Des approches multi-agents pour le contrôle des réseaux autonomes

Hugo Pommier, Benoît Romito, Grégory Bonnet, François Bourdon

Des approches multi-agents pour le contrôle des réseaux autonomes

Hugo Pommier, Benoit Romito, Grégory Bonnet, François Bourdon

UMR CNRS 6072 GREYC – Équipe Modèles, Agents, Décision (MAD)
Université de Caen Basse-Normandie
Boulevard du Maréchal Juin BP 5186, 14032 Caen Cedex, France
prenom.nom@unicaen.fr

Résumé : Aujourd'hui, les réseaux ont cru en complexité en raison de leur taille, leur hétérogénéité et leur dynamisme ; à tel point qu'un contrôle central n'est plus possible. Cet article de positionnement a pour vocation de montrer en quoi le paradigme des systèmes multi-agents peut contribuer aux problématiques des réseaux, et plus précisement, des réseaux pair-à-pair. Nous y présenterons trois applications : l'utilisation d'algorithmes de nuées pour le stockage décentralisé de documents, une approche coopérative pour la répartition de charge et la détection de coalitions d'agents malveillants pour contrecarrer les attaques Sybil.
Mots-clés : agents malveillants, algorithmes bio-inspirés, réseaux autonomes, réseaux pair-à-pair

1. Introduction

Depuis quelques années, un nouveau domaine de recherche dans le cadre des réseaux, les *réseaux autonomes*, a vu le jour. Ces travaux se placent dans la lignée du calcul autonome proposé par (Kephart, 2005) et visent à améliorer la robutesse des réseaux et des services face à des événements imprévus comme une transformation de topologie, de charge de travail ou de caractéristiques physiques. En effet, les réseaux ont cru en complexité en raison de leur taille, leur hétérogénéité et leur dynamisme ; à tel point qu'un contrôle central et/ou par le biais d'un opérateur humain n'est plus possible.

Les algorithmes et les protocoles décentralisés sont prometteurs car ils permettent de décomposer les problèmes, de s'adapter à l'augmentation en taille des réseaux et de considérer des systèmes ouverts. De plus, la décentralisation

vise à réduire les coûts de maintenance et d'opération en ne considérant que localement le système. Plus généralement, nous pouvons remarquer que ces questions touchent au *paradigme multi-agent* de par leur distribution, décentralisation, partage et décomposition des tâches, coopération et adaptation.

Cet article de positionnement a pour vocation de montrer en quoi les systèmes multi-agents peuvent contribuer à répondre aux problématiques des réseaux, et plus précisement, des réseaux pair-à-pair qui présentent toutes les caractéristiques des systèmes traités par le paradigme agent. Nous montrerons de quelles manière ces deux domaines sont liés (cf. section 2.) puis nous présenterons trois applications sur lesquelles nous travaillons : l'utilisation d'algorithmes de nuées pour le stockage décentralisé de documents (cf. section 3.) ; une approche coopérative pour la répartition de charge (cf. section 4.) et la détection de coalitions d'agents malveillants pour contrecarrer les attaques Sybil (cf. section 5.).

2. Des réseaux pair-à-pair aux systèmes multi-agents

Il existe une analogie intuitive entre les réseaux pair-à-pair et les systèmes multi-agents. Comme le rappellent (Lua *et al.*, 2004), un réseau pair-à-pair est un modèle distribué où les entités appelées pairs jouent le double rôle de client et de serveur et interagissent afin d'offrir à une communauté un service décentralisé. Ces services sont des applications informatiques distribuées comme les grilles de calcul, le stockage distribué (OceanStore), le partage de fichier (BitTorrent), la diffusion de vidéos (PPLive) ou la téléphonie (Skype).

Caractéristiques	**Dans un contexte pair-à-pair**
Décentralisé	pas de contrôleur central
Dynamique	un taux d'attrition important
Large échelle	plusieurs dizaines de milliers de pairs
Ouvert	arrivée de nouveaux pairs

TABLE 1: Caractéristiques des réseaux pair-à-pair

Les réseaux pair-à-pair présentent des caractéristiques, résumées dans le tableau 1, que l'on retrouve dans les systèmes multi-agents. De plus, un pair est un agent logiciel dont les caractéristiques identifiées par (Boissier *et al.*, 2004) sont mises en perspective dans le tableau 2. En raison de cette analo-

gie, les réseaux pair-à-pair sont une instance de systèmes multi-agents et les techniques multi-agents peuvent servir à les dôter de plus d'autonomie.

Agent	**Pair**
Adaptatif	un pair s'adapte aux changements locaux de topologie
Autonome	un pair n'est pas contrôlable par un autre pair
Mobile	un pair se déplace au sein du réseau
Persistant	un pair est un processus continu
Pro-actif	un pair joue le rôle de client
Réactif	un pair joue le rôle de serveur
Social	un pair communique et coopère avec les autres pairs

TABLE 2: Analogie entre pairs et agents

Dans ce contexte, trois problématiques inhérentes à la gestion de réseaux pair-à-pair, nous intéressent :

1. la modélisation de l'information dans un cadre totalement décentralisé. Dans le contexte du stockage de données réparti, nous proposons une approche où la représentation d'un fichier s'effectue sous la forme d'un système multi-agent bio-inspiré. Nous traitons alors des questions de placement et de recherche de l'information ;
2. nous étudions, au travers d'une application de gestion de charge d'un service distribué d'analyse de traces, comment une coopération entre agents autonomes, basée sur des mécanismes décentralisés d'apprentissage, de restructuration et de synchronisation spatio-temporelle, produit une forme d'auto-organisation constructive d'un réseau ouvert ;
3. si l'utilisation de systèmes de réputation peut être efficace pour détecter un unique pair malveillant, ils ne peuvent détecter des coalitions de pairs malicieux. Dans ce contexte, nous nous intéressons à la présence d'agents dissimulant leurs accointances, et nous intéressons à la découverte de ces accointances et de ces organisations implicites dans le réseau.

Bien entendu, ces trois points ne couvrent pas l'ensemble des problématiques liée aux réseaux pair-à-pair et d'autres peuvent être envisagées comme l'adaptation du réseau au comportement de ses utilisateurs (Bonnet *et al.*, 2011), la mise en cache coopérative (Bonnet & Doyen, 2010) ou la simulation des comportements égoïstes (Navarrete Gutierrez *et al.*, 2010).

3. Modélisation bio-inspirée de l'information

La gestion de l'information dans un réseau pair-à-pair peut avoir de multiples supports (mail, stockage, communication). Dans tous les cas, un besoin de robustesse est nécessaire, ce qui se traduit par la mise en place de :

1. *une politique de redondance*, obtenue par la réplication ou en utilisant un code d'effacement (Plank, 1997) dans lequel la donnée est découpée en m blocs puis encodée en $m + n$ fragments. Le système est alors en mesure de tolérer n disparitions de fragments ;
2. *une politique de supervision*, suivant l'évolution du nombre de fragments d'un document pour le réparer en cas de faute (Giroire *et al.*, 2009) ;
3. *une politique de réparation*, définissant le seuil à partir duquel il faut restaurer des fragments (Dalle *et al.*, 2009) ;

Dans les approches classiques décentralisées (Haeberlen *et al.*, 2005; Druschel & Rowstron, 2001), l'application des ces politiques est laissée à la responsabilité des pairs en qui une confiance complète est accordée. Nous proposons d'ouvrir sur de nouvelles perspectives en terme d'autonomie et de flexibilité en proposant un modèle de gestion sécurisée de l'information fondé sur un système d'agents mobiles réactifs. Chaque fragment généré par une politique de redondance est encapsulé dans un agent mobile : le document est alors représenté par le groupe de ses agents fragments. Ainsi, les politiques de supervision et de réparation ne repose plus sur les pairs mais sont transférées sur les agents qui ont la possibilité d'appliquer leur propre politique. Il est alors possible d'avoir plusieurs documents évoluant dans le même environnement mais ayant des comportements différents réduisant ainsi le rôle des pairs à de simples hébergeurs. L'apport de la mobilité offre notamment aux documents la possibilité d'adapter leur placement dans le réseau lorsque l'état de leur environnement ne satisfait plus certains critères.

Pour garantir des propriétés de localité et de décentralisation, nous avons adapté les trois règles (cohésion, séparation, alignement) de (Reynolds, 1987) pour régir le déplacement des agents mobiles dans notre modèle (Pommier & Bourdon, 2009). Le comportement global émergeant est alors une nuée de fragments similaire aux nuées d'oiseaux.

3.1. Sûreté et disponibilité de l'information

Nous avons ajouté au modèle de flocking une politique de réparation de type *lazy* consistant à définir un seuil de tolérance r tel que $m \leq r < m + n$. Si le nombre d'agents passe en-dessous du seuil r alors un agent, choisi par un mécanisme d'élection de chef, ordonne à m agents de reconstruire le nombre de fragments manquants. Toutefois, l'application de règles locales asynchrones associées aux temps de transit durant les déplacements peuvent provoquer des ruptures dans la cohésion d'une nuée. La valeur de cohésion est vue comme la taille de la plus grande composante connexe d'une nuée. Les conséquences des variations de la cohésion sont directes : dans le cas où la nuée se sépare en sous-nuées avec une cohésion inférieure à m, la donnée devient temporairement indisponible et engendre des réparations de fragments.

Nous avons montré qu'il est possible de trouver, pour un schéma de fragmentation et un seuil de réparation donné, le nombre de fragments suffisant pour que la rupture de cohésion d'une nuée ne vienne pas perturber la politique de réparation. De plus, nous observons que la nuée trouve un équilibre à la suite de réparations successives. Nous pouvons ainsi paramétrer m et n en fonction de la typologie du réseau.

3.2. Recherche de l'information

Pour retrouver un document stocké sous la forme d'une nuée en mouvement, il est nécessaire de proposer une méthode de recherche qui soit efficace même en cas de forte mobilité. Les méthodes usuelles à base d'inondations, bien qu'efficaces pour rechercher de l'information, sont beaucoup trop coûteuses en termes de communication. Pour retrouver un groupe d'agents nous nous appuyons sur deux algorithmes décentralisés construit à partir d'agents mobiles. Le premier repose sur une marche aléatoire, et le second utilise les capacités de stigmergie des agents. Ces derniers sont capables de déposer des phéromones pour marquer leurs déplacements. Les règles de flocking précédemment mises en place, assurant une connexité entre les agents stockant des morceaux d'information, doivent ainsi permettre de récupérer à moindre coût l'ensemble ou une partie des fragments d'un document puis de le reconstruire.

Les expérimentations menées dans (Pommier *et al.*, 2011) ont permis de montrer la faible influence de la vitesse de déplacement d'une nuée sur les différents type d'agent de recherche. Bien que l'algorithme utilisant des phéromones couvre le réseau plus rapidement qu'une marche aléatoire, les deux algorithmes proposent des résultats assez proches.

4. Coopération et auto-organisation dans les réseaux

Nous appelons *fermeture efficiente* les mécanismes internes d'un système qui le conduisent, via un processus auto-organisé, vers un état stable. Ce processus met en jeu des mécanismes d'apprentissage, de transformation structurelle et de synchronisation spatio-temporelle, produisant une coopération implicite entre les entités du système. On retrouve ces dimensions dans des contextes très différents comme par exemple celui où des personnes doivent se synchroniser pour pousser efficacement le véhicule en panne.

Nous illustrons notre approche avec la mise en place d'un service décentralisé d'analyse de traces. Ce problème $\mathbf{NP}$-complet (Cornuéjols *et al.*, 1990) a été identifié en recherche opérationnelle sous le nom du *problème de positionnement*, dont l'une des multiples variantes (Hamacher *et al.*, 1996) est celui du positionnement discret sans contrainte de capacité [1].

4.1. Formulation du problème

Nous considérons un grand nombre n de clients $\{c_1, \ldots, c_n\}$ possèdant des besoins changeant et imprévisibles en matière d'analyse de traces. Afin de ne pas être tributaire des aléas du réseau et de faire face à des situations évolutives au meilleur coût, le prestataire du service dispose d'un nombre m de serveurs $\{s_1, \ldots, s_m\}$ placés sur d machines du réseau. Les machines se déplacent et anticipent la puissance de calcul nécessaire pour faire face aux demandes qu'elles auront dans leur avenir proche.

On peut simplifier ce problème en considérant une grille de positions sur lesquelles se déplacer les serveurs, sachant que les clients sont considérés comme étant sur des points fixes de la grille. Optimiser notre système consiste à trouver la meilleure position des serveurs sur la grille afin de satisfaire les demandes des clients. Cette simplification suppose que la situation optimale recherchée consiste à placer l'ensemble des serveurs en minimisant une distance globale qui correspondrait au coût du service pour le prestataire et conséquemment pour les clients.

4.2. Complexité du problème

En pratique, le calcul de la solution optimale est très vite impraticable. Pour parcourir tous les cas possibles et trouver l'optimum, il faudra faire nd^{2m} opé-

1. Uncapacitated Facility Location Problem (UFLP)

rations. On remarque que le nombre de serveurs m est extrêmement sensible. Avec 10 clients, 20 machines et 2 serveurs il faudra 2 millions d'opérations. Avec la même configuration mais 4 serveurs, ce qui est loin d'un réseau de type Internet, il faudra alors 500 mille milliards d'opérations, soit une journée entière de calcul pour une machine classique. Même si l'on utilise des techniques de recherche heuristique permettant de réduire de large sous-espaces des solutions non-optimales, avec l'algorithme de *séparation-valuation* par exemple, il est difficile de s'attaquer à la résolution exacte d'un tel problème.

Une autre complexité de calcul s'ajoute à la première. En effet le coût global des échanges entre les composants du système dépend fortement de la répartition des clients sur les serveurs. Cette deuxième complexité s'exprime par le nombre de Stirling de seconde espèce $S(n, k)$, qui correspond aux k-partitions d'un ensemble N à n éléments. Suivant la formule exacte (Comtet, 1970), ce nombre croît très rapidement en fonction du nombre de clients (n) et du nombre de serveurs (k-partitions). Par exemple, pour $S(6, 2)$ nous avons 31 cas, pour $S(15, 2)$ nous avons 16383 cas et pour $S(15, 4)$ nous avons 42 355 950 cas possibles de partitionnement des clients.

Chaque serveur de traitement de traces suit un protocole comportemental précis fondé sur : un ajustement interne à sa perception du contexte changeant, une modification de sa position (structurelle) dans le dispositif et une synchronisation de son processus décisionnel avec les autres serveurs qui interagissent avec lui. Pour illustrer les pistes ouvertes par ces approches, citons l'heuristique décentralisée de recherche d'un serveur de traces au plus près du client dont l'effet sur le système est de conduire systématiquement, pour une position donnée des serveurs, à la partition optimale des clients, et ce parmi le k-partitionnement possible. C'est un premier pas conduisant le comportement local vers l'aboutissement d'un comportement global recherché.

5. Détection d'agents malveillants

5.1. Des coalitions de faux agents

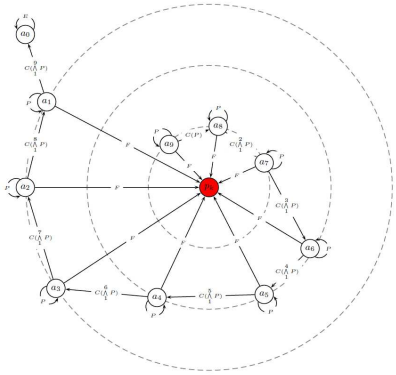
L'objectif de ces travaux est d'étudier les solutions multi-agents aux attaques Sybil sur les réseaux pair-à-pair. En effet, comme le montre (Douceur, 2002), une des menaces récurrentes envers ces réseaux est la présence de Sybil : de faux pairs qui soutiennent un ou plusieurs pairs malveillants pour accomplir différentes attaques. De telles coalitions de pairs peuvent, par exemple, contrecarrer les systèmes de réputation mis en place pour s'assurer

du bon comportement des utilisateurs (Hoffman *et al.*, 2009) ou les mécanismes d'incitation à la coopération (Morge & Mathieu, 2007). De nombreux travaux ont été proposés pour définir des protocoles insensibles aux Sybil mais, comme le montrent (Levine *et al.*, 2006), ces approches souffrent toutes d'une des limites suivantes : elles proposent de centraliser à nouveau le réseau, d'introduire un coût (monétaire ou en termes de calcul) pour entrer dans le réseau, ou préconisent l'utilisation d'un réseau social pour renforcer les liens de confiance au détriment des pairs nouveaux entrant.

5.2. Hypothèses de travail

Afin de pallier ces limites, nous proposons d'étudier les structures remarquables présentes dans un graphe d'interaction en faisant l'hypothèse que ces structures permettent de détecter des Sybil. Plusieurs intuitions nous permettent de formuler cette hypothèse : (1) les Sybil contrôlées par un même attaquant forment un amas de pairs fortement connexes (voisinage, ressources partagée, confiance mutuelle, interactions) ; (2) une Sybil peut fournir des informations contradictoires lorsqu'elle est interrogée par deux pairs distincts. Ceci peut fournir des heuristiques pour détecter des coalitions d'agents malveillants.

5.3. Identification de schémas d'interaction



Pour cela, nous nous intéressons à la simulation de ces attaques par des coalitions d'agents à partir de graphes d'interactions comme celui représenté ci-contre. Chaque nœud représente un agent et chaque arc est une interaction étiquettée par le rôle que joue l'un des agents envers l'autre.

Nous considérons par la suite les agents honnêtes du réseau. Ces derniers utilisent un protocole d'agrégation décentralisé pour reconstruire un graphe partiel en faisant l'hypothèse que les Sybil peuvent dissimuler des interactions. À partir de ce graphe partiel, nous envisageons l'utilisation de tech-

niques de découverte de liens (Getoor & Diehl, 2005) pour ensuite identifier des structures s'apparantant à une coalition malveillante.

6. Conclusion

Cet article de positionnement a présenté trois problématiques inhérentes à la gestion de réseaux pair-à-pair que nous traitons à l'aide d'approches multi-agents. Dans le contexte du stockage de données réparti, afin de rendre robuste aux défaillances le système, nous proposons une approche où les fichiers sont découpés en fragments encapsulés dans des agents mobiles. Ces agents mobiles se répartissent alors sur le réseau à l'aide d'un algorithme de nuée afin de maintenir une cohésion facilitant leur recherche, tout en évitant un placement statique de l'information. Nous nous intéressons également à la répartition automatique de charge en étudiant des protocoles décentralisé d'optimisation. Enfin, en raison de l'ouverture des réseaux, la présence de coalitions d'agents malveillants est envisagée. Dans ce contexte, nous nous intéressons à la découverte d'accointances et d'organisations cachées dans les réseaux. Ces problématiques ne sont toutefois pas les seules et nous pouvons citer la mise en cache coopérative pour le contrôle de trafic ou le placement décentralisé des pairs dans le réseau en fonction du comportement de leur utilisateur.

Références

BOISSIER O., GITTON S.-S. & GLIZE P. (2004). Caractéristiques des systèmes et des applications, systèmes multi-agents. *ARAGO*, **Vol. 29**, 25–54.

BONNET G. & DOYEN G. (2010). Coopération entre systèmes multi-agents appliquée au contrôle de trafic sur les réseaux pair-à-pair. In *Actes des 18es JFSMA*, p. 181–190.

BONNET G., ULLAH I., DOYEN G., FILLATRE L., GAÏTI D. & NIKIFOROV I. (2011). A semi-markovian individual model of users for P2P video streaming applications. In *Proceedings of the 4th NTMS*.

COMTET L. (1970). *Analyse combinatoire*, volume 2. Presses universitaires de France.

CORNUÉJOLS G., NEMHAUSER G.-L. & WOLSEY L.-A. (1990). *Discrete Location Theory*. John Wiley and Sons.

DALLE O., GIROIRE F., MONTEIRO J. & PERENNES S. (2009). Analysis of failure correlation impact on peer-to-peer storage systems. In *Proceedings of the 9th IEEE P2P*, p. 184–193.

DOUCEUR J.-R. (2002). The sybil attack. In *Proceedings of the 1st IPTPS*.

DRUSCHEL P. & ROWSTRON A. (2001). PAST : a large-scale, persistent peer-to-peer storage utility. In *Proceedings of the 8th HotOS*, p. 75–80.

GETOOR L. & DIEHL C. (2005). Link mining : a survey. *SIGKDD Explorations Special Issue on Link Mining*, **Vol. 7(2)**, 3–12.

GIROIRE F., MONTEIRO J. & PÉRENNES S. (2009). P2P storage systems : How much locality can they tolerate ? In *Proceedings of the 34th IEEE LCN*, p. 320–323.

HAEBERLEN A., MISLOVE A. & DRUSCHEL P. (2005). Glacier : highly durable, decentralized storage despite massive correlated failures. In *Proceedings of the 2nd NSDI*.

HAMACHER H.-W., NICKEL S. & SCHNEIDER A. (1996). *Classification of locations problems*. Rapport interne, Universitat Kaiserslautern.

HOFFMAN K., ZAGE D. & NITA-ROTARU C. (2009). A survey of attack and defense techniques for reputation systems. *ACM Computing Survey*, **Vol. 42(1)**, 1–31.

KEPHART J.-O. (2005). Research challenges of autonomic computing. In *Proceedings of the 27th ICSE*, p. 15–22.

LEVINE B.-N., SHIELDS C. & MARGOLIN N.-B. (2006). *A Survey of Solutions to the Sybil Attack*. Rapport interne, University of Massachussets.

LUA E.-K., CROWCROFT J., PIAS M., SHARMA R. & LIM S. (2004). A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Survey and Tutorial*, p. 1–22.

MORGE M. & MATHIEU P. (2007). Mécanisme de rétribution pour les systèmes P2P d'échange de fichiers. *Revue d'Ingénierie des Systèmes d'Information*, **Vol. 12**, 61–84.

NAVARRETE GUTIERREZ T., SIEBERT J., CIARLETTA L. & CHEVRIER V. (2010). Impact des dimensions spatiale et temporelle dans la modélisation d'un phénomène collectif de type « free-riding ». In *Actes des 18es JFSMA*, p. 119–128.

PLANK J. S. (1997). A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems. *Software – Practice & Experience*, **27**(9), 995–1012.

POMMIER H. & BOURDON F. (2009). Agents mobiles et réseaux pair-à-pair : Vers une gestion sécurisée de l'information répartie. *Revue d'Intelligence Artificielle*, **23**(5-6), 697–718.

POMMIER H., ROMITO B. & BOURDON F. (2011). Searching flocks in peer-to-peer networks. In *Proceedings of the 9th PAAMS*.

REYNOLDS C. W. (1987). Flocks, herds and schools : A distributed behavioral model. In *Proceedings of the 14th SIGGRAPH*, p. 25–34.