ParTAC: A Partition-Tolerant Atomic Commit Protocol for MANETs

Brahim Ayari, Abdelmajid Khelil, Neeraj Suri
Computer Science Department
Technische Universität Darmstadt
Darmstadt, Germany
Email: {brahim, khelil, suri}@informatik.tu-darmstadt.de

***Abstract*—The support of distributed atomic transactions in mobile ad-hoc networks (MANET) is a key requirement for many mobile application scenarios. Atomicity is a fundamental property that ensures that all nodes decide a consistent outcome. As MANETs are characterized by frequent perturbations due to network partitioning and the fragility of nodes, providing atomicity is challenging. Existing protocols that ensure strict atomicity in MANETs are either bound to specific mobility pattern or based on building blocks such as consensus or group membership, not allowing arbitrary partitions or requiring exact knowledge about the members of a partition. These assumptions limit the deployment of these protocols to very restricted MANET scenarios, and may lead to poor commit rate, high message overhead or blocking related to intolerably long Commit/Abort decision times.**

In this paper, we present the first Partition-Tolerant Atomic Commit protocol (ParTAC) for MANETs which does not rely on consensus or group partition membership. As a consequence, ParTAC supports a significantly wider range of mobility patterns and partitioning scenarios than existing protocols. To reduce Commit/Abort decision times and prevent the protocol from blocking, ParTAC follows a best-effort strategy by defining a lifetime for every transaction after which the transaction is aborted. Further, we introduce a new coordination strategy based on a flexible pre-selection of *multiple* coordinators among the participating nodes. Thus, the failure of a single coordinator can be tolerated in the presence of network partitioning. Moreover, transactions can be aborted by *any* coordinator based on lifetime expiration. ParTAC is evaluated using simulations to demonstrate the performance of the protocol in terms of commit rate, message efficiency and Commit/Abort decision time.

***Keywords*-transaction processing; distributed databases; MANET; network partitioning; atomicity;**

I. INTRODUCTION

The pervasiveness and functionality of portable devices, equipped with wireless network interfaces is continuously increasing. Mobile devices are also increasingly equipped with small-footprint databases such as Oracle Database Lite 10g [1] and IBM DB2 Everyplace [2]. For consistent mobile data management, often mobile users require transactional services which are not explicitly planned in advance. Examples include the spectrum of mobile commerce scenarios, mobile DBs, and increasingly cooperative or autonomous driving through vehicle-to-vehicle communication [3]. Often a connection to a wide-area network such as the Internet may be unavailable due to lack of infrastructure or may be inconvenient or impractical due to the costs/expenses required for such a connection. Mobile Ad-hoc Networks (MANET) are mainly deployed to maintain a certain level of service availability when an infrastructure is unavailable. For instance they are used to support transactional services, such as in Vehicular Ad-hoc Networks, where communication between different entities should be set quickly. The achievable level of transactional service delivery in MANETs depends essentially on the basic transaction services provided by atomic commit protocols. Atomic commit protocols ensure strict atomicity of transactions and consequently play a major role in transaction processing.

Network partitioning is a major perturbation and characteristic of MANETs [4], [5]. Mobile atomic commit protocols need to cope with this perturbation as network partitioning usually leads to service unavailability if consistency is desired [6]. While commit protocols that are designed for fixed networks, such as the traditional Two-Phase Commit (2PC) protocol [7] rely on reliable and continuous communication between the transaction participants, their applicability in MANETs is limited as network partitioning prevents such reliable and continuous communication. Commit protocols developed for infrastructure-based mobile environments such as [8] rely on some nodes in the fixed network to coordinate mobile transactions. Therefore, these protocols are also not directly applicable for MANETs. We refer to [9], [10] for a comprehensive characterization of the atomic commit problem in different mobile environments and its major design requirement issues along with some sketches of possible solutions.

There exists a few atomic commit protocols in MANETs [11], [12], [13], [14]. These are either based on consensus, or show limited partition-tolerance, or require global view assumptions such as partition group membership decreasing their applicability in MANETs. [11] introduces a considerable overhead because it uses consensus in order to assure strict atomicity in MANETs. [12], [13] assume a very specific mobility pattern of a subset of the mobile nodes which makes these solutions applicable only to some specific MANET scenarios. [14] sketches an idea of a commit protocol that presumes partition membership knowledge, which is hard to realize in typical dynamic MANETs.

Research supported in part by DFG GRK 1362 (TUD GKMM), EU INSPIRE, EU CoMiFin, and CASED (www.cased.de).

Contributions and Paper Organization

We propose ParTAC, the first partition-tolerant atomic commit protocol for MANETs which unlike existing protocols, *(a)* does not rely on consensus, *(b)* is independent of the mobility patterns of mobile nodes, *(c)* does not require partition membership knowledge and *(d)* delivers best-effort transactional service availability. Table I summarizes the main advantages and contributions of ParTAC compared to the existing related work which will be detailed in Section IV.

TABLE I
BRIEF COMPARISON TO RELATED WORK

Protocol	Uses consensus	Uses partition/ group membership	Requires specific mobility pattern
ParTAC	*No*	*No*	*No*
[11]	Yes	No	No
[12], [13]	Yes	No	Yes
[14]	No	Yes	No

ParTAC adapts the lifetime concept for mobile transactions introduced in [8] to MANETs in order to reduce transaction decision times. A further key idea is to use multiple coordinators and thus to replicate the coordinator role in order to tolerate unavailability of any subset of coordinators and communication failures. Therefore, ParTAC does not block when some of the coordinators are unavailable for a longer period of time than the lifetime. Furthermore, ParTAC leverages the mobility patterns characteristic for MANETs by having coordinators collect votes from other participants while moving. These votes are shared and merged once multiple coordinators meet by electing a single coordinator. Our analysis shows that ParTAC reduces the Commit/Abort decision time of initiated transactions and helps in trading-off the desired level of the availability, latency and efficiency of the transactional service by adapting the parameters such as the transaction lifetime and the number of coordinators.

The paper is organized as follows. In Section II, the system model is described along with a comprehensive classification of perturbations. The challenges for mobile transaction protocols in MANETs are presented in Section III with related work appearing in Section IV. In Section V, a detailed description of the ParTAC protocol is provided. The protocol is evaluated in Section VI. Section VII concludes the paper and briefly presents our future work.

II. SYSTEM MODEL AND PERTURBATIONS

We present the system model of the mobile environment where strict atomicity is desired for the transactional services to be valid. Next, we identify the relevant perturbations, i.e., constraints and failure modes that can occur in the mobile ad-hoc environment to affect atomic commit functionality.

A. System Model

We consider a generic MANET that consists of a set of mobile nodes (MNs). We do not put any assumptions on the MN density or on the mobility pattern of MNs. MNs are usually equipped with sensors to sense their environment and update their databases. We assume that every MN in this environment has a unique ID. The MNs can communicate with each other in an ad-hoc manner for instance using Bluetooth, WLAN or WAVE. MNs are generally battery-powered and resource-restricted for both computational capabilities and device lifetime.

We refer to a distributed transaction among MNs as a *Mobile Transaction (MT)*. MNs participating in the execution and commit of a MT are called participant MNs (P-MNs). Commit protocols are generally based on the existence of at least one *coordinator (CO)*, which is responsible for coordinating the execution of the corresponding transaction. For different transaction and mobile system models, different nodes may play the CO role which requires special capabilities such as stable storage. More than one CO in ad-hoc transaction scenarios may then be needed. This key issue of CO selection is detailed in Section III. The CO is responsible for storing information concerning the state of the transaction execution. Based on the information collected from and about the P-MNs of the transaction, the CO takes the decision to either commit or abort the transaction and shares this decision with all P-MNs. In this paper, we consider that the application/user is able to specify an appropriate (tolerable) *lifetime* for each initiated MT. The lifetime of a MT is defined as the maximal timeout the CO should wait (as long as there is no final decision) before deciding about the outcome of the MT.

B. Classification of Perturbations

Within this ad-hoc mobile system supporting transactional applications, we consider two main classes of perturbations: *Operational constraints* (battery power, computing, connectivity etc.) and *failures*. The environmental constraints relevant to mobile transactions are mainly characteristics of mobile nodes and wireless links. Failures of the mobile environment are classified into communication and node failures.

1) Operational Constraints: The considered mobile environment is constrained by the characteristics of both *MNs* and *wireless links*. MNs inherently possess restricted computational capabilities such as computational and storage capacity. Especially MNs usually possess limited storage which restricts the amount of storable data. These resource constraints increase the time MNs need to execute transaction fragments or may even lead to execution failures. MNs may also run in different energy modes or might be turned-off to save energy. Additionally, wireless links are characterized by high latency and restricted bandwidth. These characteristics lead to considerably varied reliability/availability and connectivity of MNs.

2) Failures: We now outline the common failure modes and classify them into classes of communication and node failures.

a) Communication Failures: These constitute the majority of failures in MANETs. We consider two types of communication failures: *Message loss* and *network partitioning*. Messages exchanged across MNs are highly vulnerable to loss due to the high bit error rate of wireless links and possible network congestion and collisions. Also high node mobility often disrupts routes and causes message loss. Message loss is highly probable to occur in MANETs and needs to be explicitly taken into consideration in the design of atomic commit protocols. Due to the inherent node mobility and autonomicity, the MANET can easily get partitioned and reconnected. As shown in [4], [5], network partitioning is frequent and unpredictable in MANETs. Another cause for network partitioning is MN disconnection either due to the user turning the MN on-off or due to MN failures that we discuss next. As network partitioning often occurs over the normal mode of MANET operations, it needs to be explicitly considered in the design of atomic commit protocols.

b) MN Failures: In this work, we consider only *transient* MN failures. Transient MN failures occur from either software or hardware faults and usually disappear if the MN reboots. A further common cause of transient failures is the lack of battery power to sustain operation of the mobile device. Transient failures are the most probable failures of MNs. Transient MN failures can manifest for the transaction commit problem as a transient network partitioning, i.e., the MN disconnects from the network if a transient failure occurs and reconnects once this failure disappears and the MN recovers.

III. COMMIT DESIGN CHALLENGES FOR MOBILE TRANSACTIONS

Given frequent network partitioning, the first challenge for transaction processing in ad-hoc scenarios is to disseminate the fragments of the MT to their corresponding MNs. For this, partition-aware dissemination protocols such as Hypergossiping [15] can be used.

As MNs do not connect to any dedicated infrastructure, the CO of the MT must be a MN or a set of MNs. Given the operational constraints and failures discussed in Section II-B, a single MN is constrained to play the CO role alone. Especially in case of network partitioning, communication failures of this single mobile CO usually may lead to the blocking of all P-MNs. The appropriate number of COs depends on the total number of P-MNs and in particular on the perturbation level of the MANET. For volatile and frequent perturbations a higher number of COs might be needed. If the perturbations are rare, a lower number of COs may suffice. In the best case every network partition should have its own CO to be able to take a preliminary decision about the outcome of the MT especially in case of an Abort decision. This can be achieved by deploying as many COs as possible, i.e., every P-MN is a CO. From the efficiency point of view, the number of COs should be minimized as the CO role implies significant message traffic to and from CO nodes which depletes valuable power.

Using the MT lifetime concept introduced in [8] may reduce the decision time of these MTs while also allowing the application to impose/define a tolerable delay-awareness for their initiated transactions. However, the appropriate lifetime value depends on multiple factors. A key issue is network connectivity, which primarily depends on mobility parameters such as speed of MNs, and their communication parameters. These variables make estimating lifetime in ad-hoc scenarios a challenge. Applications initiating delay-aware MTs in MANETs should be at least able to compute how long they are willing to wait before receiving the results of the initiated MT. This time can be used as the lifetime of the initiated MT or can be adapted to the current state of the underlined MANET. In this work we do not assume synchronized clocks across the mobile entities. Thus the lifetime can elapse at different times for different entities.

IV. RELATED WORK

We outline existing solutions along with their limitations and subsequently present our proposed approach. In [11], a cross layer commit protocol for mobile ad-hoc networks (CLCP) is presented. This protocol employs all participants as coordinators and uses consensus to ensure failure tolerance. CLCP is directly instantiated from the application layer, but operates on both network and application layers. Consensus introduces a considerable message overhead to MANETs which makes it undesirable. So an atomic commit protocol that does not use consensus is aimed.

In [12], [13], the authors propose the use of a *cluster of coordinators* preferably in single-hop distance from each other to avoid blocking of P-MNs in case one CO fails. The cluster of COs elects a single *main coordinator* and uses the 3PC protocol [16] to agree on a consistent decision either to commit or abort the MT. If the cluster of COs is partitioned or the main CO fails the authors use a termination protocol based on the Paxos Consensus protocol [17] to elect a new main CO. The assumption in [12], [13] that the COs are moving together in a group (forming a one hop cluster) is not valid in most of ad-hoc scenarios. Targeting a more generic solution, this assumption on the mobility pattern of a subset of the MNs in the MANET needs to be relaxed to consider a generalized arbitrary mobility model.

In [14] a commit solution is presented which assumes that every MN in a partition knows all the members of the partition it belongs to. However this solution is briefly sketched and lacks a detailed description of the proposed group based commit protocol. Given the *partition membership information*, every partition elects a leader and uses the 2PC protocol [7] inside the partition to decide whether the transaction should be tentatively committed or aborted. This temporary decision is communicated to all P-MNs within the partition. When a P-MN joins a new partition, the tentative decision (obtained in its original partition) is communicated to the new partition. As described in [14], the correctness of the proposed solution is assured by the partition membership assumption, i.e., the fact that partitions can be detected. The assumption that every MN in a partition knows all the members of its partition is crucial for a generalized MANET. Some works [18], [19] addressed the problem of group membership in MANETs, however, a generic solution remains a challenge.

Furthermore, the blocking time of P-MNs is often not considered and in worst case all P-MNs may be blocked forever if one of the P-MNs disconnects. As shown in [20], there exists no non-blocking atomic commit protocol if network partitioning may occur for an unpredictable duration. Fortunately, the number of blocked P-MNs can be minimized as we will discuss later. The approach sketched in [14] as based on partition membership information does not use consensus and is independent from the mobility of nodes in contrast to [11], [12], [13]. However, it is based on the assumption that partition membership is available to all its members. Partitions in MANETs are usually very dynamic as nodes may leave and join partitions arbitrarily. Therefore, acquiring the global partition membership information becomes very inefficient.

Our developed approach *(a)* does not use consensus, *(b)* is independent from the underlying mobility pattern and *(c)* does not require any partition membership information as highlighted in Table I presented in Section I.

V. The Proposed ParTAC Protocol

In MANETs, network partitioning is the dominant operational case to consider. We propose a new approach that minimizes and controls the decision time of MTs despite network partitioning and tolerates communication failures of P-MNs. Our approach provides for efficiency and strict atomicity in presence of network partitioning. It limits the decision time of MTs by defining a lifetime for each initiated MT. To tolerate CO unavailability and failures, a set of COs from the P-MNs of the MT is preselected allowing for the replication of MT management data on more than one CO.

A. Protocol Overview

We briefly describe the main building blocks of the ParTAC protocol considering: *(1)* The lifetime of the MT is defined upon its initialization. The selected transaction lifetime value information along with a complete list of P-MNs and a list of pre-selected coordinators is communicated to every P-MN (the pre-selection of COs out of the P-MNs can be random or based on node properties such as IDs, mobility, connectivity, storage capabilities etc.). However, the transaction lifetime information can only be used by the preselected COs as will be shown in the description of the protocol operations. Each CO can safely abort the MT if its lifetime expires. *(2)* The preselected COs are required to collect votes from MT P-MNs. *(3)* When two COs encounter each other, they exchange their collected votes and elect a single active CO among themselves. The other CO immediately stops playing an active CO role and behaves like other normal P-MNs. *(4)* As a result, if all COs transitively encounter each other before the expiration of the MT lifetime, only one active CO remains which will take the final decision for the MT.

In the following we distinguish between normal P-MNs and COs. The term "all P-MNs" includes also COs as they are also P-MNs in the MT execution.

B. Protocol Operations: Activities of P-MNs

The activities of P-MNs are detailed in Alg. 1. As we do not assume the existence of partition membership information, we require that P-MNs send their votes to each CO they encounter as long as there is no final decision (Alg. 1, Line 11 and Lines 13-15). P-MNs know that they are encountering a CO when they receive a beacon from that CO as described in the next section (Section V-C). Hence even if one CO was not aware about the P-MN's vote, e.g., due to message loss, then the vote information is not lost, but communicated to the next encountered CO. It is important to mention here that a P-MN is not allowed to change its vote once it is sent to at least one CO. So all the votes sent to the COs are the same.

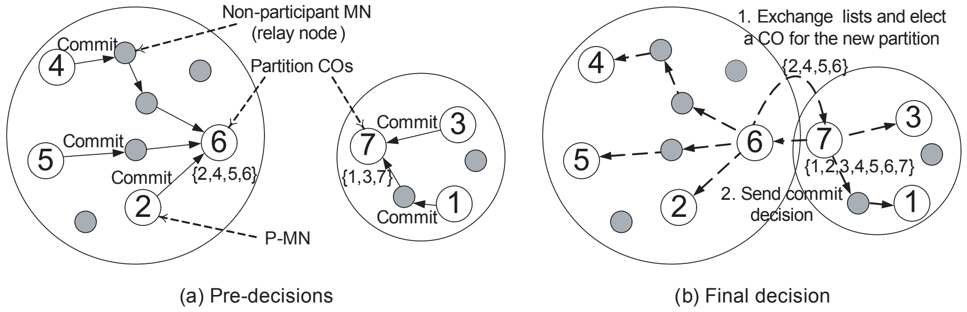
Alg. 1: P-MN's Activities in ParTAC

```
1  wait for receiving a mobile transaction $T_i$;
2  extract the corresponding execution fragment, the set of P-MNs and
   preselected COs;
3  let $P_n = \{P\text{-}MN_1, \ldots, P\text{-}MN_n\}$ the set of all P-MNs;
4  let $C_m = \{CO_1, \ldots, CO_m\}$ the set of all preselected COs;
5  start executing the received execution fragment;
6  if P-MN decides to abort $T_i$ then
7      abort $T_i$;
8      send "No" vote to all CO in $C_m$;
9      exit;
10 else    // P-MN decides to commit $T_i$
11     send "Yes" vote to all CO in $C_m$;
12     while waiting for the final decision about the outcome of $T_i$ do
13         if beacon is received from a CO then
14             send "Yes" vote to the CO from which the beacon was
               received;
15         end
16     end
17     if final decision is Commit then
18         commit $T_i$;
19         exit;
20     else    // decision is Abort
21         abort $T_i$;
22         exit;
23     end
24 end
```

C. Protocol Operations: Activities of Preselected COs

Alg. 2 details the activities of preselected COs. Each CO (as it is also a P-MN in the MT) starts executing its execution fragment upon receiving the MT T_i. The preselected CO starts a timer to detect/watch the expiration of the lifetime of the MT (Alg. 2, Line 7). If it decides to vote for aborting the MT, it sends an Abort decision to all P-MNs of the MT. The COs periodically send presence beacons to allow other P-MNs and COs in their partition to discover their presence (Alg. 2, Line 8). These beacons are those already being sent by the underlying routing protocol so as not to add additional messages.

Every preselected CO maintains a commit-list L of all P-MNs from which it has received a "Yes" vote. For example let's consider the scenario depicted in Fig. 1. In this scenario the COs are preselected based on their IDs (highest ID). For example, in Fig. 1 (a), Nodes 6 and 7 are preselected as COs in the given scenario because they have the highest IDs among the P-MNs involved in the MT. Node 6 maintains in this example the commit-list $L = \{2, 4, 5, 6\}$. If the CO decides to vote for committing the MT, it adds also its ID to its own



(a) Pre-decisions

(b) Final decision

Fig. 1. Partition-tolerant commit in MANETs

commit-list as it is also a P-MN in the MT (Alg. 2, Line 15). As soon as a CO receives a “No” vote it decides to abort the MT and sends an Abort decision to all P-MNs. If the lifetime of the MT expires on a CO before receiving a final decision, the CO decides also to abort the MT (Alg. 2, Lines 49-51).

If two COs encounter each other (e.g., if the corresponding network partitions join) these two COs exchange their commit-lists (Alg. 2, Lines 23-33 and 36-37) and elect one CO among themselves (in the example scenario of Fig. 1, the CO with the highest ID, i.e., Node 7 (Fig. 1 (b)) is elected). The other CO becomes a normal P-MN (Alg. 2, Line 34) and behaves from this point in time and onwards according to Alg. 1. In Alg. 2, we use a schema based on the highest ID to elect the remaining active CO, however, existing election algorithms for MANETs like [21], [22] can also be used. COs are allowed to give their list of votes only to other COs and only after they complete the election process. The non-elected CO (e.g. Node 6 in Fig. 1 (b)) sends it commit-list to the elected one (e.g. Node 7 in Fig. 1 (b)) that merges it with its own list. Thus, the lists are merged only if the election succeeds. If the list does not arrive at the elected CO because of message loss e.g., this information is lost. The protocol can still commit the MT because the non-elected CO will send its vote to every CO it encounters after changing its role to a P-MN. The rest of the votes in the lost commit-list might be collected by another CO since each P-MN sends its vote to every encountered CO and not to a single one of them.

The election process as described above guarantees the uniqueness of the taken decision. From the description of our approach we observe that the votes of COs can only be given to other COs after the election process. Using this schema for the election of a new and single CO guarantees that no two or more COs have the complete knowledge about which P-MNs voted to commit the MT. In the latter case these COs could take different decisions about the outcome of the MT which violates the correctness of the proposed solution.

Every time a CO election is performed, the new elected CO checks whether its new list contains all P-MNs of the MT ((Alg. 2, L 39). If this is the case it decides to commit the MT and sends a final Commit decision to all P-MNs. If all P-MNs voted for committing the MT and only one CO remains for the MT, then this unique remaining CO might have a list that does not contain the IDs of all P-MNs because some votes were lost or the corresponding P-MN did not send any vote due to a transient MN failure or communication failure. In this case the expiration of the transaction lifetime will lead to a MT Abort. P-MNs share the final decisions on encounter. The final decision is inherently replicated onto the CO that turned to a P-MN since the lists of the COs are exchanged (Alg. 2, Lines 26 and 37) before electing a new CO among them. This replication is needed to recover from a failure of the last remaining CO. For the dissemination of the decision and for the communication between the P-MNs inside a single partition, either flooding or a MANET routing protocol like AODV [23] are used depending on the ratio of P-MNs to non-participant MNs. The efficiency and availability of ParTAC can be enhanced by using partition aware dissemination and routing mechanisms like Hypergossiping [15].

Our proposed approach reduces the transaction decision time. Consequently the resource blocking time of P-MNs is reduced as the COs do not wait arbitrarily long to connect to decide the outcome of the MT but have bounded waiting time given by the transaction lifetime. If the transaction lifetime expires at one CO before reaching a final decision, the MT is aborted. This is not viable in any existing solution as P-MNs have to meet asynchronously to be able to reach a final decision.

D. Correctness Basis

To show the correctness of the proposed ParTAC protocol composed of Alg. 1 and 2, we demonstrate that it satisfies the required five *atomicity properties* [24]:

- *Stability:* A participant cannot reverse its decision after it has reached one.
- *Consistency:* All participants that reach a decision reach the same one.
- *Validity:* The Commit decision can only be reached if *all* participants voted “Yes”.
- *Non-Triviality:* If no failure occurs and all participants voted “Yes”, then the final decision should be Commit.
- *Termination:* At any point in execution, if all existing failures are repaired and no new failures occur for sufficiently long time, then all participants will eventually reach a decision.

It follows directly from the specification of the ParTAC protocol in Section V that it satisfies the stability and the

Alg. 2: CO's Activities in ParTAC

```
1  wait for receiving a mobile transaction $T_i$;
2  extract the corresponding execution fragment, the lifetime of the MT,
   the set of P-MNs and preselected COs;
3  let $P_n = \{P\text{-}MN_1, \ldots, P\text{-}MN_n\}$ the set of all P-MNs;
4  let $C_m = \{CO_1, \ldots, CO_m\}$ the set of all preselected COs;
5  let $L = \emptyset$ the list of all P-MNs which sent "Yes" vote to the CO;
6  start executing the received execution fragment;
7  while waiting for lifetime to expire do
8      broadcast periodically beacons containing own ID;
9      if CO decides to abort $T_i$ or receives "No" vote then
10         abort $T_i$;
11         send Abort decision to all P-MNs in $P_n$;
12         exit;
13     end
14     if CO decides to commit $T_i$ then
15         add own ID to $L$;
16         checkList($L$);
17     end
18     switch message $M$ is received do
19         case $M$ is a "Yes" vote from a P-MN
20             add ID of sending P-MN to $L$;
21             checkList($L$);
22         endsw
23         case $M$ is a beacon from another CO
24             compare the received ID with the own ID;
25             if Own ID $>$ received ID then
26                 send request to CO asking for list $L$ (include own
                   list $L$ in the request);
27             else
28                 send own list $L$;
29                 change role to normal P-MN;
30             end
31         endsw
32         case $M$ is a request to send list $L$
33             send own list $L$;
34             change role to normal P-MN;
35         endsw
36         case $M$ contains a list $L$ from another CO
37             send own list $L$ if not already done;
38             add all IDs of received $L$ to own list;
39             checkList($L$);
40         endsw
41         case $M$ is a Commit decision
42             commit $T_i$; exit;
43         endsw
44         case $M$ is an Abort decision
45             abort $T_i$; exit;
46         endsw
47     endsw
48 end
49 abort $T_i$;    // $T_i$ is aborted if lifetime expires
   before reaching a decision
50 send Abort decision to all P-MNs in $P_n$;
51 exit;

52 procedure checkList($L$)
53     if $L$ contains the IDs of all P-MNs then
54         commit $T_i$;
55         send Commit decision to all P-MNs in $P_n$; exit;
56     end
57     return;
```

non-triviality properties. We now show that it also satisfies the consistency, validity and termination properties.

Consistency: The consistency property is satisfied due to the fact that only the last active CO decides about the outcome of the transaction in case the final decision is Commit and distributes the same final decision to every P-MN. In this case the last remaining CO is the single one which can have the final eventual complete list since at least its vote was not communicated to any other CO or P-MN according to the specification of the ParTAC protocol. If more than one CO are still remaining in the system, they can only take an Abort decision and no Commit. Thus the consistency property is guaranteed by our protocol.

Validity: We assume that one of the preselected COs decides to commit the transaction when at least one of the P-MNs has not decided yet. Since this P-MN has not voted yet, its ID can not appear in any list L (Alg. 2, Line 5) of the preselected COs according to the specification of the ParTAC protocol. Obviously, no preselected CO can then take the decision to commit the MT since this contradicts with the protocol specification (Alg. 2, Lines 52-57). In the case that at least one of the P-MNs decides to abort the transaction, the preselected COs can not decide to commit the whole transaction because this decision will violate the protocol specification (Alg. 2, Lines 9-13). Hence, the commit decision can only be reached if all P-MNs voted "*Yes*", i.e., decided to commit the transaction.

Termination: We consider any execution containing the failures listed in the perturbation model detailed in Section II-B. From the ParTAC protocol specification, we can observe that because we are using a timeout concept the protocol can not block forever (the blocking of the protocol forever leads to a non-termination of the protocol). If at any point in execution all existing failures are repaired and no new failures occur for sufficiently long time, then all P-MNs will eventually reach a decision. Especially in this situation all P-MNs (including COs) can meet each other eventually and progressively the lists of COs are filled and the number of COs is reduced until only one CO remains having a list L containing the IDs of all P-MNs. This complete list allows this CO to take a commit decision (Alg. 2, Lines 52-57) and the protocol terminates. If the lifetime expires at any CO before reaching the final decision, the MT is aborted (Alg. 2, Lines 49-51) leading also to the termination of the protocol.

VI. PERFORMANCE EVALUATION

We use simulations to validate our approach. We now present the measured performance metrics, the simulation model and our results that ascertain the high commit rate, the bounded decision time and the efficiency of ParTAC.

A. Performance Metrics

For the evaluation of the ParTAC protocol, we focus on three major performance metrics: *(a) Commit rate* as it determines the service availability, *(b)* commit latency or *transaction decision time* as it determines the service response time, and *(c) message complexity* as it determines the scalability and efficiency of our approach. We measure the commit rate as the ratio of number of successfully committed MTs to total number of initiated MTs. The transaction decision time is the time needed to take a decision about the outcome of the initiated MT, i.e., the time between the initiation of the MT and the time where the final decision is reached at the CO. The blocking time of P-MNs is determined by the transaction decision time and the time needed for the final decision to reach the P-MNs. This time is dependent on the implementation of the dissemination protocol of the final decision and therefore will not be further investigated in our performance evaluation. The message complexity of ParTAC is defined as the number of messages sent and received in average by each P-MN during the execution of the MT.

The performance of our approach is evaluated based on the service delivery level assured by the protocol and defined basically by the commit rate and the decision time. The costs of the assurance of the service delivery level are measured in terms of message complexity. We focus in our performance evaluation on the impact of network partitioning on the performance metrics.

B. Simulation Settings

For our simulation studies we have used J-Sim [25], [26], a component-based, compositional simulation environment that is entirely developed in Java and increasingly used in the MANET community [27]. For the performance evaluation of the ParTAC protocol, we consider a representative range of parameter values to assess the described approach. We selected the commonly used Random Waypoint mobility model [28] and the Reference Point Group Mobility (RPGM) model [29]. We fix the mobility area and the communication range, and vary the number of nodes to consider scenarios where the network is heavily partitioned and others where the number of partitions is low over time. We vary also the node speed to investigate its impact on the performance of ParTAC. We generate the mobility scenarios using the BonnMotion mobility simulator [30]. Given its importance, for all our simulation studies we vary the partitioning degree through varying the number of nodes (note that for RPGM we need to use more nodes to reach the same partitioning degree). The partitioning degree is provided by BonnMotion and reflects how likely that two randomly chosen nodes are within the same partition at a randomly chosen point in time.

TABLE II
SIMULATION SETTINGS

Parameter	Values
Geographical area	$2km$ x $2km$
Communication range	$250m$
Mobility models	RandomWaypoint (RWP), RPGM
Node speed	LOW uniform in [0.5, 1.5] m/s MEDIUM uniform in [3, 10] m/s HIGH uniform in [10, 25] m/s
#Nodes	$\in$ [20,200] for Randomwaypoint $\in$ [60,380] for RPGM
#COs	$\in$ {2,3,4,7,10}
#P-MNs	10
lifetime	$\in$ {60,120,300,900} s

We generate transactions of similar length and with execution fragments of P-MNs of similar length also. We initiate one transaction at the beginning of each simulation. We fix the number of P-MNs to 10 and vary the number of preselected COs and the lifetime. Each simulation is repeated 140 times for statistical significance of the results. Table II summarizes our simulation settings.

C. Simulation Results

We present now the results of our conducted simulation studies for the defined performance metrics. As mentioned before, we simulate ParTAC under different network conditions and vary all protocol parameters to study the behavior of our protocol in a wide range of possible deployment scenarios. Overall, we split the results for Abort and Commit cases to have have better insights to ParTAC.

Impact of Transaction Lifetime: We fix in this scenario *(a)* the number of preselected COs to 3, *(b)* the mobility model to random waypoint, *(c)* the speed to LOW and *(d)* vary the transaction lifetime value. We choose the number of COs to be 3 to keep the number of exchanged messages low as will be shown when the impact of the number of preselected COs will be investigated later in this section. Fig. 2 shows how the commit rate behaves when the number of nodes or the partitioning degree varies. We observe that the commit rate is inversely proportional to the partitioning degree. If the partitioning degree decreases the number of partitions decreases and the number of committed transactions increases. Fig. 2 illustrates also that an increasing transaction

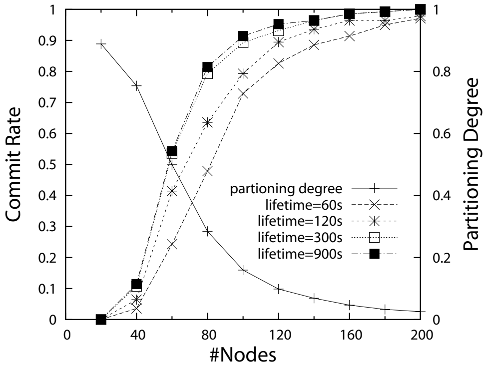


Fig. 2. Impact of Partitioning Degree and Lifetime on Commit Rate

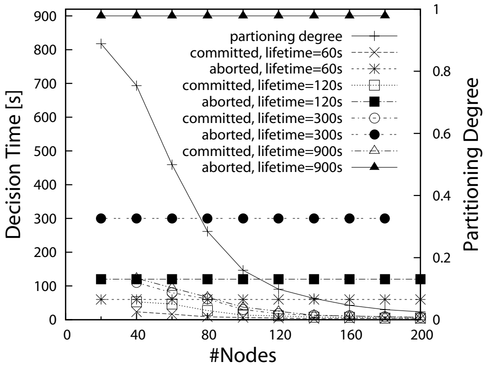


Fig. 3. Impact of Lifetime on Decision Time

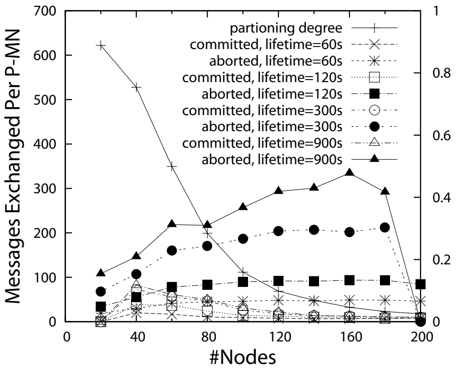


Fig. 4. Impact of Lifetime on Message Complexity

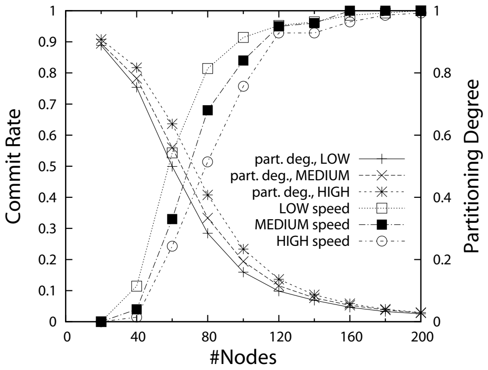


Fig. 5. Impact of Speed of MNs on Commit Rate

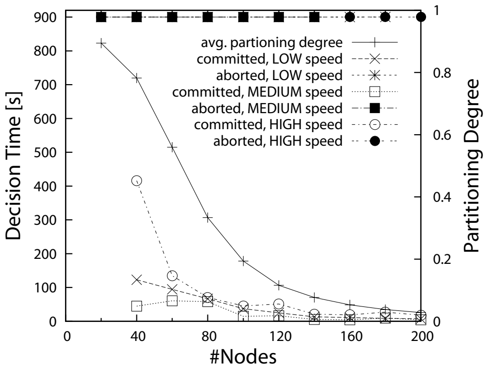


Fig. 6. Impact of Speed of MNs on Decision Time

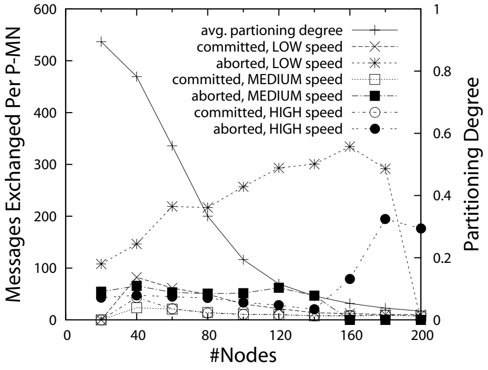


Fig. 7. Impact of Speed of MNs on Message Complexity

lifetime value results in a higher commit rate. Therefore, an appropriate selection of the lifetime value is important to reach a higher commit service availability, however at the cost of a higher commit service latency as shown in Fig. 3. This figure illustrates the existence of a tradeoff between the commit service availability and latency. Especially, in the case when the MTs are aborted the COs need to wait for the expiration of the lifetime to abort the MT which increases the commit latency considerably. In the Abort case the efficiency of ParTAC decreases as shown in Fig. 4. The number of exchanged messages increases considerably since during this time when a P-MN encounters a CO it sends its vote to this CO. The message efficiency in the Abort case can be improved by adding an acknowledgement sent by the CO every time it receives a vote. This is part of our future steps.

Our simulations show the existence of a transaction lifetime which trades off the commit rate and the transaction decision time with a moderate message complexity. The value of the transaction lifetime is dependent on different network parameters and especially the expected partitioning level of the MANET over time.

Impact of Mobility: We arbitrarily fix in this scenario the number of preselected COs to 3 and the transaction lifetime value to 900 s. To assess the influence of mobility on the ParTAC protocol, we vary the speed of the MNs and their mobility models. Fig. 5 shows that the partitioning degree increases if we increase the speed of MNs because of more partition split and join dynamics inside the network [5]. This explains why the commit rate decreases if we increase the speed of MNs. Based on this observation, we conclude that the commit rate of ParTAC does not depend directly on the speed of the MNs but only on the partitioning level of the network, which can be affected by the speed of the MNs, especially, if insufficient number of MNs are deployed in the MANET scenario. Figs. 6 and 7 illustrate that a MEDIUM speed is the best for low decision time and high efficiency (low number of exchanged messages) of ParTAC.

Fig. 8 shows that the commit rate is not dependent on the mobility model of the MNs or on the number of MNs in the simulated area but depends only on the partitioning level or degree of the network. The overhead in terms of transaction decision time (Fig. 9) and messages exchanged between the P-MNs (Fig. 10) is higher for RPGM than random waypoint. For RPGM more nodes are deployed in the same simulation area to reach similar levels of partitioning degree. This increase in the number of MNs leads to a higher message losses and higher network congestion, which explain the higher transaction decision time and higher number of exchanged messages in the case of committed transactions.

Based on these overall results described above, we highlight that our approach allows to efficiently reach maximal commit rates independent from the mobility pattern of the MNs (mobility model and speed). This confirms our claim in Section IV. We highlight also the scalability of our approach since the increase of the numbers of nodes in the simulated area does not result in an over proportional increase of the MT costs in term of decision time and message complexity.

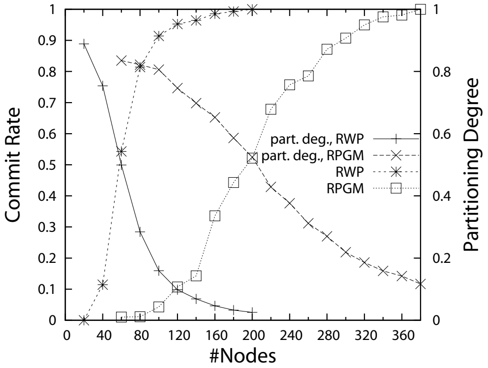


Fig. 8. Impact of Mobility Model on Commit Rate

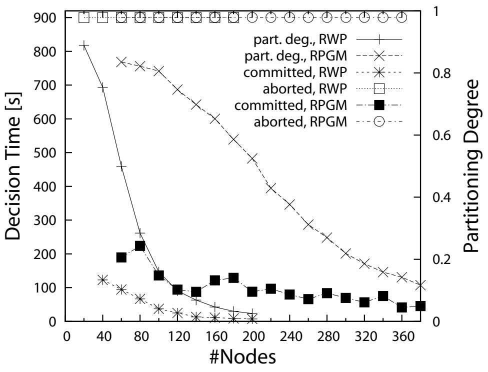


Fig. 9. Impact of Mobility Model on Decision Time

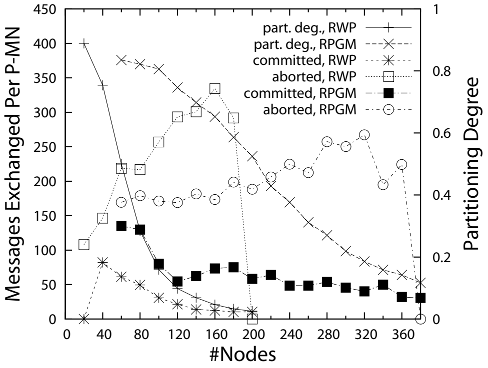


Fig. 10. Impact of Mobility Model on Message Complexity

Impact of Number of Preselected COs: We arbitrarily fix in this scenario the transaction lifetime value to 120 s, the mobility model to RWP and the speed to LOW and vary the number of preselected COs. The number of preselected COs does not impact the commit rate of ParTAC as illustrated in Fig. 11. This is due to the fact that as soon as two COs encounter each other only one of them remains active and the other one becomes a normal P-MN. After a certain point in time only few (2 to 3) COs remain and all the simulated scenarios behave from this instant onwards similarly. This point in time is nearer to the initiation time of the MT in Commit case as from all the COs present in one partition only one remains active as soon as they receive beacons from each other. However, the number of preselected COs have a minor

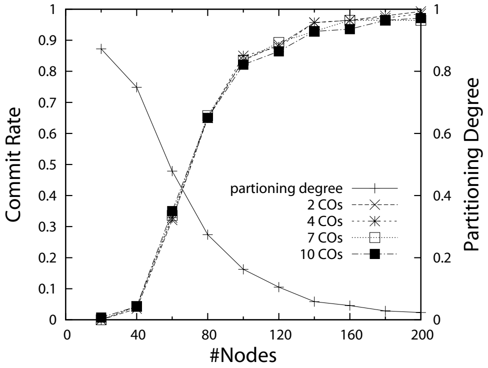


Fig. 11. Impact of Number of COs on Commit Rate

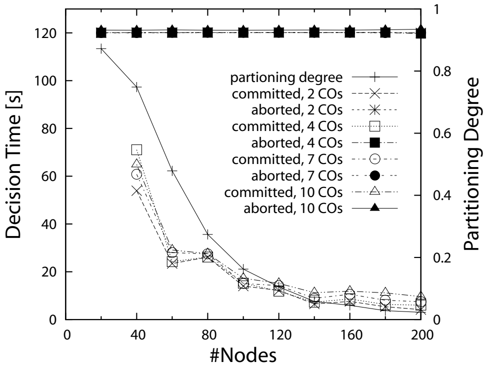


Fig. 12. Impact of Number of COs on Decision Time

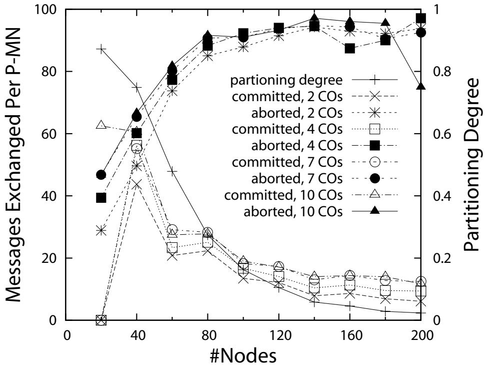


Fig. 13. Impact of Number of COs on Message Complexity

impact on the decision time and the efficiency of the ParTAC protocol as shown in Fig. 12 and Fig. 13, respectively. The decision time shows a slight increase as the number of COs increases due to the time needed to elect an active CO every time two COs encounter each other. The slight increase of the number of messages exchanged per node is due to the fact that every P-MN needs to send its vote to more COs as the number of COs increases. It is noteworthy to mention that selecting higher number COs is primarily to tolerate CO failures during the MT execution. Our simulations show that a higher CO failure-tolerance does slightly impact the transaction decision time and message efficiency.

D. Discussion

Transactional services represent a key part of service oriented architectures and increasingly for mobile environments such as MANETs, Vehicular Ad-hoc Networks etc. The user/application requires to perform a certain number of atomic transactions with a maximized commit rate and within a certain tolerable response time. Data consistency and high transactional service availability should be provided despite the frequent perturbations during the service operational conditions in MANETs. Our ParTAC commit protocol considers the application requirements by defining a transaction lifetime for each initiated MT. Within the MT lifetime, our approach guarantees consistency of data and maximizes the commit rate. This is achieved for MANETs showing an arbitrary degree of perturbations with respect to network partitioning. Therefore, our commit solution provides for a best effort transactional service availability for the challenging MANET environment. Furthermore, the ParTAC approach helps in reducing the transaction decision time resulting in a better transactional throughput and consequently in a better scalability. This allows to maximize the number of users that can use the database resources on resource-limited mobile nodes.

VII. CONCLUSION

In this work, we have shown how delay-awareness can help in reducing the costs of mobile transactions and in decreasing the number of aborted transactions in MANETs. Delay-awareness can also help in providing perturbation-resilience in generalized MANETs. We have presented the main challenges for designing atomic transaction protocols faced in MANETs. We presented ParTAC, a novel atomic transaction commit protocol that provides strict atomicity in spite of frequent MANET perturbations and especially network partitioning. Our protocol is independent from the considered MANET and it is generalized since it is not based on hard assumptions like consensus and group membership. Being atomic and efficient, and maximizing the commit rate, ParTAC guarantees data consistency while allowing for high transactional service availability and scalability. In future work, we plan to combine our previous work for infrastructure-based mobile environments [8] with the current solution to provide an integrated commit solution for a generalized mobile scenario where some of the mobile devices have access to the infrastructure.

REFERENCES

[1] Oracle Database Lite 10g. Oracle Corporation. [Online]. Available: http://www.oracle.com/technology/products/lite/index.html

[2] IBM DB2 Everyplace. IBM Software Products. [Online]. Available: http://www-01.ibm.com/software/data/db2/everyplace/

[3] E. Huang, W. Hu, J. Crowcroft, and I. Wassell, "Towards commercial mobile ad hoc network applications: A radio dispatch system," in *Proc. of MobiHoc*, 2005, pp. 355–365.

[4] J. Hähner, D. Dudkowski, P. J. Marrón, and K. Rothermel, "A quantitative analysis of partitioning in mobile ad hoc networks," in *Proc. of SIGMETRICS/Performance*, 2004, pp. 400–401.

[5] ——, "Quantifying network partitioning in mobile ad hoc networks," in *Proc. of MDM*, 2007, pp. 174–181.

[6] S. B. Davidson, H. Garcia-Molina, and D. Skeen, "Consistency in a partitioned network: a survey," *ACM Computing Surveys*, vol. 17, no. 3, pp. 341–370, 1985.

[7] J. Gray, "Notes on data base operating systems," in *Operating Systems, An Advanced Course*, 1978, pp. 393–481.

[8] B. Ayari, A. Khelil, and N. Suri, "FT-PPTC: An efficient and fault-tolerant commit protocol for mobile environments," in *Proc. of SRDS*, October 2006, pp. 96–105.

[9] ——, "Delay-aware mobile transactions," in *Proc. of SEUS*, 2008, pp. 280–291.

[10] ——, "Exploring delay-aware transactions in heterogenous mobile environments," *Journal of Software*, vol. 4, no. 7, pp. 634–643, 2009.

[11] S. Obermeier, S. Böttcher, and D. Kleine, "CLCP-A distributed cross-layer commit protocol for mobile ad hoc networks," in *Proc. of ISPA*, 2008, pp. 361–370.

[12] J.-H. Bose *et al.*, "An integrated commit protocol for mobile network databases," in *Proc. of IDEAS*, 2005, pp. 244–250.

[13] S. Böttcher, L. Gruenwald, and S. Obermeier, "A failure tolerating atomic commit protocol for mobile environments," in *Proc. of MDM*, 2007, pp. 158–165.

[14] W. Xie, *Supporting Distributed Transaction Processing Over Mobile and Heterogeneous Platforms. PhD-Thesis*. Georgia Institute of Technology, 2005.

[15] A. Khelil, P. J. Marrón, C. Becker, and K. Rothermel, "Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks," *Ad Hoc Networks*, vol. 5, no. 5, pp. 531–546, 2007.

[16] D. Skeen and M. Stonebraker, "A formal model of crash recovery in a distributed system," *IEEE Trans. on Software Engineering*, vol. 9, no. 3, pp. 219–228, May 1983.

[17] L. Lamport, "The part-time parliament," *ACM Trans. on Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.

[18] G.-C. Roman, Q. Huang, and A. Hazemi, "Consistent group membership in ad hoc networks," in *Proc. of ICSE*, 2001, pp. 381–388.

[19] L. Briesemeister and G. Hommel, "Localized group membership service for ad hoc networks," in *Proc. of IWAHN*, 2002, pp. 94–100.

[20] D. Skeen, "Nonblocking commit protocols," in *Proc. of COMAD*, 1981, pp. 133–142.

[21] N. Malpani, J. L. Welch, and N. Vaidya, "Leader election algorithms for mobile ad hoc networks," in *Proc. of DIALM*, 2000, pp. 96–103.

[22] S. M. Masum, A. A. Ali, and M. T. youl Islam Bhuiyan, "Asynchronous leader election in mobile ad hoc networks," in *Proc. of AINA*, 2006, pp. 827–831.

[23] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of WMCSA*, 1999, pp. 90–100.

[24] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[25] J-Sim. The J-Sim Website. [Online]. Available: http://j-sim.cs.uiuc.edu/

[26] J-Sim Wireless. The J-Sim Wireless Extension Tutorial. [Online]. Available: http://j-sim.cs.uiuc.edu/v1.3/wireless/wireless_tutorial.htm

[27] H.-Y. Tyan and J. Hou, "Design, realization, and evaluation of a component-based, compositional network simulation environment," in *Proc. of CNDS*, 2002.

[28] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. of MobiCom*, 1998, pp. 85–97.

[29] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proc. of MSWiM*, 1999, pp. 53–60.

[30] BonnMotion. The BonnMotion Mobilty Scenario Generation and Analysis Tool. [Online]. Available: http://iv.cs.uni-bonn.de/wg/cs/applications/bonnmotion/