000, environ 55% des solutions induisent un nombre de reconstructions inférieur à l'espérance. Dans le cas particulier où $h \leq m_0$, 60% des solutions ne nécessitent aucune reconstructions. Nous avons également observé que 90% des solutions induisent un nombre de reconstructions inférieur à 1.5 fois l'espérance. Ces résultats montrent que la majorité des solutions induisent un nombre de reconstructions proche de l'espérance.

Conclusion et perspectives

Nous avons étudié les performances moyennes d'un algorithme simple et pratique de mise à jour d'un arbre de connexion pour un groupe dynamique. Ces premiers résultats nous ont permis d'affiner considérablement l'étude analytique du nombre de reconstructions, jusqu'ici uniquement évalué dans le pire cas.

Nous espérons par la suite donner une expression analytique exacte ou des encadrements de $E(m_0,h)$ permettant une manipulation plus aisée des valeurs de l'espérance et aider les opérateurs gérant ce type de groupes d'utilisateurs à estimer au mieux le coût global induit par les reconstructions pour le maintient d'arbres de diamètre limités.

Références

[IW91] M. Imase and B.M. Waxman. Dynamic steiner tree problem. *SIAM J. Discr. Math.*, 4(3) :369–384, 1991.

[Thi06] N. Thibault. *Algorithmes d'approximation pour l'optimisation en ligne d'ordonnancements et de structures de communications*. PhD thesis, Université d'Evry, 2006.

[TL] N. Thibault and C. Laforest. An optimal rebuilding strategy for an incremental tree problem. *Journal of interconnection networks (à paraître)*.

[TL04a] N. Thibault and C. Laforest. Deux méthodes incrémentales pour le maintien d'un arbre de connexion. In *AlgoTel*, pages 63–67, 2004.

[TL04b] N. Thibault and C. Laforest. An optimal online strategy to increment connection trees. In *IEEE Workshop on Adaptive Wireless Networks, (in conjuction with GLOBECOM)*, 2004.

[TL06] N. Thibault and C. Laforest. An optimal rebuilding strategy for a decremental tree problem. In *SIROCCO*, LNCS 4056, pages 157–170. Springer, 2006.

[TL07] N. Thibault and C. Laforest. Minimizing the number of critical stages for the on-line steiner tree problem. In *INOC (à paraître)*, 2007.