JENAER SCHRIFTEN

ZUR

MATHEMATIK UND INFORMATIK

Eingang: 01.12.2008 Math/Inf/08/08 Als Manuskript gedruckt

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-

Mohamed Hamdy and Birgitta König-Ries
{hamdy|koenig}@informatik.uni-jena.de
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-

Mohamed Hamdy and Birgitta König-Ries
{hamdy|koenig}@informatik.uni-jena.de

Technical Report Nr.: Math/Inf/08/08

Institute of Computer Science
Friedrich-Schiller-University Jena

December, 2008.
Jena, Germany.
Abstract

The service distribution protocol for mobile networks is an interest-based service replication protocol based on two complementary mechanisms namely replication and hibernation. It allows for the self-organized propagation of service copies (replicas) in an ad hoc network and thus serves to overcome the challenging characteristics of ad hoc networks in particular the frequent occurrence of network partitioning. It increases the service accessibility over time based on the interest of a set of clients with regard to that service. In [1, 2] it was shown that the protocol indeed reaches high service availability in all network partitions. In this report, we investigate a global view of the protocol performance with respect to the related important parameters. Since the protocol utilizes periodically the service popularity (interest) during its replication/hibernation processes, we introduce and discuss a quantifying criterion for the interest in the given service which is its gross interest at a certain time. Two extremely different (rich and poor) gross interest scenarios will be evaluated. The correctness of the replica placement (allocation) process is modeled in different ways depending on the resultant number of replicas inside a given partition. Moreover, we extend our analysis trying to find out what the number of produced replicas inside a specific network partition will be. For that, we introduce the concept of "important partition". The simulation results we show provide a detailed view on the performance of the service distribution protocol and how it behaves against very different and challenging constraints and settings.
Contents

1 Introduction 4

2 Overview of the Service Distribution Protocol for Mobile Services 4
   2.1 Replication Mechanism ........................................... 5
   2.2 Hibernation Mechanism ........................................... 5
   2.3 Service Caching Mechanism ...................................... 6

3 Network Model 6
   3.1 Mobility Model ................................................... 6
   3.2 Calling Model .................................................... 6
   3.3 Service Model .................................................... 7

4 Gross Interest and the Proposed Scenarios 7
   4.1 Active Clients .................................................... 7
   4.2 Scenario 1: Rich Interest Scenario .............................. 7
   4.3 Scenario 2: Poor Interest Scenario ............................... 7
   4.4 Quantifying the Gross Interest Difference Between Scenarios . . . . 7

5 Allocation Correctness 8

6 Simulation 9
   6.1 Performance Parameters and Settings .......................... 9
   6.2 Primary Evaluation .............................................. 10
   6.3 Alternative Allocation Correctness Computation Methods ......... 12
   6.4 Further Evaluation ............................................. 13
   6.5 The Important Partition ........................................ 14

7 Related Work 16

8 Conclusions 17

List of Figures

1 Service availability and success ratio against network size for Scenario 1 and 2 .... 8
2 Different allocation correctness computations against network size for Scenario 1 .... 9
3 Different allocation correctness computations against network size for Scenario 2 .... 10
4 Number of replicas in the Important Partition against network size for Scenario 1 and 2 .... 11
5 Service prevalence and residence time against network size for Scenario 1 and 2 .... 12
6 Replication degree of both numbers of client groups for Scenario 1 and 2 .............. 13
7 Linear allocation correctness against network size for both Scenario 1 and 2 .......... 14
8 Size of the Important Partition and the number of partitions against the network size ... 16

List of Tables

1 Average (Avg.) success ratios and availability with the related standard deviation (Avg.StD.) against different hibernation thresholds ................................................. 11
2 Average (Avg.) residence time, prevalence, and replication degree with the related standard deviation (Avg.StD.) against different hibernation threshold ......................... 12
3 Average number of the computed concurrent active replicas inside the important partition 15
4 Different (Avg.) allocation correctness ratios with related standard deviation (Avg.StD.) .... 15
1 Introduction

Wireless mobile ad hoc networks (MANETs) have been an active topic for research for the last few years. Such networks consist of a number of mobile hosts, where each of these hosts is free to move anywhere and covers a certain transmission range around itself. A wireless link is formed between two nodes, if two mobile nodes are near enough to be in the transmission range of each other. Since MANETs are infrastructure-less, beside being a network client, each mobile host is supposed to work in a collaborative manner as a router for the rest of the network participants without any centralized help or administration. Ad hoc mobile networks have many challenging features like a poor wireless link quality, low availability and the ever-changing topology. Since the mobile hosts are usually moving, the network typically consists of an ever-changing number of network partitions. This makes not all of the participants/resources available for each other.

On the other hand, sharing resources is vital for this kind of collaborative network. Resources are not only computing power, memory buffers, or link utilization but also data and functionality. In general, one of the most popular paradigms for functionality sharing is service orientation in which functionalities are represented as services. To apply service orientation in ad hoc networks the service availability needs to be ensured. Regarding the continuous partitioning of the network, any mobile host’s access to the service is not guaranteed.

In cases in which availability is vital, replication of resources is a popular solution applied in many systems like DBMS, RAID, and DNS. Unfortunately, none of the replication methods known from these areas take the constraints of MANETs into consideration. Service replication in ad hoc networks requires a continuous replication decision in order to keep services as available as possible for all participants despite ever-changing topologies and statuses of the network.

The service distribution protocol (SDP) for mobile networks applies an optimistic replication mechanism for a specified service in order to keep it as available as possible over the changing network partitions. Replication decisions are based on the popularity of the service. Interesting services will be replicated and thus made available in the whole network, while those services obtaining lower interest will be shut down. Besides the replication mechanism the protocol triggers also the hibernation mechanism which is responsible for shutting down the unpopular (uninteresting) services everywhere in the network. Popular (very interesting) services may loose their popularity over time, new (equivalent but cheaper) services may gain a higher interest and so on. The tightly coupled two mechanisms of replication and hibernation guarantee good performance for service execution and accessibility over ad hoc networks.

In this report the effects of different settings for the replication/hibernation mechanisms are studied regarding the general performance of SDP. As a key function of a successful service replication in ad hoc networks, the replica placement process should answer the question of where to put the resultant replica. The better replication approach should show a more correct replica placement process. This report aims also to highlight different ways to quantify the correctness of the replica placement process and to present and quantify the replica placement (allocation) correctness and the related effects by the hibernation mechanism of the SDP.

The rest of this report is organized as follows: the basic idea of SDP and the related mechanisms are summarized in Section 2. The network used in this work is modeled in Section 3. The service requesting model (calling model), the gross interest, and two related scenarios are presented in Section 4. The correctness of the related replica placement process is discussed in Section 5. Then our detailed simulation and results are shown in Section 6. Finally, Section 7 presents the related work.

2 Overview of the Service Distribution Protocol for Mobile Services

The basic idea of this protocol is that the common interest of a group of clients determines the replication process of a service. The most interesting service (of a set of equivalents) should be replicated when needed in the network, others should be hibernated, those hibernated services could be reactivated after a while.
Initially, when a single service is published, a set of clients starts evaluating this service. After a while, some of these clients show enough interest to receive their own replicas of the service. After copying the service to those clients, a number of replicas will run simultaneously. Either these new replicas will leave their current partition joining new partitions and let the service prevail into other network partitions or they stay in the same partition. In this work, the term “replicas” refers to the set of an original service and its replicas. As clients find one most interesting service and direct their requests to it, the other replicas will not be interesting enough to continue and will therefore be hibernated. These differences in interest may stem from different cost associated with the usage of a specific replica, e.g., communication cost, delay, or expected reliability.

The service popularity or interest is indicated by some value which can be used to rank the alternative replicas. As mentioned above, this value could be a vector of a set of many quality criteria. Also, depending on circumstances of the network operations and the service meaning itself, the most interesting replica could be a set of equivalent replicas taking into consideration many parameters like the network size, service throughput and volume of requests.

Replication, hibernation, and service caching are the three basic components in the protocol and will be discussed in detail below.

2.1 Replication Mechanism

The replication mechanism is triggered by a specified service provider upon a replica hosting request by one of its clients. If a client achieves a certain replication threshold, it is supposed to ask for its own replica. The main functionalities of this mechanism are as follows:

- On the provider side:
  - Pass a replica: Upon a client request, passes a replica of the currently running service. Keeps information on where the replica has been passed to.
  - Check correctness: Checks the current correctness of a new replica placement based on the allocation correctness computation method discussed later.
  - Publish: Publishes the new status of the services locally and for the remote repositories. The protocol assumes presence of a local service repository for each mobile host. The publishing process is done through all accessible node repositories [3].

- On the client side:
  - Find the least expensive service: The service cost is an index labeling each of the replicas. Clients are supposed to find the lowest cost service and call it. As mentioned before, this index (the service cost) could represent a combination of criteria.
  - Count and ask: Counts the requests of the client regarding a specified service and asks for its own replica, if it achieves a specified number of calls (replication threshold). The number of calls has been selected for simplicity as an indication for the interest. In other cases, the call length or the amount of transferred data could be used.
  - Switch into a provider: Switches the client to be a provider after receiving a replica and publishes the current replica status.

2.2 Hibernation Mechanism

The Hibernation mechanism is running always by the provider side. It is triggered in cases when the noticed common interest by a service is less than a specified level (hibernation threshold). The main functionalities of the hibernation mechanism are as follows:

- Count and decide: During a predefined time interval, this functionality counts the number of requests to its provider by the clients, if this number is not sufficient let this service be hibernated.
- Cache it: Enable the service provider to cache its hibernated service based on its available resources.
- Publish: Publishes just the new hibernated status of the deactivated replica.

2.3 Service Caching Mechanism
Based on the currently available resources of the mobile host, or alternatively on some usage profiles, the mobile host can cache the hibernated service for later use. For simplicity, in this report, the mobile hosts always accept to cache hibernated replicas. The service caching functionalities are:
- Restore a service: Restores a hibernated replica from the cache when the current node (as a client) achieves the replication threshold instead of activating the replication mechanism.
- Publish: Publishes just the new (un)hibernated status of the restored replica.

3 Network Model
In this section, the network used to evaluate our concepts is presented. At a certain time $t$, the network is modeled as an undirected unweighed graph $G(N,E)$, where $N$ is the set mobile nodes and $E$ is the set 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 $G(N,E) = G_1(N_1,E_1) \cup G_2(N_2,E_2) \cup \ldots \cup G_k(N_k,E_k)$, $N = N_1 \cup N_2 \cup \ldots \cup N_k$, and $E = E_1 \cup E_2 \cup \ldots \cup E_k$ each mobile host has a fixed range of wireless transmission with radius $R$; if the distance between two mobile hosts is less than $R$, then a connection between them is established. All mobile hosts are placed in a square area $[2]$.

3.1 Mobility Model
The random way point mobility model [4] is used in this research, in which each of nodes should pick a constant speed between $[0, V_{max}]$ meter/second, then chooses a random location to go. After reaching its desired location, it is supposed to wait for a pause time between $[0, P_{max}]$ and so on.

3.2 Calling Model
The calling model presents the request behavior of the mobile hosts regarding a specified service. Initially, all mobile hosts seek the initial (original) service provider node. Only those nodes in the same partition with the provider node are assumed to start evaluating the service calling (requesting) and be involved in the related replication-hibernation processes. The calling frequency is the main output of the calling model, the number of calls (requests) per a certain time interval dominate the calling behavior of the clients toward a specified service. Of course, many other parameters could replace or be combined with the calling frequency like the call length. For simplicity, the proposed calling model is based on the calling frequency (calling rate).

For each mobile host, a calling rate is generated between $[0..Max-Rate]$ calls per minute, The "Max-Rate" indicates the number of clients’ groups maintained by the network, for example if Max-Rate = 3 calls per minutes then there are four clients’ groups (0 calls/minute, 1 calls/minute, 2 calls/minute, and 3 calls/minute) in the network. The calling rate is constant during a complete calling period. The calling period could be a single "Calling-Length" and its multiplications between $[1..Max-Calling-Units]$, and after a pause time equals a single "Pause-Length" or its multiplications $[0..Max-Pause-Units]$ in minutes the node selects another calling rate and so on. 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-Units) will be discussed and set in the following sections.
3.3 Service Model

The network maintains one service at the beginning of the operation time. This service is placed randomly in
the network. All mobile nodes can participate in the replication mechanism. The original service is replicable.
All participants can cache a replica in case of service hibernation. Each replica is indicated by a requirement
index which quantifies the requirements needed to run this service (the cost of requesting/hosting this
service). These values are generated normally about 20% of a general requirement index. The requirement
index is a mimic of the reality. Normally, even if two providers provide the same service, requirements by
each of them to use their service (or get a replica) will differ [2]. As one of the service distribution protocol
main moves, clients are supposed to find the minimum requirement index from the neighboring services to
communicate with. Over time, this will result in popular and unpopular service replicas.

4 Gross Interest and the Proposed Scenarios

The calling behavior of a group of clients regarding a specified replica generates a common interest for this
service which is the gross interest. The service replication/hibernation processes are dependent on the gross
interest. In the following subsections the matter of how to quantify this gross interest is addressed.

4.1 Active Clients

Active clients are those clients which are in a calling period (see Section 3.2). The expected number of active
clients at a certain time in a network partition is tightly coupled to the proposed calling model. At time \( t \),
the expected number of active clients \( E_x \) is:

\[
E_x = P_z \frac{E_x_n}{E_x_n + R \cdot E_x_m} \quad \text{(1)}
\]

where \( E_x_n \) is the expected value of Max-Calling-Units, \( E_x_m \) is the expected value of Max-Pause-
Units, and \( R \) is the ratio between a single ”Pause-Length” to a single ”Calling-Length” and \( P_z \) is the size
of the \( i^{th} \) network partition. Based on computing the \( E_x \) value, two extremely different scenarios are
proposed to be used.

4.2 Scenario 1: Rich Interest Scenario

In the first scenario, a very active calling behavior with relatively short periods of pausing is modeled. The
mobile nodes have 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 3 minutes and the max-pause-units equals 5 units. So, the pause periods
will be 0 minutes, 3 minutes, 6 minutes, ..., or 12 minutes.

4.3 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 10 minutes and the max-pause-units equals 5 units,
which means that the pause periods should be 0 minutes, 10 minutes, 20 minutes, ..., or 40 minutes.

4.4 Quantifying the Gross Interest Difference Between Scenarios

For Scenario 1, assume a partition with size equal to 25 mobile hosts, this leads to about 20.8 mobile hosts
can be active clients 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, as mentioned in Section 6, we found that for a network size equals 70 nodes, the expected partition size will be about 17 nodes. Again, for Scenario 1, there will be about 14.17 active clients, and for Scenario 2 there will be just about 2.8 active client nodes.

5 Allocation Correctness

The number of mobile hosts with active replicas is the main parameter of determining the correctness of the allocation (placement) process. The correctness of a replica allocation should indicate the relation between the number of active replicas to the size (number of mobile hosts) of a network partition at a specific time. The proposed correctness ratio will be a value between 1 and 0.

If no replica is available within a partition, that is obviously not good. Thus, the correctness should be 0. One replica inside each network partition is assumed to be enough to achieve 100% service availability (our proposed protocol does not take into consideration the related QoS issues yet and assumes that one active replica can serve the whole partition). Therefore, the correctness of such a placement should be 1. Since we are depending on a replication protocol to ensure high availability, we must allow for the creation of replicas. When a new replica is created, initially, it will be in the same partition as the copy of the service it was replicated from. Therefore, we have to allow the existence of two replicas within one partition without penalizing this. Thus, we define the correctness of two replicas within one partition to be 1 also, regardless of the partition size. A linear relation between the partition size and the active replicas inside it is proposed for the other cases. The linear correctness allocation ratio, \( LCR_t \) at a specified time \( t \) of a specified \( (i^{th}) \) network partition \( P_i \) is linearly based on partition size \( P_z_i \) and the number of replicas \( R_i \):

\[
LCR_t(P_i) = \begin{cases} 
0 & R_i = 0 \\
1 & R_i \in \{1, 2\} \\
\frac{P_z_i - R_i}{P_z_i - 2} & R_i > 2
\end{cases}
\]  

More about the allocation correctness ratio and how it indicates the replica allocation process is discussed during the next evaluation section.
6 Simulation

The proposed evaluation depends on an extensive simulation. A detailed simulator was built using C++\(^1\). The proposed simulations are divided into two main categories: first, the general performance of the service distribution protocol is evaluated against different replication/hibernation behaviors, then the effects on the linear replica allocation correctness are drawn. Second, regarding the resultant linear allocation correctness, we extended three other replica correctness computation methods to overcome the deficiencies of the linear allocation correctness and evaluated them.

6.1 Performance Parameters and Settings

The simulation run time for each experiment was set to be 2 hours. The network is placed in a squared shaped (600\(\times\)600\(^2\) meter\(^2\)) area, varying its size from 10 to 140 mobile hosts, each mobile node can cover a transmission range with a fixed radius of 75 meters. The maximum allowed speed for the mobile nodes is 6\(\text{meters/second}\) with a maximum residence (pause) time equal to 15 minutes. Each plotted point in the presented curves comes from the average of 20 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 a number of calls (requests) equal to the maximum call rate (Max-call-rate: 1). The hibernation threshold for a provider is gaining less than one call(request) in a given length of minutes. In this report this given interval will be either 1, 3, or 5 minutes. The performance parameters in this evaluation are the service availability, success ratio, prevalence, residence time, and replication degree.

- Service Availability: Average ratio of the summation of time intervals that the service or one of its replicas was active somewhere in the network to the overall lifetime of the network.

- Success Ratio: Average ratio of the number of succeeded calls (request-queries) of the whole mobile hosts in the network to the total number calls.

- Service Prevalence: Ratio of the number of mobile nodes having an active replica to the total number of the network participants.

- Service Residence time: The average time that the service remained active on the hosting mobile node.

\(^1\)Microsoft Visual C++, version 8.50727
It represents a general indication for the service operation time by all service providers (current and former).

- **Replication Degree**: The average of the ratio of the number of available replicas in a partition to its size of all the network partitions at a time.

The concepts of the service prevalence, replication degree, and the replica allocation correctness ratio are closely related and could be interpreted as different terms for indicating the number of running replicas inside the network. Each of these parameters has its own perspective in highlighting the number of the concurrently running replicas: while the service prevalence measures that number on the scale of the whole network, the replication degree conclude the status of the service distribution in the network on the level of the network partition, and the computation methods of the replica allocation correctness represent a judgement for how each of the network partition is near to the optimum situation of the optimum replica distribution in a given network.

### 6.2 Primary Evaluation

As shown in Figure 1, in Scenario 1, the service availability increases as the network size increases and decreases as the number of the client groups increases for the three hibernation thresholds. This is due to the smaller set of the participants ready to host a replica at a time because, as one of the settings, the replication threshold is set to be $1$：“the maximum call rate”. In case of 4 groups that value is 3 which means that just $\frac{1}{4}$ of the participants can achieve that calling rate. On the other hand in the case of just two groups, $\frac{1}{2}$ of the participants can achieve the maximum calling rate. The same behavior can be noticed in Scenario 2 but the growing of both of availability and success ratio are slower and later. This is due to the poorness of the gross interest in this scenario.

Although keeping longer time intervals for the hibernation threshold is better to keep higher service availability and success ratio, the effect of shortening the hibernation threshold has a relative small effect on both availability and success ratio for both scenarios. Table 1 clarifies these differences.

Figure 5 shows the prevalence ratio of the network participants. For Scenario 1, the higher the number of client groups the lower the prevalence ratio. The effect of longer hibernation thresholds increases the prevalence ratio. The notable result here is the curve of the (1:1) hibernation threshold, it seems to be usually steady under (0.20). On the other hand, in Scenario 2, the same behavior can be noticed for the curves of the different hibernation thresholds for both client groups. But generally, the effect of the poor
Figure 4: Number of replicas in the Important Partition against network size for Scenario 1 and 2

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>2 Groups</th>
<th>Success Rate</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Avg</td>
<td>Avg.StD.</td>
</tr>
<tr>
<td>1:1</td>
<td>0.79</td>
<td>0.21</td>
<td>0.88</td>
</tr>
<tr>
<td>1:3</td>
<td>0.81</td>
<td>0.15</td>
<td>0.90</td>
</tr>
<tr>
<td>1:5</td>
<td>0.85</td>
<td>0.11</td>
<td>0.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 2</th>
<th>2 Groups</th>
<th>Success Rate</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Avg</td>
<td>Avg.StD.</td>
</tr>
<tr>
<td>1:1</td>
<td>0.64</td>
<td>0.17</td>
<td>0.80</td>
</tr>
<tr>
<td>4 Groups</td>
<td>1:3</td>
<td>0.69</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>1:5</td>
<td>0.70</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 1: Average (Avg.) success ratios and availability with the related standard deviation (Avg.StD.) against different hibernation thresholds.

gross interest makes the prevalence ratio very low.

The service residence time decreases as the hibernation threshold test interval increases and as the network size increases for all client groups in both scenarios. As the number of the client group increases, the residence time increases. This is very logical because as the number of client groups increases, the number of clients interested to have a replica will decrease since the replication threshold will be set to the higher call rate which will be assigned for a fewer number of the clients. As a clear note, this effect is reduced as the network becomes more dense and this is logical, too, because the number of the clients interested to host a replica starts increasing in denser networks.

Figure 6 shows the resultant replication degree of the protocol in both proposed scenarios with respect to the number of client groups. The effect of the gross interest type is clear. The replication degree is varying between about 0.3 in the rich gross scenario with 2 client groups to less than 0.05 in the poor gross interest scenario for 4 client groups, as in figure 6. The replication degree increases as the gross interest increases and number of client groups decreases. This result confirms our concept that the more interesting services should prevail and stay more in the network. The average values and the related standard deviation of Figures 5, 6 are presented in Table 6.2.

In Figure 7 the resultant linear replica allocation correctness is examined against different hibernation behavior. As seen, the linear allocation correctness increases as the network size increases. In Scenario 1, the different resultant correctness ratios reach about 0.6 as maximum ratios at starting from about 50 nodes network size for 2 and 4 client groups. On the other hand, the correctness ratio is slower in growing and requires higher dense situation of the higher network sizes to reach about 0.5 for 2 and 4 client groups.
Figure 5: Service prevalence and residence time against network size for Scenario 1 and 2

Table 2: Average (Avg.) residence time, prevalence, and replication degree with the related standard deviation (Avg.StD.) against different hibernation threshold

These results have shown the deficiencies of the linear computation method for the allocation correctness. Also, in shorter hibernation thresholds, like 1:1, the uninteresting replicas should be terminated quicker than with other longer threshold. This quick hibernation should be reflected as higher allocation correctness which could not be seen or interpreted by these results. The concluded deficiencies are:

- Not sensitive to the partition size, since the computing of the average value is giving the different partition sizes the same weight.
- Not sensitive to the number of replicas in cases of large partition sizes. For example, in case of partition size equal to 50 nodes, the allocation correctness ratio of 20 replicas will be 0.63 which is very high.

In the following subsections different allocation correctness computations are proposed and their results are analyzed.

6.3 Alternative Allocation Correctness Computation Methods

Three different alternative computation methods are proposed to overcome the deficiencies of the linear computation method (LCR), namely, the weighted linear allocation correctness ratio (WLCR), the rational allocation correctness ratio (RCR), and the weighted rational allocation correctness ratio (WRCR).

WLCR is the weighted sum of the linear allocation correctness ratios in the different partitions, this computation method is supposed to be sensitive to size of the network partition, where $WLCR_t$ at given time $t$ is:
Figure 6: Replication degree of both numbers of client groups for Scenario 1 and 2

\[ WLCR_t = \sum_{i=0}^{n} \left( \frac{Pz_i}{\text{NetworkSize}} \cdot \text{LCR}(P_i) \right) \] ... (3)

RCR is computed in the same fashion as in equation 3 but on a rational basis to the number of the active replicas as follows:

\[
RCR_t(P_i) = \begin{cases} 
0 & R_i = 0 \\
1 & R_i \text{ in } \{1, 2\} \\
\frac{Pz_i - R_i}{R_i} & R_i > 2 
\end{cases}
\] ... (4)

RCR is supposed to be more sensitive to the higher number of replicas in side a partition, for example, a partition size equals to 50 nodes, the allocation correctness ratio of 20 replicas will be (0.09).

WRCR is the weighted sum of the rational allocation correctness ratios in the different partitions. This method is highly sensitive for both partition size and the higher number of replicas inside a network partition, \( WRCR_t \) at given time \( t \) is:

\[ WRCR_t = \sum_{i=0}^{n} \left( \frac{Pz_i}{\text{NetworkSize}} \cdot \text{RCR}_t(P_i) \right) \] ... (5)

6.4 Further Evaluation

In this subsection, we evaluate the different allocation correctness methods (LCR, WLCR, RCR, WRCR) as an indication for the goodness of the replica placement process of the service distribution protocol.

As shown in Figure 2, in Scenario 1, the higher number of client groups, the higher ratios of LCR and WLCR. In large number of partition sizes and higher number of the number of the allowed to host replicas (replication threshold), both LCR and WLCR are directly proportional to the size of the network. At higher numbers of client groups the difference between LCR and WLCR is noticeable. On the other hand, both RCR and WRCR start increasing during a certain interval of network sizes then the curves reflect their behavior and start a slow (for RCR) or quicker collapsing. The wideness of this interval of the network sizes depends not only on the length of the hibernation evaluation time interval but also on the number of client groups. As the hibernation threshold increases the wideness of the stable RCR and WRCR observed interval decreases, and as the allowed number of participants allowed to host replica decreases this interval decreases as well.

In case of Scenario 2, with the poor gross interest, as shown in Figure 3, the effect of the low number of produced active replicas inside the different partitions is clear. The growing of all the curves is very slow.
until a certain network size and then takes a similar behavior to those of Figure 2 except the WRCR at hibernation threshold 1:1 for both of client groups. As the network size increases the WRCR increases, too (in contradiction behavior against all the other WRCR curves in the other cases), this contradiction can be explained as follows: The shorter hibernation threshold enforces the concurrent active, but uninteresting replicas to be hibernated quicker, since the available number of the active replicas is really small (regarding the poor gross interest in Scenario 2), the active number of replicas will be smaller too. The other main parameter affected by that behavior is the number of the client groups, as the replication threshold, as the large number of participant allowed to host an active replica decreases, the average WRCR ratio increases (from about 0.4 starting from 80 node in 2 client groups to about 0.6 in 4 client groups). Table 6.5 shows the average vales of the four allocation correctness ratios and the related resultant average standard deviation.

As general notes, first, the differences between the three proposed computing ratios are very clear: while WLCR is telling that everything is very nice (in Scenario 1, values of the allocation correctness reaches about more than 0.9 in some cases like at hibernation threshold 1:1 in 4 client groups starting from about 80 node network size), both RCR and WRCR are saying something very different (in Scenario 1, values of the allocation correctness reaches about less than 0.35 and decrease in the same case of at hibernation threshold 1:1 in 4 client groups starting from about 80 node network size). This difference is due to the sensitivity of both RCR and WRCR to the number of the replicas inside each partition compared to both LCR and WLCR.

Second, in Scenario 2, as we can see in Figure 3, no allocation correctness ratios can be achieved till higher network sizes are available which ensure smaller number of formed network partitions.

In the following subsection, the issues of identifying and sampling the partitioning behavior and of the network and deducing the related produced number of replicas inside these partitions are shown.

### 6.5 The Important Partition

The Important partition is a network feature representing the size of a "typical" network partition. It indicates the size of the most weighted partition in the network. The important partition is depending on the mobility model and its settings (see Section (3.2)). Let \( \text{Imp.Part}_{Net} \) be the important partition of a given network size \( \text{size}(Net) \) consisting of \( n \) partitions where the \( i_{th} \) partition size is \( \text{size}(P_i) \), then

\[
\text{Imp.Part}_{Net} = \sum_{i=1}^{n} \left( \text{size}(P_i) \cdot \frac{\text{size}(P_i)}{\text{size}(Net)} \right) \quad (4)
\]
By observing the size of the important partition, as a sample for the partitions of this network, and using the measured WRCR ratio, the number of the currently active replicas inside the important partition can be computed. Keep in mind that we do not consider the issue of how many concurrent replicas are supposed to satisfy a connected partition from QoS perspectives. Figure 8 shows the number of network partitions and the size of the important partition against the network size. The number of partitions observed in this evaluation was varying between about 8 partitions for the 10 nodes network size to about 15 nodes in the interval of [40,70] nodes networks sizes and decreases to be about less than 3 network partitions at network size of 140. As the network sizes increases the size of the important partition increases too and indicates that at higher network density most of the network participants belong to the same partition most of the network operation time. For example, at network size of 120 nodes, the observed important size is about 98 node.

Figure 4 shows the computed number of replicas inside the important partition for both scenarios versus different settings and against the network size. The most important results comes out, if we consider two intervals for network sizes: the first interval is the "low-moderate" interval which includes network sizes less than or equal to 90 nodes, the other interval containing the "higher" network sizes. Table (6.5) shows the difference between the average number of replicas per each of the low-moderate interval "Low.Int." and the higher interval "High.Int" network sizes. Generally, the number of the generated replicas inside the important partition is usually less than 3 active replicas in the low-moderate interval against all of the other settings. This result is considered a very important result, especially from our ongoing research about managing these concurrently running active replicas. The proposed protocol can for lower and moderate network sizes (till 90 nodes) produce a very limited concurrent active replicas number at time. On the other hand, and for the high network densities in the higher intervals of sizes (starting form 100 nodes), the number of produced replicas is directly proportional to the network size, it grows faster as the hibernation threshold increases. As the number of the client groups increases, the number of the generated replicas decreases. In Scenario 2, the number of the concurrent active replicas in general is less then Scenario 1.
7 Related Work

According to the taxonomy of data replication in P2P systems in [5], the proposed protocol should be categorized as a "partial-multimaster optimistic asynchronous" replication protocol.

As more directly related work, examples for service replication over ad hoc networks, [6, 7] introduce algorithms for replication and synchronization for services in mobile ad hoc networks. Assuming presence of a global view of all nodes and network status, the original service node triggers and controls the replication process (in case a new partition formation being predicted), replicas are established and passed before the formation of the predicted partitions. These replicas are supposed to be hosted by an elected node. The used service model assumes presence of master nodes in order to keep services synchronized. [8, 9] estimate the link quality and employ a partition prediction mechanisms based on TORA [10] supplying two mechanisms for pull based replication: (a) replication (pre-partition formation) and (b) merging (after two partitions merged) mechanisms, the work is extended in [11] by more details and consideration about the leader election problem [11], and how the resultant replicas should be deployed. [12, 13] use the same concepts of link evaluation and based on a global view of the ad hoc network, they can deploy their resultant replicas inside the predicted partitions. [8] presented a comparison between these approaches.

As examples of data replication and management for wireless networks, [14] introduces an adaptive pull protocol for data dissemination for mobile networks by estimating the data freshness and considering the data load, then being compared to a flooding approach. [15] proposes a method for data replication in mobile networks, starts a recovery stage to overcome the effects of the mobility and ever-changing topology. This work also is based on the frequency of accessing and moving averages equation. [16], as in [15, 17], is considering the data accessibility frequency to introduce data replicas in many approaches of replicating a specified data item on the whole mobile hosts. The work of [17] is very relevant to our research regarding the definition of interest (until now, the calling frequency) and dismissing un-accessed data segments.

[18] introduces a survey over many data consistency models, then introduces full and partial replication algorithms taking into consideration the data consistency based on ordering the observer’s graphs. Based on a Petri net model, [19] presents an evaluation of a replica placement process (based on a "best" communication matrix) in a dynamic ad hoc network and the availability.

Excluding the proposed protocol, most of the other available approach for service replication are tightly coupled to evaluating the wireless link quality among the mobile hosts and different partitioning prediction schemes. This coupling makes these approaches very dependent on very low level data acquitted from the lower network layers, specified pre-required network architectures and network components like the routing component, on reality, as wireless mobile networks are very open environment, many of different and heterogeneous network architectures and components could be gathered in the same network, of course this open environment represent a very huge challenge for these approaches. In opposite, the proposed protocol does not deal at all any information from the lower network layers, instead of that, it utilizes information available just in the application (service oriented) layer. The proposed protocol could be considered as a network architecture independent protocol.

Again, excluding the proposed protocol, most of the approaches give the service a fixed importance degree, which means, even if the participants in a specified network partition are not interested at all in
this service, the network will anyway enforce the replication mechanism. In contrast, the proposed protocol assume varying gross interest for the service during the network operation time, more interesting services are replicated more others are hibernated.

8 Conclusions

In this report we introduced the service distribution protocol for MANETs as a solution to increase the service accessibility based on a replication/hibernation approach. We showed that this proposed protocol is valid and efficient for two extremely different scenarios with respect to the calling behavior. The effects of different replication thresholds and hibernation thresholds on the general performance and the replica allocation process are discussed. Based on different computation schemes for the ratios of correctness of the allocation, the advantages and disadvantages of these proposed correctness ratios were drawn. Finally, an extended analysis for indicating the number of the concurrent active replicas inside a sample network partition was presented and shown how the concurrent running number of active replicas is really limited for most of the network setting, especially for the moderate network size.

The introduced protocol represents a framework that answers and manages the question/problem of when to replicate/hibernate a service (based on its gross interest by a certain group of clients). The related important question of how to replicate a specified service is still missing. Many areas of concurrent management of services like service synchronization and concurrent mobile data/transactions controlling are being studied by our group in order to complement the proposed protocol to be able to answer both of the previously mentioned questions.

References


