Routage pair-à-pair dans les réseaux spontanés à large échelle

Aline C. Viana[1,2], Marcelo D. de Amorim[1], Serge Fdida[1], and José F. de Rezende[2] †

[1]*Laboratoire d'Informatique de Paris 6 – Université Pierre et Marie Curie*
8, rue du Capitaine Scott – 75015 – Paris – France
Mél : {viana,amorim,sf}@rp.lip6.fr

[2]*Grupo de Teleinformática e Automação – COPPE/EE – Universidade Federal do Rio de Janeiro*
P.O. Box 68504 – 21945-970 – Rio de Janeiro – RJ – Brasil
Mél : rezende@gta.ufrj.br

Le découplage entre l'identifiant et l'adresse topologique d'une entité permet aux systèmes communicants de faire face, d'une manière simple, au facteur d'échelle. Cet article présente Tribe, un protocole de routage pair-à-pair basé sur le concept de table de hachage distribuée. Dans cette approche, l'adresse topologique courante d'un nœud est stockée dans un autre nœud qui joue le rôle d'une ancre. Les ancres sont responsables pour traduire l'identifiant d'un nœud en une adresse dépendante de la topologie. Tribe est une solution originale, robuste, autonome et capable de gérer la mobilité des utilisateurs.

Keywords: Routage, auto-organisation, tables de hachage distribuées.

1 Introduction

Actuellement, les systèmes téléinformatiques se dirigent vers une architecture où l'utilisateur est nomade et l'accès au réseau ne dépend plus d'un ordinateur fixe. Les utilisateurs s'équipent de plus en plus avec des équipements mobiles, tels que les ordinateurs portables, les organisateurs personnels et les téléphones portables. Ce scénario permettra aux utilisateurs d'établir des réseaux spontanés indépendamment d'une infrastructure fixe. Le point crucial dans ces systèmes de communication est de garantir la connectivité entre les nœuds communicants. En autres mots, le *routage* redevient le composant principal car on ne peut plus dépendre d'une infrastructure telle que l'Internet. Cette nouvelle conception des réseaux a pour résultat de rendre le réseau aux usagers, en leur proposant un accès plus large aux contenus. Ainsi, il n'y a plus de distinction entre client et serveur mais une parfaite symétrie, un système où tous les terminaux sont des sources potentielles de données ; les communications se développent de manière autonome et spontanée entre les terminaux.

Dans ce papier, nous présentons une nouvelle stratégie de routage dans les réseaux spontanés exigeant l'auto-organisation des nœuds. Il s'agit du découplage de l'*information de localisation* d'un nœud de son *emplacement* courant dans le réseau. Un nœud quelconque maintient son information de localisation à jour dans un nœud qui jouera le rôle de point de rendez-vous – le nœud *ancre*. À chaque fois qu'une source désire communiquer avec une destination, elle doit d'abord demander à l'ancre correspondante quelle est l'adresse courante de la destination. Cette stratégie est au cœur du protocole Tribe proposé dans cet article.

Tribe est un système distribué sans aucun contrôle centralisé, où tous les nœuds ont les mêmes responsabilités et toute communication est symétrique. Les nœuds créent une infrastructure réseau qui décrit la

†Ce travail est financé en partie par CAPES, COFECUB, UFRJ, CNRS et EuronetLab.

localisation relative du nœud par rapport à son voisinage. Chaque nœud a un identificateur universel qui est associé à un identificateur temporaire. Ce dernier décrit la localisation relative du nœud dans le réseau.

La suite de cet article détaille le protocole Tribe. Dans la Section 2 nous décrivons la stratégie de routage indirect employée dans le protocole Tribe. La Section 3 discute quelques propriétés de l'infrastructure réseau. Le protocole Tribe est présenté dans la Section 4. Finalement, le Sections 5 présent une discussion sur les travaux liés et la conclusion de cet article.

2 Routage indirect

Dans le cadre de l'Internet, une solution a été proposée pour gérer la mobilité des nœuds : le Mobile IP. Mobile IP est un protocole qui permet à des terminaux mobiles de se déplacer dans différents réseaux tout en gardant valide son adresse IP, ce qui évite toute reconfiguration de la part de l'utilisateur final. Lorsqu'un terminal mobile change de sous-réseau, son adresse doit être également mise à jour. Le protocole Mobile IP est donc une bonne solution s'il existe une infrastructure supportant le concept de "*home agent*" (agent mère) fixe. Si tous les nœuds sont susceptibles de bouger, *y compris l'agent mère*, cette stratégie ne peut pas être directement appliquée.

Cet article s'appuie sur une nouvelle technique de communication que l'on appelle *routage indirect*. L'essence de cette approche est de manipuler les informations de routage d'une façon originale par rapport à la technique utilisée par le protocole IP traditionnel. Dans IP, chaque routeur possède une information permettant l'acheminement du message de bout en bout – soit par une interface de sortie bien définie, soit par une sortie par défaut. Dans le routage indirect, le processus est divisé en deux parties essentielles :

1. **Localisation :** avant toute communication, la destination doit être localisée et mappée en une adresse valable et identifiable dans l'espace d'adressage connu par tous les nœuds susceptibles d'intégrer le réseau ;
2. **Communication :** la source peut communiquer avec la destination en utilisant l'adresse obtenue dans la phase de localisation.

Les avantages de cette technique sont multiples. Les nœuds ne sont pas obligés de stocker des informations de routage dans tous les autres nœuds du système. Chaque nœud est responsable pour une partie des informations de localisation des nœuds existants. Ainsi, ces informations sont complètement distribuées parmi les éléments du réseau – le passage à l'échelle est donc assuré. Un autre avantage de cette technique est son faible coût de maintenance. Chaque nœud maintient sa localisation exacte mise à jour uniquement chez son nœud ancre – qui joue le même rôle que l'agent mère au Mobile IP –, et uniquement chez celui-ci. Si le nœud change de place, il doit informer sa nouvelle localisation à son nœud ancre. [‡]

3 Propriétés du support réseau

De façon à couvrir tous les aspects nécessaires au déploiement des futurs systèmes de communication pervasifs, l'infrastructure réseau associée au protocole Tribe doit être capable d'assurer un certain nombre de propriétés :

- **Contrôle distribué :** un certain nombre de modèles de réseaux, comme par exemple l'interconnexion d'équipements intelligents interactifs, les applications pair-à-pair et les réseaux ad hoc sont intrinsèquement décentralisés. Une des principales caractéristiques de ces systèmes est la procédure de distribution d'informations et de responsabilités, qui doit être robuste au facteur d'échelle et aux failles. La technique de distribution retenue dans Tribe est le mapping `clé` $\rightarrow$ `valeur` au travers de tables de hachage distribuées (DHT – *distributed hash tables*). Une DHT est utilisée aussi bien pour distribuer les informations parmi les nœuds que pour permettre à un nœud de déterminer, d'une façon précise, son nœud ancre ;

‡ Il est clair qu'il faut également que l'identité d'un nœud soit connue par ses voisins. Mais cet échange se fait naturellement (environnement sans fil) et ne compromet pas le fonctionnement global du réseau.

- **Association physique – logique :** si la représentation logique du positionnement d'un nœud dans une topologie reflète sa localisation physique par rapport à ses voisins, le service de distribution d'information pour le routage est largement simplifié. Ceci peut être constaté au travers des nombreuses propositions de routage en utilisant l'emplacement physique (coordonnées) des nœuds (par exemple, par le biais de GPS – *Global Positioning System*) ;
- **Routage indépendant d'IP :** l'architecture IP est adaptée (elle a déjà fait ses preuves) à des environnements reliés à une infrastructure fixe. Même pour le traitement de la mobilité, elle exige la présence d'un agent mère fixe ;
- **Auto-organisation :** les réseaux auto-organisables doivent être indépendants de la technologie sous-jacente [1]. Ceci est la conséquence naturelle de l'avancement des techniques et du comportement des usagers. Contrairement aux modèles précédents, qui étaient développés autour de l'entité réseau, les futures générations seront dirigées par les besoins des utilisateurs ;
- **Support à la mobilité :** les nœuds devront être capables de se déplacer librement dans le réseau, mais aussi pouvoir le joindre ou le quitter, selon leur volonté, d'une façon robuste. Ces opérations ne doivent pas nuire au bon fonctionnement global du réseau et doivent résister au facteur d'échelle. De la même manière, les nœuds ayant une partie de responsabilité dans l'ensemble du réseau, ils doivent pouvoir déléguer et/ou se charger des responsabilités des autres nœuds.

4 Le protocole Tribe

Dans Tribe, un nœud x possède trois identificateurs. Le premier, appelé identificateur universel U_x, représente l'identité du nœud. Cet identificateur est connu de tous les nœuds du systèmes (au moins des nœuds susceptibles de communiquer avec x) et est complètement indépendant d'un adressage au niveau réseau. L'identificateur universel peut être un nom, un numéro ou même (rien n'empêche) une adresse IP.

Le deuxième identificateur est l'adresse virtuelle V_x, obtenue à partir de la traduction de U_x en une valeur dans un espace d'adressage $\mathcal{V}$ connu. La fonction de traduction est également rendue publique et, de cette manière, connue de tous les nœuds. La fonction de cet identificateur est d'acheminer une requête de localisation vers le nœud ancre du nœud recherché.

Le dernier identificateur est l'adresse relative du nœud dans le réseau, E_x. Cette adresse est dépendante de sa localisation et est obtenue par l'intermédiaire du voisinage temporaire du nœud. L'utilisation de l'adressage relatif est le seul moyen de router dans le réseau.

La figure 1 illustre le fonctionnement de base du processus de routage indirect. Soit A le nœud voulant établir une communication avec le nœud B. Tout au début, A ne sait comment atteindre B car la localisation de B est inconnue. A doit donc contacter le nœud qui connaît cette localisation, c'est-à-dire le nœud ancre de B. Appelons ce nœud H. H connaît la localisation de B simplement parce que B maintient sa propre localisation dans le réseau à jour chez H. Autrement dit, H agit comme un point de "rendez-vous" entre B et n'importe quel autre nœud voulant contacter B. La particularité de cette approche est que les points de rendez-vous sont distribués et dépendants uniquement de chaque nœud. Après avoir reçu la demande de A, le nœud H répond avec un message contenant la localisation actuelle de B. La communication entre A et B peut maintenant être établie.

4.1 Le système d'adressage

Pour qu'un système supportant les fonctionnalités décrites ci-dessus soit mis en place, un certain nombre de besoins doit être respecté. Comme décrit auparavant, un nœud x est identifié d'une façon unique dans le réseau par une adresse relative, E_x. Cette adresse lui est attribuée par un de ses nouveaux voisins lorsque le nœud arrive dans la topologie. L'adresse relative est donc dépendante de la position physique du nœud. À l'origine, ce même nœud possède un identificateur publique et universel (connu de tous), noté U_x. Cet identificateur est complètement indépendant de n'importe quelle caractéristique de la configuration physique du réseau. En appliquant une fonction de hachage sur cet identificateur (comme par exemple SHA-1), un numéro de m-bits, $\hat{U}_x$, peut être associé au nœud x. La taille m doit être suffisamment grande pour minimiser la probabilité que deux traductions donnent le même résultat. Ensuite, $\hat{U}_x$ est traduit, par l'intermédiaire

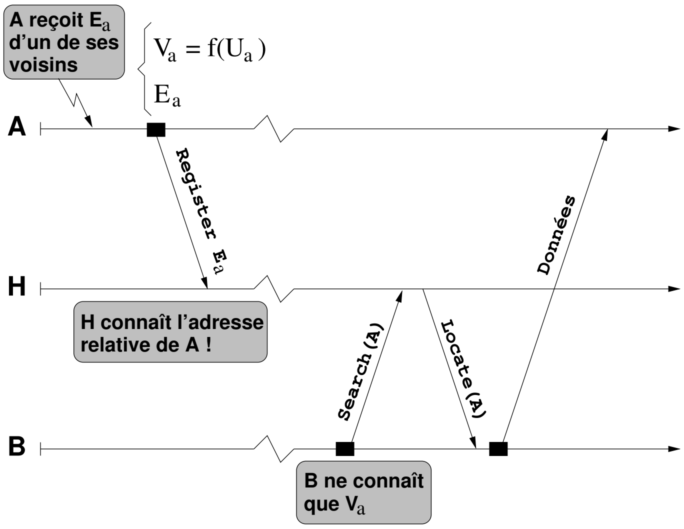

FIG. 1 – *Principe du routage indirect.*

d'une fonction de hachage linéaire congruentielle quelconque, en une valeur contenue dans un espace de adressage connu $\mathcal{V}$. On appelle le résultat de cette traduction l'identificateur virtuel $V_x \in \mathcal{V}$.

Le nœud x est donc complètement identifiable par le triplet (U_x, V_x, E_x). Étant donné que l'adresse relative est dépendante de la localisation, l'acheminement des messages peut être fait de manière unique, comme nous allons le voir par la suite. Le nœud x doit enregistrer son adresse relative E_x chez son nœud ancre, que nous appellerons H_x. Il est important de noter que même le nœud x ne connaît pas la vraie identité de H_x, c'est-à-dire son adresse relative R_h.[§] Pour bien comprendre comment x procède pour contacter h (jusqu'à présent inconnu), il faut tout d'abord spécifier le système d'adressage.

Le point central du système d'adressage est la distribution de l'espace d'adressage $\mathcal{V}$ entre les nœuds du réseau. Supposons que $\mathcal{V}$ suit un modèle circulaire de taille m, c'est-à-dire $\mathcal{V} = [0, 2^m - 1]$. À tout moment, chaque nœud sera responsable d'une partie de cet espace d'adressage. Le nœud x sera donc responsable de la région $\mathcal{R}_x = [R_x^{\ominus}, R_x^{\oplus})$, où $[R_x^{\ominus}, R_x^{\oplus}) \subset [0, 2^m)$.[¶] Pour n'importe quel autre nœud y, nous avons $\mathcal{R}_x \bigcap \mathcal{R}_y = 0$. La limite inférieure de la région détenue par un nœud sera l'identificateur relatif du nœud. Ainsi, pour le nœud x, nous avons $E_x = R_x^{\ominus}$.

4.2 Arrivée d'un nœud

Lorsqu'un nœud arrive dans le réseau, il met en marche un algorithme de *bootstrap* pour identifier ses voisins. Comme l'environnement en question est du type sans fil, les voisins d'un nœud sont tous ceux qui se situent dans sa zone de couverture. L'un de ses voisins lui donnera une partie de sa propre région de contrôle. Soit $\mathbf{A}_x = \{a_1, a_2, \ldots, a_k\}$ l'ensemble de k voisins du nœud x. La *liste de voisins*, $\mathbf{L}_x$, du nœud x est définie comme étant $\mathbf{L}_x = \{[E_{a_1}, \mathcal{R}_{a_1}], [E_{a_2}, \mathcal{R}_{a_2}], \ldots, [E_{a_k}, \mathcal{R}_{a_k}]\}$, où $\mathcal{R}_{a_j}$ est la région sous la responsabilité du nœud a_j, c'est-à-dire $\mathcal{R}_{a_j} = [R_{a_j}^{\ominus}, R_{a_j}^{\oplus})$, $\forall a_j \in \mathbf{A}_x$.

Si le nombre d'éléments dans $\mathbf{A}_x$ est égal ou plus grand que deux, le système doit appliquer une méthode pour décider lequel des voisins partagera sa région avec le nouvel arrivant. Jusqu'à présent, deux critères ont été définis de façon à sélectionner de manière unique le nœud qui partagera sa région. En premier lieu, le nœud qui vient d'arriver recevra la région du voisin qui possède la plus grande région de contrôle. Ce nœud sera son *voisin père*. En second lieu, la région attribuée sera la moitié supérieure de la région du voisin père. Ainsi, le voisin père p_x est choisi de façon que :

$$S(\mathcal{R}_{p_x}) = \max\{S(\mathcal{R}_{a_j})\}, \quad p_x \in \mathbf{A}_x, \forall a_j \in \mathbf{A}_x,$$

§ Sachant que l'unique manière de router dans le réseau est par l'intermédiaire des adresses relatives.

¶ Pour une question de simplicité de notation, on utilise $[X, Y)$ à la place de $[X, Y-1]$.

où $S(\mathcal{R}_{a_j})$ est la taille de la région de a_j's, c'est-à-dire, d'une forme générale, $S(\mathcal{R}_z) = R_z^{\oplus} - R_z^{\ominus}$.

À la fin de l'opération, le nouvel arrivant et son voisin père se retrouvent dans la situation suivante :

$$\mathcal{R}_x = \begin{cases} R_x^{\ominus} = & \frac{R_{p_x}^{\oplus} + R_{p_x}^{\ominus}}{2}, \\ R_x^{\oplus} = & R_{p_x}^{\oplus}. \end{cases}$$

et :

$$\mathcal{R}_{p_x} = \begin{cases} R_{p_x}^{\ominus} = & R_{p_x}^{\ominus}, \\ R_{p_x}^{\oplus} = & \frac{R_{p_x}^{\oplus} + R_{p_x}^{\ominus}}{2} - 1. \end{cases}$$

La méthode présentée ci-dessus assure que toutes les adresses relatives sont uniques dans tout le réseau. En plus, les adresses ne changent pas si les nœuds deviennent voisin père d'un nouvel arrivant (sauf, bien évidement, si le nœud se déplace).

4.3 Enregistrement

Après avoir reçu sa région de contrôle et son adresse relative, le nouvel arrivant (le nœud x) doit s'enregistrer auprès de son nœud ancre, h_x. Il est important de noter encore une fois que x ne connaît pas la vraie identité de h_x – les nœuds ancres sont découverts à la demande. Par ailleurs, le fait qu'un nœud n'ait pas besoin de connaître son nœud ancre est le principe fondamental qui fait de cette approche une architecture auto-organisable et autonome. Le nœud calcule, à partir de son identificateur universel, son adresse virtuelle. Ceci est équivalent à la procédure de localisation d'un nœud. Un message contenant son adresse relative est envoyé *vers* le nœud dont la région de contrôle contient l'adresse virtuelle de x. Cette procédure d'envoyer "vers" le nœud ancre est possible grâce au mécanisme de routage saut-à-saut (expliqué par la suite).

Lorsqu'un nœud désire communiquer avec x, il doit d'abord demander à h_x quel est la localisation actuelle de x. Le principal avantage de ce système est que h_x est une entité logique (c'est-à-dire, le nœud qui possède la région contenant l'adresse virtuelle de x). Il peut changer dans le temps sans que x ou celui qui veut communiquer avec x soient importunés. Il suffit que, lors d'un changement de nœud ancre, quelqu'un d'autre s'occupe de la région de contrôle abandonnée.

4.4 Procédure de routage

Chaque nœud connaît sa région et les régions de ses voisins. Ces informations sont suffisantes pour router un message et composent les tables de routage des nœuds. Les nœuds acheminent les message vers la destination en envoyant le message au voisin dont la région de contrôle s'approche le plus de la région de contrôle qui contient l'identificateur de la destination. La figure 2 en montre un exemple. Dans la topologie représentée, l'espace d'adressage est de 128 bits, $\mathcal{V} = [0, 2^{128} - 1]$.

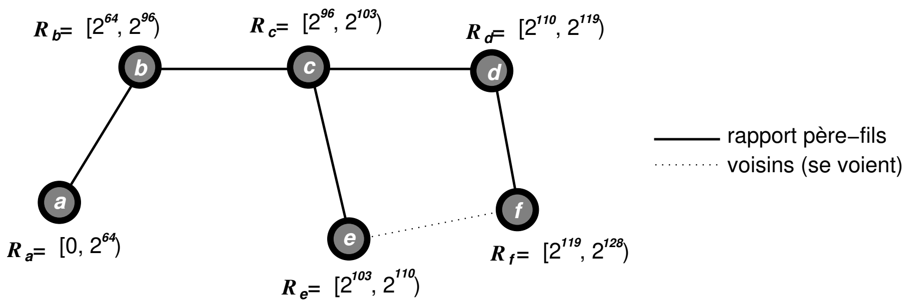

FIG. 2 – *Exemple de topologie basée sur des régions.*

Supposons que le nœud b souhaite envoyer un message à f en utilisant l'adresse relative $E_f = 2^{119}$. Au départ, b a le choix d'acheminer le message soit à a, soit à c (ses seuls voisins). Comme la région de c

s'approche plus de 2^{119}, b envoie le message à c. Ce dernier, à son tour, doit faire face à la même décision. Entre e et d, il finit par envoyer à d. Finalement, d peut livrer le message à la destination f.

5 Discussion

Dans cet article, nous avons présenté Tribe, un protocole de routage indirect basé sur le concept des tables de hachage distribuées. Tribe est générique, auto-organisable et indépendant des limitations imposées pour un adressage du type IP. Le protocole Tribe emploie une stratégie de routage distribuée où l'identificateur de m-bit d'un nœud détermine le nœud ancre responsable pour stocker l'information de localisation du nœud.

Des architectures comme CAN [2], Chord [3, 4], Tapestry [5], Pastry [6] et *i*3 [7] emploient un concept similaire de la stratégie de routage indirect. Néanmoins, différemment du protocole Tribe, elles sont traitées comme des infrastructures sur-couche (*overlay*) et créent une topologie au niveau applicatif.

Les projets Terminodes [8, 1] et Grid [9, 10] proposent une distribution d'informations de routage qui est similaire au protocole Tribe. La différence par rapport Tribe, c'est qu'ils utilisent d'informations géographiques pour créer la topologie et effectuer le routage.

Tribe est une solution originale, robuste, autonome et capable de gérer la mobilité des utilisateurs. Des caractéristiques de Tribe comprennent la simplicité et l'indépendance d'un positionnement géographique. L'article [11] présent plus des détails sur le protocole Tribe. Des travaux futurs prendront en compte, entre autres, la mobilité, le traitement de panne des nœuds, l'hétérogénéité des nœuds, la fusion de deux ou plusieurs réseaux créés indépendamment et la répartition de charge.

Références

[1] L. Blazevic, L. Buttyan, S. G. S. Capkun, J. P. Hubaux et J. Y. L. Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," *IEEE Computer Communications Magazine*, juin 2001.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp et S. Shenker, "A scalable content-addressable network," dans les *Proceedings of ACM SIGCOMM'01*, août 2001.

[3] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek et H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, fevrier 2003.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek et H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," dans les *Proceedings of ACM SIGCOMM'01*, août 2001.

[5] B. Y. Zhao, J. D. Kubiatowicz et A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," tech. rep., University of California, Berkeley - UCB/CSD-01-1142, EECS, mai 2001.

[6] A. Rowstron et P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," dans les *Proceedings of IFIP/ACM Middleware'01*, novembre 2001.

[7] I. Stoica, D. Adkins, S. Zhuang, S. Shenker et S. Surana, "Internet indirection infrastructure," dans les *Proceedings of ACM SIGCOMM'0*, août 2002.

[8] J. P. Hubaux, T. Gross, J. Y. L. Boudec et M. Vetterli, "Towards self-organized mobile ad hoc networks: the terminodes project," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 118–124, janvier 2001.

[9] Grid Project. `http://www.pdos.lcs.mit.edu/grid/`.

[10] B. A. Chambers, "The grid roofnet: a rooftop ad hoc wireless network," Master's thesis, Deparment of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology - MIT, juin 2002.

[11] A. C. Viana, M. D. Amorim, S. Fdida et J. F. Rezende, "Indirect routing using distributed location information," dans les *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PERCOM'03)*, (Dallas-Fort Worth, Texas), mars 2003.