

# An Adaptive Approach to Service Discovery in Ad Hoc Networks

Carlos Henrique Pereira Augusto and José Ferreira de Rezende

Grupo de Teleinformática e Automação - PEE - COPPE  
Universidade Federal do Rio de Janeiro (UFRJ)  
{chenrique,rezende}@gta.ufrj.br

**Abstract.** Service discovery allows the interaction between network nodes to cooperate in activities or to share resources in client-server, multi-layer, as well as in peer-to-peer architectures. Ad hoc networks pose a great challenge in the design of efficient mechanisms for service discovery. The lack of infrastructure along with node mobility makes it difficult to build robust, scalable and secure mechanisms for ad hoc networks. This paper proposes a scalable service discovery architecture based on directory nodes organized in an overlay network. In the proposed architecture, directory nodes are dynamically created with the aim of uniformly covering the entire network while decreasing the query latency for a service (QoS) and the number of control messages for the sake of increased scalability.

## 1 Introduction

Ad hoc networks are characterized by the lack of infrastructure and spontaneous topologies. This complicates the use of previous configured and specific servers to perform service discovery, as occurs when a DNS server is used. In addition, the use of directory servers, such as LDAP or X.500, is not a simple task in an infrastructureless environment. Some solutions are available for service discovery in wired networks, such as SLP[1], Jini[2] and Upnp[3]. However, they assume the existence of special nodes acting as servers.

Nonetheless, there are proposals for wired networks and Internet that do not rely on specific servers, and are already used in Peer-to-Peer (P2P) environments. Researches are being performed in these environments, which present great challenges in scalability and dynamic behavior. Some proposals are Pastry[4], Tapestry[5], CAN[6], Chord[7] and Symphony[8], and there are some successful applications, such as Gnutella[9] and JXTA[10]. However, in P2P, scalability indicates from thousands to millions of nodes, and dynamic behavior refers to the ingress and egress of nodes in the network in a timescale from minutes to hours of sojourn without changing of location.

In contrast, ad hoc networks, due to their lack of infrastructure and scarcity of resources, present scaling challenges to attain a few hundreds of nodes collaborating in an efficient way. Further, dynamic behavior is replaced by mobility that leads to link breakages and even network partition.

This paper proposes a mechanism to service discovery in ad hoc networks, using concepts of P2P networks. However, it is not made a direct use of a P2P solution in ad hoc network due to their distinct scope. The proposed architecture makes use of three abstractions planes: the real ad hoc network; the overlay network, composed of directory nodes only; and the service table ring.

This paper presents the general design concepts involved in the proposal, and details and evaluates the first abstraction plane and the mechanism of construction of second plane. The reminder of the paper is organized as follows. Section 2 presents related works, firstly in P2P and after in ad hoc networks. Section 3 makes a theoretical description of the proposal. In Section 4, an analytical model for the second plane construction is presented. Section 5 discusses implementation details of first and second planes, and present some simulation and analytical results. Finally, the Section 6 draws conclusions and discusses the future of this work.

## 2 Related Works

This section presents some works related to the proposal presented in this paper. Firstly, a brief overview of service discovery schemes used in P2P architectures are presented with the purpose of clarifying some design choices made in this proposal. Next, we describe some service discovery schemes specially tailored to mobile ad hoc networks (MANETs).

### 2.1 In P2P

In P2P systems, such as Chord [7], Symphony [8], CAN [6], Pastry [4], and Tapestry [5], a service or resource description is hashed to a key that is used in the lookup for that particular service or resource. The ownership of the keys, i.e. the responsibility over the location of the associated services, is partitioned among participating nodes using the concept of DHT (Distributed Hash Table). In these systems, when a query is issued, it is routed through an overlay network formed by the DHT nodes to the node responsible for that key. The geometry structure of the overlay network, i.e. the DHT nodes organization, is the key difference between the existing proposals. In Chord and Symphony, nodes are organized in a virtual ring. Pastry and Tapestry maintain a tree-like data structure, and CAN uses a  $d$ -dimensional Cartesian coordinate.

Therefore, a great challenge is how to distribute keys, consistently and efficiently, among the nodes of the network. In Chord and Symphony, each node should have at least the knowledge about its successor and predecessor nodes in the ring. However, a query using only this information may cause a circular search in the entire ring. A lookup should traverse on average  $N/2$  nodes when random variables are uniformly distributed and  $N$  is the number of DHT nodes.

Chord[7] creates connections between nodes nearly located to the  $N/2$ ,  $N/4$ ,  $N/8$  positions of the ring in order to diminish the search space. These connections enable jumps in the ring, and decrease the search space to  $O(\log N)$ ,

making the query more efficient. However, when there are many ingress and egress of nodes in the network, the cost associated to the management of these connections largely increases.

Alternatively, Symphony[8] proposes the establishment of random connections, using a probability distribution function in a similar way of the small-world Kleinberg's work[11]. Since there is no need of recalculating ring fractions when nodes ingress or egress to the network, the overhead cost due to a dynamic behavior does not increase.

## 2.2 In Ad Hoc Networks

In this subsection, we describe some proposals for service discovery in ad hoc networks. These works are organized in two types, the ones that are independent of ad hoc routing protocols being used and the ones that rely on a specific routing protocol with the purpose of improving scalability.

**Independent of Routing Protocol** In [12], it is proposed Nom, where clients actively flood the network with service query messages that are handled by passive servers. This approach is normally referred as *pull method*. In opposition to this method, there is the *push method*, where servers broadcast service advertisements, and clients cache it for later invocation. It is possible to merge these two methods in a hybrid one, where both, servers and clients, participate actively, advertising or registering their services, or sending queries, respectively.

The Card proposal[13] uses *small world* concepts to establish long-range contacts in the ad hoc network. However, since it has not any location mechanism, contacts are established in a random way, contradicting the Kleinberg work[11]. Further, it requires a routing protocol that permits to know an  $H$ -hops neighborhood of a node.

[14] can be classified as orthogonal to the above approaches, since it does not rely on any routing protocol, but still can use cross-layer mechanisms to enhance performance and scalability. It models each server instance as an electrostatic charge that produces a field in the network, and it models each query message as a charge with inverse polarity that is attracted to the service by the field gradient. However, for the field information to be propagated to every node, it needs a network flooding, as in the *push method*. A proposed solution to this problem [14] is to cache information in intermediate nodes. However, this solution is only effective when multiple instances of the same service exist. Considering that a network can make available several distinct services, there will be a field value to each service type. Further, some services, such as file sharing or Web Services, require the announcement of file name descriptions, what leads to a large increase in the service per node ratio.

In [15], despite it has been classified as independent, the mechanism strongly uses typical OLSR protocol information, such as MultiPoint Relays (MPR) selection, to perform a two-hops bordercast, making it more complex to be implemented with another routing protocol. This restriction is in part overcome

by the use of an overlay network. Applying a typical concept from P2P networks, some nodes are elected as directory nodes, which are responsible for services registering. Thus, queries are driven to them, which will answer the queries, acting as super peers.

**Routing Protocol Dependent** A good option in terms of scalability for service discovery in ad hoc networks is the use of routing protocol functions. The main reason is the similarity between both tasks. In this approach, register and query messages may be piggybacked on control messages of the routing protocol or may be routed through paths expressly maintained by the routing protocol for service discovery. However, the main distinction between these tasks is that in service discovery the search is not univocal as in routing, i.e. service discovery aims to find some of the available instances of queried services. In addition, there is no need to find the best path to these instances, since this task is lately achieved by the routing protocol.

In [16], service queries are piggybacked on route discoveries launched by the AODV routing protocol. In [17], it is proposed the Konark protocol to service discovery and delivery, where services are described through XML, increasing the flexibility. However, the discovery process is dependent on a previously established multicast tree in the ad hoc network.

### 3 Proposed Architecture

In this section, it is made a brief description of the entire service discovery architecture proposed in this work. The design of this architecture is influenced by the following evaluation of work presented in [14]. The average sending rate of control traffic per node ( $NrMsg/(node \times s)$ ) has a linear increase with the number of service instances (see Figure 2(a)). Therefore:

$$NrMsg/(node \times s) = O(Nrof\ servers) \quad (1)$$

Considering that the number of service providers is a constant ratio of the nodes in the network, i.e.  $Nrof\ servers = K \times Nrof\ nodes$ , then:

$$NrMsg/(node \times s) = O(K \times Nrof\ nodes) = O(Nrof\ nodes) \quad (2)$$

This implies a scalability constraint. Further, for some service types, according to considerations in Section 2.2, one can have a high number of services per node ratio. This claim supports the use of an overlay network and suggests the use of a single service type - the directory node service - in a similar way of the proposal in [15]. An advantage of directory nodes adoption is the multiple answer possibility. If there are several service instances, the directory node can answer with all possible instances, leaving to the client the choice of the more adequate one.

In this model, a service query must arrive at a directory node, which will answer or forward this query to another node in the directory nodes overlay

network. This mechanism assumes that each client node has knowledge of at least one directory node, and that each directory node has knowledge of other directory nodes to forward messages to them.

Our proposal avoids flooding the whole network with control messages. Each directory node floods their neighborhood for  $H$  hops, so that every node within this neighborhood has knowledge about its existence. Then, each node computes the field contribution of this directory node, which is the charge announced by this directory node divided by the hop distance from it. The messages used to this purpose are called announcement messages.

In a similar way of the work in [14], each node computes a sum of all contributions of all those directory nodes that advertise to it. This sum is inserted in a hello message and sent to 1-hop neighbors. When there is a query or subscribe message needing to be forwarded, each node sends it to its neighbor that has the highest contribution value advertised. This way, this message follows the gradient to the target, such as in an anycast routing.

However, directory nodes should provide a full coverage of the network and maintain a certain degree of connectivity between them in the overlay network. Otherwise, some client nodes will not be able to send their queries. In our proposal, common nodes within uncovered areas (or with small contributions) should get promoted to directory nodes. We choose to call this procedure as *promotion process* to differ from an election process as performed in the proposal in [15]. Our promotion process does not require the exchange of extra messages to this purpose and each node makes a local decision. Further, the process is adaptive in the sense that it seeks to uniformly cover the entire network with a limited number of directory nodes, as proved in Section 4.

Nodes may establish a willingness to be promoted, which could be set by configuration or according to the residual energy, processing power, mobility, or interest of the node (e.g., a node with a large number of services). In the promotion process, the node willingness affects the window time within which a random value is chosen. At the end of this random time, the node verifies whether an announcement message was received from other nodes. Otherwise, it becomes a directory node. Nodes that are not willing to become directory nodes should wait for larger times. In our initial implementation, we did not make any distinction between nodes willingness.

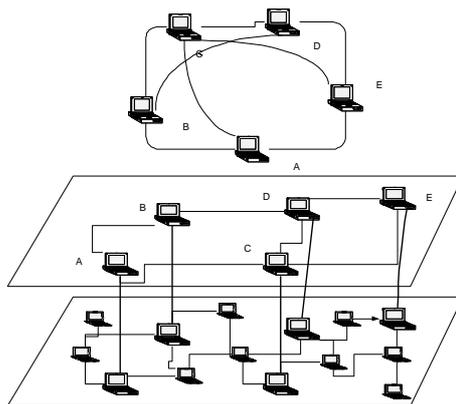
Each directory node keeps two tables, namely the virtual neighbors table and the service table. The first one is composed by neighbors in the overlay network, and is maintained through the reception of announcement messages. This table is used to forward queries that a directory node cannot answer in a similar process used in P2P network solutions. The service table maintains the services registered in the directory nodes by the service providers. The distribution of the service table is made through a simplification of the proposal in [8], which is explained below.

By these definitions, the proposed architecture has three planes of abstraction. First, the real ad hoc network composed by nodes in a topographic distribution. Second, the overlay network composed by a subset of the nodes, i.e.

directory nodes. Finally, the DHT ring. Using the idea presented in [8], this ring is divided in: sectorial communities, or directory nodes into the same ring sector that keep the same DHT fragment; ring neighborhood, between nodes of adjacent sectors; and the long-range contacts that are nodes from far sectors used to aid in the ring search. However, this long-range contacts are also virtual neighbors in the overlay network, and, hence, they are  $H$  hops neighbors in the first abstraction plane, i.e. the real network.

Service providers must subscribe their services in a directory node using registering messages, which are also forwarded by field gradient. So, a service is subscribed in a directory by proximity. However, with DHT utilization an associated entry must be inserted in directory nodes of the corresponding sector. Then, the directory node, upon the reception of a subscribe request, must forward it to nodes in the adequate sector, using the procedure in [8], but keeping a local cache to make easier a local search. Therefore, each directory node keeps two service tables: a sector services table and a neighborhood services table.

Directory nodes must know their virtual neighbors and their respective sectorial community in the ring. Selective flooding in the overlay network is used to this purpose. Note that this network has much fewer nodes than the real network. Thus, flooding can be performed by unicast messages exchanged between directory nodes. All tables maintained by directory nodes are soft-state.



**Fig. 1.** Abstraction layers

These three network planes can be seen in Figure 1, where the real network is the bottom plane. The middle plane is the overlay network that is composed by directory nodes only, in this example they are named A, B, C, D and E. In the higher plane is the DHT ring, where the nodes are arranged according to a sector sequence. However, they establish long-range contacts as reported in [8], but these long-range contacts are only created when these nodes are virtual neighbors in the second plane. For example, the A-C link in the ring is

created because there is a virtual neighborhood between A and C in the overlay network, since these nodes are  $H$ -hops neighbors in the first plane.

### 4 Analytical model

This section presents an analytical model for the process of directory nodes promotion, which is used to construct the overlay network. This model uses parameters of the network and table 1 presents the adopted notation. The network topology is modeled as a circular area.

**Table 1.** Adopted notation for directory nodes distribution

$N$	total number of nodes
$R$	network radius
$A = \pi R^2$	network area
$\delta = \frac{N}{A}$	density
$r$	transmission range radius
$H$	TTL of DN flooding
$Nv_H$	number of $H$ hops neighbors
$N_{DN}$	total number of directory nodes
$\delta_{DN} = \frac{N_{DN}}{A}$	directory nodes density
$f(\delta, H, r)$	average area of $H$ hops neighborhood of a node
$Tx$	rate of announcement messages per second sent by a DN

Considering that directory nodes are uniformly distributed in the network, we can assume that  $DN$  is a random variable with binomial distribution expressed by:

$$P(x = k) = \binom{n}{k} p^k (1 - p)^{n-k} \tag{3}$$

where  $p$  is the probability of a node to be a directory node, that is equal to  $\frac{N_{DN}}{N}$ ,  $k$  is the number of directory nodes ( $N_{DN}$ ) and  $n$  is the total number of nodes ( $N$ ) in the area. Considering as a sample the number of nodes distributed in an area of  $H$  hops, the average number of nodes in this sample is:

$$Nv_H = f(\delta, H, r) \times \delta = f(\delta, H, r) \times \frac{N}{\pi R^2} \tag{4}$$

By letting  $k = 0$ , we have the probability that there is not any DN in the  $H$ -hops neighborhood of a node:

$$P(x = 0) = \left(1 - \frac{N_{DN}}{N}\right)^{Nv_H} \tag{5}$$

However, if a node has not encountered a DN within its  $H$ -hops neighborhood, it must launch a promotion process. Assuming that the mechanism converges, we should not have other nodes being promoted, then:

$$\left(1 - \frac{N_{DN}}{N}\right)^{N v_H} \times N < 1 \quad (6)$$

Solving this inequality:

$$N_{DN} > N \left(1 - \left(\frac{1}{N}\right)^{\frac{1}{N v_H}}\right) \quad (7)$$

In this case, we have less than one node without DN in its  $H$ -hops neighborhood, and no more promotions should occur. Then, this equation gives a close approximation of the number of DN nodes in the network. Each DN sends  $Tx$  messages per second, and they are forwarded by each  $H - 1$  hops neighbors. Then, the total announcement messages transmitted per second is:

$$AnnouncementMsg/s = \left(N \left(1 - \left(\frac{1}{N}\right)^{\frac{1}{N v_H}}\right)\right) \times Tx \times \delta \times f(\delta, H - 1, r) \quad (8)$$

Substituting (4) and  $\delta$ :

$$AnnouncementMsg/(s \times N) = f(\delta, H - 1, r) \times Tx \times \frac{N}{\pi R^2} \times \left(1 - \left(\frac{1}{N}\right)^{\frac{1}{f(\delta, H, r) \times \frac{N}{\pi R^2}}}\right) \quad (9)$$

Since we can prove that:

$$\lim_{N \rightarrow \infty} \frac{N \times \left(1 - \left(\frac{1}{N}\right)^{\frac{k}{N}}\right)}{\ln(N)} = k \quad (10)$$

Then,  $AnnouncementMsg/(s \times N) = O(\ln(N))$ , and for high values of  $N$ :

$$AnnouncementMsg/(s \times N) = \frac{f(\delta, H - 1, r)}{f(\delta, H, r)} \times \ln(N) \times Tx \quad (11)$$

and

$$N_{DN} = \frac{\pi R^2}{f(\delta, H, r)} \times \ln(N) \quad (12)$$

To estimate the area given by  $f(\delta, H, r)$  observe that its maximum value occurs when the nodes are in the border of the transmission range ( $r$ ). Then:

$$f(\delta, H, r) = \pi(rH)^2 \quad (13)$$

On the other hand, we can consider as an inferior limit to  $f(\delta, H, r)$  when the nodes are placed in  $\frac{r}{2} + dr$ . Despite this is not being the worst case, we can admit this as an inferior limit to  $f(\delta, H, r)$  in a network with a random distribution and a density not too small. Then:

$$f(\delta, H, r) = \pi\left(\frac{r}{2}H\right)^2 \quad (14)$$

Using the maximum value for  $f$  and  $H > 1$ :

$$AnnouncementMsg/(s \times N) = \frac{(H-1)^2 r^2}{R^2} \times N \times \left(1 - \left(\frac{1}{N}\right)^{\frac{R^2}{N H^2 r^2}}\right) \times Tx \quad (15)$$

And, for large  $N$ , we have:

$$AnnouncementMsg/(s \times N) = \frac{(H-1)^2}{H^2} \times Ln(N) \times Tx \quad (16)$$

To obtain the total number of control messages in the first plane, to these announcement messages should be added the hello messages sent by all nodes at a  $Tx$  rate.

The  $N_{DN}$  is more dependent on  $f(\delta, H, r)$ , but it can be approximated by:

$$N_{DN} = \frac{R^2}{r^2 H^2} \times Ln(N) \quad (17)$$

In (16) and (17) the product  $H \times r$  cannot be much higher than  $R$  since not the network is entirely flooded. In this case, these fractions should be replaced by 1.

## 5 Performance Evaluation

We use the ns-2 simulator, version 2.29, to evaluate our architecture. We made two implementations, both using IEEE 802.11 standard at MAC layer configured with its default parameters. The first one is equivalent to the proposal in [14]. Using this implementation, we obtain similar results to the original work in the same scenario of  $1300 \times 1500$  meters, 100 nodes, 10 clients sending 4 queries per second, random way point mobility with velocity up to 20m/s without pause and 1000 seconds of simulation.

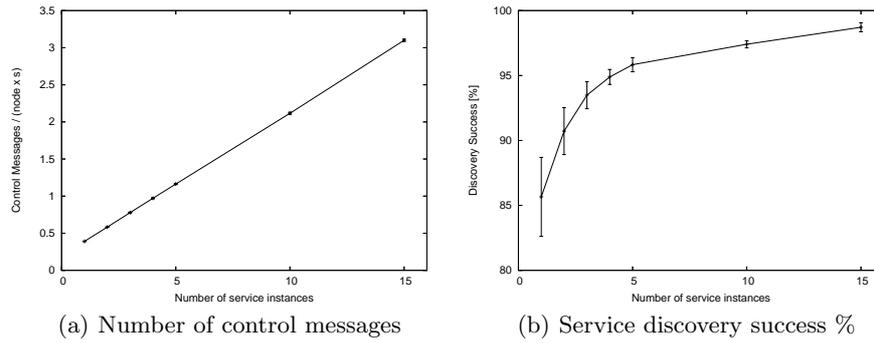
With this specification, we have the following values applicable to our analytical model:

**Table 2.** Values used in simulations

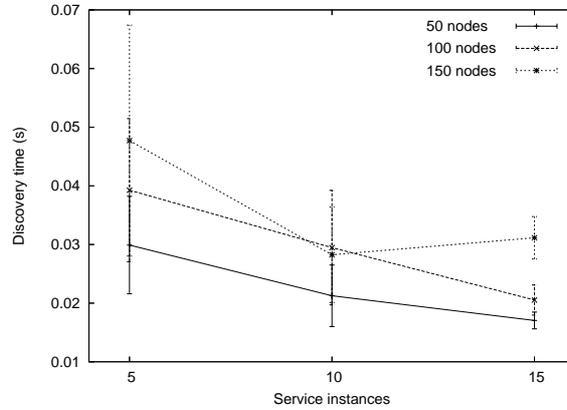
$N$	= 100 nodes
$A$	= $1300 \times 1500 = 1950000 \text{ m}^2 = 1.95 \text{ km}^2$
$\delta$	= $51.28 \text{ nodes/km}^2$
$r$	= 250 m
$Tx$	= $1 \text{ message/5s} = 0.2 \text{ messages/s}$

In addition, we run some other simulations with different scenarios to understand the detailed behavior of this mechanism. We used a simulation time of 200 seconds with clients sending queries after an initial convergence time, estimated in 16 seconds, when DNs get promoted and state tables are constructed. Figures 2(a), 2(b) and 3 show the overhead imposed by control messages, the

discovery success rate and the average response time when the number of service instances is varied, respectively. The discovery success rate is the ratio of the number of service query messages that arrive at directory nodes to the total number of query messages sent. The control message overhead is measured as the sending rate of control messages per node. Finally, the average response time is the time interval between sending a query message and receiving it at directory nodes or service instances. This metric is different from search time, which must include the transmission in the overlay network plus the latency of the query response to the requesting node.



**Fig. 2.** Control messages overhead and discovery success rate per service instances



**Fig. 3.** Average response time

These curves show that their proposal suffers with the decrease of the number of service instances, presenting a smaller service discovery rate with a greater

confidence interval. Therefore, we conclude that the proposal in [14] has a better performance to a specific number of service instances/nodes ratio. If the number of instances is too low, the efficiency in service discovery decreases. Otherwise, if it is high, there is a large increase in network overload. This happens even when the mechanism of flooding reduction (cache information in intermediate nodes) is used. In this case, the number of messages is limited but the number of overhead bytes increases linearly to the number of service instances.

The second implementation applies to our architecture. In this implementation the flooding control is made by a time to live (*t**tl*) in each packet, as *H* variable of analytical model, which defines a flooding area. The same performance metrics defined above are evaluated.

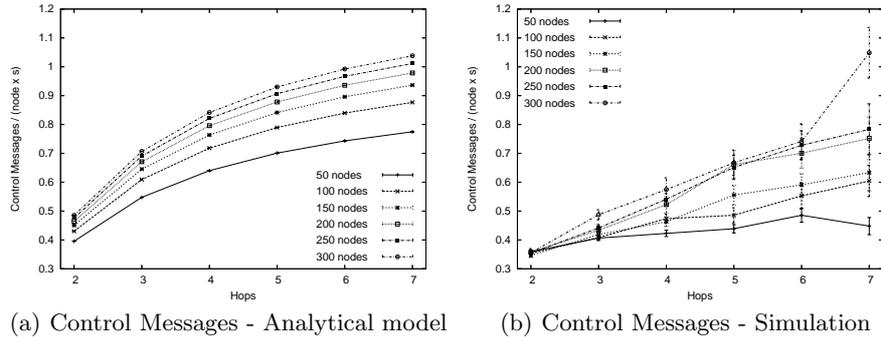


Fig. 4. Analytical and simulation results

In the first simulations, we used the same parameters of table 2, but without mobility and varying the number of nodes (*N*). The control message overhead obtained by the analytical model and simulations are presented in Figures 4(a) and 4(b), respectively. The correspondence between the results demonstrates the effectiveness of our analytical model.

In the same simulations we obtained the two other metrics, discovery success rate and average response time, shown in Figures 5(a) and 5(b). These figures show that the first plane of our architecture presents coherent values, when compared to Figures 3 and 2(b), in a scenario without mobility.

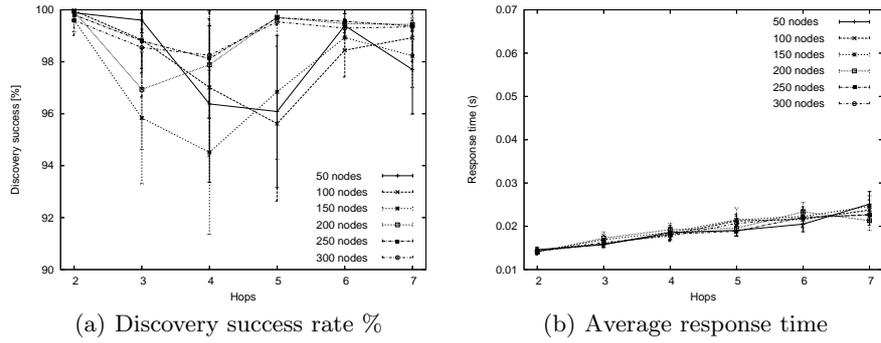


Fig. 5. Discovery success and response time without mobility

In second simulation runs, we evaluate geometrical aspect of the network. In the first simulations, we used an almost square area and we obtained equivalent results to the analytical model, which used a circular area. In Figure 6, simulation results using a rectangular area with  $3000m \times 650m$  are shown. Similar results and a control message overhead limited to  $O(LnN)$  are obtained.

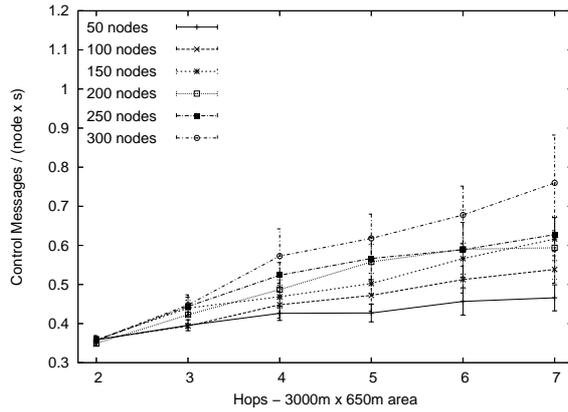


Fig. 6. Control Messages - 3000m x 650m area

After these validations, we made new simulations in scenarios with all nodes using random waypoint mobility with velocity up to 20m/s, without pause. Despite the fact that our analytical model do not consider mobility aspect, we can see in Figure 7 only a small increase of control message overhead when compared to the static scenario. This increase occurs because neighborhood changes, provoking new DN promotions. However, the overall behavior is similar to the previous results and they demonstrate the effectiveness of our model.

We also measured discovery success rate and average response time, shown in Figures 8(a) and 8(b), respectively. These results show a great robustness to mobility of the mechanisms proposed, since they are very similar to the ones obtained with static scenarios.

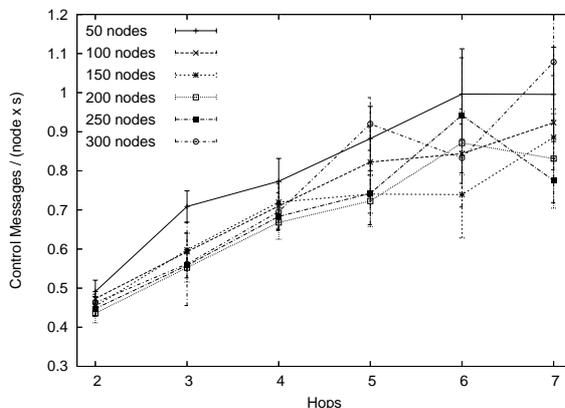
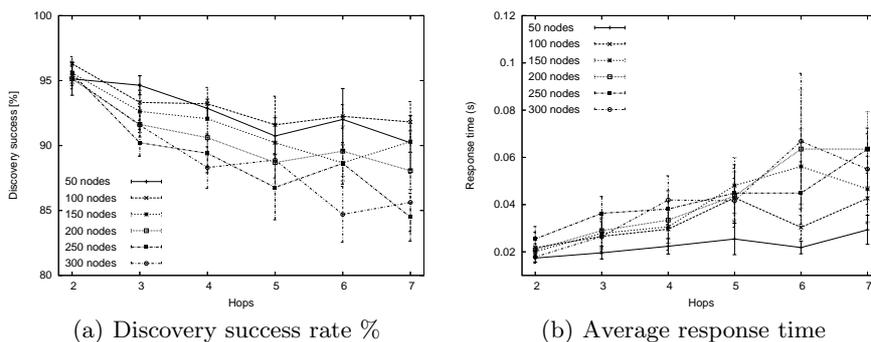


Fig. 7. Control Messages



(a) Discovery success rate %

(b) Average response time

Fig. 8. Discovery success and response time with mobility

## 6 Conclusion

From analytical and simulation results, we conclude that our architecture for service discovery on ad hoc networks presents a good performance. The control message overhead in the first abstraction plane of this architecture is  $O(Ln(N))$ , which implies a good scalability and efficiency. Another main feature of our proposal is its independence from network parameters, as number of nodes,

number of service instances and density. Besides, the proposed architecture achieve a good robustness against nodes mobility even when moderate to high degrees of mobility are used

A future work is the implementation of the second plane and its validation. With this implementation, a performance evaluation in different scenarios, adjusting the promotion process to several network densities and different mobility models and patterns will be done. The architecture will be also evaluated against node removal, network partitioning and rejoining.

## References

1. RFC2608. Service Location Protocol, Version 2 (1999).  
<ftp://ftp.rfc-editor.org/in-notes/rfc2608.txt> - Último acesso em 13/12/2005.
2. Jini. Jini Network Technology (2005).  
<http://www.sun.com/software/jini/> - Último acesso em 13/12/2005.
3. UPnP. UPnP Forum (2005).  
<http://www.upnp.org/> - Último acesso em 13/12/2005.
4. Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329 (2001).
5. B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley (2001).
6. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content addressable network. Technical Report TR-00-010, Berkeley, CA (2000).
7. Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160 (2001).
8. G. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in a small world. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003)* (2003).
9. Gnutella. Gnutella Protocol Development (2005).  
<http://rfc-gnutella.sourceforge.net/> - Último acesso em 13/12/2005.
10. JXTA. JXTA technology (2005).  
<http://www.jxta.org/> - Último acesso em 13/12/2005.
11. Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *STOC '00: Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170 (2000). ISBN 1-58113-184-4.
12. D. Doval and D. O'Mahony. Nom: Resource location and discovery for ad hoc mobile networks. In *1st Annual Mediterranean Ad Hoc Networking Workshop, Medhoc -Net* (2002).
13. Ahmed Helmy, Saurabh Garg, Priyatham Pamu, and Nitin Nahata. Contact-based architecture for resource discovery (card) in large scale manets. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 219.1 (2003).

14. Vincent Lenders, Martin May, and Bernhard Plattner. Service discovery in mobile ad hoc networks: A field theoretic approach. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 120–130 (2005).
15. Françoise Sailhan and Valérie Issarny. Scalable service discovery for manet. In *3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'2005)*, pages 235–244 (2005).
16. J. Antonio Garcia-Macias and Dante Arias Torres. Service discovery in mobile ad-hoc networks: better at the network layer? In *ICPP 2005 Workshops. International Conference Workshops on Parallel Processing*, pages 452–457 (2005).
17. Sumi Helal, Nitin Desai, Varun Verma, and Choonhwa Lee. Konark - a service discovery and delivery protocol for ad-hoc networks. In *Third IEEE Conference on Wireless Communications and Networking (WCNC)*, volume 3, pages 2107–2113 (2003).