A Service Distribution Protocol For Mobile Ad Hoc Networks

Mohamed Hamdy and Birgitta König-Ries
Institute of Computer Science, Friedrich-Schiller-University Jena
07743 Jena, Germany
hamdy|koenig@informatik.uni-jena.de

ABSTRACT

Achieving high service availability is a key requirement when deploying service-based applications over ad hoc networks. Without high availability, reliable service execution cannot be achieved. Ad hoc network characteristics like ever-changing topology and limited resources are the main challenges that need to be overcome in order to achieve high availability. In this work we introduce our service distribution protocol that aims to increase the availability of the services over ad hoc networks based on two mechanisms: a service replication and a service hibernation mechanism. These mechanisms are using measurements of both clients' and providers' interests in a specified service to obtain a good service distribution over participant nodes. A detailed simulation of our approach was done which shows promising results.

Categories and Subject Descriptors: C.2.2 [Computer-Communication Network]: Network Protocols.

General Terms: Algorithms.

Keywords: Mobile service, Service replication.

1. INTRODUCTION

1.1 Motivation

Mobile ad hoc networks represent the only feasible solution in many applications and situations. Usage of ad hoc environments is varying from mobile networks in small offices to the most aggressive situations like in rescue and disaster recovery operations. In this network type, not only sharing of data is an important issue but also sharing of functionality. Also, ad hoc participants have to rely on shared computations to achieve the basic core processes of the network like routing and transportation. Service orientation offers an excellent basis for ad hoc networks to achieve this sharing of functionality both on the application and the "infrastructure" level. Service-orientation is a paradigm that inherently allows for loose coupling and dynamic changes. If semantics are added to service-orientation, the treatment of changes can be further improved. We explore the general feasibility of service-orientation for mobile applications in general in [1].

In general, in order to realize a service oriented architecture, some basic components, i.e., service description, discovery, matching, and execution components, are needed. Unfortunately, not all of these components can be directly mapped from the web service domain to ad hoc networks. Instead, some adaptations are necessary. In previous work, we have developed a semantic service description language that is lightweight enough to be usable in ad hoc environments [2, 3], and overlay mechanisms to support service discovery in dynamically changing ad hoc networks. In this paper, we are taking a closer look at the service execution phase, which turns out to be the phase that needs to be most radically adapted in order to be able to deal with typical ad hoc network characteristics. Suppose a client node achieved a successful service discovery, and started to execute some service remotely on another provider node. In an ad hoc network, is has to be considered normal that the provider node becomes either temporary or even permanently unavailable for the client node. Over the network operation time, the network topology always varies. The network could be partitioned in more than one partition. A provider node can easily disjoin the network. Of course, from a client's perspective it is unacceptable, if a high percentage of service calls never results in a successful execution because the service provider becomes unavailable while working on the request. Note, that this unavailability can have two reasons: Either the provider leaves the network or consumer and provider move apart and end up in different network partitions. From the client's point of view it doesn't really matter which of these causes occurred – both have the same undesirable result.

What is needed is therefore a mechanism that increases service availability. Since it is neither feasible to forbid a provider or consumer or even the nodes on the route between the two to move, and thus to avoid network partitions, nor to forbid providers to leave the network, service replication represents the only solution to increase service availability. Sufficient replication can ensure the availability of a specified service. Let us assume that the network is partitioned into more than one partition, if there is a "replica" in each network partition, then client nodes can continue the service execution from another replica-provider even if they move between partitions.

The main questions that need to be answered when developing a replication scheme are: Where to put replicas?

ICPS'08, July 6–10, 2008, Sorrento, Italy.

Who is allowed to maintain a replica? When to invalidate a replica? How will clients choose the suitable replicas to communicate with? How to manage replica consistency? How to merge two replicas into one service?.

1.2 Basic Idea

In this paper we present a service distribution protocol that mainly answers the first three of the previously mentioned questions. For the time being, we make some simplifying assumptions regarding the other questions which we will describe in the next section. Our protocol is based on service replication over the participant nodes. The proposed protocol uses two mechanisms: The *replication* mechanism is the one that facilitates the placing of a new service "replica" on one of the service clients using the "client-service-interest" measurement. On the other hand, the *hibernation* mechanism is responsible to kill and remove services or replicas from provider nodes based on the "provider-service-carelessness" measurement.

Consider the following very simple scenario: We have an ad hoc network, where just one service, namely a multiplication service is offered. To ensure availability of this service on all nodes, we could simply make a copy of the service available on each node, i.e. replicate the service to every node. Now, regardless of how the network changes over time, the service will be available to each node. Unfortunately, the price to pay is rather high: Replication itself requires some effort and running the service will deplete each node's resources. Thus, we should somehow restrict replication. A number of reasonable factors come to mind: First, their might be nodes that will never use the service. Why should they keep a replica? Second, there may be nodes where lots of replicas are in the neighborhood. Again, why should they keep their own replica? Thus, obtaining replicas should depend on the interest a node has in the service. This is what our replication mechanism does. Since a node's interest in a service may vary over time, it should be possible to get rid of replicas that are no longer wanted/needed. This is, what our hibernation mechanism is used for.

Both client-service-interest and provider-service-carelessness could be a combination of many features like service publishing time, calling frequency, number of connected sessions (load), and the pre-requirements of this service. The client-service-interest defines how much a client is interested to host a specified service; the higher the interest the higher the possibility to receive a replica. The provider-service-carelessness measures how much the provider does not like to host that service any more, the higher the carelessness the lower the possibility to maintain a replica. So the client-service-interest is associated with the replication mechanism and the provider-service-carelessness is associated with the hibernation mechanism.

Service vitality for the network participants is determined by their interests in it. This interest is not supposed to be constant during the whole of the network operation time. So, the dynamic changing interest of the client should be considered by the service placing mechanism. Services with high interests are supposed to traverse and be copied until the interest does not exceed a certain level. Most services will not be vital over the entire operation time, these services will survive and be placed wherever interest is high enough, otherwise it will be hibernated.

Using just the service interest measurements (interest, and carelessness) in the replication mechanism can easily save many exhaustive operations of predicting the affecting topology changes of the network. Actually, the dependence on these prediction processes makes the replication mechanism tightly coupled with some other network specified core components (namely the routing component). By this coupling, replication needs to investigate the lower network layer, which is very difficult. Moreover, the replication mechanism becomes incompatible with any network other than those applying the specified routing protocol.

Currently, by using our network model (section 2), we focus on one network partition at a time in order to evaluate the proposed protocol and to prove our concepts. Work dealing with multiple partitions is ongoing.

1.3 Structure of the Paper

The rest of this paper is organized as follows: In section 2, the network model that we use to evaluate our concepts and the proposed protocol is presented. The proposed protocol with its two mechanisms of replication and hibernation is shown in Section 3. In Section 4, the results of an elaborated simulation are discussed and analyzed. Finally, the related work and conclusions are drawn in Sections 5, and 6 respectively.

2. NETWORK MODEL

This research aims at measuring the effect of the interest based replication/hibernation mechanisms on one partition of the ad hoc network. The assumed network model at certain time is an undirected, unweighted graph G(N,E) where N represents the set of nodes and E is the set of edges. $G_x(N_x, E_x)$ represents one of the network partitions, where: $G(N,E) = G_1(N_1,E_1)\cdots\bigcup G_x (N_x,E_x)\cdots\bigcup G_k(N_k,E_k)$, $N = N_1\cdots\bigcup N_x\cdots\bigcup N_k$, and $E = E_1\cdots\bigcup E_x\cdots\bigcup E_k$. The network is placed in a square shaped area. The network topology is varying according to the node movements. Each node is supposed to cover a fixed radius of radio transmission range R, R is constant over the network operation time. Nodes are uniquely identified.

2.1 Mobility Model

The applied mobility model here is the "Random Waypoint" model [4] in which each node is supposed to move from its current location to a specified target location with a constant speed. The speed is uniformly randomly selected between [1..15] m/s; to achieve higher mobility we avoided a speed of 0 m/s. After reaching its destination, each node is supposed to wait for a pause time. The pause time is uniformly selected from [0..150] seconds. A slight modification was applied by adding a "Mobility index" which is a percentage between [0..100]. A higher mobility index enforces higher speeds and lower pause intervals to be selected. Actually, the upper limit of the speed range is high for the relatively high mobility indices, this add very restricted constraints of mobility. Lower values of mobility index make the upper limit becomes lower.

2.2 Workload Model

We assume that - and we think it is a very realistic assumption - nodes with a heavy workload will seem to be unavailable in our network partition. Workload is categorized into the values {25,50,75,100}%. Each node remains loaded with the same workload ratio for a specified time interval of [1..5] minutes, after finishing that interval, nodes are supposed to pick a new workload ratio. Both workload ratio and intervals are uniform random values.

2.3 Node Availability

The node availability is an artificial measurement that indicates either the node is available in the network partition or not. A node availability $A(n_i)$, where i is the node identifier, is based on both the node speed and workload ratio. The following equation shows how $A(n_i)$ is computed:

$$A(n_i) = \begin{cases} true & \frac{(100-workload)(R)}{speed\sqrt{Area}} \geq \frac{Constant}{(\frac{MaxSpeed}{2})(100-MobilityIndex)} \\ false & otherwise \end{cases}$$

Where: *MaxSpeed* is the maximum allowed speed to be selected by a node and if $A(n_i)$, then node n is in $G_x(N_x, E_x)$.

2.4 Service Model

We assume a rather simplistic service model: The network has just one service at its starting time. Initially, the service is placed on the first available node. Two main assumption are drawn here; (a) All nodes would like to participate in the replication (service distribution enabled nodes). (b) The original service itself is replicable. We are aware that this is a rather strong assumption that will not hold in each and every network. However, there are many realistic application scenarios, e.g., all scenarios involving nodes belonging to one organization, many scenarios involving catastrophe management and the like, all scenarios involving the usage of open source software etc., where this assumption holds. We are also aware, that the assumption will not hold for each and every service. There certainly are services that require specific hardware or software environments etc. which restrict their replicability. Again, however, there are many services that can be replicated to arbitrary devices without a problem.

Also the service "Requirement index" feature is introduced, it represents how high the requirements of the offered service on its host are. Each identical service-replica has a different value of this index, this value is normally distributed about 20% of a general requirement index value. Client nodes are supposed to find the lowest-requirement possible service or replica to communicate.

In this work, about concurrent services(replicas), the research aiming to achieve the higher possible service availability, concurrent identical services -replicas- are going to be investigated in further research. In fact, and for many communities, concurrent managing of identical services is not the highest important issue. For example, in some scientific communities, computations and sharing black-box functions (e.g. arithmetic functions) represent the major type of their offered services and does not require any concurrent managing.

2.5 Calling Model

After placing the service on the initial provider node, the available nodes in the network partition become clients for this service. Each client node maintains a calling rate (number of calls per minute) randomly uniform generated between [0..4] calls/min., a calling interval which is one of {5,10,15} minutes and a pause interval {0,5,10,15} minutes. Both of pause and calling intervals are randomly uniform selected.

3. SERVICE DISTRIBUTION PROTOCOL

The proposed service distribution protocol is mainly based on two mechanisms, as mentioned before, both of these mechanisms are using the service interest to determine either to replicate a service and generate a new replica, or to hibernate a running service from a provider node. Many criteria could be combined together to compute the service interest. For simplicity, the proposed service interest measurement is only based on the calling frequency of the service. For example, if a client node called a specified service for four times in one minute then it will be an interested node to host a replica of that service. In the same fashion, if a provider received just one call in (N,N>=1) minutes, then this provider is going to turn off (hibernate) this service.

3.1 Replication Mechanism

The replication mechanism is triggered by the provider nodes, each provider has to estimate a list of hosting its service/replica by its interesting clients, the interresting clients have to forward their requests to some service provider if they would achieved the a certain replication threshold based on their service/replica calling frequency. The core actions of the replication mechanism are as follows:

- *Restore it from cache*:
  If a client is interested enough to host a replica, it should search first if it had a replica before, if yes it restores it.
- *Find least requirement service*:
  If a client is interested enough to host a replica, then, it should discover a service or replica with the least requirement index.
- *Pass a replica*:
  If a client is interested enough to host a replica, it receives a replica from its provider.
- *Switching to the local service*:
  When a replica is received by a client node (new provider), then the node switches its calling to the local replica.
- *Publish*:
  Allows publishing the new service/replica status.

3.2 Hibernation Mechanism

The hibernation mechanism is also triggered by the provider nodes, each service or replica should be hibernated if its provider loose its interesting to allow this service/replica to run. The core actions of the hibernation mechanism are as follows:

- *Shutdown*:
  Hibernates a local service-replica.
- *Publish*:
  Allows publishing the new service/replica status.
- *Find another service*:
  Finds another identical replica of the called services, if that called service is not found or hibernated.

3.3 Service Caching

A side effect of the proposed protocol is the service caching. Caching data over the network application is a common activity that increases the availability, the new thing here is caching the functionality. By replicating a specified service the client node becomes a provider, after a while, the provider looses his interest and decides to hibernate that service. The hibernation does not mean deleting the service (regarding to the client available resources, profiles, service requirements. . .), the node may store the services/replica in its cache for probable later usage.

4. RESULTS AND DISCUSSIONS

An extensive simulation for the proposed distribution protocol has been done. A varying number of network nodes (from 10 to 100 nodes) with transmission radio range of 25 meter put uniformly together in a square shaped area of $300^2\ meter^2$. The results were obtained from averages of 20 runs, in each run the network operation lifetime was 2 hours. Three main criteria measuring the proposed protocol performance were defined (service availability , prevalence ratio and residence time). These measurements were examined against network size and mobility.

4.1 Performance Analysis

Some common settings in the different runs were fixed. The replication threshold, at which the replication mechanism is triggered, is settled to be four calls per minute. That threshold was chosen to be equal to the maximum allowed calling rate. Also, the hibernation threshold is set to be only one call in five minutes. Two groups of results were obtained here: the first group is obtained from just applying the replication mechanism (shown in 2), the second one is obtained from applying both the replication and hibernation mechanisms. The two groups were estimated against (a) a varying network size (with a fixed mobility index equals 50%) and (b)a varying mobility index (with a fixed network size = 50 nodes).

4.1.1 Service Availability:

The service availability is the ratio between the time that the service was available to the total network lifetime. This ratio is very important to measure how much the service (with its replicas) enhanced the availability. As shown in figure (1:A), the service availability increases as the network size increases, this is intuitive because if there is a larger number of nodes this means more requests - calls - and more interested nodes to host replicas. This leads to the higher the number of nodes, the higher the number of replicas will be , and thus the higher the availability that can be achieved. Starting from 40 nodes, by applying just the replication mechanism, the availability becomes bounded between 90% and 100%. By applying both replication and hibernation mechanisms, the achieved service availability is always less than with just the replication mechanism. This is due to the purpose of the hibernation, but the advantage is that the proposed protocol decreases the utilization on the network links and the service availability doesn't collapse. The average difference between the both resultant service availability was always less than 7.5%.

The effect of the mobility on the service availability is shown in Figure (1:B). The proposed protocol achieves high service availability for lower speeds (from 10% to 50% mobility index), the availability is bounded by [83..100]%. For moderate speeds (from 50% to 70% mobility index) the availability is bounded by [21..83]%, otherwise it becomes very low quickly. In fact, not in all cases, even if there are enough nodes, we can ensure the service availability, especially in very dynamic, less dense ad hoc networks.

4.1.2 Prevalence Ratio:

The ratio between the number of nodes that shares a service or a replica during the network life time to the total network size is identified by the prevalence ratio. In order not to waste the network resources, like link utilization, a low prevalence ratio together with high availability is extremely demanded. As shown in Figure (2:A), the prevalence ratio converges to be less than 20% while the network size increases, this means that the proposed protocol utilization (resultant of the network traversing replica) should be fixed (as a function in the network size).

On the other hand, Figure (2:B), the prevalence ratio is affected dramatically by very high speeds (starting from 70% mobility index). The prevalence ratio decreases slowly between slow and moderate speeds (from 10% to 70% mobility index). The average difference between applying the replication mechanism only and applying the two mechanisms is less than 2

4.1.3 Residence Time:

Residence time measures the average time that a service or a replica stayed running on a hosting node. As shown in Figure (3:A), residence time is not affected by the network size. By applying both replication and hibernation mechanisms one can make the nodes host the replicas for less time. The average difference between both grousp (starting from 30 nodes) of results is 3 minutes. The effect of mobility is shown in Figure (3:B), the residence time decreases when the mobility increases and this is an expected result.

5. RELATED WORK

In [5] S.Dustdar et al. (2007), continuing the research of [6], introduce a set of replication and synchronization algorithms for mobile services. Nodes should be informed about all the other nodes in the network and so the replication mechanism (component) is based on that global view. The original service node moderates the replication process once a partition formation is predicted (using the link quality), a powerful node in the new partition is supposed to be elected to host a replica. In order to keep services synchronized, a client should try in the first to communicate with the primary service copy.

In [7, 8] A. Derhab et al. (2005), are estimating the link quality and partition prediction, using the TORA [9] and some partition detection mechanisms, this research supplies two mechanisms for replication; (a) replication (prepartition formation) and (b) merging (after two partitions merged) mechanisms.

[10] M.Hauspie et al. (2001), other research like [11] goes also on the same concepts of link evaluation of the ad hoc network, a comparison between them is stated in [7]

Generally, excluding our proposed protocol, most of the service replication research for mobile networks, namely ad hoc networks, are based on the mobility prediction and the wireless link quality, that dependency makes the replication components tightly coupled with the lower layers of the network, this type of coupling has clearly a negative effect on the compatibility, because it is requiring presence of some specific routing and mobility prediction protocols and

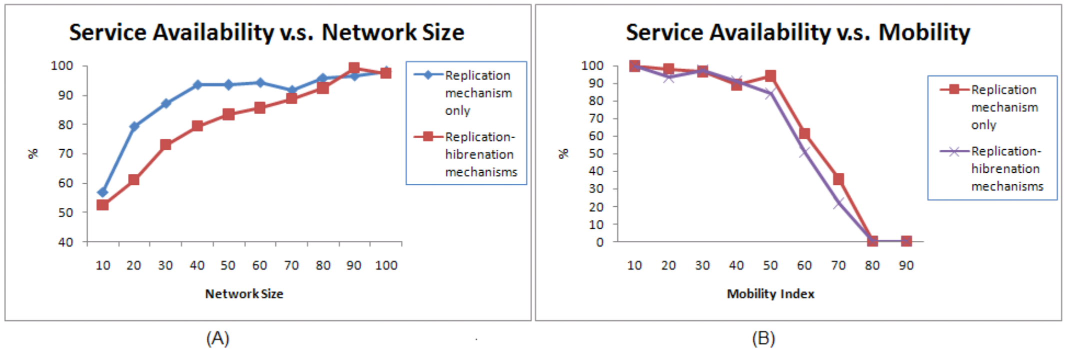

Figure 1: Service Availability

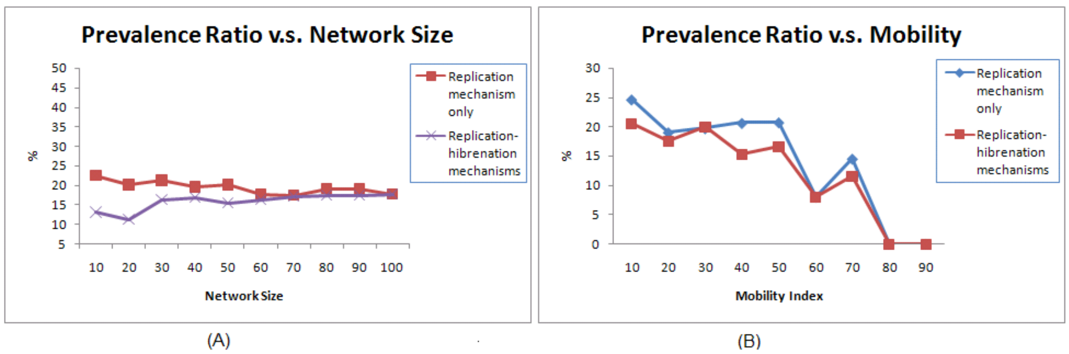

Figure 2: Prevalence Ratio

schemes.

Also, the service which is supposed to be replicated is considered to be a vital for all of the network participants, this implies that all services should be replicated in all network partitions. Instead, our approach offers another concept to weight the importance of the various services in the network. In reality, during the operation time, the importance degree of most of services is not the same for all participants, moreover, it is always changing for each single participant. The client interest may represent the realistic measurement to dominate the replication process.

Finally, according to the taxonomy of data replication in P2P systems in [12], the proposed protocol should be categorized as a "**partial-multimaster optimistic asynchronous**" replication protocol. The partial replication is that replication in which not all of the services will be replicated. Our proposed protocol - till now - is a multimaster replication approach, because the updates are allowed all over the replication sites regardless of synchronization issues, and optimistic means that it assumes no synchronization conflicts will happen. In our ongoing research synchronization, consistency, and merging services are being addressed.

6. CONCLUSION

This paper shows that the proposed service distribution protocol, with its two main mechanisms (replication and hibernation), has very promising results. The main contribution behind our work is to use the service interest based measurements to dominate the service replication in the network. Results of our experiments were categorized into two groups. The notable conclusions here are: (A) For the first group of results: applying the replication mechanism of the proposed protocol can preserve high service availability and reasonable prevalence ratio; (B) For the second group of results: applying both replication and hibernation mechanisms together, which is more restricting and realistic, can preserve a very close performance to the first group.
Also, service caching is shown as a resultant feature of using these proposed mechanisms. What is still remaining requiring more investigation is the concurrency management -synchronization- of the identical replicas, which is what we are already working on.

7. REFERENCES

[1] Mohamed Hamdy and Birgitta König-Ries. Service-orientation in mobile computing - an overview. In *Workshop on Tools and Applications for Mobile Contents (TAMC2006) in Conjunction with the 7th International Conference on Mobile Data Management (MDM'06)*, Nara, Japan, May 2006.

[2] Michael Klein, Birgitta König-Ries, and Michael Müssig. What is needed for semantic service descriptions - a proposal for suitable language constructs. *International Journal on Web and Grid Services 2005*, vol.1:328–364, 2005.

[3] Ulrich Küster and Birgitta König-Ries. Semantic service discovery with DIANE service descriptions. In *Proceedings of the International Workshop on Service Composition and SWS Challenge at the 2007 IEEE/WIC/ACM International Conference on Web*

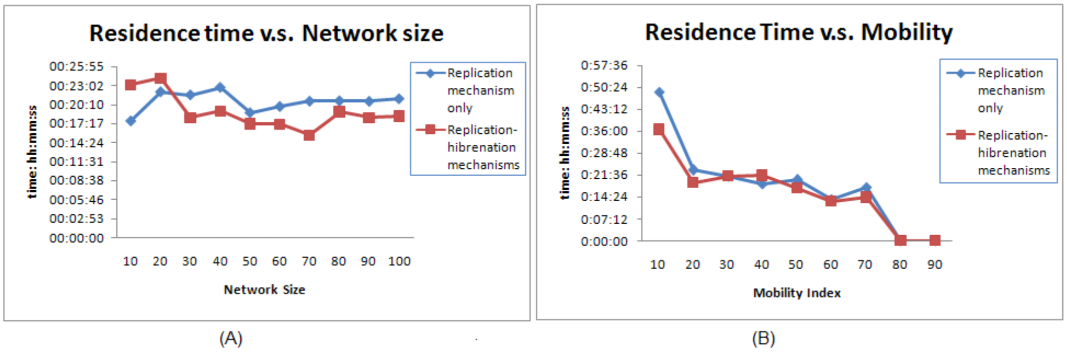

Figure 3: Residence Time

Intelligence (WI 2007), Silicon Valley, USA, November 2007.

[4] Guolong Lin, Guevara Noubir, and Rajmohan Rajaraman. Mobility models for ad hoc network simulation. In *The 23rd Conference of the IEEE Communications Society (INFOCOM 2004)*, Hong Kong, 2004.

[5] Schahram Dustdar and Lukasz Juszczyk. Dynamic replication and synchronization of web services for high availability in mobile ad-hoc networks. *Service Oriented Computing and Applications*, 1:19–33, March 2007.

[6] Lukasz Juszczyk. Replication and synchronization of web services in ad-hoc networks. Master's thesis, Technischen Universität Wien, 2005.

[7] Abdelouahid Derhab, Nadjib Badache, and Abdelmadjid Bouabdallah. A partition prediction algorithm for service replication in mobile ad hoc networks. In *Second Annual Conference on Wireless On-demand Network Systems and Services (WONS 2005)*, Swi, 2005.

[8] Abdelouahid Derhab and Nadjib Badache. A pull-based service replication protocol in mobile ad hoc networks. *European Transactions on Telecommunications*, 18:1–11, 2005.

[9] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM (3)*, pages 1405–1413, 1997.

[10] Michael Hauspie, David Simplot, and Jean Carle. Replication decision algorithm based on link evaluation for services in manet. Technical report, University of Lille, France, 2001.

[11] K. Wang and B. Li. Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks. In *IEEE INFOCOM02.*, New York,USA, 2002.

[12] Vidal Martins, Esther Pacitti, and Patrik Valduriez. Survey on data replication systems in p2p systems, Recearch Report, Inria, Rennes, France, 2006.