Leader Election Modes of the Service Distribution Protocol for 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: Ad hoc networks are characterized by high dynamics in particular with respect to the formation of network partitions. This behavior makes it difficult to (s)elect a suitable leader for a given network partition at any given time. However, such leaders or representatives are needed by many protocols for ad hoc networks. In this paper, we will illustrate the leader election process and its influence on protocol performance using the example of the service distribution protocol. Effects of different modes of leader (s)election on the replica placement (allocation) process are investigated as well as the general service availability and success ratio of the protocol.

1 Introduction

Mobile ad hoc networks (MANETs) are a kind of mobile network in which mobile nodes (hosts) move freely anywhere, join/disjoin the network, and communicate with each other without any centralized administration or help. If two mobile nodes are close enough, a wireless link between them is formed. Due to nodes' movements, the topology of the network is ever-changing. Thus, the network usually consists of a number of distinct network partitions. This partitioning behavior makes some/all of the network resources inaccessible for the network participants. In order for the decentralized network to function, the mobile nodes are supposed to share their resources in a collaborative manner in order to achieve the basic required network functionalities like routing and data transportation. However, not only computing power, memory buffers, link utilization, and data should be considered as resources, but also functionality offered by the nodes. Recently, service orientation has become a popular paradigm for sharing functionalities among connected peers. There, the functionalities to be shared are represented as services. These services are provided by providers to all interested clients. Considering the ad hoc network partitioning, however, access to any given service on the provider sites could not be guaranteed. This, of course, affects deeply the service execution between the mobile nodes and needs to be ameliorated, if service orientation is to be used as a basis for functionality sharing in ad hoc networks. Replication of resources is a classical approach to increase the availability and is successfully being used, e.g., in distributed database management systems (DDBMS), or domain name servers (DNS). Unfortunately, none of the approaches developed for these domains can be directly applied for service replication on ad hoc networks.

What we basically want to achieve is the following behavior: Functionality is offered in the network in the form of replicable services. Examples for such services are, e.g., gateway services to other networks (i.e. Internet and GPS), and ad hoc key distribution and certificate issuing [HKR08b]. For simplicity of presentation, we regard just one service being deployed on the network. We assume that all nodes are basically able to run and cache the service albeit at different costs. Initially, the service is running on just one node. Each time when a new partition is about to be formed, a replication mechanism should ensure that there will be a service replica inside this newly formed partition. The created replica should be placed on a selected node, which should be the optimal place to receive and operate the replica according to some criterion. In distributed systems [SSG02, Tel00, Kau96, Toh01] the question of where to place the new replica is answered in the context of the leader election issue.

The leader election problem in MANETs is difficult to address as the topological and resource properties are always changing. Most of the service replication protocols employ proactive leader selection approaches based on partition prediction approaches. When two or more network partitions are being merged, a leader is elected from the concurrent services in the newly formed partition to transform the multiple independent copies of the service into one synchronized service.

Most hierarchical routing protocols for MANETs like CEDAR [SSB99] and ZBR [DHY03] address the problem of leader election; mostly rich-resourced nodes are selected to host information used to produce the required routes by the nodes inside the clusters of these leaders. The TORA [PC97] routing protocol represented the milestone for the work of [MWV00] to introduce a self stabilizing leader election approach in two modes of single and multiple topological changes. In [DB08], another self-stabilizing leader election algorithm based on TORA is presented for which it has been proven that it ensures that a network partition can converge to a legitimate state within a finite time even if topological changes occur during the convergence time. [DE08] proposes a hierarchical leader election protocol which clusters the network into balanced clusters, builds a spanning tree of the locally selected cluster leaders, and ends with the election of a super-leader. [VDI$^+$03] considers the problem of trust among the participants in many applications and introduces a set of secure leader election algorithms for the two synchronous and asynchronous modes for MANETs.

Even more directly related to our work are service replication protocols for MANETs. [DJ07, Jus05] introduce mechanisms for mobile service replication by the initial service. A node moderates the replication process per predicted partition, replicas in the new formed partitions are supposed to be hosted by a powerful elected node. For consistency, a master node is kept in order to keep services synchronized among the partitions. The replication process is based on the assumption of availability of a global view for the current network status by all of the participants. [MHKS08] proposes a replication approach for mobile service which is modeled as a validated Petri net model for the dependability. A use case is illustrated to show how the approach configuration parameters could be determined and optimized. In [DBB05, DB05] , by estimating the link quality with a partition prediction mechanism based on TORA, the authors supply two mechanisms for pull based replication; (a) replication (pre-partition formation) and (b) merging (after two partitions merged) mechanism. [WL02] presents a group of algorithms that enable service replication and replica placement selection based on partition prediction and server selection mechanisms by both clients and servers using a distributed grouping mechanism at regular service discovery intervals. [HSC01], based on the concepts of link quality (robustness), evaluates the replication decision, without being involved with the velocity of the underlying network or location information. Based on the available information from the routing component, the network partitioning is not considered as the only trigger of the replication process but also required quality requirements like avoiding long routes in the high density network. As an important result, this could involve the process of replica placement and the related leader election involved in the replication decision which means that finding a good place (node) to be elected could be a trigger for the replication process. A survey on the required quality parameters which affect efficiency of the service execution over a MANET is presented in [RT06]. These parameters are defined and evaluated to a typical efficiency matrix.

In most of the approaches described above, both partition prediction and leader election require a global view of the network and/or querying the lower network layers. The first is rather unrealistic to achieve in a MANET, the latter has several drawbacks: (a) coupling to the lower layers of the network makes these approaches network architecture dependent, (b) querying specific lower layers implies a coupling to specific network components and their architecture and (c) enquiring and processing of the lower network layers is expensive. Additionally, most of these approaches assume the same importance for the service over all of the network operation time and do not take into consideration that the importance of the service is variant over time and may be different for different clients.

In order to avoid the previously mentioned deficiencies of these approaches, in our prior work, we have introduced the service distribution protocol (SDP) [HKR08c, HKR08a]. This protocol operates on the application layer, only. It utilizes the popularity of a specified service, without any partitioning behavior prediction as an indicator for the need for replication. When common interest (*Gross Interest*) for a specified service by a certain group of clients reaches a threshold, the interesting service is replicated. On the other hand those services having lower interest will be shut down (hibernated). The tightly coupled two mechanism of replication and hibernation guarantee higher service availability and better performance for service execution over MANETs, see [HKR08b] for details. Based only on the amount of interest in a service, new providers will be chosen to host replicas of this service. This choosing of providers amounts to a sort of leader election. Since the network topology is ever-changing, these new providers are likely to travel to other partitions and prevail their replicas. In a given network partition, the hibernation mechanism dominates the leader election process. In case of concurrently running replicas, those replicas that are cheapest (according to some measure) from the clients' point of view, will get more requests (interest by the clients in this partition). After a while, the hibernation mechanism will shut down all of the not interesting replicas and only one replica will continue. Again, this is implicitly a leader election process.

In this paper, we show different modes for leader election of the SDP and investigate how the general performance of this protocol is affected by the election process. Especially, we are interested in the replica allocation (placement) process. For that purpose, two use scenarios are drawn which differ in the service requesting behavior. An extensive simulation approach was employed to investigate the effects for these different use cases. The rest of the paper is organized as follows: An overview of SDP is presented in Section 2. Concepts and consideration about the process of leader election by the protocol are presented in Section 3. The used service model and the relating calling model and how they shape the service gross interest is discussed. Two very different usage scenarios are drawn in Section 5. The used network model is proposed in Section 6. Finally, the simulation and results are discussed in Section 7.

2 The Service Distribution Protocol for MANETs

The main measurement behind the replication decision in the service distribution protocol is the service interest. Interest of a group of clients regarding a specified service could be measured based on many criteria. For example, the number of requests gained by a service provider during a certain time interval, the size of transmitted data from/to a service provider, and/or the call-length could be indicators for the interest in a service at a given time. The protocol employs the frequency of calls (requests) in a certain interval of time as the measure of interest in the service. If a client achieves a certain number of calls per given time interval, it will ask for its own replica of the service. On the other hand, if a service provider does not receive at least a given number of calls (requests) during a certain interval it will shut down (hibernate) its service. The hibernated service may be cached for later activation (restoring). Of course, both the replication and caching decisions are built on the availability of resources on the host side. In this work, we make the assumption that sufficient resources are available on the mobile node to host the service. Performing these processes of replication/hibernation/caching/restoring ensures service prevalence over all network partitions. The protocol consists of three main mechanisms: *Replication*, *Hibernation* and *Caching*:

Replication Mechanism: Based on an optimistic replication model [MPV06], the functionalities of this mechanism are shared between both the service provider and the service clients as follows:
On the provider side:

- Pass a replica: Allows the provider to pass a replica to a new (proposed) provider and keep track of information about this replication.
- Check Correctness: Enables the service provider to check the correctness of a new replica placement as discussed in the following sections.
- Publish: Enables a provider to publish the new status of a service (active/hibernated).

On the client side:

- Find least cost service: The service cost is an index labeling each of the service/replicas, clients are supposed to find the least cost service and call it. This index represents a service request-cost which should come out of a combination of criteria.

- Count and ask: For simplicity, the frequency of calls (request) has been selected to represent the interest of a client in a specific service. This functionality enables a client to count the number of calls regarding a specific service and if this number achieves a certain value (replication threshold), it allows the client to ask for its own replica of that service.
- Switch into a provider: After receiving a replica, this mechanism enables the client to activate and publish the status of its new replica.

Hibernation Mechanism: This is triggered by a service provider which detected a loss of interest regarding his active service. It includes the following functionalities:

- Count and decide: For a given time interval, the number of requests to the provider by its group of clients is counted. If this number is not sufficient to let this service be hibernated.
- Cache it: Enable the service provider to cache its hibernated service based on its available resources.
- Publish: Publishes the new hibernated status of the deactivated service/replica.

Caching Mechanism: This includes the required functionalities to manage the cached service/replicas by an ex-provider, it includes the following functionalities:

- Restore a service: Restores a hibernated service/replica from the cache when the current node (as a client) achieves the replication threshold (instead of activating the replication mechanism) or based on the unavailability of the requested service (which is also cached) in the current partition. In cases where the current node restored a replica while there exist another active replica on another node (provider) site, after a while and based on the type of the used hibernation mode (mentioned in Section 3, a selection process takes place and one of these replicas is continued and the rest (if there is more than one active replicas in the same partition) will be hibernated.
- Publish: Publishes the status of the restored service/replica.

3 Leader Election Modes of the Service Distribution Protocol

At a certain time the service may be replicated over more than one hosting mobile node within the same network partition. This produces a set of equivalent concurrently running replicas. Based on the hibernation mechanism, after a given time interval (hibernation threshold), at each of the concurrent provider sites, a check for the gained interest will take place. Since all of the replicas' clients are oriented to select the least-cost per request replica in this network partition, just one replica will gain enough interest and will continue. The rest will be hibernated and cached. The main parameter in this selection process besides the request cost (which steers the common interest to a specific service) is the length of the given time interval to perform the hibernation checks and decisions. This interval is supplied by the hibernation threshold. The hibernation threshold is formed as (number-of-calls:hibernation-time-interval). The length of the hibernation time interval determines the leader election modes of the service distribution protocol: Longer lengths of the hibernation check interval leads to more simultaneously running replicas inside a given partition and vice versa. In the following, we will investigate different settings for this time interval and the implications they have on protocol performance in more detail.

3.1 Long-election Mode of Hibernation

In this mode, the hibernation time interval of the hibernation threshold is fixed determined to be either one or the multiple of the calling (requesting) time unit. In this work, the calling time unit is a minute. This mode enables replicas to remain activated for a longer time to be evaluated. We assume the long-election to be helpful for very high mobility networks, in which the frequency of network partition formation is very high. In the work of [HKR08b], effects of different (multiplies of this long mode hibernation check time intervals) are applied and analyzed.

3.2 Short-election Mode of Hibernation

The basic idea in this mode is to evaluate the initial interest regarding a given (just published) service/replica as soon as possible. This means evaluation should take place after the minimum possible time required by the service provider to receive, activate, publish, and wait for a slice time for calls. Once this very tight time interval ends, the hibernation test takes place. If the service received at least a given number of calls, this means the service will continue, any other active service will be hibernated. This election mode is very conservative regarding the number of concurrent replicas inside a network partition.

4 Service Model and Gross Interest

In this research, the proposed deployed service is an abstract service. We assume that the pre-required resources for this service are available by all of the mobile nodes, this means, all of the network nodes can participate in the replication process. A ”Requirement index” is describing each of the replicas. The requirement index represents both how much resources are supposed to be available by a client node to be eligible to ask for its own replica from a provider node and the cost of the request of this service. In reality, although all services are equivalent, not all available concurrent services should be provided with the same cost. All candidate client nodes are supposed to search for a service with the minimum available requirement index in the network partition.

4.1 Calling Model

The calling model describes the behavior of the service requesting by the clients. As in [HKR08c, HKR08a], The frequency of calls (requests) and the time of call length are the main parameters indicating the proposed calling model. For simplicity, the proposed calling model is based on only the frequency of calls. Initially, all of the created nodes seek for the initial (original) service provider node, only those nodes with at least one feasible path to the provider node are assumed to start evaluating the service calling and be involved in the related replication-hibernation-caching processes. After a while, service/replicas prevalence through the network is supposed to cover as much as possible of the ad hoc formed partitions [HKR08a]. A variant calling rates are maintained by each node, the calling rate is generated between [0..Max-Rate] calls per minute, The "Max-Rate" indicates the number of client's groups maintained by the network, for example if Max-Rate = 3 calls per minutes then there are four client's groups (0 calls/minutes, 1 calls/minutes, 2 calls/minutes, and 3 calls/minutes) are supposed to be presented in the network. The calling rate is supposed to be constant during a calling period , the calling period is supposed to be a "Calling-Length" and its multiplications between [1..Max-Calling-Units] in minutes, and after a pause time equals to a "Pause-Length" or its multiplications [0..Max-Pause-Units[in minutes the node is supposed to select another calling rate and so on. All of calling rate, calling period, and pause period are uniformly randomly selected. The values of the {Calling-Length,Max-Calling-Units} and {Pause-Length, Max-Pause-Unites} are given and evaluated as in the following sections.

4.2 Gross interest

Based on the calling behavior of a specific set of the service clients, the service will be replicated and prevail over the different network partitions. The different settings to the proposed calling model will generate different types of calling patterns and leads to a resultant gross interest regarding a specific service at certain time. Richer Gross interest indicated by higher calling frequencies with shorter pause intervals will require shorter time to produce better service prevalence over the network, on the other hand, poorer gross interest will generate the opposite situation.

5 Scenarios

In this section, two extremely different, rich and poor gross interest, scenarios are developed to reflect the concept of the gross interest, as mentioned in Section 4.2, moreover, a discussion about quantifying the difference between these two scenarios is presented in the following subsections.

Scenario 1: Rich Interest Scenario: An active calling behavior with short periods of pausing is modeled in this scenario. The mobile node has a *calling-length* equal to 10 minutes and the *max-calling-units* equal to 5 units. This means that the calling period could be either 10 minutes, 20 minutes ..., or 50 minutes. On the other hand, each of the mobile nodes is assumed not to call (wait) for a pause period. The pause period settings are as follows; the *pause-length* equals to 3 minutes and the *max-pause-units* equals to 5 units. So, the pause periods will be 0 minutes, 3 minutes ..., or 12 minutes [HKR08b].

Scenario 2: Poor Interest Scenario: The second scenario presents a lazy calling behavior with long periods of pausing. The *calling-length* is set to 2 minutes, the *max-calling-units* is set to 3 which means that the clients have 2 minutes, 4 minutes, or 6 minutes of calling periods. The *pause-length* equals to 10 minutes and the *max-pause-units* equals to 5 units, which means that the pause periods should be 0 minutes, 10 minutes, ..., or 40 minutes [HKR08b].

Counting Active Clients: Active clients are those clients which are being in a calling period as in Section 4.1. The expected number of the active clients in at certain time in a given network partition is related to the calling model. At certain time (t), the expected number of active clients (Ex) [HKR08b] ,

$$Ex = Pz_i . \frac{Ex_n}{Ex_n + R.Exm} \text{ ... (1)}$$

where (Ex_n) is the expected value of the Max-Calling-Units, (Ex_m) is the expected value of Max-Pause-Unites, and R is the ratio between a single "Pause-Length" to a single "Calling-Length" (see Section 4.1), and (Pz_i) is the "i^{th}" network partition size. Based on computing the (Ex) value, two extremely different scenarios are proposed to be used in this work evaluation.

Quantify the Gross Interest Difference Between Scenarios: Assume a partition with size equals to 25 mobile host, this leads to, for scenario 1, about 20 mobile host could be active client a time, which is a really high number of clients, on the other hand, for scenario 2, just about 4.1 clients will be active. As one of our experimental results, we computed the weighted sum (weight: size of a partition to the network size) of our resultant network partitions during our evaluations using the proposed specifications of the introduced mobility model, we found that for a network size equals to 70 node, the expected partition size will be about 17 node. Again, For scenario 1, there will be about 14 active client, and for scenario 2 there will be just about 2.8 active client nodes [HKR08b].

6 Network Model

The used network in this work is modeled at certain time t as an undirected unweighted graph $G(N, E)$, where N are the mobile nodes and E the groups of formed links between the participant nodes. The network is formed of a number of disjoint network partitions. $G_x(N_x, E_x)$ presents a specified network partition x , where $G(N, E) = G_1(N_1, E_1) \ldots \bigcup G_x(N_x, E_x) \ldots \bigcup G_k(N_k, E_k)$, $N = N_1 \ldots \bigcup N_x \ldots \bigcup N_k$, and $E = E_1 \ldots \bigcup E_x \ldots \bigcup E_k$. The wireless transmission range (R) of the mobile nodes is fixed during the whole operation time (75 meter). The network is placed in a square area

($600 by 600 m^2$) [HKR08a]. A wireless like if formed if and only if two neighbor mobile nodes are located inside the transmission range of each other. The random way point mobility model [LNR04] is the movement model of the mobile nodes. Each mobile node picks a constant speed between $[0, V_{max}] m/s$ (applied:[0..6]m/s), then chooses a random location to go, after reaching its desire location, it is supposed to wait for a pause time between $[0, P_{max}]$ (applied:[0..15]minutes) and so on.

7 Simulation

A detailed simulation has been elaborated by this work, a rather sophisticated simulator has been built for that using C++[1]. The general performance parameters namely service availability and success ratio have been discussed. The main parameters describing the quality of the replica placement process is presented and evaluated in this section.

7.1 Performance Parameters

Service Availability indicates the ratio between the summations of the whole time intervals that the service was active some where in the network, but it does not indicate if the service was accessible or not for the whole of the network participants. the importance of this parameters is that it reflects the presence of the service anywhere to be replicated and prevailed over the whole network partitions.

Success Ratio indicates the ratio between the number of succeeded request to the service/replicas overall the network. It presents the satisfaction of the clients requests by the current service distribution in the network.

Correctness of a Replica placement: One of the main issues in this research, as in [HKR08b], is to measure the correctness of the replica placement ratio which indicates the available number of the concurrent active replicas inside a given partition at certain time, moreover, this research aims to reflect also the effects of the proposed leader election modes on the service distribution protocol replica placement process. [HKR08a] suggested a linear replica allocation correctness. [HKR08b] introduced a detailed discussion on the linear computation method of the allocation correctness and concluded to another computation method (WRCR: Weighted Rational Correctness Ratio) for the ratio based on both of the weight of a given network partition to the whole network size and the number of the concurrent active replica inside it on a rational basis. The following equation shows the RCR_t (Rational Correctness Ratio) relation at given time (t) for thei^{th} network partition (P_i) :

$$RCR_t(P_i) = \begin{cases} 0 & R_i = 0 \\ 1 & R_i in \{1, 2\} \\ \frac{3}{R_i} \cdot \frac{Pz_i - R_i}{Pz - 2} & R_i > 2 \end{cases} \quad \text{... (2)}$$

[1]Microsoft Visual C++, version 8.50727

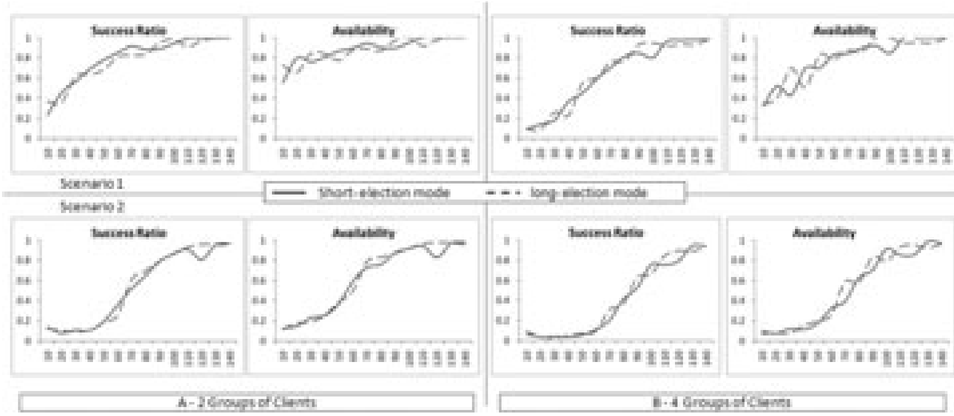

Figure 1: Success ratio and service availability (0-1) against the network size (10-140 node) for both of scenario 1 and 2 for (A)2 client groups and (B)4 client groups

where R_i is the number of the concurrent active replica inside (P_i),and (Pz_i) is the partition size of (P_i) and WRCR is as follows:

$$WRCR_t = \Sigma_{i=0}^{n}(\frac{Pz_i}{size(N)}.RCR_t(P_i)) \text{ ... (3)}$$

where n is the number of the partitions at given time (t) , and N is the network as defined in Section 6. In this research the WCRC is used as a measurement the correctness of the replica allocation process.

Typical Partition of the Network: The typical partition represent the most weighted partition regarding the size in the network, which can provide a good network partition sample. The Typical partition is a feature of a given network mobility model settings. The typical partition is used as in this work in order to analyze the number of replicas inside it. In fact it is a kind of computed (virtual) partition rather a physical one, it is computed as $Typ.Part_N$:

$$Typ.Part_N = \Sigma_{i=1}^{n}(Pz_i.\frac{Pz_i}{size(N)}) \text{ ... (4)}$$

7.2 Configurations and Settings

The simulation run time for each experiment was set to be 2 hours, the network size is varying from 10 to 140 mobile hosts. Results come out of the average of 20 times runs. The number of client groups is either 2 groups(maximum call rate 1 call / minute) or 4 groups (maximum call rate 3 call / minute). The replication threshold for a client is achieving number of calls (requests) equals to the maximum call rate in a given minute (Max-call-rate:1), the hibernation threshold for a provider is gaining less than one call(request) in a

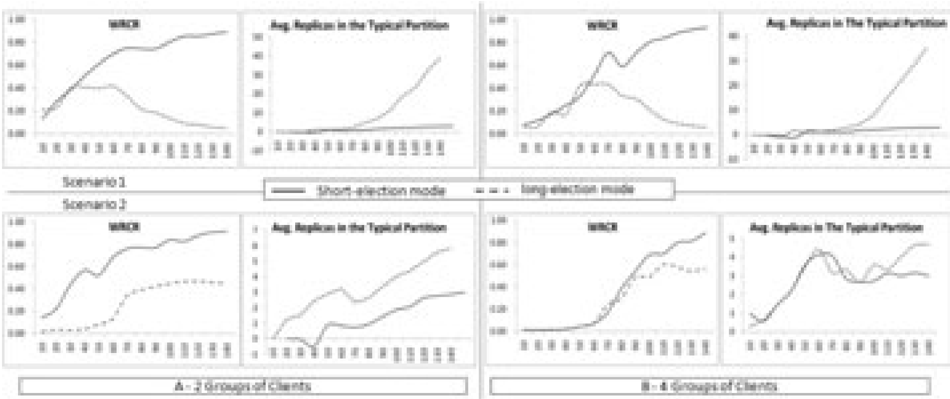

Figure 2: WRCR ratio (0-1) and Avg. number of replicas in the typical partition against the network size (10-140 node) for both of scenario 1 and 2 for (A)2 client groups and (B)4 client groups

given length of minutes, in this research, regarding the previously introduced two election modes, the hibernation threshold will be either at least one call (request) per a minute (1call:1minute, indicates the long election mode) or one call per the minimum waiting time interval shown in subsections (3.1,3.2), in our simulation we consider this required time interval to be 1 second [HKR08b].

7.3 Results

Figure 1 shows the general performance of the service distribution protocol, for both rich and poor interest scenarios, the service availability increases as the network becomes mode dense, this leads to the same behavior on the success ratio since a higher success ratio (indicates the number of succeeded requests in the network) is tightly coupled to a higher service availability. No difference in the results come from the different election modes, in fact this can be concluded as a positive point for the short election mode: Shorter hibernation check intervals do not affect the high service availability and success ratio. As the number of the client groups increases both the service availability and accordingly the success ratio decreases, this dues to the smaller set of the interesting (in hosting replicas) clients in the higher number of the client groups. Table 1 shows the values of the average service availability and success ratio with the relative observed standard deviation for both scenarios, hibernation modes and number of client groups.

Figure 2 shows both the observed WRCR ratio and the average number of concurrent replicas in the typical partition in the network regarding varying the network sizes. For Scenario 1 and for 2 client groups, as the network increases, the WRCR ratio increases too, starting from 70 node network size WRCR is about/more than (0.80), as the client groups increases, the WRCR ratio growing rate becomes slower. This is due to the lack of service prevalence in the different network partitions. The number of partitions with

			SuccessRatio		Availability	
			Avg.	Avg.StD	Avg.	Avg.StD
Scenario1	2 Groups	SM	0.78	0.20	0.88	0.19
		LM	0.81	0.15	0.90	0.16
	4Groups	SM	0.64	0.17	0.79	0.23
		LM	0.64	0.17	0.80	0.24
Scenario2	2 Groups	SM	0.53	0.14	0.62	0.16
		LM	0.55	0.11	0.64	0.13
	4Groups	SM	0.41	0.13	0.52	0.17
		LM	0.42	0.12	0.53	0.17

Table 1: Average (Avg.) success ratio and service availability with the average value of the observed standard deviation (Avg.Std.) for both Short and long election modes (SM, LM) for both scenarios and client groups

no service coverages is increased because of the lower number of clients interesting to host a replica. The same behavior is presented by the curves of the short election mode for Scenario 1. On the other hand, in the long election mode, for both numbers of client groups, the observed WRCR ratio starts increasing with the increasing network sizes til an interval of network sizes (about [30,60], and [50,70] nodes for (2,4) client groups respectively) then the WRCR ratio starts decreasing as the network size increases. This behavior could be interpreted by the curves of the average replicas in the typical partition. For 2 client groups, the average number of concurrent running replicas is slightly growing til about network size of 70 nodes (less than 5 replicas in the typical partition). Then it grows quickly to which directly affects the observed WRCR ratio. In the 4 clients groups the WRCR collapses slower, because the average number of concurrent running replicas is less then in the 2 groups case. In fact, in the high density network (large network sizes), the typical partition size is mostly about the network size [HKR08b], and in cases of high gross interest and lower replication threshold, as in Scenario 1 with 2 client groups, the number of the interested clients to host replicas is very high and produces a higher number of concurrent replicas. In Scenario 2, in cases of poor gross interest, the protocol conserves higher WRCR ratio as the network size increases for both of the hibernation modes, increasing the number of the client groups affects negatively the growth of the WRCR ratio. Generally, applying the short hibernation election mode decreases the number of the concurrently running replicas in all scenarios and client groups (about 3 replicas at most at a time). This result is pretty good because, by applying such an election mode, we can ensure a very limited number of replicas inside the network (typical) partition at a time.

8 Conclusions

The main contribution of this work is to highlight how the service distribution protocol can converge to the optimum replica distribution situation during the life time of a mobile ad hoc network based on the proposed leader election modes. As mentioned, the protocol success and unique key is using just the available information about the service use of both service provider and clients in a service oriented environment. By utilizing these information and the service prevalence by mobility and replication actions, the protocol ensures higher service availability and querying success ratio although the partitioning behavior of the ad hoc network. This paper provides a solution to how the leader election process could be met by the service distribution protocol during a continuous election process based on variant service interest generated by variant gross interest regarding a specific service. Two time based leader election modes, namely short and long election modes, are introduced for the service distribution protocol for MANETs. The two modes have been evaluated on two extremely different use scenarios which have been shown and quantified in Section 5. Then they have been evaluated using a detailed simulation. Results show that the short election mode can ensure a very limited number of concurrently running replicas from both the whole network and a network partition point of view. Although the long mode election can achieve high performance metrics regarding service availability and success ratio, it produces a relatively high number of concurrent replicas in the network. On the other hand, the short election mode can also achieve a similar service availability and success ratio with relatively constant number of concurrent replicas inside the network and its partitions. While these results were obtained for the SDP, we believe that they also provide some guidance for similar decisions in other protocols - at the very least they show the need for a careful examination of the choice of time interval for decision making in leader election.

References

[DB05] Abdelouahid Derhab and Nadjib Badache. A pull-based service replication protocol in mobile ad hoc networks. *European Transactions on Telecommunications*, 18:1–11, 2005.

[DB08] Abdelouahid Derhab and Nadjib Badache. A Self-Stabilizing Leader Election Algorithm in Highly Dynamic Ad Hoc Mobile Networks. *IEEE Transactions on Parallel Distributed Systems*, 19(7):926–939, 2008.

[DBB05] 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.

[DE08] Orhan Dagdeviren and Kayhan Erciyes. A Hierarchical Leader Election Protocol for Mobile Ad Hoc Networks. In *8th International Conference on Computational Science (ICCS 2008)*, pages 509–518, Kraków, Poland, 2008.

[DHY03] H. Du, H. Hassanein, and C. Yeh. Zone-based routing protocol for high-mobility MANET. In *Canadian Conference on Electrical and Computer Engineering*, 2003.

[DJ07] 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.

[HKR08a] Mohamed Hamdy and Birgitta König-Ries. An Extended Analysis of an Interest-Based Service Distribution Protocol for Mobile Ad-hoc Networks. In *Proc. of the International Conference on Wireless Information Networks and Systems (WINSYS 2008)*, Porto, Portugal, 2008.

[HKR08b] Mohamed Hamdy and Birgitta König-Ries. Service Availability, Success Ratio, Prevalence, Replica Allocation Correctness, Replication Degree, and Effects of Different Replication/Hibernation Behavior Effects of the Service Distribution Protocol for Mobile Ad Hoc Networks -A Detailed Study-. Technical report, Institute of Computer Science, Friedrich-Schiller-University Jena, Germany, December 2008.

[HKR08c] Mohamed Hamdy and Birgitta König-Ries. A Service Distribution Protocol For Mobile Ad Hoc Networks. In *the proceedings of the International Conference on Pervasive Services (ICPS 08)*, Sorrento, Italy, 2008.

[HSC01] 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.

[Jus05] Lukasz Juszczyk. Replication and Synchronization of Web Services in Ad-hoc Networks. Master's thesis, Technischen Universität Wien, 2005.

[Kau96] Morgan Kaufmann. *Distributed Algorithms*. Nancy A. Lynch, 1996.

[LNR04] 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)*, 2004. College of Computer & Information Science Northeastern University Boston.

[MHKS08] Erling Matthiesen, Ossama Hamouda, Mohamed Kaaniche, and Hans-Peter Schwefel. Dependability Evaluation of a Replication Service for Mobile Applications in Dynamic Ad-Hoc Networks. *Lecture Notes in Computer Science*, 5017/2008:171–186, 2008.

[MPV06] Vidal Martins, Esther Pacitti, and Patrik Valduriez. Survey on data replication systems in P2P systems. Technical report, INRIA, Rennes, France, 2006.

[MWV00] N. Malpani, J. L. Welch, and N. Vaidya. Leader Election Algorithms for Mobile Ad Hoc Network. In *Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 96–103, 2000.

[PC97] Vincent D. Park and M. Scott Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *INFOCOM (3)*, pages 1405–1413, 1997.

[RT06] Pablo Rossi and Zahir Tari. Software Metrics for the Efficient Execution of Mobile Services. In *The Processdings of a workshop on Emerging Web Services Technology (WEWST'06), in conjunction with 4th European Conference on Web Services (ECOWS'06)*, 2006.

[SSB99] Raghupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan. CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 17:1454–1465, 1999.

[SSG02] S. Panchapakesan Shanti S. Gupta. *Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations (Classics in Applied Mathematics)*. Society for Industrial Mathematics, 2002.

[Tel00] Gerard Tel. *Introduction to Distributed Systems*. Cambridge University Press, 2000.

[Toh01] C-K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall Publishers, 2001.

[VDI+03] Sudarshan Vasudevan, Brian Decleene, Neil Immerman, Jim Kurose, and Don Towsley. Leader Election Algorithms for Wireless Ad Hoc Networks. In *in Proceedings DARPA Information Survivability Conference and Exposition*, pages 22–24, 2003.

[WL02] K. Wang and B. Li. Efficient and guaranteed service coverage in partitionable mobile ad hoc networks. In *IEEE INFOCOM02*, New York,USA, 2002.