Service Location Protocol Overhead in the Random Graph Model for Ad Hoc Networks

Hend Koubaa, Eric Fleury

HAL Id: inria-00107566
https://inria.hal.science/inria-00107566

Submitted on 19 Oct 2006

Service Location Protocol Overhead in the Random Graph Model for Ad Hoc Networks

Hend Koubaa   Eric Fleury

LORIA - INRIA

Campus Scientifique - BP239

54506 Vandœuvre-Les-Nancy - FRANCE

{ Hend.Koubaa, Eric.Fleury}@loria.fr

Abstract

To deploy a protocol in a network, a deep performance study has to be curried. In particular, in the case of ad hoc networks, this performance is principally tied to overhead optimization. In fact, bandwidth limitation and frequent collisions in ad hoc networks represent constraints upon deploying applications which overhead is important. In this paper, we are interested in minimizing overhead due to the deployment of an application we think is relevant to be deployed in ad hoc networks. This application, called service location, aims to make it possible for a mobile user to know about provided services in the network. In this context, we propose an analysis to estimate consumed overhead in different service location approaches. Our analysis is motivated by network and ad hoc application modeling and estimation of overhead due to service covering and request handling.

1   Introduction

To get benefit from a network service, a user still has to know its network address. The current network evolution covers a broad spectrum of services and enables to switch from an infrastructure offering a minimum service set to more sophisticated and programmable architectures offering a complex service set. The number of services will grow in the same manner as service demand is growing. With this evolution perspective, the current manual configuration will not be practical anymore when we aim scalability. Service location protocols [1], [2], [10], [11] try to overcome this drawback: a user has only to provide the service type and its attributes. It is the role of the service location protocol to search for the host address where the user can find the service he/she wants.

To the best of our knowledge, current service protocols are designed for wired networks or for piconets. Nevertheless, it seems important to continue to offer this kind of services in ad hoc networks. The intrinsic characterization of ad hoc networks is the node mobility which induces that nodes may van-

ish at any time. The notion of *server* that is present in some service location protocols is not valid any more. We have developed a service location protocol that ensures two principal tasks: Service Covering Set construction and Request Handling. The Service Covering Set construction consists of a backbone creation and the Request Handling specifies how a client request is handled. The backbone in our context groups server entities we call mediators and which role is to locally cover services. In this paper we focus our analysis on the overhead study of the service location protocol. This analysis is based on an overhead comparison between the Service Covering Set approach, the One_server approach which is an approach based on the use of one server, and broadcast approaches.

The remainder of this paper is organized as follows. In Section 2, we present related work. Section 3 presents different approaches for a service location protocol in ad hoc networks. The overhead study of the proposed protocols is investigated through network and application modeling. The section 4 presents the considered network and application models. The analyse results are presented in the sections 5 and 6. Finally, conclusions are presented in Section 7.

2   Related work

Performance study of protocols in ad hoc networks is most of the time based on simulations. Even though, some works [4], [7] give analysis and simulation results. Their analysis are based on indoor and outdoor network models. They use the random graphs, the sparse grid, the dense strip and a one dimension model. Recently, in [5], the same authors present a new model called the random unit graph for outdoor networks. This model is different from the random graph since a link between two nodes is not henceforth based on a fixed probability but is based on the distance separating them: a link exists if the distance is smaller than the unit length. This model is justified by the fact that signal attenuation in outdoor networks is principally influenced by distance. We

think that analysis and simulation results can help giving optimized conclusions about protocol performances, especially when simulation results confirm the analysis ones.

3 Proposed approaches

In wired networks, the introduction of a server aims to guarantee scalability: it makes it possible the transmission of requests and the service declarations from a network to another. The problem that raises is to know if we have to deploy server/servers in ad hoc networks. This problem is very tied to the approaches used in the ad hoc routing: the reactive approach and the proactive approach. If we think that the answer is negative, two approaches are possible: the provider proactively broadcasts its services in all the network, or the clients reactively broadcast their requests. In this section, we present first, the main characteristics of a protocol based on one server, second, protocols without servers and third a multi server protocol.

3.1 One server approach

Each client and provider has to know the address of the server. A possible manner a client/provider may be informed about the server identity is that server declaration message is periodically broadcast. Another manner that overcomes the broadcast problem is based on the reactive approach. The client or the provider which needs to locate the server initiates a discovering process based on flooding. Each provider is periodically declared to the server. The robustness of the one server protocol is very tied to the robustness of the server. If the server moves away, the first provider discovering its absence is declared server. In the case of a conflict, the provider with the minimum address is elected. Client requests are sent to the server. This protocol has the disadvantage that in case of network partition, more than one server will be responsible for provider covering. When the network is recovered, we have to introduce a mechanism that ensures the election of one among these servers.

3.2 Broadcast approaches

The service location protocols that do not introduce servers are not interesting from a scaling point of view. Their implementation and application are simple and they can be used in small ad hoc networks . Two approaches are possible in this case. The main characteristic of the first approach consisting of request broadcast is that it doesn't include signaling traffic. This approach, we call client broadcast approach is a reactive approach since the service location protocol overhead is added to the network in an on demand manner. In the second approach, all services are periodically broadcast. In this case, each client can save the service information in its cache. This

approach, we call service broadcast approach is a proactive approach since service location overhead is proactively added to the network.

Both approaches proposed above are based on broadcast which is very expensive in ad hoc networks. In fact, most unicast protocols do not optimize broadcast. The minimization of the broadcast overhead is even though considered in some works [3], [6]. For this reason, we propose another protocol based on local covering by many servers we call mediators as we explain next.

3.3 Mediator approach

Mediator backbone construction Our goal is to scatter mediators only in clusters containing services in order to not unnecessarily disturb parts of the network that do not contain any services. We present in the following the main characteristics of mediators that must allow services to be declared without having recourse to broadcast and that must allow a client to search for services in their local area.

Let model an ad hoc network by a graph $G = (V, E)$. We are interested by the sub-graph $G' = (P, E_p)$ induced by $P \subseteq V$, the set of all providers (see the figure 1). A provider cluster related to a given provider is the set of nodes in its neighborhood that are providers. This provider is called a cluster-head (in the example of the figure 1, the cluster of the node p_3 contains p_3, p_6, p_1 and p_9). Each provider is then a cluster-head of its cluster and it is a member of its neighbor provider clusters (as an example, p_3 belongs to the cluster of p_9). An ad hoc network can be viewed as a set of clusters. Note that we can extend the definition of provider neighbors to nodes that are less than $k > 1$ hops away from the provider.
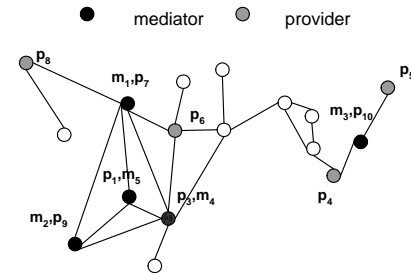

Figure 1. Example of a mediator configuration

The idea of our protocol is analogue to a MAC protocol in which all nodes share the same access medium. In our case, the network is considered as a set of provider clusters in which each provider shares the medium with providers of its cluster, *i.e.*, only one of them can use the medium. The first not covered provider accessing the medium will be elected as a mediator.

Each provider periodically declares its identity and its services to all its neighbors and it also listens to messages sent by

them. After a period of time `Election Timeout`, it can deduce that it is in one of the following three situations: (1) not connected to any mediator; (2) unidirectionally connected to one or more mediators or (3) bidirectionally connected to at least one mediator. In the both first situations, the provider is not covered and then can be declared mediator. In the last case, it is covered and then it does nothing. Each mediator in the network periodically declares its identity and the identities of providers it covers.
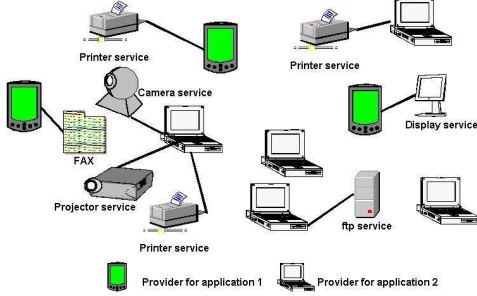

Figure 2. Service deployment by ad hoc applications

Global management of mediators A client must be able to locate a mediator since this last one represents an entry point to the set of services available in the network. Routing clients' requests must be provided since both clients and mediators may move. Even though, in some cases, clients can simply locate needed services if they are in their neighborhood. In fact, client requests are handled as follow: *(1)* If the service is declared in the client neighborhood then this service can be found in its cache and the client can directly send its request to the provider offering this service. Note that the use of a cache inside each client accelerates the request response time but does not allow to perform global service management. *(2)* If not, the client looks for mediators in its zone only if a mediator can respond. *(3)* In all other cases, the request routing problem is raised. Request handling is ensured by using mediator multicasting. More details of the mediator multicast approaches can be found in [8].

4 Network and application models

In our analysis, we considered the random graph model. This model is particularly relevant for mobile networks where nodes are virtually in range of each other but random perturbations make that some pair of nodes can not receive each other. In this model, we consider a mobile network made of n nodes and assume that the events "a node A correctly receives the packets transmitted by node B" are iid. when A and B vary in the graph and each event occurs with fixed probability p. We denote $q = 1 - p$. In our analysis, n and p are the parameters modeling the network. Other parameters are considered. These parameters are specific to ad hoc applications.

We consider the following parameters: *(i)* P_s is the probability to be a provider, *(ii)* b is the number of providers in the network: $b = P_s n$, *(iii)* P_c is the probability to be a client, *(iv)* c is the number of clients in the network: $c = P_c n$ and *(v)* $p_{response}$ is the probability that a provider is able to respond a client request. More clearly, this probability means that if there are n_s services provided in the network, $p_{response}$ is equal to the number of services offered by a node divided by n_s. For example, if $n_s = 10$ and $p_{response} = 0.1$, this means that each provider offers in average only one service. We have considered this assumption in order to simplify and for the reason that in an ad hoc application, the application organizes its services among its members to ensure load balancing. In this case, each member offering some services is viewed as a provider. We take for example an ad hoc conference. In this application, the services offered can be printing, projection of a remote laptop display, remote control of the projector used and voice recognition system. If we say that $p_{response} = 0.5$, this means that each provider in the ad hoc conference contains in average two services (among the four services). We think that this model is relevant for mobile applications in which a main objective is to balance load because transmission power and wireless bandwidth are limited. The figure 2 presents an example of service deployment of two ad hoc applications.

In addition, we assume a linear model of clients and providers as function of the network size. In fact, we think that an ad hoc network deployment is curried out to serve one or more applications. Each application can use the service location protocol to locate needed services. The application is a concern of a specific group which size is linear with the network size. The group contains clients and providers that are present to realize a specific objective. An application can get benefit from other application services as we explain in [9].

5 Overhead analysis

The overhead generated by a protocol based on one server is given by the cost of the server identity broadcast, the provider registrations, the client request transmission to the server and the replies transmitted by the server. We note that we do not take into account the overhead due to the server election and the update of this information within all the network. This simplification is justified by the fact that the server changing is not as frequent as the overhead added due to client requests and provider registrations. In the random graph model with parameters (n, p), the broadcast costs n and the transmission of a packet costs the average distance between two nodes, i.e. $1 + q$ or $1 - 2p$. Thus, the total overhead is equal to $n + (b + 2c)(2 - p)$.

The overhead generated by a protocol based on provider declaration broadcast is given only by the cost of provider identity broadcast. Thus, it is equal to bn.

The overhead generated by a protocol based on request

3

broadcast is due to the client request broadcast and the reply transmission of the potential providers. Thus, it is equal to $cn + cbp_{response}(2 - p)$.

Let take M the number of mediators in the network.

The evaluation of the overhead due to mediators is based on the study of four possible situations in which a client request is handled: *(i)* the client finds its response in its cache, *(ii)* the client finds its response within its mediator, *(iii)* the client does not find its service within its mediator and *(iv)* the client has not a mediator.

We start our analysis by giving some definitions: *(i)* P_1 is the probability that a client has not the service information in its cache and is served by one of its mediators, *(ii)* P_2 is the probability that a client has not the service information in its cache and is not served by its mediators, *(iii)* P_0 is the probability that a client has not the service information in its cache and has not a mediator and *(iv)* P_3 is the probability that a client has the service information in its cache.

To estimate these probabilities, we need the probability $P_{NoCache}$ which is the probability that the client has not the service information in its cache. It is the probability that the client is not connected to all providers that offer the requested service. Let N_{PoPr} be the number of these providers and P_{Cache} be the probability that the client has the service information in its cache. P_{PoPr} is the probability to be a possible responder. We have then: *(i)* $N_{PoPr} = P_{PoPr}n = p_{response}b$, *(ii)* $P_{NoCache} = (1 - p)^{N_{PoPr}} = (1 - p)^{(P_{response}b)}$ and *(iii)* $P_{Cache} = 1 - P_{NoCache} = 1 - (1 - p)^{(P_{response}b)}$.

Let P_{NoMed} be the probability that the client has not a mediator and P_{Med} be the probability that the client has a mediator. We have *(i)* $P_{NoMed} = (1 - p)^M$ and *(ii)* $P_{Med} = 1 - P_{NoMed} = 1 - (1 - p)^M$.

Let P_{MedRes} be the probability that a mediator responds to the client and $P_{MedNoRes}$ be the probability that a mediator does not respond to the client. P_{MedRes} is the probability that a mediator has the response in its cache, thus it is equal to P_{Cache}. We have *(i)* $P_{MedRes} = P_{Cache} = 1 - (1 - p)^{(P_{response}b)}$ and *(ii)* $P_{MedNoRes} = 1 - P_{MedRes} = (1 - p)^{(P_{response}b)}$.

We use these probabilities to estimate the four probabilities P_1, P_2, P_0 and P_3. To simplify M estimation, we consider the worst situation in which each provider will declare itself a mediator. In this case, $M = b$. Then, we use these four probabilities to estimate the overhead of the mediator protocol. This overhead is given by the cost of the locally mediator identity broadcast, the locally provider identity broadcast and the overhead used to the request handling. This last overhead is equal to the sum of four overheads having the following probabilities: *(i)* the overhead with probability P_3 is used by clients finding the service in the cache, *(ii)* the overhead with probability P_1 is used by clients finding the service within their mediators, *(iii)* the overhead with probability P_2 is used by clients that do not find the requested services within their mediators and finally *(iv)* the overhead with probability P_0 is used by clients that do not have locally mediators.

The overhead due to the locally mediator identity broadcast is equal to $M.1$. The overhead due to the locally provider identity broadcast is equal to $b.1$. The overhead needed to make it possible to clients to find the services in the local cache is equal to 0 since no messages are transmitted for this purpose. The overhead needed to make it possible to clients to find the services within the local mediators is equal to 2. In fact, we suppose that only one mediator is asked. This assumption is based on the fact that the average number of mediators covering a provider is not high. The overhead needed to make it possible to clients to find the services within mediators other than their local ones is equal to $2 + 2n$. The overhead needed to make it possible to clients that do not have locally mediators to find the services within mediators in the network is equal to $2n$.

The mediator protocol overhead is thus equal to:

$$2b + c$$
$$(2(1 - p)^{(P_{response}b)}(1 - (1 - p)^{(b)})(1 - (1 - p)^{(P_{response}b)})$$
$$+ (2 + 2n)(1 - p)^{(2P_{response}P_s n)}(1 - (1 - p)^{(b)})$$
$$+ 2n(1 - p)^{(P_{response}P_s n)}(1 - (1 - p)^{(b)})$$

A comparison between the different overheads is based on varying n, p, $p_{response}$, P_s and P_c. The objective of our analysis is to prove that in case of specific ad hoc applications the overhead generated by the service location protocol based on mediators is better than the other approaches. The result of our analysis is encouraging since in most cases, the service location protocol based on mediators is the best in terms of overhead minimization. In addition it makes it possible to clients to be served by the nearer provider.

Our analysis is based on fixing a parameter and varying the other parameters. We varied n from 0 to 600 and P_s from 0 to 1. For p, $p_{response}$ and P_c, we took low and high values.

5.1 Overhead comparison between the One_Server and the mediator approaches

p **and** $p_{response}$ **are low** At first we have chosen low values of p and $p_{response}$ ($p = 0.1$, $p_{response} = 0.1$).

We present in the figure 3 a graph visualizing zones within which an approach outperforms another. We remark that the One_Server approach outperforms the Mediator approach in small size networks. The Mediator approach outperforms the One_Server approach in large size networks when the percentage of providers is high. For example, if 50% of nodes are providers, the Mediator approach outperforms the One_Server approach only if the network size is superior to 500. Thus, in sparse networks, when $p_{response}$ is low, the One_Server approach outperforms the Mediator approach when the network size is low and the Mediator approach outperforms the One_Server approach when the network size and the provider percentage are both high.

We note that the increase of client percentage favors the One_Server approach. This is justified by the fact that the term
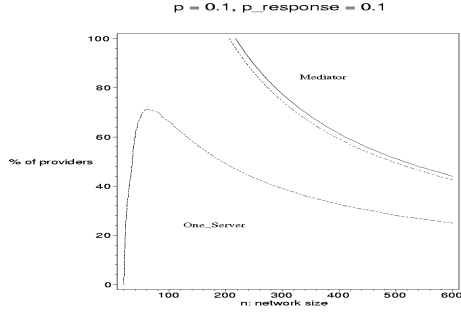
4

p = 0.1, p_response = 0.1

p = 0.9, p_response = 0.1

Figure 3. Comparison between One_Server and mediator protocol overheads for different values of client percentage: 2% **in dash line,** 50% **in dot line and** 100% **in solid line**

Figure 5. Comparison between One_Server and mediator protocol overheads for different values of client percentage: 2% **in dash line,** 10% **in dot line and** 100% **in solid line**

in c in the One_Server overhead is $2c(2-p)$ which is low if it is compared to the term in c in the mediator overhead which is in n. This is confirmed in our model: in one hand, if the number of clients increases, the overhead in the One_Server approach is increased due to unicast client/server communication overhead which is independent of network size, provider percentage and $p\_response$. In another hand, the overhead in the mediator approach is increased due to client/mediators communication which depends on network size. We remember here that $p_{response}$ is low and the probability a client will look for a service in all the network is not low.
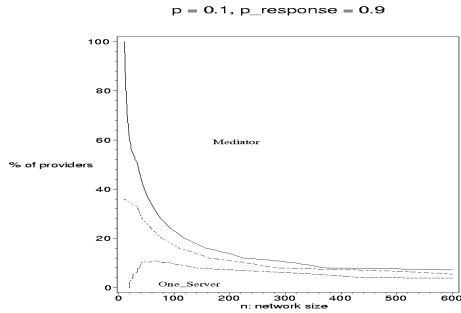
p **is high and** $p_{response}$ **is low** We can remark that the figure 5 is similar to the figure 4. Thus, we have to maximize the probability, a client is connected to more than one provider or the probability, a provider contains more than one service. Even though, we note that the influence of p is superior to the influence of $P_{response}$: for example, in the case $n = 100$ and the client percentage is 100%, if $p = 0.9$ and $P_{response} = 0.1$, the minimum P_s for the Mediator approach to outperform the One_Server approach is 10% but it is 20% when $p = 0.1$ and $P_{response} = 0.9$. We note that the One_Server approach outperforms the mediator approach if P_s is very low.

p **and** $p_{response}$ **are high** In dense networks, the Mediator approach outperforms the One_Server one when $P_{response}$ is high (see the figure 6).

6 Overhead comparison with broadcast approaches

Service broadcast approach We present in the figure 7 zones within which an approach outperforms the other approaches. We remark that the service broadcast approach outperforms the One_Server and mediator approaches only in small size networks. This is justified by the fact that service broadcast overhead is in n^2. Thus, the service broadcast protocol performs badly in case of large size networks and when the provider percentage is high. We note that in this section, we didn't vary the values of p and $p_{response}$ since the service broadcast protocol cost is independent of these values.

p = 0.1, p_response = 0.9

Figure 4. Comparison between One_Server and mediator protocol overheads for different values of client percentage: 2% **in dash line,** 10% **in dot line and** 100% **in solid line**

p **is low and** $p_{response}$ **is high** In this section, we take a high value of $P_{response}$. We present in the figure 4 a graph visualizing zones within which an approach outperforms another. We remark that in most of cases, the mediator approach outperforms the One_Server approach. The influence of client percentage increase is less significant than the case $P_{response}$ is low. This is due to the fact when $P_{response}$ increases, the probability a client is locally served increases and thus the overhead incurred is lower.
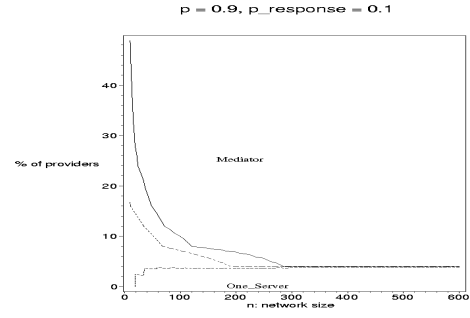
Client broadcast approach The behavior of the client broadcast protocol is not better than the service broadcast protocol when it is compared to the One_Server and mediator protocols. It is less performant in case of large size networks and when the provider percentage is high (see the figure 8). In this figure, we present zones within which the mediator approach outperforms the one_Server approach and the client broadcast approach. The client broadcast approach does not

5

outperform the mediator approach unless we are in the case of small size networks and a low number of providers. We note that the client broadcast approach does never outperform the One_Server approach. In fact, the One_Server overhead is not high even in the case of large size networks. Even though, the mediator approach overhead is high when the number of providers is low. For this reason, in the case of a low number of providers, the client broadcast outperforms only the mediator approach.

When we compare both broadcast approaches, we remark that the client broadcast performs better in the case of small client number and worse if we increase the number of clients.
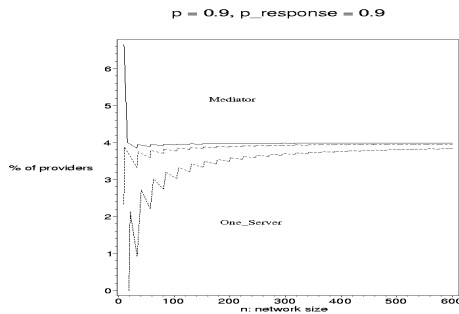

Figure 6. Comparison between One_Server and mediator protocol overheads for different values of client percentage: 2% **in dash line,** 10% **in dot line and** 100% **in solid line**
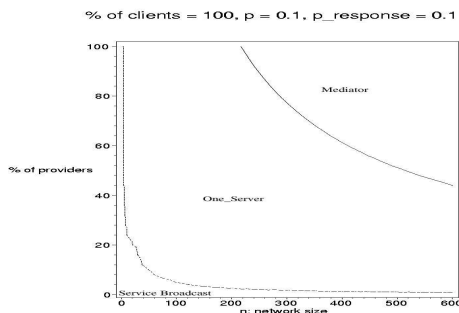

Figure 7. Comparison between One_Server, mediator and service broadcast overheads: service broadcast/ One_Server in dot line and One_Server/mediator in solid line

7   Conclusion

We resume that the mediator approach is more performant than the one_server approach in sparse large size networks when the percentage of providers is high. It is also more performant, independently of network size and provider percentage when the network is dense or the probability a provider
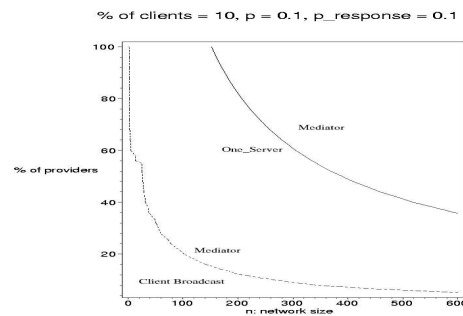

Figure 8. Comparison between One_Server, mediator and client broadcast overheads: client broadcast/ One_Server in dot line, client broadcast/ mediator in dash line and One_Server/mediator in solid line

contains the requested service is high. The Mediator approach is thus the most scalable. Even though, the other approaches can be deployed in the case of small size networks when the percentage of providers is not high. The service broadcast is performant only in small size networks when the percentage of providers is low. The client broadcast approach is usually outperformed by the mediator one or the one_server one.

References

[1] S. Consortium. Salutation architecture. http://www.salutation.org/.

[2] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz. An Architecture for a Secure Service Discovery Service,. In *Proc. IEEE Mobicom'99*, Seattle, Washington, USA, 1999.

[3] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual International Conference on Mobile Computing (MobiCom'99)*, volume 31,5, pages 78–91, Washington, Aug. 1999.

[4] P. Jacquet and A. Laouiti. Analysis of mobile ad hoc network routing protocols in random graph models. Technical Report 3835, INRIA, 1999.

[5] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance analysis of olsr multipoint relay flooding in two ad hoc wireless network models. Technical Report 4260, INRIA, September 2001.

[6] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, and T. Clausen. Optimized link state routing protocol. Internet Draft draft-ietf-manet-olsr-00.txt, IETF MANET Working Group, March 2001.

[7] P. Jacquet and L. Viennot. Overhead in mobile ad-hoc network protocols. Technical Report 3965, INRIA, 2000.

[8] H. Koubaa and E. Fleury. A Fully Distributed Mediator Based Service Location Protocol in Ad Hoc Networks. In *Proc. IEEE SAWN, Globecom*, San Antonio, Texas , USA, november 2001.

[9] H. Koubaa and E. Fleury. Reflections on Ad Hoc Cooperative Teams. In *Workshop on "Mobile Ad Hoc Collaboration" CHI*, Minneapolis, Minnesota, USA, April 2002.

[10] S. Microsystems. Jini technology specifications. white paper. http://www.sun.com/jini/specs/.

[11] U. Plug and P. Forum. Universal plug and play device architecture. white paper. http://www.upnp.org/resources.htm.