Localized Probabilistic Algorithm for Efficient Information Dissemination in Ad Hoc Networks

Michaël Hauspie, Amandine Panier, David Simplot-Ryl

To cite this version:

Michaël Hauspie, Amandine Panier, David Simplot-Ryl. Localized Probabilistic Algorithm for Efficient Information Dissemination in Ad Hoc Networks. [Research Report] RT-0294, INRIA. 2004, pp.9. inria-00069886

HAL Id: inria-00069886
https://hal.inria.fr/inria-00069886

Submitted on 19 May 2006

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Localized Probabilistic Algorithm for Efficient Information Dissemination in Ad Hoc Networks

Michaël Hauspie — Amandine Panier — David Simplot-Ryl

N° 0294

Avril 2004

THÈME 1

Rapport technique

ISSN 0249-0803

Localized Probabilistic Algorithm for Efficient Information Dissemination in Ad Hoc Networks

Michaël Hauspie * , Amandine Panier * , David Simplot-Ryl*

Thème 1 — Réseaux et systèmes
Projet POPS

Rapport technique n° 0294 — Avril 2004 — 9 pages

Abstract: Ad hoc networks are autonomous dynamic networks composed of mobile devices like personal digital assistants (PDA) for instance. In such mobile networks, lack of infrastructure leads to non trivial information discovery and dissemination. A scheme in which a unique object centralizes information is not efficient for many reasons. Indeed, it is difficult to maintain a centralized structure because of network topology variation.

In this report, we propose a probabilistic algorithm to satisfactorily distribute an information token among nodes forming the network, that is to say only a given number of nodes will memorize. We do not consider a particular type of information and we discuss on how efficiently realize the dissemination. This kind of dissemination would find application in service discovery where mobiles may need to access services or in routing for example.

Key-words: Ad Hoc Networks, Information Dissemination, Service Discovery, Geographical Routing

---

* IRCICA/LIFL, Univ. Lille1, INRIA Futurs, France. Email: {hauspie, panier, simplot}@lifl.fr

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 —Télécopie : +33 1 72 92 59 ??

Algorithme probabiliste localisé pour une dissémination d'information efficace dans les réseaux ad hocs

Résumé : Les réseaux ad hoc sont des réseaux dynamiques, autonomes, constitués de d'objets tels que des PDA. Dans un tel environnement, le manque d'infrastructure rend non trivial la publication et la recherche d'informations. Une solution utilisant un objet unique centralisant l'information n'est pas efficaces pour de nombreuses raisons. En effet, la maintenance de la cohérence d'une telle structure est difficile à maintenir à cause de la dynamicité du réseau.

Dans ce rapport, nous proposons un algorithme probabiliste qui permet de distribuer efficacement une information dans le réseau, c'est à dire que peu de nœuds vont mémoriser cette dernière tout en assurant une optimisation du coût de la recherche de cette information. Ce type de dissémination d'information peut trouver son application dans des domaines aussi variés que la découverte de services ou le routage par exemple.

Mots-clés : Réseaux ad hoc, dissémination d'information, découverte de service, routage géographique

1 Introduction

Ad hoc networks are autonomous dynamic networks composed of mobile devices like personal digital assistants (PDA) for instance. In such mobile networks, lack of infrastructure leads to non trivial information discovery and dissemination. A scheme in which a unique object centralizes information is not efficient for many reasons. For instance, let us consider a location server which provides geographic location of nodes. Such server can be used in several layers: in geographical routing [7] or context-aware applications.

Indeed, it is difficult to maintain a centralized structure because of network topology variation. Firstly, if we do not suppose any fixed access point, maintaining the server location is expensive. Secondly, even if we have a fixed access point, nodes which are far from the server are penalized since request time increases with distance. Thirdly, this kind of architecture raises an issue: the server may become the seat of a bottleneck. As a matter of fact, all requests are sent to the single server. This issue is known in wire infrastructures. This is why research focuses on proposing distributed versions of known centralized algorithms in the wired world. Finally, and it is again a well known issue in wired infrastructures, if the server falls down, the entire network is paralyzed. The ideal dissemination would be each node knows all available information in the network but considering that mobiles are limited in memory space and energy, it is not a realistic solution.

In this paper, we propose a probabilistic algorithm to satisfactorily distribute an information token among nodes forming the network, that is to say only a given number of nodes will memorize. We do not consider a particular type of information and we discuss on how efficiently realize the dissemination. This kind of dissemination would find application in service discovery where mobiles may need to access services or in routing for example.

The paper is organized as follows. In the next section we will review some architectural supports which have been set up in this two particular fields of application. In Section 3, we will expose our proposal. Then, in Section 4, we will analyze the results of the experiments. Finally, in Section 5, we will sum up our proposal and give details for future work.

2 Literature Review

Information dissemination is something important for service discovery. An efficient information dissemination can allow to reduce message overhead and request time.

In the literature, proposed service discovery architectures are generally centralized although it is not a suitable solution for ad hoc networks. In [4], the authors try to show that it is not inevitably a bad choice. They deal with Directory Agents hosted by nodes forming a dominating set [3]. This DAs allow servers to record services they provide. The disadvantage of this solution is that the memorizing nodes are always the same. The nodes forming the "virtual backbone" are penalized.

A performance evaluation of different *Post-query strategies* is proposed in [6]. A *Post-query strategy* executes *Post-query protocols* (servers publish services they provide according to the *Posting protocol* and clients request for services according to the *Querying protocol*) in rounds. Five *Post-query strategies* are confronted : the *greedy strategy*, the *incremental strategy*, the *uniform memoryless strategy*, the *with memory strategy* and the *conservative strategy*. The *greedy strategy* consists in making all nodes publish to or query all nodes in the network; the *incremental strategy* consists in making the set of nodes to which other nodes publish and the set of nodes other nodes query bigger each round; the *uniform memoryless strategy* consists in making all nodes publish to or query a random set of nodes; the *with memory strategy* consists in making all nodes publish to or query each round nodes not yet contacted; the *conservative strategy* consists in making all nodes publish to or query its one-hop neighbors. The *greedy strategy* is obviously unsuitable since nodes are limited in memory space. The *uniform memoryless strategy* may be suitable if the set of nodes is small and the way memorizing nodes are distributed is interesting, but the dissemination of information does not follow any reproducible property. The *with memory strategy* tends to make all nodes knowing about all available information in the network. The *conservative strategy*, implies a non satisfactory dissemination of information since nodes wishing to get the information token provided by the source will reduce the cost of the request by one hop in the best case.

In the same way as for service discovery, information dissemination is important for geographical routing since mobiles need to know the location of the destination they want to reach. To access such an information token, the sender node can broadcast a request for location and wait for a reply for example. Each node knows its location thanks to a positioning system such as GPS [1].

Grid Location Service [5] is a distributed service location. Each node updates its location to location servers and is able to be a location server as well. As we move away from a node, its location servers are scattered more and more; this is possible since the network is divided into bigger and bigger squares. A level-0 square corresponds to the unit square. A level-n square is composed of four level-$(n-1)$ squares. Each node has to choose one node as location server in each of the three squares taking part to the formation of a higher level-square with it. Figure 1 shows in which squares node B will have to choose its locations servers. Since the number of location servers for a node decreases as we move away from it, the total number of a node location servers tends to be minimized. A node updates its location to its location servers when it has moved a given distance d away from its last updated location. The main disadvantage of this proposal is that it implies inevitably the use of GPS; this is not suitable for information dissemination in the general case. Moreover, it may happen that a node goes far away to find a data instead of finding it near to it, increasing the overhead and the query delay; this is the case for border nodes within a square since the distance between a border node whishing to reach another border node standing next to him in a different square and a location server of this node, is ineluctably longer than the distance between these two nodes.

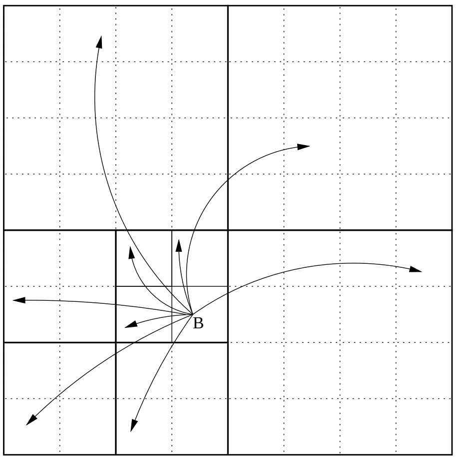


Figure 1: Example of GRID distribution for node B location.

In Tribe [8], a node updates its location only to one another node which is discovered on demand: its anchor. Each node in the network has a universal identifier, a virtual identifier, a relative address and a control area. A node calculates its virtual identifier in the addressing space from its universal identifier. Its anchor is then the node whom control area contains its virtual address. The anchor knows the relative address of the node and its current location. When a node needs to communicate with another node, it first has to request for the current location of the destination. To do so, the node has to calculate, in the same way as the destination node, the virtual address to find the node knowing the desired location. This is possible since the universal addresses and the conversion function in respective virtual addresses are supposed to be known by each node forming the network. When a node has to forward a packet, it chooses in its neighborhood the node whom control area is closed to the control area containing the identifier of the destination. Contrary to Grid, Tribe does not use geographical information to determine the nodes which will memorize the location of other nodes and to route the packets. The disadvantage is that the anchor of a node may be faraway from the node thus increasing message overhead and the request time.

All reviewed proposals do not take care about available memory space of objects acting as servers of information. Moreover, the designed architectures may be difficult to maintain due to the mobility of the network. We propose a localized probabilistic algorithm to efficiently disseminate information across the network.

3 Probabilistic dissemination

In this section, we will focus on describing the property hidden behind the idea of an *efficient* information dissemination and how we achieve to verify this property using only local data.

3.1 Efficient information dissemination

We believe that an efficient dissemination should balance the number of nodes needed to cache information and the cost induced by requesting it. We also believe that a node looking for an information token should find it or a reference to it on a node closer to the source than itself. On one hand, the closer is the node that caches the token from the node that makes the request, the lower will be the cost of the research. On the other hand, it grows up the number of nodes that are needed to disseminate the token and thus, the cache size needed if more than one token is disseminated.

We can formalize this idea by the following definition. We consider a network modeled as a unit disk graph [2] $G(V, E)$ where V is the set of nodes and E is the set of edges. An edge exists between two nodes if they can communicate through one direct wireless link (*i.e.* they are within each other's radio radius).

Definition 1 *Let u be a node that has an information token I_u to disseminate across the network. Let v be a node and let $Cache(v)$ be the set of tokens that are stored in the cache of node v. I_u is said well disseminated if the following property is verified :*

$$\forall v \in V, m < d(u,v) \leq M, \begin{cases} I_u \in Cache(v) \\ \text{or} \\ \exists w \in V \text{ s.t.} \\ I_u \in Cache(w), \\ d(v,w) < \lambda d(v,u), \\ d(u,w) < d(u,v), \end{cases}$$

where m and M are the minimum and the maximum distance between which we consider the criterion, $d(x, y)$ is the distance between node x and node y in hops and λ is a distance factor ($\lambda \in]0, 1[$).

The parameter λ in the above definition is the factor *tweaking* the balance between memory size and research cost. The lower it is, the lower is the cost of research but the higher is the number of nodes needed to cache the token. For instance, a fair balance would be achieved with $\lambda = \frac{1}{2}$.

3.2 Perfect case

Considering a perfect case, where we can place nodes wherever we want, we can evaluate where to place memorizing nodes so as to verify the property using as few nodes as possible. The main idea of our algorithm will then be to deduce a probabilistic law for selecting the nodes that will cache the token in order to average the perfect locations as accurately as possible. So, the first thing to do is to find the perfect locations of the nodes that will cache the token. In that perfect but theorical case, we assimilate hop-distance to euclidian distance.

As we consider the criterion starting at distance m, the first nodes that need to cache the token will be located at a distance m to the source. Those nodes are at the first rank, called r_0 in the following. Each of those nodes will *cover* a zone located at a distance λm around them. A zone is said covered if all nodes located inside it satisfy the property given in Definition 1. Thus, we need to distribute a number a_0 of nodes on the circle of center the source node and radius m. From here, we can give the rule for constructing rank r_{n+1} from rank r_n (which will increase the covered area). The construction of rank r_1 from rank r_0 is depicted in Figure 2. Rank r_{n+1} is constructed from rank r_n by placing a node on the intersection of each couple of circles of centers each successive pair of nodes of rank r_n and radius λm_n where m_n is the distance between the nodes of rank r_n and the source. A full view for ranks r_0 and r_1 is given Figure 3.a and 3.b.

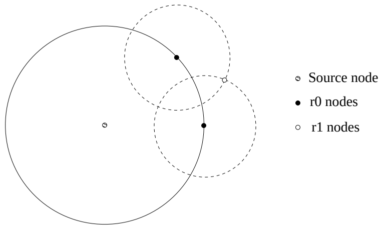


Figure 2: Constructing r_1 from r_0.

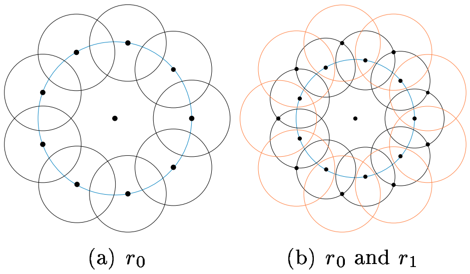

(a) r_0 (b) r_0 and r_1

Figure 3: Area covered by ranks r_0 and r_1.

This construction algorithm introduces a condition on the minimal distance between nodes of a same rank. Nodes of rank r_n must be at most at distance $2\lambda m_n$ to each other to ensure that the circles intersect. We can then define a condition on the minimum number of nodes to place on rank r_0:

$$a_0 > a_{min} = \frac{2\pi}{arccos\,(1 - 2\lambda^2)}. \tag{1}$$

To satisfy the condition, we consider using $a_0 = a_{min} + k, k \in \mathbb{N}^*$. We need to find the best value of k so that the whole desired area is covered by a minimum number of nodes. This can be easily done by computing it for some values of k. Figure 4 shows the number of nodes needed to cover an area going from 2 hops to the source to 20 hops with $\lambda = \frac{1}{2}$. In that case, we can see that the good values of k are 3 or 4.

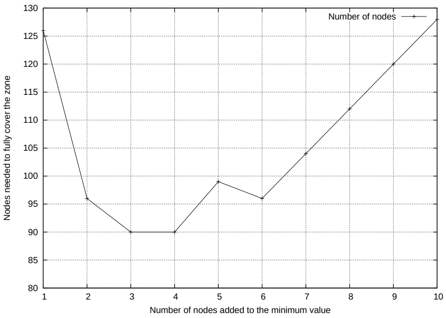


Figure 4: Number of nodes to add to a_{min} to cover all the zone.

In our construction, each rank has the same number of nodes, that is $a_0 = a_1 = \cdots = a_n$. Thus, we are able to know for each rank, how many nodes should cache the token so that the full area is covered (*i.e.* each node in the area satisfies the property given in Definition 1). The last thing needed is to know, for each rank, its distance to the source. Again from simple geometrical properties, we can deduce m_n (the distance of rank r_n to the source):

$$m_n = m_0.\tau^n, \tag{2}$$

where τ is given by

$$\begin{aligned} \tau \quad = \quad & \sqrt{1 - \frac{1}{2}\left(1 - cos\frac{2\pi}{a_0}\right)} + \\ & \sqrt{\lambda^2 - \frac{1}{2}\left(1 - cos\frac{2\pi}{a_0}\right)}. \end{aligned}$$

Given the equations (1) and (2), we know where to ideally place nodes that will cache the token so as to verify the property given in definition 1. It is then easy to define a probabilistic rule $P(d)$ so that a node u located at a distance d to the source S can decide by itself either it should cache the token or not :

$$P(d) = \begin{cases} \frac{a_0}{NC_d} & i \text{ s.t. } m_i = d, \\ \frac{a_0}{NC_d + NC_{d+1}} & \text{s.t. } m_i \notin \mathbb{N}, \\ & \text{and } \lfloor m_i \rfloor = d(S,u), \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where NC_d is the number of nodes located at distance d to the source. NC_d can be approximated simply by using the surface of the ring of outer radius Rm_i and inner radius $R\,(d-1)$. This surface is given by

$$S = \pi R^2 d^2 - \pi R^2 \left(d-1\right)^2 = \pi R^2 \left(2d-1\right)$$

where R is the transmission radius. Thus, NC_d can be approximated by the following formula

$$NC_d = D\left(2d-1\right) \tag{4}$$

where D is the number of nodes per communication zone (*i.e.* the density of the network). The accuracy of this formula is shown Figure 6.

Thus, the publication algorithm to run is given Figure 5.

```
foreach node $u \in G(V,E)$
  $p := rand()$;
  if $p < P(u)$
    Cache the token in the node $u$
  fi
end
```

Figure 5: Information publication algorithm.

3.3 Localized protocol

As the global density can be estimated by the local density, each node has the required knowledge to execute the algorithm given Figure 5 in a localized manner. The publish is done using a broadcast protocol. Each node receiving the publication packet generates a random number and checks against its probability. The distance to the source is given by the publication packet.

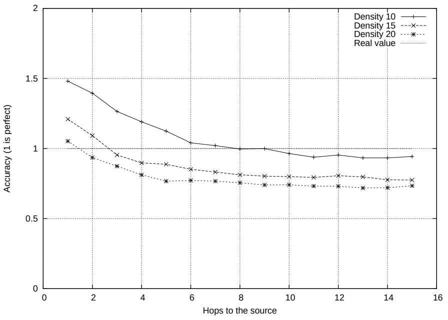
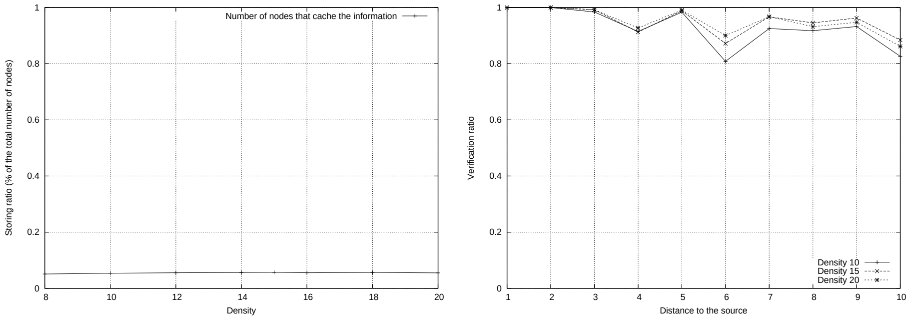


Figure 6: NC_i evaluation accuracy.

The main advantage of the algorithm is that publication is done using only one broadcast. If a node needs to look for a token, it broadcasts a request in a small area around him. A node that receives this request and that has cached the token can reply by providing the information token or sending the address of the token publisher. If no node receives the request, another broadcast is done with a bigger area. The process is iterated until a node that knows something about the requested token replies. As information is well disseminated, the request process will not iterate much and the cost of research will remain low.

4 Experiments

In order to evaluate our protocol, we run experiments on a 500 nodes unit graph. Nodes are distributed uniformly in a square area. The communication range is 10 meters. A node is selected at the center of the area to be the source of the information token. It broadcasts a publication packet that includes its identifier and some data about the token he wishes to publish. This data depends on the type of information that is published. For instance, in the case of service publication, it can include the service description which will be stored with the id of the publisher. Nodes forwarding the broadcast packet run the algorithm given Figure 5 to decide either they should store the token or not. First, we present the results obtained with $\lambda = \frac{1}{2}$ as it is representative of a fair balance between cache size and research cost. Then, we will show how we can modify the cache needs through the change of the value of λ.



(a) Number of nodes caching the information token (b) Number of nodes that verify the property after publication

Figure 7: Efficiency of the publication algorithm.

Figures 7.a and 7.b, show the proportion of nodes that are storing the token and how the property given in Definition 1 is verified across the network. We can see that we achieve to strongly verify the property around the source using only a small amount of nodes. With this results, we can expect to be able to manage much more sources and verify the property for all these sources without leading to huge cache size needs.

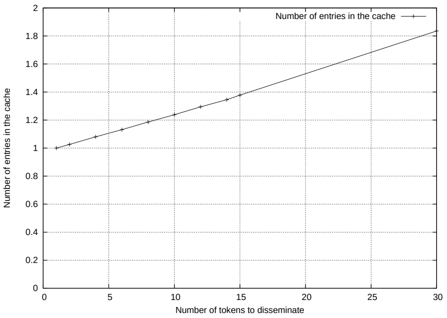


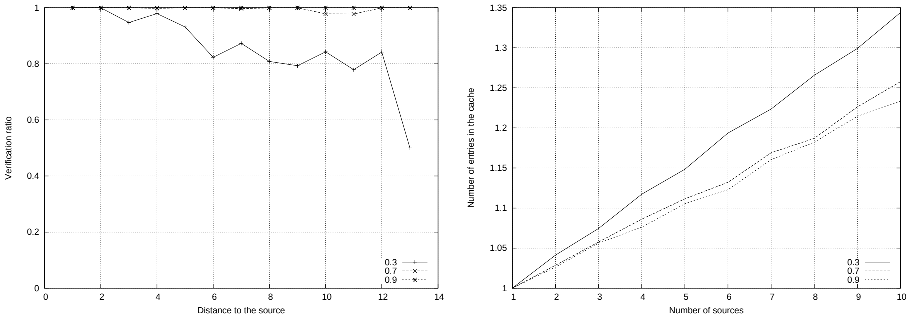
Figure 8: Average number of entries in the cache of nodes that are storing.

Figure 8 shows that our expectation is verified as the number of entries in the cache of the nodes remains very low while increasing the number of nodes.

The influence of λ for the publication is shown Figure 9. The values are given for density 15 and $\lambda = 0.3, 0.7$ and 0.9. We can see that, as we expected, modifying λ can change the average cache size. Moreover, the property is still verified in each case.

5 Conclusion

In this paper, we proposed a probabilistic method for performing an efficient information dissemination in a mobile ad hoc network. We first propose a criterion of efficient information dissemination which depends on the balance of request and memory cost we want to address. We shown through experimental results that our



(a) Influence on property verification (b) Influence on cache size

Figure 9: Influence of λ.

proposal fits our objective which was to verify our criterion using little memory and a simple algorithm that can run on very small nodes.

Our experiments shown that our algorithm should be suitable for a large population which disseminates a large amount of information across the network. For this purpose, we must work on an efficient cache policy to use when no more memory is available on small objects.

Our next work is to evaluate the cost of the request to verify that our protocol actually optimize it. Meanwhile, we are working on enhancing this protocol by solving a problem induced by this fully probabilistic algorithm. As the nodes take the decision by themselves without being aware of their neighborhood, more than one node may store the information token where only one would be enough. We believe that the use of dominating sets or cluster head sets should lower the number of nodes storing and thus the memory use induced by the dissemination.

References

[1] *The Global Positioning System FAQ*, July 1997.
URL: http://www.gpsy.com/gpsinfo/gps-faq.txt.

[2] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[3] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. In *Proc. of IEEE International Conf. on Communications (ICC'03)*, May 2003.

[4] U. C. Kozat and L. Tassiulas. Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues. *Adhoc Networks*, 2:23–44, 2004.

[5] J. Li. A scalable location service for geographic ad hoc routing. Master's thesis, Massachusetts Institute of Technology, January 2001.

[6] H. Luo and M. Barbeau. Performance evaluation of service discovery strategies in ad hoc networks. In *2nd Annual Conference on Communication Networks and Services Research (CNSR 2004)*, Canada, may 2004.

[7] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers. *ACM SIGCOMM '94 Computer Communications Review*, 24(4):234–244, October 1994.

[8] A. C. Viana, M. D. de Amorim, S. Fdida, and J. F. de Rezende. Indirect routing using distributed location information. In *Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, Dallas-Fort Worth, Texas, March 2003.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803