

# Approche Auto-adaptative Inspirée du Système Immunitaire pour la Mise en Oeuvre de l'Ubiquité

J. Gaber and M. Bakhouya  
Laboratoire SeT

Université de Technologie de Belfort-  
Montbéliard, 90010 Belfort, France

[gaber@utbm.fr](mailto:gaber@utbm.fr)

## RÉSUMÉ

Dans cet article, une approche auto-adaptative et auto-organisationnelle, basée sur les agents mobiles et inspirée du système immunitaire de l'Homme, pour la découverte de services dans un réseau dont l'évolution est aléatoire, est présentée. Cette approche est basée sur l'organisation des serveurs en communautés par la création de réseaux d'affinités pour représenter les services dans le réseau. L'établissement des liens d'affinités entre les serveurs permet de résoudre, par collaboration, les requêtes des utilisateurs. Cette approche permet au système de découverte de s'adapter d'une manière dynamique aux changements du réseau, à la disponibilité des services et aux requêtes des utilisateurs. L'objectif de cette approche est de mettre en oeuvre un paradigme alternatif, inverse au paradigme traditionnel client/serveur, qui permet la mise en oeuvre de l'ubiquité numérique; c'est le service qui vient vers le client et non le client qui prend l'initiative et demande une ressource ou un service en connaissant a priori son existence et sa localisation.

**Mots clefs:** Ubiquité numérique, Environnement pervasif, Découverte de services, Système immunitaire, Agents mobiles, Communautés de serveurs.

## ABSTRACT

In this article, a self-adaptive and a self-organizing approach based on the mobile agent paradigm and inspired by the human immune system for dynamic service discovery in large-scale network is presented. This approach is based on the organization of servers into communities by the creation of affinity relationships in order to represent services in the network. The establishment of relationship affinities between servers allows to solve, by collaboration, the user requests. This approach permits to the service discovery system to cope with the dynamic changes in the network, the services availability and the user requests. The aim of this approach is to carry out an alternative paradigm, opposed to the Client/server paradigm, which allows the design and the implementation of ubiquitous applications; it is the service that come to the user not the user who should initiates a request, should know a priori that the required service exists and should be able to provide the location of a server holding that service.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UbiMob'06, September 5-8, 2006, Paris, France.

Copyright 2006 ACM 1-59593-467-7/06/0009&#65533;\$5.00

## Categories and Subject Descriptors

C.2.4 [Computer-communication networks]: Distributed Systems-Client/server, Distributed applications.

## General Terms

Algorithms, Theory, Management, Performance, Experimentation

**Keywords:** Ubiquitous computing, Pervasive computing, Service discovery, Immune system, Mobile agents, Community of servers.

## 1. INTRODUCTION

L'évolution des télécommunications sans fil, des réseaux mobiles et leurs croisements avec les réseaux de communication fixes permettent aujourd'hui d'envisager l'ubiquité des communications, des services et des ressources [1]. En effet, les utilisateurs peuvent accéder à des services quel que soit l'endroit, le moment et ce à partir d'un terminal fixe ou mobile. Cependant, l'environnement dans lequel évoluent les utilisateurs et les services est aléatoire à cause de la mobilité de ces utilisateurs et des modifications dynamiques du nombre et de la disponibilité des ressources [2]. Par exemple, des ressources peuvent être retirées du réseau et de nouvelles ressources peuvent être introduites d'une manière aléatoire. Il est dès lors impossible aux utilisateurs d'avoir une connaissance globale et a priori de tous les services disponibles dans le réseau. La mise en oeuvre de nouvelles applications dans le cadre de l'ubiquité numérique nécessite alors le développement de nouvelles approches possédant des capacités d'auto-organisation, d'auto-adaptation et d'émergence [2,3]. Le paradigme traditionnel client/serveur (C/S) devient inefficace et inapproprié dans un environnement dynamique à large échelle dont l'évolution est aléatoire et imprévisible [2].

Dans ce cadre, deux nouveaux paradigmes alternatifs au paradigme traditionnel client/serveur ont été proposés dans [2,3] : le premier paradigme est inverse au paradigme traditionnel client/serveur; c'est le service qui vient vers le client et non le client qui prend l'initiative et demande une ressource ou un service en connaissant a priori son existence et sa localisation. Le deuxième paradigme met en oeuvre le principe de l'auto-organisation pour permettre l'émergence spontanée de nouveaux services adhoc dans un environnement sans aucune planification au préalable et d'une manière imprévisible. Le premier paradigme permet la mise en oeuvre de l'ubiquité numérique (Ubiquitous

computing) dans un réseau dynamique à grande échelle issus du croisement des réseaux filaires, non filaires et mobiles. Le deuxième paradigme est plus adapté à l'informatique ambiante ou diffuse (Pervasive computing) pour la mise en œuvre de l'auto-organisation et l'émergence dans le contexte des réseaux mobiles ad hoc et des réseaux de capteurs sans fil [2,3].

L'objectif de l'approche présentée dans cet article est de mettre en œuvre le premier paradigme alternatif, inverse au paradigme traditionnel client/serveur, qui permet la mise en œuvre de l'ubiquité numérique. Plus précisément, c'est le service qui vient vers le client et non le client qui doit connaître à priori son existence et sa localisation.

Ce paradigme alternatif peut être mise en œuvre par l'intermédiaire d'un middleware adaptatif à l'évolution dynamique du réseau, des ressources et des utilisateurs en s'inspirant des concepts du système immunitaire de l'Homme [2,3]. L'analogie avec le système immunitaire est effectuée de la manière suivante [2]. La requête d'un utilisateur est considérée comme un virus qui attaque le réseau. Le middleware réagit en développant une réponse immunitaire pour éliminer le virus agresseur c'est-à-dire satisfaire la requête de l'utilisateur. En d'autres termes, la ressource ou le service recherché correspond à la réponse immunitaire. Ceci représente une approche alternative, inverse au paradigme traditionnel client/serveur. Le système immunitaire possède un ensemble de principes fonctionnels, tels que l'auto-régulation, l'auto-organisation, la coopération et la mémoire adaptative [2]. Ces caractéristiques permettront au middleware d'élaborer des solutions décentralisées et adaptatives dans un environnement dynamique et incertain.

## 2. DESCRIPTION DE L'APPROCHE

Les systèmes de découverte de services proposés dans la littérature peuvent être classés en deux grandes familles [4,5]: les systèmes structurés et les systèmes non-structurés. Les systèmes structurés sont des systèmes basés sur une organisation structurelle qui leur permet de localiser les services [6,7]. Cependant, dans le contexte d'un réseau dynamique, la mise en œuvre de l'adaptation de cette structure organisationnelle à l'évolution et aux modifications du réseau nécessite des protocoles additionnels coûteux de mise à jour et de maintenance. A l'opposé, les systèmes non structurés qui ne s'appuient sur aucune organisation structurelle ont la possibilité de s'adapter à l'évolution et aux modifications du réseau, mais la localisation des services fait appel à des techniques de recherche coûteuses en terme de ressources et en terme de temps pour la résolution des requêtes des utilisateurs [4,8,9].

Dans ce qui suit, l'approche distribuée présentée est basée sur les agents mobiles et s'inspire du système immunitaire pour la découverte de services dans les réseaux à grande échelle. Elle peut être classée dans la famille des systèmes structurés, en étant une approche auto-organisationnelle et auto-adaptative [4]. Le principe de fonctionnement du système immunitaire est le principe de la sélection par le clonage. Ce dernier est mis en œuvre par l'intermédiaire de réseaux d'affinités entre des cellules du système immunitaire, les B-Cells, et par une mémoire adaptative. Dans cette approche, l'analogie est effectuée de la manière suivante. Les réseaux idiotypiques sont représentés par des communautés de serveurs. La mémoire adaptative est représentée par un mécanisme de renforcement des réponses aux

requêtes. En d'autres termes, l'approche proposée est fondée sur la construction et la suppression des communautés de serveurs en fonction des requêtes des utilisateurs et la disponibilité des services.

### 2.1 La construction des communautés

Jerne [10] avait introduit l'idée que les B-cells communiquent entre elles et créent des réseaux idiotypiques (i.e., réseaux d'affinités) même en l'absence d'antigènes. Plus précisément, les cellules B ne sont pas des cellules isolées, mais elles sont reliées par des chaînes de stimulation/suppression dans lesquelles une B-cell est considérée comme un antigène pour une autre B-cell. Par conséquent, un réseau idiotypique peut se créer de deux manières: une manière proactive en l'absence d'un antigène et d'une manière réactive par l'introduction et la stimulation de ce dernier. Par analogie avec le système immunitaire, un réseau idiotypique est représenté par une communauté de serveurs. Les ressources du réseau correspondent aux B-cells. La requête d'un utilisateur pour un service correspond à un antigène. La résolution de la requête correspond à la réponse immunitaire. En d'autres termes, une communauté de serveurs est un réseau d'affinités liant entre eux des serveurs capables de fournir, ensemble, un service. La création d'une communauté peut s'effectuer d'une manière proactive ou réactive. Les créations proactives des communautés sont initiées par des serveurs alors que les créations réactives sont effectuées lors de la résolution des requêtes émises par des utilisateurs [4].

Plus précisément, lorsqu'un service est constitué d'un ensemble de ressources, un serveur disposant d'une partie de ces ressources ne peut fournir ce service d'une manière individuelle. Mais, grâce au processus proactif qui est un processus de dissémination des ressources et de détection des communautés, les serveurs disposant des ressources nécessaires pour mettre en œuvre le service, se découvriront mutuellement et pourront fournir conjointement ce service. En d'autres termes, le processus proactif est un processus de détection de communautés qui permet aux serveurs de se découvrir mutuellement dans le but de créer des relations d'affinités pour représenter les services présents dans le réseau. De plus, si toutes les ressources nécessaires à un service existent dans le réseau, alors une ou plusieurs communautés représentant ce service seront créées entre les serveurs fournissant ces ressources.

Pour mettre en œuvre ce processus de détection, nous adoptons une technique basée sur des agents mobiles qui représentent les anticorps du système immunitaire. Ces derniers exécutent des marches aléatoires pour la découverte des services présents dans le réseau. Rappelons qu'une marche aléatoire dans un graphe  $G(n,m)$ , où  $n$  est le nombre de sommet et  $m$  est le nombre de liens, est un processus stochastique dans lequel un mobile se déplace aléatoirement entre les nœuds du graphe. Broder and Karlin dans [11] ont montré qu'un mobile effectuant une marche aléatoire dans  $G$  visite tous les sommets de  $G$  au moins une fois quand la longueur moyenne de sa marche est en  $O(n^2 \log(n))$ . Ce temps de couverture peut être amélioré en permettant aux agents mobiles de se cloner dans le réseau [4]. Un algorithme de régulation de la taille de la population d'agents est nécessaire pour éviter la surcharge du réseau. Nous utilisons l'algorithme de régulation distribué, également inspiré du système immunitaire de l'Homme, qui a été présenté dans [4,12,14].

Plus formellement, une communauté est un réseau d'affinités qui peut être représenté par un graphe directe pondéré  $G(S,L)$ , où  $S$  est l'ensemble des serveurs,  $L$  est l'ensemble des liens logiques qui représentent les relations d'affinités entre ces serveurs. Un lien entre un serveur  $i$  et un serveur  $j$  est caractérisé par un poids  $m_{ij}$  qui représente l'affinité entre ces deux serveurs au sein de la communauté représentant le service  $s$ . Notons que chaque serveur peut fournir une ou plusieurs ressources pouvant permettre la création de services. Par conséquent, un serveur peut appartenir à une ou plusieurs communautés simultanément.

## 2.2 La suppression des communautés

Afin de défaire les communautés de serveurs inutilisées et éliminer les agents mobiles disséminateurs (i.e., les anticorps) correspondants du réseau, nous adoptons le même mécanisme utilisé dans le système immunitaire. Plus précisément, dans ce dernier, quand un antigène attaque le corps humain, les B-cells activées par cet antigène se clonent et produisent des anticorps jusqu'à l'élimination de l'antigène, c'est la réponse dite "primaire". Ainsi, la présence de ces B-cells est renforcée par le mécanisme du clonage et lorsque le même antigène attaque une nouvelle fois le corps, le système immunitaire réagit plus rapidement à cette nouvelle exposition, c'est la réponse dite "secondaire". En revanche, les B-cells, dont la présence n'est pas renforcée par des antigènes, finissent par être éliminées [2,4], c'est l'apoptose. En d'autres termes, les B-cells non renforcées par des expositions au même antigène sont éliminées progressivement et les réseaux idiotypiques correspondant disparaissent [2,4].

De la même manière, dans cette approche, lorsqu'une communauté existe dans le réseau mais qu'elle n'est plus ou peu utilisée, elle doit être défaite (i.e., les liens d'affinités doivent être supprimés). De plus, les agents mobiles qui créent cette communauté doivent également être éliminés du réseau. Dans le cas contraire, ces agents continueront à parcourir le réseau et à construire les communautés pour représenter leur service. Pour renforcer ou affaiblir les liens d'affinités un mécanisme d'apprentissage par renforcement est proposé dans [4,15]. Ce mécanisme est appliqué pendant la résolution des requêtes. Plus précisément, un agent mobile, correspondant à un antigène ou virus et représentant une requête, une fois créé, initie une marche aléatoire dans le réseau jusqu'à atteindre un serveur pouvant fournir une ou plusieurs ressources recherchées pour composer le service. Dans le cas où le serveur rencontré dispose d'une ou plusieurs ressources pouvant satisfaire la requête mais ne possède pas de liens d'affinités pour ce service (i.e., il n'est pas le point d'entrée d'une communauté représentant tout le service recherché), alors la création réactive est déclenchée. Pour ce faire, un agent mobile anticorps est créé pour la création des communautés correspondant au service recherché. En revanche, si le serveur rencontré est le point d'entrée d'une communauté correspondante au service recherché, l'agent effectue un parcours glouton au sein de cette communauté en fonction des valeurs des liens d'affinités. Il emprunte un lien lorsque sa valeur d'affinité est la plus élevée. De plus, lors de son parcours, il renforce la valeur de l'affinité des liens sélectionnés et diminue celles des liens non sélectionnés. Nous désignons ce processus de renforcement par le terme de renforcement local des liens. Plus précisément, l'affinité  $m_{ij}^{(s)}$  entre une ressource du serveur  $i$  et une

ressource du serveur  $j$  par rapport à un service  $s$ , est ajusté à une étape  $k+1$  de la manière suivante :

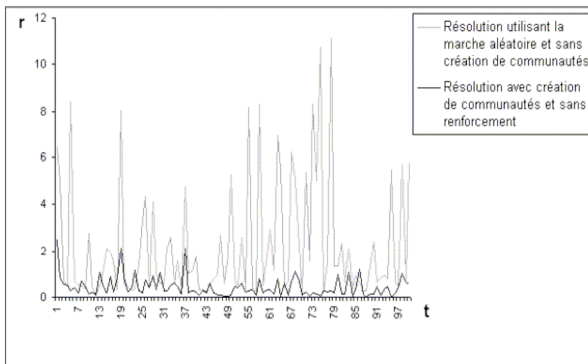
$$m_{ij}^{(s)}(k+1) = m_{ij}^{(s)}(k) + \Delta m_{ij}^{(s)}(k) = m_{ij}^{(s)}(k) + \tau(u_{ij}^{(s)}(r) - f(m_{ij}^{(s)}(k))) \quad (1)$$

$f$  étant la fonction logistique et  $\tau$  est une constante de proportionnalité comprise entre 0 et 1.  $u_{ij}^{(s)}$  représente la satisfaction locale par la présence des ressources  $i$  et  $j$  recherchées par la requête  $s$ . Cette valeur vaut 1 pour une satisfaction positive et 0 pour une satisfaction négative. Lorsque l'agent termine son parcours au sein de la communauté et constitue la liste des serveurs fournissant le service, il procède au renforcement du chemin sélectionné  $\varphi$ . Plus précisément, il emprunte le chemin à l'envers jusqu'au point d'entrée en renforçant une deuxième fois les liens choisis. Nous désignons ce deuxième processus de renforcement par le terme de renforcement global [4,15]. La variation des valeurs des liens d'affinité est calculée en utilisant l'équation (1), où  $u_{ij}^{(s)}$  est substitué par  $u_{\varphi}^{(s)}$  qui représente la satisfaction globale par rapport à la résolution de la requête  $s$ . Cette valeur vaut 1 pour une satisfaction positive 0 pour une satisfaction négative.

En utilisant ce mécanisme de renforcement, les valeurs des liens d'affinités d'une communauté de serveurs diminuent, au fur et à mesure que des agents requêtes ne demandant pas le service représenté parcourent les serveurs de cette communauté. En effet, une communauté de serveurs qui n'est pas invoquée par les requêtes des utilisateurs doit être défaite lorsque les valeurs de ses liens d'affinités correspondants deviennent négatives. Il est important de noter que, grâce à ce mécanisme d'apprentissage, lorsqu'un service est souvent invoqué par les requêtes, la construction réactive de nouvelles communautés de serveurs et la présence de celles qui existent déjà sont renforcées. Par conséquent, la rencontre entre une requête recherchant un service souvent sollicité, et des communautés qui le représente s'effectuera plus rapidement. Il est important de noter également que ce renforcement des liens d'affinités constitue la mémoire auto-organisationnelle de cette approche auto-adaptative. Ceci correspond aussi, par analogie, à la mémoire adaptative du système immunitaire qui permet de développer des réponses immunitaires secondaires plus rapides aux antigènes déjà rencontrés auparavant.

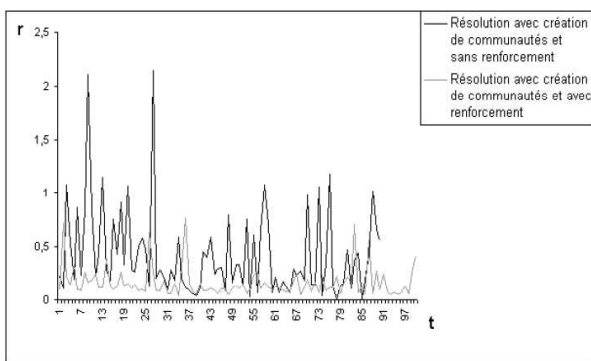
## 3. SIMULATION

Les simulations présentées ci-dessous ont été développées par Network Simulator (NS2) [4,13]. Pour ce faire, un réseau de 100 serveurs, dont la topologie est aléatoire, est généré. Chaque serveur dispose d'une ressource. Il y a 10 types de ressources distribuées aléatoirement dans les serveurs. Le résultat de la première simulation présenté dans la figure 1 montre que la création des communautés, avec une résolution des requêtes basée sur un parcours aléatoire au sein de ces communautés, améliore le temps nécessaire à la résolution des requêtes en comparaison avec une stratégie de résolution basée uniquement sur la marche aléatoire dans le réseau et sans création des communautés.



**Figure 1. Comparaison en terme de temps de résolution  $r$ , avec ou sans création des communautés de serveurs.**

Le résultat de la simulation présenté dans la figure 2 montre que l'utilisation du mécanisme de renforcement améliore le temps nécessaire à la résolution des requêtes en comparaison avec une stratégie de résolution basée uniquement sur la marche aléatoire au sein des communautés créées.



**Figure 2. Comparaison en terme de temps de résolution  $r$ , avec et sans renforcement des liens d'affinités entre les serveurs.**

#### 4. CONCLUSION

Dans cet article, une approche auto-organisationnelle et auto-adaptative pour la découverte de services dans les réseaux à grande échelle en s'inspirant du système immunitaire de l'Homme est présentée. L'auto-organisation est mise en oeuvre par la création de réseaux d'affinités. Ces derniers s'auto-organisent et s'adaptent dynamiquement en fonction du type et de la fréquence des requêtes des utilisateurs via un mécanisme d'apprentissage et de renforcement des liens d'affinités entre les serveurs. Actuellement, nous étudions la nature des graphes émergents liant les serveurs entre eux (i.e., les communautés de serveurs). Cette approche peut être également utilisée pour la mise en oeuvre du deuxième paradigme pour l'émergence spontanée de nouveaux services dans le cadre des réseaux ad hoc et des réseaux de capteurs [2,3].

#### 5. RÉFÉRENCES

[1] Weiser, M. Hot topics: Ubiquitous computing. *IEEE Computer*, October 1993.

[2] Gaber, J. *New paradigms for ubiquitous and pervasive computing*, Research Report RR-09-00, Université de Technologies de Belfort Montbéliard (UTBM), pp. 1-6, Septembre 2000.

[3] Gaber, J. *New paradigms for ubiquitous and pervasive applications*, *Proc. of the First Workshop on Software Engineering Challenges for Ubiquitous Computing*, Lancaster, UK, June 2006.

[4] Bakhouya, M. *Approche auto-adaptative à base d'agents mobiles et inspirée du système immunitaire de l'Homme pour la découverte de services dans les réseaux à grande échelle*, *Thèse de doctorat*, Université de Technologie de Belfort-Montbéliard, 2005.

[5] Bakhouya, M., and Gaber, J. *Approches de mise en oeuvre de l'ubiquité numérique (chapitre 6), Réseaux mobiles ad hoc et réseaux de capteurs sans fil (Traité IC2, série Réseaux et télécommunications)*, Ed. H. Labiod, Hermes, 02/2006, ISBN 2-7462-1292-7, pp.129-163.

[6] Robert, M. *Discovery and Its Discontents : Discovery Protocols for Ubiquitous Computing*, UIUCDCS-R-99-2132, March 25 2000.

[7] Czerwinski, S., Zhao, B., Hodes, T., Joseph, A. and Katz, R. *An architecture for a secure service discovery service. Proceeding of ACM MobiCom '99*, Sep. 1999.

[8] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. *Search and replication in unstructured peer-to-peer networks*, in *Proceedings of 16th ACM International Conference on Supercomputing (ICS'02)*, New York, USA, June 2002.

[9] Wang, C., and Li, B. *Peer-to-peer overlay networks: A survey*. <http://comp.uark.edu/~cgwang/Papers/TR-P2P.pdf>, 2003.

[10] Jerne, N.K. *Towards a network theory of the immune system. Ann. Immunol. (Inst. Pasteur) 125C. 373*, 1974.

[11] Broder, A., and Karlin, A. *Bounds on the Cover Time*, *Journal of Theoretical Probability*, 2(1):101-120, January 1989.

[12] Bakhouya, M., and Gaber, J. *Distributed auto-regulation approach of a mobile agent population in a network*. Research Report RR-12-02, Université de Technologies de Belfort Montbéliard (UTBM), pp. 1-14, December 2002.

[13] *NS-2 patch*, available at the Faculty of Information Technology, Mathematics and Electrical Engineering, Department of Telematics web site: <http://www.item.ntnu.no/~wittner/ns/index.html>

[14] Bakhouya, M., and Gaber, J. *Adaptive approach for the regulation of a mobile agent population in a distributed network*, In *5th International Symposium on Parallel and Distributed Computing (ISPD'06)*. IEEE Press. Timisoara, Romania.

[15] Bakhouya, M., and Gaber, J. *A reinforcement learning of link affinities and user requests for Self-adaptive graph emergence from an arbitrary graph*, Research Report RR-07-02. Université de Technologies de Belfort Montbéliard (UTBM), pp. 1-12, 2002.