

Ensuring Strong Data Guarantees in Highly Mobile Ad Hoc Networks via Quorum Systems

Daniela Tulone

MIT Computer Science and Artificial Intelligence Lab

Abstract

Ensuring the consistency and the availability of replicated data in highly mobile ad hoc networks is a challenging task because of the lack of a backbone infrastructure. Previous work provides strong data guarantees by limiting the motion and the speed of the mobile nodes during the entire system lifetime, and by relying on assumptions that are not realistic for most mobile applications. We provide a small set of mobility constraints that are sufficient to ensure strong data guarantees and that can be applied when nodes move along unknown paths and speed, and are sparsely distributed.

In the second part of the paper we analyze the problem of conserving energy while ensuring strong data guarantees, using quorum system techniques. We devise a condition necessary for a quorum system to guarantee data consistency and data availability under our mobility model. This condition shows the unsuitability of previous quorum systems and is the basis for a novel class of quorum systems suitable for highly mobile networks, called *Mobile Dissemination Quorum* (MDQ) systems. We also show a MDQ system that is *provably optimal* in terms of communication cost by proposing an efficient implementation of a read/write atomic shared memory.

The suitability of our mobility model and MDQ systems is validated through simulations using the random waypoint model and the restricted random waypoint on a city section. Finally, we apply our results to assist routing and coordinate the low duty cycle of mobile nodes while maintaining network connectivity.

Key words: Data guarantees, linearizability, mobile ad hoc networks, mobility model, quorum systems, system performance, scalability, coordination, cooperative tasks.

1. Introduction

Ensuring the availability and the consistency of shared data is a fundamental task for several mobile network applications. For instance, nodes can share data containing configuration information, which is crucial for carrying out cooperative tasks. The shared data can be used for example to coordinate the duty cycle of mobile nodes to conserve energy while maintaining network connectivity. The consistency and the availability of the data plays a crucial role in that case since the loss of information regarding the sleep/awake cycle of the nodes might compromise network connectivity. The consistency and availability of the shared data is also relevant when tracking mobile objects, or in disaster relief applications where mobile nodes have to coordinate distributed tasks without the aid of a fixed communication infrastructure. This can be attained via read/write shared memory provided each node maintains a copy of data regarding the damage assessment and dynamically updates it by issuing write operations. Also in this case it is important that

the data produced by the mobile nodes does not get lost, and that each node is able to retrieve the most up-to-date information. Strong data consistency guarantees have applications also to road safety, detection and avoidance of traffic accidents, or safe driving assistance.

The atomic consistency guarantee introduced by Herlihy and Wing (7) is widely used in distributed systems because it ensures that the distributed operations (e.g., read and write operations) performed on the shared memory are ordered consistently with the natural order of their invocation and response time, and that each local copy is conforming to such an order. Intuitively, this implies that each node is able to retrieve a copy showing the last completed update, which is crucial in cooperative tasks. However, the implementation of a fault-tolerant atomic read/write shared memory represents a challenging task in *highly mobile* networks because of the lack of a fixed infrastructure or nodes that can serve as a backbone. In fact, it is hard to ensure that each update reaches a subset of nodes that is sufficiently large to be retrieved by any node and at any time, if nodes move along unknown paths and at high speed. Figure 1 shows a scenario of unsuccessful update. In that ex-

Email address: tulone@csail.mit.edu (Daniela Tulone).

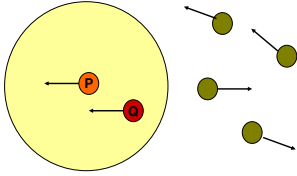


Fig. 1. A scenario of unsuccessful data update.

ample node P broadcasts an update, which is received by its neighbor Q but not by the other nodes since by the time node Q forwards it they are outside its radio broadcast.

The *focal point model* introduced by Dolev et al. in (4), provides a first answer to this challenge since it masks the dynamic nature of mobile ad hoc networks by a static model. More precisely, it associates mobile nodes to fixed geographical locations called *focal points*. According to this model, a focal point is *active* at some point in time if its geographical location contains at least one active mobile node. As a result, a focal point becomes *faulty* when each mobile node populating that subregion leaves it or crashes. The merit of this model is to study node mobility in terms of failures of *stationary abstract points*, and to design coordination protocols for mobile networks in terms of static abstract nodes. The latter task is clearly easier than the former. However, the model proposed in (4) assumes that only a fraction of focal points can become faulty during the *entire* system lifetime. This implies that only a fraction of subregions becomes empty during the system lifetime. Clearly, this assumption poses strong limitations on the motion of the mobile nodes over the system lifetime, and on the density of the network. Moreover, this condition is very difficult to ensure in mobile *sparse networks* where a node can trigger a focal point failure each time it leaves a focal point region to join another one. For these reasons the implementation of a read/write atomic shared memory proposed in (4) cannot be applied to networks where nodes are not densely deployed and move from one geographic location to another with relevant speed following unknown paths.

In this paper we investigate a small set of mobility constraints that are necessary to ensure strong data guarantees. Our work uses and extends the focal point model (4) because it allows us to study node mobility in terms of failures of static abstract points and to apply fault-tolerant techniques. Our goal is to devise mobility conditions that are sufficient to derive strong data guarantees and are realistic for most mobile applications in which the motion and the speed of nodes is unknown, such as vehicular applications. The key idea of our proposal consists of transforming the problem of tolerating high node mobility into the problem of tolerating *continuous* focal point failures, and applying fault-tolerance techniques, such as proactive recovery. In contrast with (4), we tolerate an *unlimited* number of focal point failures during the entire system lifetime, that is, we allow mobile nodes to move according unknown paths and speed. We achieve that by devising a small set of mobility constraints that are sufficient to ensure strong data guarantees. Our mobility constraints define a *minimum coverage* of the mobile nodes

over the geographic system area and limit the node motion only during the expected maximum round trip delay. Note that assuming a minimum node coverage is weaker than assuming a specific node density since some subregions can be more populated than others, and some of them can be empty. As a result, our model allows higher node mobility with respect to previous work (4), and it is more realistic.

As mentioned before, our mobility constraints are sufficient to ensure strong data guarantees, and more precisely to implement a fault-tolerant read/write atomic shared memory. Our implementation is built on top of the focal point abstraction and tolerates an unbounded number of focal point failures during the system lifetime. The *recovery* of the focal point after a failure represents a crucial point in our implementation to guarantee data availability and atomic consistency. Our recovery protocol allows a previously faulty focal point to become active by retrieving the most up-to-date copy. Note that since the motion of the mobile nodes is continuous over the time, nodes can leave a geographical subregion and join another one, thus causing continual failures of the focal points. Therefore, it is crucial for the availability of the data that each focal point successfully recovers its state, and that at any time there is a *sufficient number* of *active* focal points. In fact, if the failure rate of the focal points exceeds the recovery rate at some point in the execution, the system can fall into a *stale condition* where the number of active focal points is not sufficient for the recovery to complete. In this case the data becomes unavailable. As a result, the availability of the data is strictly related to the *liveness* of the recovery protocol, and to its *response time*. This observation along with the need of designing energy-efficient protocols motivated us to investigate quorum systems under high node mobility. In (16) we have shown that quorum systems can be regarded as a tool for energy conservation in sensor networks if properly adapted, here we study quorum systems under our mobility model. More precisely, we devise a condition that is necessary for a quorum system to guarantee data consistency and availability. This condition shows that previous quorum systems are not always able to guarantee data consistency and availability under our mobility model. We propose a class of quorum systems, called *Mobile Dissemination Quorum* (MDQ) systems, that is resilient to high node mobility, and show a MDQ system, called Q_{opt} , that is *provably optimal* in terms of communication cost. We prove the optimality of Q_{opt} by showing an implementation \mathcal{I} of a read/write shared memory consisting of a suite of read/write/recovery protocols that guarantee atomic consistency and data availability. As shown in Figure 2, \mathcal{I} is built on top of the following abstractions: the focal point regions introduced in Section 3.1, the (revised) focal points defined in Sections 3.2 and 4, and the MDQ systems presented in Section 6.

We evaluate our mobility constraints and MDQ systems through simulations using the random waypoint and the restricted random waypoint on a city section (25), and different node speed and node sets. Our simulation results show the suitability of our model and the energy-efficiency of our quorum system. Finally, we discuss the applicability of our results and show that our implementation \mathcal{I} can be applied to coordinate

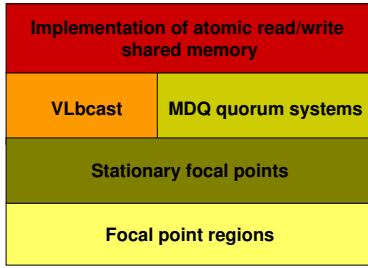


Fig. 2. Abstractions used in \mathcal{I} .

the low duty cycle while maintaining network connectivity and to assist routing.

Our contributions can be summarized as follows:

- **Mobility model.** We propose a small set of mobility constraints necessary to ensure strong data guarantees in highly mobile networks. These conditions use an extension of the focal point abstraction proposed in (4), and do not limit the motion of the nodes during the entire system lifetime. Simulation results performed using the random waypoint model and the restricted random waypoint on a city section confirm their suitability in several cases.
- **Quorum systems for mobile networks.** We study quorum systems under our mobility constraints and prove that previous quorum systems fail to guarantee data consistency and availability by showing a condition necessary for a quorum system to guarantee these properties. Then, we propose a class MDQ of quorum systems satisfying this condition, and compute a quorum system Q_{opt} that is *provably optimal* in terms of the communication cost. We prove the optimality of Q_{opt} by showing an efficient implementation \mathcal{I} of an atomic memory built on top of it. Our simulation results show that Q_{opt} leads in most cases to a reduction of message transmissions larger than 40% if the number of mobile nodes is at least twice the number of focal point regions.
- **Applications.** We apply our implementation \mathcal{I} to assist routing to improve energy conservation and reliability and to coordinate the low duty cycle of mobile nodes while maintaining network connectivity.

Structure of the paper. In Section 2 we compare our proposal with previous work, and in Section 3 we describe our system model. Section 4 illustrates briefly our implementation of the focal point. In Section 5 we illustrate our mobility model, and in Section 6 we analyzed quorums systems in mobile networks, and define the MDQ systems and Q_{opt} . In Section 7 we show the optimality of Q_{opt} by presenting an implementation of a read/write atomic shared memory built on top of it. In Section 8 we validate our results through simulations, and in Section 9 we discuss some applications, and then conclude the paper.

2. Related works

In this section we compare our results with previous work related to the implementation of an atomic memory in MANETs, and more generally to data consistency.

As mentioned in the Introduction, our work employs the fo-

cal point model proposed by Dolev et al. (4), which associates abstract mobile nodes to fixed geographical locations. However, their work assumes a weak mobility model that imposes strong limitations on the node motion and density over the entire system lifetime. This model is not realistic for most mobile applications and for low density networks where nodes move along unknown paths (e.g., robotic applications, vehicular networks). Our implementation relaxes these assumptions, and assumes *arbitrary* node motion. Moreover, our work does not rely on reconfigurations to guarantee fault-tolerance in mobile settings as in RAMBO (5). This leads to a simpler and more efficient implementation, features that are relevant in sensor networks due to their limited energy source.

Several solutions have been proposed for data dissemination in MANETs, such as (29; 15; 18; 19). However, their perspective is different than ours since they do not provide strong data consistency guarantees, such as atomic consistency and data availability. On the other side, some of these proposals (17; 19; 32) address the problem of network partitions, which we do not consider at this stage. Some of these proposals use node mobility to deliver information opportunistically: mobile nodes can exchange information when they meet (20), or move to deliver messages (17), thus improving the network connectivity. A well-known technique to increase the availability of the data is to replicate data across a set of nodes. Clearly, this approach brings up the problem of ensuring consistency among the replicas in the presence of node mobility. This problem has been analyzed in several papers such as in (27; 29; 31), however their perspective is different than ours since they do not analyze the problem of ensuring strong data guarantees. Another related problem is the location management problem, which has many applications in MANETs. For instance, it is relevant to locate cache of Internet-based services (11; 26), or improve the data accessibility service, thus enhancing QoS in MANETs to access desired data with high success rate (18).

In addition, we analyze quorum system techniques in highly mobile networks and devise a framework and theoretical bounds to ensure data consistency and availability. To the best of our knowledge this is the first systematic study of quorum systems in the specific context of highly mobile networks. In fact, previous work on quorum systems in MANETs rely on strong assumptions regarding the motion of the nodes and their distribution, such as (4; 5; 28), or provide probabilistic guarantees as in PAN (15).

3. System model

In this section we describe our system model and abstractions, which consist of the focal point regions and the stationary focal points. These abstractions share some similarity with (4).

Our model consists of a bounded region \mathbf{G} of a two-dimensional plane, populated by a dynamic set of mobile nodes (e.g., nodes can be replaced after they run out of battery). The mobile nodes can move on any continuous path in \mathbf{G} , and may fail at any time due to battery depletion or physical damage. They communicate with their neighbors through radio broad-

cast medium. We assume that each mobile node has a clock that is synchronized, and that each node is aware of its current position. This can be ensured by equipping the node with a GPS, or by applying location services such as (21).

Let us denote the physical broadcast radius of the nodes by r . We assume a reliable broadcast service, called the Lbcast service, built on top of the physical radio broadcast and whose broadcast radius is equal to r . Note that by doing that we assume symmetric and reliable radio links. Although both assumptions are not entirely realistic, recent publications (33; 22) have proposed solutions for providing reliable broadcast in case of node mobility, and have shown that careful neighborhood management and retransmissions can provide loss rates as low as 1-2 percent in sensor networks, which should be sufficient for our purposes. We denote by d the maximum transmission delay of the Lbcast service. Therefore, after d time units each neighbor receives any broadcast message.

3.1. Focal point regions

We consider a set of geographical subregions of \mathbf{G} , called *focal point regions*, populated by mobile nodes. A mobile node is in a focal point region at a certain time if its position falls in that region. As mentioned above, our focal point model diverges from (4) for its underlying geometry. For instance, while the focal point regions proposed in (4) are defined as *non-intersecting regions* of \mathbf{G} , our focal point regions intersect. As shown in Section 5.1, this assumption makes our implementation resilient to high node mobility and low density. We define the *diameter* of a closed geographic region $A \subset \mathbf{R}^2$ of the plane, as the maximum Euclidean distance between any two points in A , and denote by $C(P, c)$ the disk whose center is point P and radius $c \in \mathbf{R}$. The *proximity region* $Prox(A, \nu)$ of width ν of a closed region A with $\nu \in \mathbf{R}$, is defined as follows:

$$Prox(A, \nu) = \left\{ P : [P \in G \setminus A] \wedge [C(P, \nu) \cap A \neq \emptyset] \right\}$$

It consists of points in $\mathbf{G} \setminus A$ whose distance from the border of A does not exceed ν . The proximity region is important to justify our failure model in case of high node mobility and low density (see Section 5.1). We define now the focal point region and the n -region vector.

Definition 1 A *focal point region* of \mathbf{G} and $\nu \geq 0$, is a closed geographic region contained in \mathbf{G} and whose diameter does not exceed $r - 2\nu$. A n -region vector $\langle G_1, \dots, G_n \rangle$ for \mathbf{G} and ν , is a vector of n -focal-point regions of \mathbf{G} such that $\mathbf{G} = \bigcup_{i=1}^n G_i$.

Figure 3 graphically illustrates an example of focal point regions, more specifically a 28-region vector. The geographic system region \mathbf{G} is partitioned into a square grid of focal point regions, and the proximity region of the focal point region P is the surrounding section indicated in Figure 3.

Note that there is a strict connection between geometric properties and underlying mobile nodes. For instance, since the diameter of a focal point region does not exceed r , all mobile nodes in a focal point region can communicate each other. Definition 1 implies that each mobile node is always contained in

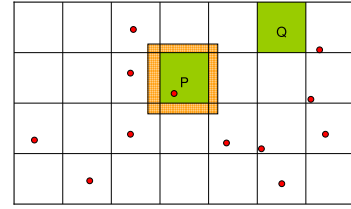


Fig. 3. Focal point and proximity regions.

at least one of the focal point regions G_1, \dots, G_n . Clearly, for a given choice of \mathbf{G} , n and ν there might exist more than one n -region vector depending on the geometry of the focal point regions (e.g., disk, square).

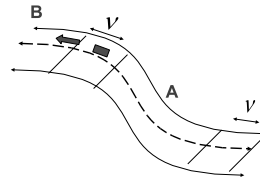


Fig. 4. An example of focal point region.

An example: we can apply our model to vehicular networks. In this example the system region \mathbf{G} is the union of the roads of a given geographical area. The focal point region is represented by road segments. Figure 4 shows two focal point regions A and B and their proximity region of width ν , and a car traveling in the proximity regions of A and B .

3.2. Focal points

A focal point is an abstraction consisting of a focal point region and the mobile nodes that populate it. More precisely, a focal point $F_i(t) = \langle G_i, M_i(t) \rangle$ at time t represents the focal point region G_i and the mobile node that are contained in G_i at time t . Since G consists of n focal point regions, we consider a set of n *stationary* focal points F_1, \dots, F_n . Each focal point can be in one of the following modes: *faulty* if there are no correct nodes in G_i , *recov* if a node has joined its empty region G_i , and *active* otherwise. For instance, in Figure 3 P is an active focal point and Q is a faulty one. We say that F_i is *adjacent* to F_j if its associate focal point region G_i is adjacent to G_j .

A focal point (or node) communicates with the other focal points using a virtual communication service, called VLbcast, which is built on top of the Lbcast service and forwards messages to its adjacent focal point regions. Note that by doing that we assume that the radio broadcast of the nodes can be set to $2r$. Similarly to the Lbcast service, the VLbcast service guarantees reliable delivery. It satisfies the following connectivity property: a focal point F_i (or node) is *connected* to F_j via VLbcast during $\Delta = [t, t + \delta]$ with $\delta > 0$, if there exists a path of nodes $C_0, \dots, C_j, \dots, C_k$ during Δ , such that C_{i+1} is within the radio broadcast of C_i for any $1 \leq i < k$, and such

that C_0 is contained in G_i and C_k in G_j . We do not discuss in this paper the implementation of the VLbcst service, but refer to previous work such as (2; 14).

In (4) a focal point F_i is *faulty* at time t if its location G_i contains no active node at time t . However, this definition seems incompleting since it does not take into account network connectivity. In fact, by active focal point we mean a focal point that is able to participate to the protocols. However, according to the previous definition F_i can be active but unable to communicate with other focal points (e.g., because its adjacent focal points they are all faulty). For this reason, we use the following definition:

Definition 2 A focal point F_i is faulty at time t if G_i does not contain any active node or if it is not connected to M focal points, where M is a network implementation-dependent parameter contained in $(1, n)$.

As a result, at any time F_i is in one of the following modes: (1) *faulty*, according to Definition 2, (2) *recov*, if at least one active client node enters an empty focal point region G_i , or its connection is recovered, and F_i is in the process of recovering its state, (3) *active*, otherwise.

4. Implementing focal points

The implementation of the focal point has an important role in the efficiency of our read/write/recovery protocols that are built on top of this abstraction. In this section we briefly describe two implementations proposed in (4), and then sketch our randomized implementation that reduces the amount of message transmissions and collisions.

Intuitively our goal is to “collapse” at any time t the mobile nodes contained in a focal point region G_i at t into a *virtual static* node associated to G_i . This implies that at the copy maintained by each mobile node contained in G_i at time t must be consistent with the others. This can be guaranteed since each node contained in G_i can reliably transit and receive messages sent by another node in G_i using the Lbcst service (see Section 3). Note that the mobile nodes contained in G_i can follow different strategies each time a read/write request reaches G_i via the VLbcst service. A simple approach requires that each node in G_i replies. However, this approach is energy-consuming since the number of broadcasts performed is equal to the number of nodes currently contained in G_i , and it is likely to cause message collisions. An alternative implementation sketched in (4) that addresses both problems relies on a *leader* node that is elected locally among the nodes contained in G_i at that time. However, the cost of computing and maintaining a leader is noticeable in case of high node mobility.

Our approach is probabilistic. Upon receiving a message each node waits for a random delay c before performing a broadcast, where c is uniformly chosen at random in $[0, \Lambda]$, and Λ is a network parameter (e.g., much larger than the maximum expected number of nodes in a focal point region). More precisely, a mobile node transmits a message m only if none of its neighbors has transmitted m yet. This simple approach reduces the

number of transmissions and collisions without the overhead of maintaining a leader since a node transmits only if required, that is if none of its neighbors has broadcast a reply. However, this is done at the cost of higher communication latency of the VLbcst service. We refer the reader to (16) for the analysis.

Note that the `join` protocol, run by each node upon entering a focal point region, is very important to determine the recovery of a focal point. A mobile node C triggers a recovery *as soon as* it passes the proximity region and enters into an *empty* focal point region (previous faulty region). More precisely, as soon as C enters a new region G_i , it broadcasts a *join request* via Lbcst, and waits for a reply. Since the Lbcst service guarantees reliable delivery, C triggers a focal point recovery only if it does not receive any reply message within $\Lambda + d$ time units, where d is the time critical path for the Lbcst service defined in Section 3.

5. Our mobility model

As mentioned in the Introduction, we transform the problem of ensuring data consistency in case of high node mobility into the problem of ensuring data consistency in case of continuous static node failures (failures of the stationary focal points). More precisely, we tolerate high node mobility and the unknown motion and speed of nodes by tolerating continuous and unbounded focal point failures. Clearly, in order to derive strong data guarantees we need to define some conditions regarding the failure rate of the focal points. In the following section we describe the focal point failure model, which represents our mobility constraints, and then discuss the case of low density networks.

5.1. Focal point failure model

The focal point failure model (our mobility constraints) consists of two conditions. The first condition defines the minimum number of focal points (active and recovering) that are connected at any time, while the second condition limits the number of failures that can occur during the expected round trip delay between any two nodes in the network. In fact, limiting the number of faulty focal points at any time is not sufficient to guarantee data availability since a focal point recovery completes only if a sufficient number of active focal points remains available during the time interval elapsed between its invocation and response time. Therefore, in order to guarantee data availability we also need to bound the failure rate between the invocation and response time of a distributed operation. In fact, if the failure rate exceeds the recovery rate at some point, the system can fall into a *stale* condition where focal points cannot complete their recovery and data is unavailable.

Our failure model is parametric in the maximum number f of faulty focal points that are tolerated by the system. This parameter depends on the specific implementation and varies in $[0, n - 3)$ (we will discuss it in depth in Section 8). Our failure model consists of the following assumptions:

- A_1 : at any time, there are at most f faulty focal points.

- A_2 : at most α focal point failures can occur during τ time units, where $\alpha \geq 1$.

Although these assumptions look very similar, they are different in nature. In fact, assumption A_1 regards a *snapshot* of the system taken at a specific point in time. It provides an upper bound f on the number of faulty focal points present in the system at any time. This assumption is related to the geographical coverage of the mobile nodes in G since it assumes the existence of $n - f$ active and recovering focal points. Note that this condition does not imply that nodes are uniformly distributed in the remaining $n - f$ regions since some subregions can be highly populated and others could contain only one mobile node. Assumption A_1 says that the mobile nodes *cover* at any time at least a $\frac{n-f}{n}$ fraction of the system region G and that $n - f$ subregions are connected. This assumption is realistic for most mobile applications where a majority of nodes move approximately according to some pattern or are task-driven (e.g., disaster relief applications, or monitor of animals, such as herd). In Section 8 we show that our assumptions are reasonable also in case nodes move independently. Note that at this stage of the work we do not consider network partitions.

Assumption A_2 provides an upper bound on the failure rate during τ time units, which is the expected maximum round-trip delay between any two nodes. This assumption is related to the density of the nodes and their maximum speed. In fact, the probability that an active focal point F_i fails during τ time units is equal to the probability that each node contained in G_i crashes, or leaves the region during those τ time units. Therefore, if the node speed is smaller than $\frac{a}{\tau}$, then the probability that F_i fails during τ time units is smaller than the probability that G_i contains only one node whose distance from the border of G_i is smaller than a .

5.2. Sparse networks

Assumption A_2 is reasonable in most mobile networks, but it could be invalidated in case of sparse networks. (e.g., a mobile node travels across the border of some focal point region, thus causing frequent failures.) This occurs if during τ time units the only mobile node contained in G_i crosses more than α times the border of some focal point region leaving more than α empty subregions. This problem can be solved using the *proximity regions* defined in Section 3. In fact, if the maximum speed of the node is $\frac{a}{\tau}$, then we can consider a set of proximity regions with $\nu = a$.

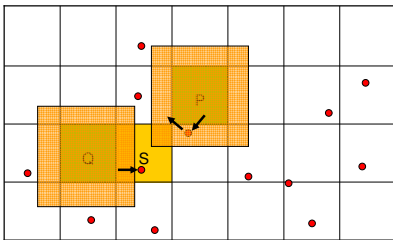


Fig. 5. Mobile nodes in the proximity regions.

According to our system model each mobile node in a proximity region can communicate with each node in G_i and in any adjacent region. Therefore, if a node C contained in G_i leaves G_i and joins its empty adjacent region G_j , then C does not trigger a failure of F_i and a recovery of F_j if it is in $Prox(G_i, a)$. Figure 5 graphically illustrates the motion of two nodes during τ time units. The node motion is indicated by black arrows. The node previously contained in the focal point region of P enters the proximity region of P and then changes direction. Note that in both cases the node does not trigger any focal point recovery. The other node is initially contained in the proximity region of Q (it has not triggered yet a recovery of S). It triggers a recovery of S only upon leaving the proximity region of Q . In Figure 4 of Section 3, focal point A remains active until the car leaves its proximity region.

6. Mobile quorum systems

In this section we investigate quorum systems in highly mobile networks in order to reduce the communication cost associated with each distributed operation. Our analysis is driven by two main reasons: (1) guarantee data availability, and (2) reduce the amount of message transmissions, thus conserving energy. As mentioned in the Introduction, the availability of the data is strictly related to the liveness and response time of the recovery protocol since the focal point failures occur continuously, as they are triggered by the motion of nodes.

Quorum systems are well-known techniques designed to enhance the performance of distributed systems, such as to reduce the access cost per operation and the load. A quorum system of a universe U is a set of subsets of U , called *quorums*, such that any pair of quorums do intersect. In (16) we have analyzed quorum system techniques in the specific context of wireless sensor networks and showed that quorum systems can lead to noticeable energy savings if appropriately adapted. In this paper we analyze quorum systems in condition of high node mobility, and more precisely under our mobility model. Note that the universe U of our quorum systems is \mathcal{FP} , a set of n stationary focal points. This choice allows us to study node mobility in terms of continuous failures of stationary nodes.

In Section 6.1 we analyze two examples of quorum systems and show that they are not always able to guarantee data consistency and availability under our mobility constraints, and provide in Lemma 1 a condition on the size of the minimum quorum intersection that is *necessary* to guarantee these properties. This condition is the basis for our class of quorum systems, which is defined in Section 6.2. Note that for simplicity of presentation in assumption A_2 we set α to 1.

6.1. Quorum systems under our mobility model

We show that quorums proposed for static networks are not able to guarantee data consistency and availability if assumptions A_1 and A_2 hold, because the minimum quorum intersection is not sufficiently large to cope with the mobility of the nodes. In fact, since read/write operations are performed over

a quorum set, in order to guarantee data consistency each read quorum must intersect a quorum containing the last update. We show that there are scenarios that invalidate this condition in case of quorum systems \mathcal{Q}_g with non-empty quorum intersection, and in case of *dissemination quorum systems* \mathcal{Q}_d (10) with minimum quorum intersection equal to $f + 1$, where f is the maximum number of failures.

Generic quorum system. It is a set of subsets of a finite universe U such that (1) any two subsets (*quorums*) intersect (consistency property), and (2) there exists at least one subset of correct nodes (availability property). The second condition ensures data availability and poses the constraint $f < \frac{n}{2}$. In our system model where nodes continuously fail and recover, this condition is not sufficient to guarantee data availability. For instance, in an implementation of a read/write atomic memory based on \mathcal{Q}_g , the liveness of the read protocol can be violated since it terminates only after receiving a reply from a *full quorum* of active focal point. Therefore, since the recovery operation involves a read operation, data can become unavailable.

Dissemination quorum systems (10). They satisfy a stronger consistency property, but insufficient if failures occur continuously. The following definition was used in (10) to introduce a dissemination quorum system. An f -fail-prone system $\mathcal{B} \subset 2^U$ of U is defined as a set of subsets of faulty nodes of U none of which is contained in another, and such that some $B \in \mathcal{B}$ contains all the faulty nodes (whose number does not exceed f).

Definition 3 A dissemination quorum system \mathcal{Q} of U for a f -fail-prone system \mathcal{B} , is set of subsets of U with the following properties:

- (i) $|Q_1 \cap Q_2| \not\subseteq B \quad \forall Q_1, Q_2 \in \mathcal{Q}, \forall B \in \mathcal{B}$
- (ii) $\forall B \in \mathcal{B} \exists Q \in \mathcal{Q} : Q \cap B = \emptyset$.

As shown in (10), dissemination quorum systems tolerate less than $\frac{n}{3}$ failures. Unfortunately, since in our system model an additional focal point might fail between the invocation and the response time of a distributed operation (see Assumption A_2), more than f focal points in a quorum set can be non-active at the time they receive the request. As a result, data availability can be violated.

The following lemma provides a condition on the minimum quorum intersection size (lower bound) that is *necessary* to guarantee data consistency and availability under our system model, provided nodes fail and recover. We refer the reader to the Appendix for the proof.

Lemma 1 An implementation \mathcal{I} of a read/write shared memory built on top of a quorum system \mathcal{Q} of a universe \mathcal{FP} of stationary nodes that fail according to assumptions A_1, A_2 and recover, guarantees *data availability* only if $|Q_1 \cap Q_2| > f + 1$ for any $Q_1, Q_2 \in \mathcal{Q}$. It ensures *atomic consistency* only if $|Q_1 \cap Q_2| > f + 2$ for any $Q_1, Q_2 \in \mathcal{Q}$.

6.2. The MDQ quorum systems

We introduce here a new class of quorum systems, called *mobile dissemination quorum systems* (MDQ) that satisfies the condition in Lemma 1.

Definition 4 A MDQ system \mathcal{Q} of U is set of subsets of U such that $|Q_1 \cap Q_2| > f + 2 \quad \forall Q_1, Q_2 \in \mathcal{Q}$.

Note that in contrast with \mathcal{Q}_g the liveness of the distributed operations performed over a quorum set is guaranteed by the *minimum number* of correct nodes contained in any quorum. As a result in case of failures the sender does not need to access another quorum in order to complete the operation. This improves the response time in case of faulty nodes and reduces the message transmissions. Let us consider now the following MDQ system:

$$Q_{opt} = \left\{ Q : \left(Q \subseteq \mathcal{FP} \right) \wedge \left(|Q| = \left\lceil \frac{n+f+3}{2} \right\rceil \right) \right\}$$

Lemma 2 Q_{opt} is a MDQ system and $f \leq n - 3$.

Proof Since $|Q_1 \cup Q_2| = |Q_1| + |Q_2| - |Q_1 \cap Q_2|$ for any $Q_1, Q_2 \in Q_{opt}$, and $|Q_1 \cup Q_2| \leq n$, then $|Q_1 \cap Q_2| \geq n + f + 3 - n$. In addition, Q_{opt} tolerates up to $n - 3$ failures since the size of a quorum cannot exceed n , that is $\left\lceil \frac{n+f+3}{2} \right\rceil \leq n$ which implies $\frac{f+3-n}{2} \leq 0$. Qvd

Note that Q_{opt} is highly resilient (in the trivial case $f = n - 3$, $Q_{opt} = \{U\}$). Clearly, there is a trade-off between resiliency and access cost since the access cost per operation increases with the maximum number of failures. Moreover, our assumption of connectivity among active focal points becomes harder to guarantee as f becomes larger.

It is important to note that the minimum intersection size between two quorums of Q_{opt} is equal to $f + 3$. We prove in the following section that there exists an implementation of atomic memory built on top of Q_{opt} . This shows that $f + 3$ is the minimum quorum intersection size necessary to guarantee data consistency and data availability under our mobility model. Therefore, Q_{opt} is *optimal* in the size of the minimum quorum intersection, that is in terms of message transmissions since the sender can compute a quorum consisting of its $\left\lceil \frac{n+f+3}{2} \right\rceil$ closest nodes. This is particularly advantageous in sensor networks because it can lead to noticeable energy savings.

7. An implementation of read/write atomic memory

In this section, we show that Q_{opt} is the quorum system with minimum intersection size $f + 3$ that is able to guarantee data consistency and availability under our system model and mobility constraints. We prove that by showing that there exists an implementation \mathcal{I} of atomic read/write memory built on top of the focal points and Q_{opt} . Our implementation consists of a suite of read, write and recovery protocols similar to (4) and built on top of the focal points and on the Qbcst abstraction.

7.1. The Qbcst service

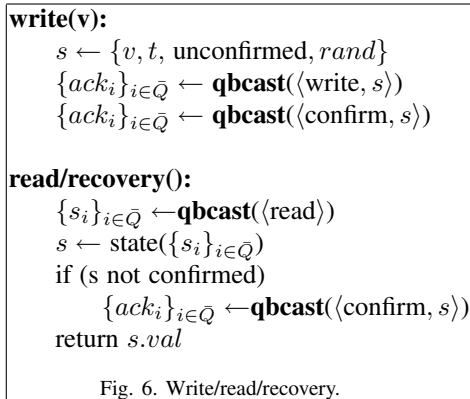
In this section we refine the VLbcst service and the definition of faulty focal points in Section 3 for the MDQ systems. We say that a focal point F_i is *faulty* at time t if G_i does not

contain any active node at time t , or F_i is not connected to a quorum of focal points. In our implementation each read, write and recovery request is forwarded to a quorum of focal points. This task is performed by the *Qbcast* service, which is a refinement of the VLbcast service. It is tailored for the MDQ system and designed for hiding lower level details. Similarly to VLbcast, Qbcast guarantees reliable delivery. It is invoked using interface $\text{qbcast}(m)$, where m is the message to transmit containing one of these request tags `write`, `read`, `confirm`. In Figure 6 the notation $\{s_i\}_{i \in \bar{Q}} \leftarrow \text{qbcast}(m, Q)$ denotes the Qbcast invocation, $\{s_i\}_{i \in \bar{Q}}$ the set of replies, where $\bar{Q} \subseteq Q$. We call the subset \bar{Q} the *reply set* associated with request m . This set plays a crucial role to prove data availability and atomic consistency (see Section 7.3). Upon receiving a request m , Qbcast computes a quorum $Q \in Q_{opt}$ and transmits message m to each focal point in Q using the VLbcast service. It is important to note that $\text{qbcast}(m)$ returns *only if* the node receives within τ time units at least $|Q| - (f + 3)$ replies from Q . If this does not occur, it waits for a random delay and retries later since if this happens the focal point is faulty by our definition.

Note that if read (or write) operations occur more frequently than write (or read) operations, we can reduce message transmissions by distinguishing between read and write and making read (or write) quorums smaller. However, for simplicity of presentation we do not distinguish between read and write quorums.

7.2. Protocols

The high level description of the read/write/recovery protocols is illustrated in Figure 6. Each mobile node maintains a copy of the state s associated with the shared variable x , which is a compound object containing the value $s.val$ of x , a timestamp $s.t$ representing the time at which a node issued update $s.val$, and a `confirmed` tag that indicates if $s.val$ was propagated to a quorum of focal points. Each node can issue `write`, `read` and `recovery` operations. A new state is generated each time a node issues a write operation.



Write protocol. A node C requesting a write v computes a new state s consisting of value v , the current timestamp t , tag `unconfirmed`, and a random identification `rand`. It transmits its update to a quorum of focal points via the Qb-

cast service by invoking $\text{qbcast}(\langle \text{write}, s \rangle)$, and successively $\text{qbcast}(\langle \text{confirm}, s \rangle)$ to make sure that a quorum of focal points received such an update (4).

Upon receiving a write request, each non-faulty focal point (including recovering) replaces its state with the new state s only if the associate timestamp $s.t$ is higher than the timestamp of its local state, and sets its write tag to `unconfirmed`. This tag is set to `confirmed` upon receiving the confirm request sent in the second phase of the write protocol, or sent in the second phase of the read protocol in case the node that issued the write operation could not complete the write operation due to failure.

Read protocol. In the read protocol, a node C invokes $\text{qbcast}(\langle \text{read} \rangle)$, which forwards the read request to a quorum Q of focal points. Each non-faulty focal point in Q replies by sending a copy of its local state s . Upon receiving a set of replies from the Qbcast service, node C computes the state with highest timestamp and returns the corresponding value. Similarly to (4), if the tag of s is equal to `unconfirmed`, it sends a confirm request. This is to guarantee the linearizability of the operations performed on the shared data in case a write operation did not complete due to client failure.

Recovery protocol. It is invoked by a node C upon entering an empty region G_i . More precisely, C broadcasts a `join` request as soon as it enters a new focal point region and waits for replies. If it does not receive any reply within $2d$ time units, where d is the maximum transmission delay, it invokes the recovery protocol which works in the same way as the read protocol.

7.3. Analysis

In this section we define a partial order on the set of operations for a given execution, and show the key steps to prove the atomic consistency and data availability of the implementation presented in the previous section. We refer the reader to the Appendix for the proofs. Note that the main difficulty of proving these properties comes from the fact that data availability and atomic consistency strictly depend on each others since failures occur continuously. We brake this tie by having each node reply to a recovery request with its local state, and by proving first data availability.

Partial order on the state of focal points. We associate a state to each operation performed during any system execution Γ as follows: let $\mathcal{O}_\Gamma = \{o_1, \dots, o_j, \dots\}$ be the set of the write/read/recovery operations in Γ and $\mathcal{S}_\Gamma = \{s_1, \dots, s_j, \dots\}$ the set of their associated states such that s_i is the state associated to operation o_i . The state associated with a write operation is the state generated by the client that issues that write operation, and the state associated with a read operation is the state computed by the client which issues that read operation.

We order the states based on their timestamps and their unique identification numbers, in case of concurrent operations. We define a relation \leq_s over set \mathcal{S}_Γ such that $\forall s_1, s_2 \in \mathcal{S}_\Gamma$, $s_1 \leq_s s_2$ if and only if $((s_1.t < s_2.t) \vee (s_1.t = s_2.t \wedge s_1.id < s_2.id)) \vee (s_1 = s_2)$. Relation \leq_s is a *partial order* on set \mathcal{S}_Γ since it satisfies reflexivity, antisymmetry, and transitivity.

Before defining our partial order on the operations in \mathcal{O} , we recall the definition of the natural order $<_n$ among operations defined in (7): $o_1 <_n o_2$ if the response time of o_1 precedes the invocation time of o_2 , that is if $res(o_1) < inv(o_2)$. We define a relation $<_a$ on set \mathcal{O}_Γ , as follows:

$$o_1 <_a o_2 \text{ if } \begin{cases} o_1 <_n o_2, \text{ if } o_1, o_2 \text{ read operations;} \\ s_1 \leq_s s_2, \text{ else.} \end{cases}$$

Clearly, relation $<_a$ is a partial order on \mathcal{O}_Γ . We show briefly the main steps to prove data availability and atomic consistency. We refer the reader to the Appendix for the proofs.

Data availability. The availability of the data is a direct consequence of our failure model, and the Qbcast service. The following lemmas are useful to prove it and will be also used in showing atomic consistency.

Lemma 3 The Qbcast service invoked by an active focal point terminates within τ time units since the invocation time.

The following lemma and Theorem 1 is a straightforward derivation of the liveness of the Qbcast service.

Lemma 4 An active focal point recovers within τ time units.

Theorem 1 Our implementation of atomic read/write shared memory guarantees data availability.

Lemma 5 At any time in the execution there are at most $f + 1$ faulty and recovering focal points.

Atomic consistency. We need to show that the total order is consistent with the natural order of invocations and response. That is, if o_1 completes before o_2 begins, then $o_1 <_a o_2$. The following lemmas provide crucial properties to guarantee that the state returned by a read/recovery operation does not precede the state associated with the last completed update.

Lemma 6 The reply set \bar{Q} associated with a request satisfies the following properties:

- (i) $|\bar{Q}| \geq \lceil \frac{n-f}{2} \rceil$;
- (ii) $|\bar{Q} \cap Q| \geq 2 \quad \forall Q \in \mathcal{Q}_m$.

Lemma 7 Let o_1 be a write operation whose state is s_1 . Then, at any time t in the execution with $t > res(o_1)$ there exists a subset M_t of active focal points such that,

- (i) $|M_t| \geq \lceil \frac{n-f}{2} \rceil - 1$ (equality holds only if f focal points are faulty and one is recovering);
- (ii) the state \bar{s} of its *active* focal points at time t is such that $s_1 \leq_s \bar{s}$;

Theorem 2 Our implementation of an atomic shared read/write memory satisfies atomic consistency.

7.4. Remarks

Our implementation relies on the assumption that f is an upper bound for the number of focal point failures (empty focal point subregions and focal points that are not connected to a quorum of focal points). However, as our simulation results show in Section 8.2, the number of faulty focal points can noticeably change during the system lifetime depending on the

distribution of nodes and their motion, especially in low density networks. Moreover, assuming a large upper bound during the entire system lifetime is not energy-efficient since the size of quorums grows according to f . Therefore, the choice of a conservative upper bound results in high communication cost and energy consumption.

This problem can be addressed by using a dynamic upper bound f that is adjusted when needed, according to the node motion and distribution. This can be achieved using our implementation \mathcal{I} in which the shared memory is the current upper bound of the faulty focal points. In fact, each focal point can get an estimate f_{est} (partial view) of the current faulty focal points each time it invokes the Qbcast service (at no additional communication cost). In case of unfrequent read/write or recovery operations the focal point can proactively monitor the number of faulty focal points. Therefore, a focal point updates the current upper bound as soon as its estimate f_{est} , obtained by the Qbcast service, exceeds the current bound f_c (value of the shared variable) minus a system parameter γ . That is, if $f_{est} - (f_c - \gamma) \geq 0$ the focal point sends an update for the current upper bound and sets it to $f_{est} - f_c + \gamma$. Similarly, the upper bound can be reduced in case of persistent reduction of the faulty focal points.

8. Simulation results

In this section we analyze the suitability of our mobility constraints and the efficiency of Q_{opt} through simulations using the random waypoint and the restricted random waypoint on a city section (25). More precisely, in Section 8.1 we describe our simulations and the tool (23) used to simulate the motion of the nodes. Then, we study the parameter f and related metrics using different node density and node speed. In Section 8.3 we discuss the efficiency of Q_{opt} .

8.1. Simulation setting

We simulate the motion of the nodes using a tool ² (23) that implements the random trip model, which is a generic mobility model that generalizes random waypoint and random walk to realistic scenarios. More precisely, it implements the perfect sampling model proposed in (24) and the perfect sampling algorithm, which has the benefit of not requiring known geometric constants or the average distance between two random points in a graph that maybe difficult to compute. In fact, this tool suffices to know an upper bound on the distance between any two points in the domain. It generates a mobility trace file in ns2-compatible format containing the following commands: the position of each node N is initialized to $(X1, Y1, Z1)$ at time TIME and the trip destination point is set to $(X2, Y2, 0)$ and numeric speed is SPEED, which is drawn uniformly at random in $[v_{min}, v_{max}]$

```
ns at TIME node(N) set X1
ns at TIME node(N) set Y1
```

² It is available at <http://www.cs.rice.edu/santa/research/mobility>.

```
ns at TIME node(N) set Z1
ns at TIME node(N) setdest X2 Y2 SPEED
```

We use the trace generated by this tool to monitor the number of empty focal point regions $f_r(t)$ at time t , and the number of faulty focal points $f_p(t)$. This task is performed every 10 seconds. More precisely, we partition the geographical area of the system into a square grid, as discussed in Section 3, and detect the subregions that do not contain any node using the position of the nodes described in the output trace. Since $f_p(t)$ is equal to $f_r(t)$ plus the number of focal points that are not connected to a quorum of focal points (see Section 7), we set $f_p(t)$ to $n - c$, where n is the total number of focal points and c is the size of the maximum connected component of a graph whose vertex are the subregions that are populated at time t . Note that since the focus of this paper is on the analysis of our mobility constraints and mobile quorums in terms of focal points, we do not simulate at this stage message transmissions and the read/write protocols. In our simulation results we use the random waypoint and the restricted random waypoint on a city section implemented in (23). We refer the reader to (25) for a survey on mobility models.

Random waypoint. At a trip transition instant, a node picks a trip destination uniformly at random on a rectangular area and samples a numeric speed from a uniform distribution in $[v_{min}, v_{max}]$, where v_{min} and v_{max} are user-defined parameters. The trip path is the straight line that connects node positions at this and next trip transition instant. Upon reaching the trip destination, the node may pause for a random time drawn from a uniform distribution in $[t_{min}, t_{max}]$, where t_{min} and t_{max} are user-defined parameters. This trip selection rule repeats. A default initialization rule is to set the node at time 0 to either move or pause phase and specify time 0 as a trip transition instant. This tool allows the user to define the system region G , the number of mobile nodes contained in G , $v_{min}, v_{max}, t_{min}, t_{max}$, and the execution time.

Our simulation results described in the following sections refer to a squared 200×200 meters system region G partitioned into a 10×10 squared grid of length 20 meters, which assumes a radio broadcast of $40\sqrt{2}$ meters. Our simulation results refer to one day execution.

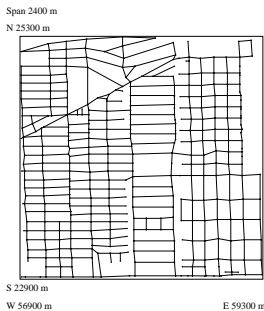


Fig. 7. Realistic street scenario corresponding to a square area of size 1200×1200 m. The scenario consists of 383 intersections and 594 road segments.

Restricted random waypoint on a city section. This is a particular instance of random waypoint on a general connected area. The domain is the union of line segments defined by the edges of a given space graph. The tool generates a ns-2 trace for a given number of mobile nodes, simulation duration and space graph containing road id, average road speed and coordinates of the road endpoints. We analyze traces relative to a real road map of a residential 1200×1200 meter area closed to Rice University¹ provided by (23) and illustrated in Figure 7. We partition this area into 12×12 squared grid of length 100 meters. Note that this area contains several dead end streets that increase the probability of network partitions. We denote this model as CityRW, and the random waypoint over a 10×10 grid as RW.

8.2. Focal point failures

In this section we analyze our mobility model over different node densities and node speed, and show that assumptions A_1 and A_2 defined in Section 5.1 in terms of focal points are realistic in several cases. More precisely, we study the parameter f (see assumption A_1), and the impact that the number of mobile nodes contained in G and their speed and variance have on f . Our goal is to show that it is possible to compute an estimate of f and adapt it when needed, as discussed in Section 7.4. From now on we denote the focal points as FPs, and the focal point regions as FP regions.

In this section we analyze the variation of $f_p(t)$ (number of faulty FPs at time t) over the system lifetime using RW and CityRW. Then, we study the impact that node density and node mobility have on f using the following 5 metrics:

- the maximum and average number of faulty FPs f during the system lifetime;
- the maximum and average number of empty FP regions f_r during the system lifetime;
- the average increment/decrement of f_p during a time interval of 10 seconds under different node density and node speed.

Variation of Faulty FPs over the Time. Figure 8 shows function $f_p(t)$ over one day using the RW model described in Section 8.1 and different sets of mobile nodes. It refers to a 10×10 grid (100 FPs) populated by mobile nodes whose speed is chosen uniformly at random in $[5, 7]$ m/sec and whose pause is chosen uniformly at random in $[40, 60]$ seconds. Figure 8 shows the variation of the faulty FPs over the time in the presence of 75, 100, 200, 500 mobile nodes (i.e., with an initial average distribution of 0.75, 1, 2 and 5 mobile nodes per FP region). Figure 8 shows that the number of faulty focal points decreases as the number of mobile nodes increases, with a dramatic drop going from 100 to 200 nodes. We can also observe that the variation of faulty FPs during the system lifetime is larger for small sets of nodes (e.g., $f_p(t)$ has several spikes in case of 75 nodes). This fact is not surprising since the number of faulty FPs highly

¹ The detailed maps are available from the United States census Bureau's TIGER (Topologically Integrated Geographical Encoding and Referencing database.)

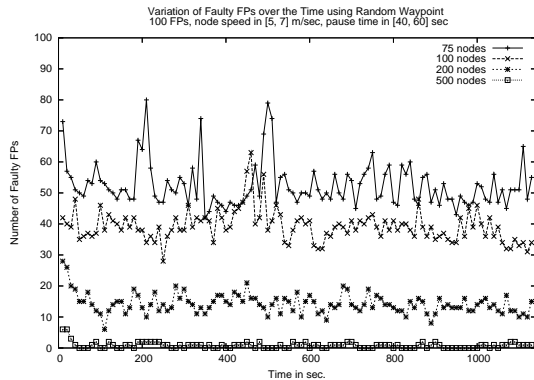


Fig. 8. Variation of faulty FPs over the time using RW.

depends on the distribution and motion of the nodes in case of sparse networks (see Section 5.1). In our experiments we have also noticed that the variability of the time pause affects function $f_p(t)$ and the increment/decrement of $f_p(t)$ during a small time interval (e.g., 10 seconds). For instance, function $f_p(t)$ is smoother if the time pause is contained in [8, 12] seconds. This is not surprising since the variability of the nodes increases in case of larger pause intervals.

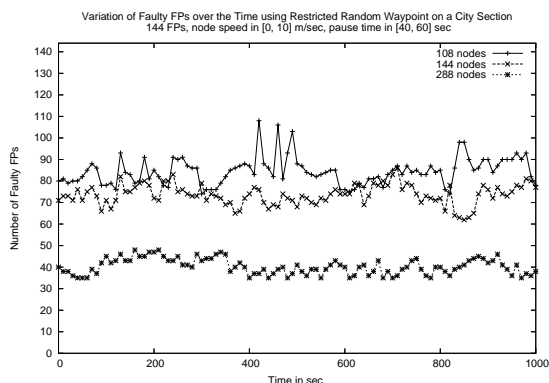


Fig. 9. Variation of faulty FPs over the time using CityRW.

We run the same experiments using the CityRW model described in Section 8.1 consisting of 144 FPs. Figure 9 shows function $f_p(t)$ during one day using 108, 144, 288 mobile nodes, which corresponds to an initial average distribution of 0.75, 1, 2 nodes per FP region. A comparison between Figure 8 and Figure 9 shows that the topological restrictions of CityRW reduces the variability of $f_p(t)$ in case of low density nodes.

Faulty FPs vs. Mobile Nodes. We analyze more in depth the impact that the number of mobile nodes have on the faulty FPs and connectivity by distinguishing between empty FP regions and faulty FPs. More precisely, we compute for each set of mobile nodes the percentage of the maximum and average number of faulty FPs, denoted as $max(f_p)$, $avg(f_p)$, the percentage of the maximum and average number of empty FP regions denoted as $max(f_r)$, $avg(f_r)$, and the percentage of the average increment/decrement of faulty FPs during 10 seconds denoted as $\Delta(f_p)$. Figure 10 shows the variation of these parameters.

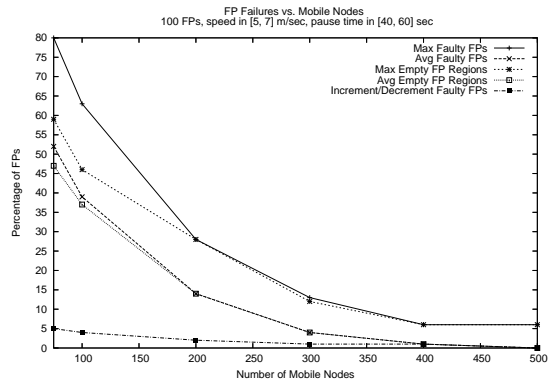


Fig. 10. Variation of faulty FPs over node sets using RW.

Note that $max(f_p) - max(f_r)$ dramatically decreases as the set of mobile nodes increases (i.e., it is negligible in the presence of at least 200 nodes). This is obvious since $f_p(t) - f_r(t)$ represents the number of FPs that are not connected to a quorum of FPs, and the network connectivity increases as the number of mobile nodes increases. We can also observe that the difference between the maximum and average number of faulty FPs $max(f_p) - avg(f_p)$ decreases as the number of mobile nodes increases. In fact, it represents the variability of the number of focal points, which is larger in case of sparse networks since the FP failures are strongly dependent on the distribution and speed of nodes.

We run the same experiments using the CityRW model and observed a similar behavior. However, $max(f_p)$ is slightly smaller in CityRW than in RW, while $avg(f_p)$ is slightly larger. This implies that the variation of the number of faulty FPs during the system lifetime is smaller in our CityRW model than in our RW model. Moreover, $max(f_p) - max(f_r)$ is smaller in CityRW. All of these observations are motivated by the restricted topology of CityRW that improves the network connectivity compared to RW where nodes moves without any restrictions.

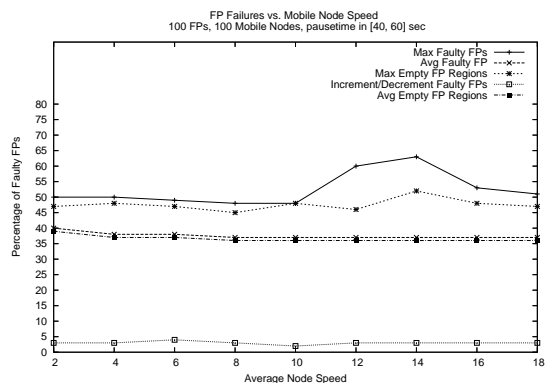


Fig. 11. Focal point failures vs. node speed.

Faulty FPs vs. Node Speed. We study the impact that node speed have on the FP failures by simulating the motion of 100

nodes over different speed using RW. More precisely, we vary the average speed of the nodes from 2 m/sec to 18 m/sec and use 1 m/sec as maximum speed variability. Figure 11 shows the variation of $max(f_p)$, $avg(f_p)$, $max(f_r)$, $avg(f_r)$, $\Delta(f_p)$ over different speed. We observe no significant variation of $avg(f_p)$, which is encouraging for our approach and confirms that mobility can improve network connectivity as shown by recent papers (8). Note that we observe a significant variation of $max(f_p)$ and $max(f_r)$ from one execution to another, especially for high average speed. This fact is not surprising since the maximum number of faulty FPs depends on the mobility pattern of the nodes that is random in our simulations. In the graph of Figure 11 the values of $max(f_p)$ and $max(f_r)$ are an average over 6 executions.

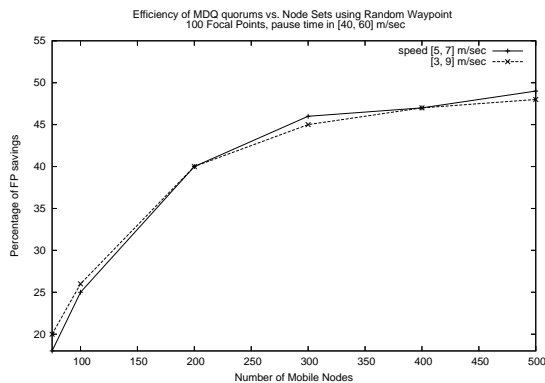


Fig. 12. Quorum size vs. node density and node speed using RW.

8.3. MDQ systems

Our MDQ system allows the sender to choose the subset containing the closest FPs using geographic information. As a result, since only one mobile node for each FP region is in charge of forwarding a message (see Section 4), the amount of message transmissions involved in each distributed operation strictly depends on the number of FPs in a quorum.

In this section we analyze the quorum size of Q_{opt} under different node density and node speed. For each set of parameters we compute the size of quorums in Q_{opt} that is equal to $\lceil \frac{n+f+3}{2} \rceil$, where $f = avg(f_p) + 2\Delta(f_p)$. In fact, as discussed in Section 7.4 we consider a dynamic upper bound. Figure 12 illustrates the percentage of FPs that are *not* involved in the protocol in cases the node speed is contained in $[5, 7]$ m/sec and it is contained in $[3, 9]$ m/sec. This percentage indicates the reduction of message transmissions. Figure 12 shows that the FPs saving is larger than 40% if the average initial distribution accounts of at least 2 nodes per FP region.

9. Applications

As discussed in the Introduction the implementation \mathcal{I} of an atomic read/write shared memory has several applications to

MANETs. For instance, it is crucial in network tasks requiring coordination among nodes. In this section we briefly illustrate how to apply \mathcal{I} to coordinate the low duty cycle of sensor nodes while maintaining network connectivity and to assist routing.

Energy conservation represents a crucial issue in wireless sensor networks because of the limited energy source. A technique used to reduce energy consumption is to periodically turn off the node's radio for a given interval. Clearly, the low duty cycle of nodes must be coordinated in order to maintain network connectivity. In fact, the lack of node coordination can cause message lost and affect the high-level application. Our implementation \mathcal{I} can be applied to address this problem by sharing information regarding the sleep/awake nodes.

Let us consider a sensor network where sensor nodes move within a geographic region G (e.g., sensors placed on moving objects, such as in ZebraNet (34)), and a n -region vector $\langle G_1, \dots, G_n \rangle$ of G . Let us suppose that each sensor node maintains a n -vector V such that $V[i]$ contains information associated with the focal point F_i , regarding for instance the low duty cycle of the nodes contained in G_i and their energy budget. For instance, $V[i]$ may consist of the following fields associated with a specific point in time t :

- $V[i].s$, a boolean variable that is equal to 1 if G_i contains at least a node whose radio is turned on at time t ;
- $V[i].e$, the sum of the energy budget of the nodes contained in G_i at that time;
- $V[i].r$, the compound reliability of the focal point, which is equal to 1 minus the probability that each node in G_i will fail or leave G_i in the next Γ time units.

Our implementation \mathcal{I} can be applied to guarantee the consistency of vector V whose data changes according to the node motion as follows. Each time a node leaves a FP region G_j and joins an adjacent region G_i it performs a write/read operation. It broadcasts a join message along with its energy budget. Nodes contained in G_i and G_j update entries $V[i]$ and $V[j]$, and a random node in G_i acting as a focal point leader (see Section 4) performs a write operation to propagate the updated $V[i]$ and $V[j]$. Note that $V[j].s$ is set to 0 if no other active node is contained in G_i .

The information contained in vector V can be applied to a number of critical network tasks, such as the coordination of the low duty cycle of focal points and of its associated nodes, or energy management. The data contained in V can be used in routing to choose the most reliable route and whose nodes have highest energy budget. Moreover, the information contained in vector V can be used to study the distribution of mobile nodes and find patterns (e.g., in ZebraNet the behavior of zebras).

10. Conclusions and future work

We have devised a small set of mobility constraints that are necessary to ensure strong data guarantees in highly mobile networks. Our mobility model improves previous work for relaxing assumptions on the motion and speed of the nodes during the system lifetime. We have also investigated quorum systems in highly mobile networks and devised a condition that is

necessary for a quorum system to guarantee data consistency and availability under our mobility constraints. This condition shows that previous quorum systems are not able to guarantee data consistency under our mobility model. We have proposed a class of mobile quorum systems and have computed a quorum system that is provably optimal in terms of communication cost. Our simulation results performed using the random waypoint and the restricted random waypoint on a city section, confirm our theoretical study.

Our work leaves several open questions such as the problem of dealing with network partitions and periods of network instability in which our set of assumptions are invalid. We are currently working on defining weaker consistency guarantees and on extending our protocols to address these cases.

References

- [1] S. Bhattacharya. *Randomized location service in mobile ad hoc networks*. In Proc. of the 8th Intl. ACM Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 66–73, Sep 2003.
- [2] T. Camp, Y. Liu. *An adaptive mesh-based protocol for geocast routing*. J.Parallel and Distributed Computing: Special Issue on Mobile Ad-hoc Networking and Computing, pp. 196–213, 2002.
- [3] S. Dolev, E. Schiller, and J. Welch. *Random walk for self-stabilizing group communication in adhoc networks*. In Proc. of the 21st Symp. on Reliable Distributed Systems, pp. 70–79, 2002.
- [4] S. Dolev, S. Gilbert, N. Lynch, A. Shvartsman, J. Welch. *GeoQuorums: Implementing Atomic Memory in Mobile Ad Hoc Networks*. In Proc. of the 17th Intl. Conf. on Distributed Computing, pp. 306-320, October 2003.
- [5] S. Gilbert, N. Lynch, A. Shvartsman. *RAMBO II: Rapidly reconfigurable atomic memory for dynamic networks*. In Proc. of the Intl. Conf. on Dependable Systems and Networks, pp. 259–269, June 2003.
- [6] Z. J. Haas and B. Liang. *Ad hoc mobile management with uniform quorum systems*. IEEE/ACM Transactions on Networking, 7(2):228–240, April 1999.
- [7] M. Herlihy, J. Wing. *Linearizability: A correctness condition for concurrent objects*. ACM Trans. on Programming Languages and Systems, 12(3):463–492, July 1990.
- [8] B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley. *Mobility improves coverage of sensor networks*. In Proc. of MobiHoc, pp. 300-308, May 2005.
- [9] D. Malkhi, M. Reiter, A. Wool. *The Load and Availability of Byzantine Quorum Systems*. SIAM J. Comput. 29(6), pp. 1889-1906 (2000).
- [10] D. Malkhi, M. Reiter. *Byzantine Quorum Systems*. Distributed Computing, 11(4), pp. 203-213 (1998).
- [11] R. Friedman, M. Gradinariu, G. Simon. *Locating cache proxies in MANETs*. In Proc. 5th Intl. Symp. of Mobile Ad Hoc Networks, pp. 175-186, 2004.
- [12] R. Prakash, Z. Haas, M. Singhal. *Load-balanced location management for cellular mobile systems using quorums and dynamic hashing*. Wireless Networks, 7(5), pp. 497-512, Sept 2001.
- [13] G. Karumanchi, S. Muralidharan, R. Prakash. *Information dissemination in partitionable mobile ad hoc networks*. In Proc. of Symp. on Reliable Distributed Systems, pp. 4–13, 1999.
- [14] Y. B. Ko, N. Vaidya. *Geotora: a protocol for geocasting in mobile ad hoc networks*. In Proc. of the Intl. Conf. on Network Protocols, pages 240–249, November 2000.
- [15] J. Luo, J-P. Hubaux, P. Eugster. *Resource management: PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems*. In Proc. of the 4th Intl. Symp. on Mobile ad hoc networking and computing, pp. 1-12, 2003.
- [16] D. Tulone. *Mechanisms for energy conservation in wireless sensor networks*. Ph.D. thesis, Departement of Computer Science, University of Pisa, Dec 2005.
- [17] W. Zhao, M. Ammar, E. Zegura. *A message ferrying approach for data delivery in sparse mobile ad hoc networks*. In Proc. of the 5th Intl. Sym. on Mobile ad hoc Networking and Computing, pp. 187-198, May 2004.
- [18] K. Chen, S. Shah, K. Nahrstedt. *Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks*. In Wireless Personal Communications, 21(1), pp. 49-76, Apr 2002.
- [19] H. Wu, R. Fujimoto, R. Guensler, M. Hunter. *MDDV: a mobility-centric data dissemination algorithm for vehicular networks*. In Proc. of the 1st Intl. Workshop on Vehicular ad hoc Networks, pp. 47-56, Oct 2004.
- [20] A. Vahdat, D. Beker. *Epidemic routing for partially-connected ad hoc networks*. Tech report, Duke University, 2000.
- [21] A. Smith, H. Balakrishnan, M. Goraczko, N. Priyantha. *Support for location: Tracking moving devices with the cricket location system*. In Proc. of the 2nd Intl. Conf. on Mobile systems, applications, and services, Jun 2004.
- [22] J. Polastre, J. Hill, and D. Culler. *Versatile low power media access for wireless sensor networks*. In Proc. of SenSys, 2004.
- [23] S. PalChanduri, J.-Y. Le Boudec, M. Vojnovic. *Perfect simulations for random mobility models*. Annual Simulation Symposium 2005: 72-79. Available at <http://www.cs.rice.edu/santa/research/mobility>
- [24] J.-Y. Le Boudec, M. Vojnovic. *Perfect Simulation and Stationarity of a Class of Mobility Models*. Infocom 2005.
- [25] T. Camp, J. Boleng, V. Davies. *A Survey of Mobility Models for Ad Hoc Network Research*. Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Vol. 2, No. 5, pp. 483502 (2002).
- [26] T. Hara. *Location Management of Replicas Considering Data Update in Ad Hoc Networks*. 20th Intl. Conf. AINA 2006: 753–758.
- [27] T. Hara. *Replica Location Management for Data Sharing in Mobile Ad Hoc Networks*. Journal of Interconnection Networks 7(1): 75-90 (2006).
- [28] Y. Sawai, M. Shinohara, A. Kanzaki, T. Hara, S. Nishio.

Consistency Management among Replicas Using a Quorum System in Ad Hoc Networks. MDM 2006: 128-132.

- [29] T. Hara. *Replica allocation methods in ad hoc networks with data update.* ACM-Kluwer Journal on Mobile Networks and Applications, Vol.8, No.4, pp.343-354, 2003.
- [30] L.D. Fife, L. Gruenwald. *Research issues for data communication in mobile ad-hoc network database systems.* ACM SIGMOD Record, Vol.32, No.2, pp.42-47, 2003.
- [31] F. Sailhan, V. Issarny. *Cooperative caching in ad hoc networks.* Proc. Int'l Conf. on Mobile Data Management (MDM'03), pp.13-28, 2003.
- [32] K. Wang, B. Li. *Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks.* Proc. IEEE Infocom'02, Vol.2, pp.1089-1098, 2002.
- [33] M. Mohsin, D. Cavin, Y. Sasson, R. Prakash, A. Schiper. *Reliable Broadcast in Wireless Mobile Ad Hoc Networks.* In Proc. of the 39th Annual Hawaii Intl. Conf., vol. 9, issue , Jan 2006.
- [34] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, D. Rubenstein. *Energy-efficient computing for wildlife tracking: design trade-offs and early experience with ZebraNet.* In Proc. 10th Conf. on Architectural Support for Programming Languages and Operating Systems, Oct 2002.

Appendix

Mobile quorum systems

Lemma 1. *An implementation \mathcal{I} of a read/write shared memory built on top of a quorum system \mathcal{Q} of a universe \mathcal{FP} of stationary nodes that fail according to assumptions A_1, A_2 and recover, guarantees data availability only if $|Q_1 \cap Q_2| > f + 1$ for any $Q_1, Q_2 \in \mathcal{Q}$. It ensures atomic consistency only if $|Q_1 \cap Q_2| > f + 2$ for any $Q_1, Q_2 \in \mathcal{Q}$.*

Proof We show first that if the minimum intersection size x between quorums does not exceed $f + 1$ focal points, there exists no implementation \mathcal{I} which ensures data availability.

If $x \leq f$, then the liveness of the read protocol can be compromised since the read operation completes only after receiving a full quorum of replies in order to guarantee data consistency, and each quorum can contain each time x faulty focal points. Let us suppose $x = f + 1$, and that each read/write/recovery operation completes only upon receiving at least $|Q| - f$ replies from quorum Q . Even in this case liveness can be compromised. In fact, due to different timing $f + 1$ focal points in Q can be faulty at the time they receive that request.

We prove by contradiction the condition necessary to guarantee atomic consistency. Let us suppose that there exists an implementation \mathcal{I} satisfying atomic consistency and using a quorum system \mathcal{Q} such that $x = f + 2$. Let us consider a write operation o_1 performed on quorum $W \in \mathcal{Q}$, followed by a read operation o_2 performed on quorum $R \in \mathcal{Q}$, such that $o_1 <_n o_2$. Let us denote by A the intersection $W \cap R$, where $|A| = f + 2$. Because of assumptions A_1 and A_2 , there is an execution such that $f + 1$ focal points in A become faulty by the time request o_1 reaches their focal point regions. This can happen if due to

different timing, $f + 1$ focal points are faulty at the time they receive the request o_1 , and one of these focal points, say P , recovers at time t after request o_1 reached its focal point region and before the return time of o_1 . Since the recovery operation of P and o_1 are concurrent, there is no guarantee that the P retrieves the state associated with o_1 . Let us suppose now that the remaining focal point in A that performed update o_1 , becomes faulty after the invocation time of o_2 , and right after one of the f faulty focal point recovers, such that the recovering focal point does not retrieve the state associated with o_1 . This scenario clearly violates atomic consistency. Qvd

Analysis of the implementation \mathcal{I}

In this section we prove that \mathcal{I} satisfies *atomic consistency* and *data availability*.

Data availability. The availability of the data is a consequence of our failure model, and the Qbcast service. The following lemmas are useful to prove it and will be also used in showing atomic consistency.

Lemma 3. *The Qbcast service invoked by an active focal point terminates within τ time units since the invocation time.*

Proof This is true since an active focal point or client is able to communicate with a quorum of focal points because of Definition 2, and because at most $f + 1$ focal points in Q can be faulty when the request reaches their focal point regions. In fact, because of assumptions A_1 and A_2 at most $f + 1$ focal points can appear to be faulty during τ time units. Therefore, at least $|Q| - (f + 1)$ focal points in a quorum reply. This proves our thesis since the Qbcast service guarantees reliable delivery, and the maximum round-trip transmission delay is equal to τ . Qvd

The following lemma and Theorem 1 is a straightforward derivation of the liveness of the Qbcast service.

Lemma 4. *An active focal point recovers within τ time units.*

Theorem 1. *Our implementation of atomic read/write shared memory guarantees data availability.*

Lemma 5. *At any time in the execution there are at most $f + 1$ faulty and recovering focal points.*

Proof Because of Assumptions A_1 and A_2 , and Lemma 4, there are at most $f + 1$ faulty and recovering focal points during any time interval $[t, t + \tau]$ for any time t in the execution. This can occur if there are f faulty focal points before t , and during $[t, t + \tau]$ one of these faulty focal points recovers and another one fails. Qvd

Atomic consistency. We prove atomic consistency by showing that there exists a total ordering of the operations with certain properties. We need to show that the total order is consistent with the natural order of invocations and response. That is, if o_1 completes before o_2 begins, then $o_1 <_a o_2$.

Lemma 6. *The reply set \bar{Q} associated with a request satisfies the following properties:*

- (i) $|\bar{Q}| \geq \lceil \frac{n-f}{2} \rceil$;

(ii) $|\bar{Q} \cap Q| \geq 2 \ \forall Q \in \mathcal{Q}_m$.

Proof The first property holds because the QBCast service completes only upon receiving at least $|Q| - (f + 1)$ replies from a quorum of servers. Therefore,

$$|\bar{Q}| \geq \frac{n - f + 1}{2} \geq \left\lceil \frac{n - f}{2} \right\rceil$$

Since $|Q \cup \bar{Q}| = |Q| + |\bar{Q}| - |Q \cap \bar{Q}|$ and $|Q \cup \bar{Q}| \leq n$, then

$$|Q \cap \bar{Q}| \geq \left\lceil \frac{n - f}{2} \right\rceil + \left\lceil \frac{n + f + 3}{2} \right\rceil - n$$

Therefore, since $\left\lceil \frac{a}{2} \right\rceil + \left\lceil \frac{b}{2} \right\rceil \geq \left\lceil \frac{a+b}{2} \right\rceil$ for any $a, b \in \mathcal{R}$, then

$$|Q \cap \bar{Q}| \geq \left\lceil n + \frac{3}{2} \right\rceil - n = 2. \text{ Qvd}$$

Lemma 7. *Let o_1 be a write operation with associated state s_1 . Then, at any time t in the execution with $t > \text{res}(o_1)$ there exists a subset M_t of active focal points such that,*

(i) $|M_t| \geq \left\lceil \frac{n-f}{2} \right\rceil - 1$ (equality holds only if f focal points are faulty and one is recovering);

(ii) the state \bar{s} of its active focal points at time t is such that $s_1 \leq_s \bar{s}$;

Proof Let us denote $t_1 = \text{res}(o_1)$, and $I = [t_1, t]$. We prove the lemma by induction on the number k of subintervals $W_1, \dots, W_i, \dots, W_k$ of I of size $\leq k$, such that $W_i = [t_1 + (i-1)\tau, t_1 + i\tau]$ for $i = 1, \dots, k$, and $[t_1, t_2] \subseteq \bigcup_{i=1}^k W_i$. We want to show that at any time t there exists a subset M_t satisfying properties 1. and 2.

If $k = 1$, there exists a subset M_t of active focal points whose state is larger than s_1 . It consists of the reply set \bar{Q} associated with o_1 , less an eventual additional failure occurred in $[t_1, t]$. Therefore, because of Lemma 6 and Assumption 1 and 2 of our failure model, $|M_t| \geq \left\lceil \frac{n-f}{2} \right\rceil - 1$. The equality holds only if $f + 1$ focal points in Q did not receive o_1 request and one of the focal points in \bar{Q} fails during $[t_1, t]$. This can occur only if one focal point recovers, because of Assumption 1. In addition, the state of any recovering focal point in W_1 is larger than s_1 because $M_t \cap Q \neq \emptyset$ for each $Q \in \mathcal{Q}_m$. In fact,

$$|Q \cap M_t| \geq \left\lceil \frac{n - f}{2} \right\rceil + \left\lceil \frac{n + f + 3}{2} \right\rceil - n - 1 \geq 1$$

Therefore each focal point that recovered during W_1 can be accounted in set M_t after their recovery. Therefore, $|M_t| = \left\lceil \frac{n-f}{2} \right\rceil - 1$ only if f focal points are faulty and one is recovering.

Let us suppose now that the thesis is true for $t \in W_i$ with $1 \leq i \leq j$, and show that it holds also for $\bar{t} \in W_{j+1}$. More precisely, we show that if there exists a subset M_t satisfying properties 1. and 2. at time $t \in [t_1, t_1 + j\tau]$, then there exists a subset $M_{\bar{t}}$ for $\bar{t} \in [t_1 + j\tau, t_1 + (j+1)\tau]$. Because of properties 1. and 2. of the inductive hypothesis, the state of each focal point that recovered at time $t \in W_j$ is greater than s_1 . Property 1. holds during W_{j+1} because of Property 1. of the inductive hypothesis, assumptions 1 and 2 of the failure model, and since any subset of size $\left\lceil \frac{n-f}{2} \right\rceil - 1$ intersects a quorum. Qvd

Theorem 2. *Our implementation satisfies atomic consistency.*

Proof (Sketch) We need to show that $<_a$ is consistent with the external order of invocations and response (condition 1.) since the other conditions are trivially verified. That is, if o_1

completes before o_2 begins, then $o_1 <_a o_2$. We distinguish the following four cases and show that the thesis holds for all of them.

Case 1. o_1 is a write operation and o_2 a read/recovery operation. We distinguish the following two subcases.

Subcase 1: the write operation o_1 is completed. The thesis follows from Lemma 7. In fact, because of properties 1. and 2. of Lemma 7, each quorum contains at least one focal point whose state is larger than s_1 . Therefore, the state associated with o_2 is such that $s_1 \leq_s s_2$, and $o_1 <_a o_2$.

Subcase 2: the write operation o_1 does not complete due to client crash. There are two addition subcases. In subcase (a) a read operation detects this scenario and propagates the uncomplete update to a quorum (confirm request). This does not compromise atomic consistency since updates are performed according to their timestamps. Therefore, $s_1 \leq_s s_2$ and $o_1 <_a o_2$. In subcase (b) a read operation does not detect a write crash and $s_2 \leq_s s_1$. In this case o_1 appears to be concurrent to o_2 and it can be ordered accordingly.

Case 2. o_1 is a read/recovery operation and o_2 is a write operation. This case is trivial since the timestamp associated with o_2 reflects the real time invocation time of o_2 .

Case 3. o_1 and o_2 are both read/recovery operations. This case is trivial since $o_1 <_a o_2$ by definition.

Case 4. o_1 and o_2 are both write operations. The thesis holds because by assumption clocks are synchronized and $s_1.t = \text{inv}(o_1)$ and $s_2 = \text{inv}(o_2)$. Qvd