IMPLICATIONS OF MISBEHAVING ATTACKS ON PROBABILISTIC QUORUM SYSTEM FOR MANETs

Elisa Mannes, Eduardo Silva, Michele Nogueira and Aldri Santos*
Department of Informatics, Federal University of Paraná, Curitiba, Brazil

Keywords: Quorum system, Wireless self-organized networks, Security, Attack tolerance.

Abstract: Reliable storage supports different data sharing services, such as mobility and cryptographic key management and distributed naming. Mobile Ad hoc NETworks (MANETs) present issues in guaranteeing the consistency of data on concurrent read and write due to dynamism of nodes, the inexistence of a central control entity and support infrastructure. Probabilistic quorum systems, as PAN (*Probabilistic Ad Hoc Quorum System*), were designed for MANETs to improve the efficiency of data replication by relaxing consistency constraints, comprising a set of quorums with relaxed intersections among themselves. PAN ensures high probability of consistency between replicated data by an asymmetric quorum construction and by a gossip-based multicast protocol. However, it does not consider the presence of malicious or selfish nodes in its operations. This work assesses the impact of lack of cooperation, timeout and data manipulation attacks against PAN. Simulation results show that PAN is vulnerable to these attacks, particularly, the data manipulation attack.

1 INTRODUCTION

Mobile Ad hoc NETworks (*MANETs*) comprise mobile devices (nodes) communicating among themselves by wireless channels. These networks are dynamic, infrastructureless, and nodes individually maintain network operation in a self-organized and distributed way. Each node communicates directly with others in its range or by a multihop routing (Luo et al., 2003). Thus, network services depend on the cooperation of nodes, being necessary tolerant operations in face to malicious actions.

MANETs present issues in guaranteeing reliable storage of data with concurrent read/write accesses. Reliable storage with reasonable cost and high availability enables the support to different data sharing services, such as cryptographic key management and distributed naming (Tulone, 2007). However, their characteristics make difficult to provide the synchronism for ensuring a high level of consistency.

To relax the consistency constraints of MANETs special mechanisms are employed (Luo et al., 2003; Gramoli and Raynal, 2007; Tulone, 2007). Probabilistic quorum systems are an example of them. They improve the efficiency of data replication in MANETs by better balancing the load between nodes. A quorum system consists of a subset of nodes with data replicas, called quorums. The construction of intersections on quorums ensure properties to guarantee that nodes obtain the most recent value even if queries or updates are performed on individual quorums.

PAN (*Probabilistic Ad Hoc Network Quorum Systems*) (Luo et al., 2003) is a quorum system more appropriated to highly dynamic environments due to the use of less strict rules for creating intersections among quorums (Gramoli and Raynal, 2007; Tulone, 2007), and guarantees a high probability of consistency on replicated data by a gossip-based multicast protocol.

Albeit PAN's advantages, it does not consider the existence of misbehaving nodes in the network. Examples of misbehavior are the lack of cooperation, timeout and data manipulation. Misbehaving attacks can compromise effectiveness and reliability of quorums system, affecting the performance of network services. Hence, this paper quantifies the impact of such attacks on the reliability of PAN, highlighting which vulnerabilities need to be addressed.

We evaluate the implications of attacks on PAN's operations by simulations. We use the reliability degree and number of intermediate nodes as metrics to quantify the effects of lack of cooperation, timeout and data manipulation attacks on writing and reading operations. Results under different network mobility show that PAN is vulnerable to these attacks, and its

*Supported by CNPq/Brazil, grant 303770/2008-2

Mannes E., Silva E., Nogueira M. and Santos A. (2010).
IMPLICATIONS OF MISBEHAVING ATTACKS ON PROBABILISTIC QUORUM SYSTEM FOR MANETs.
In *Proceedings of the International Conference on Security and Cryptography*, pages 189-195
DOI: 10.5220/0002995701890195

reliability decreases with higher percentage of attackers. Attacks result in a low reliability of the system, having its worst performance under the data manipulation attack. Node mobility also affects the reliability, and write operations are more compromised by all attacks than read operations.

This paper proceeds as follows. Section 2 presents related works. Section 3 describes PAN's characteristics. Section 4 describes PAN's vulnerabilities and attacks. Section 5 depicts simulation scenarios and the metrics used for analysis. Section 6 presents results. Section 7 presents the conclusions and future works.

2 RELATED WORK

Quorums systems were first introduced as a voting scheme to ensure data consistency (Gifford, 1979). Generally they own strict rules for intersection constructions and must ensure that at least one quorum is free of faulty nodes. Recently, authors have proposed improvements in quorum systems to make them more flexible and reliable (Herlihy, 1987; Naor and Wieder, 2003). In (Herlihy, 1987), they designed a system of dynamic quorums, whereas in (Naor and Wieder, 2003), authors created a reconfigurable quorum system. However, Friedman et al. (Friedman et al., 2008) have proved that such approach is not applicable to MANETs due to the overhead generated by the messages for reconfigurations. They conclude that the ideal strategy for MANETs lies on probabilistic quorums because of their capability to relax restrictions of intersection (Malkhi et al., 2001).

Among the quorums systems proposed for MANETS, we highlight three of them. The **PAN** (Luo et al., 2003) that employs propagation mechanisms based in *gossip*. The *timed* quorum system (Gramoli and Raynal, 2007) that aims to ensure intersections during an amount of time, and the *mobile dissemination* quorum system (Tulone, 2007) that uses the geographic location of nodes to construct quorums. Strategies for probabilistic quorums on MANETs are listed in (Friedman et al., 2008). Despite those quorums systems consider characteristics as dynamic topology and collaboration, they do not consider misbehaving nodes, being susceptible to them.

Misbehaving nodes in MANETs have been addressed in several works, such as (Hu and Burmester, 2009; Marti et al., 2000), but most of them consider attacks in routing and application levels. Quorums system have been evaluated with focus on the Internet (Amir and Wool, 1996; Owen and Adda, 2006), and for the best of our knowledge, there are no studies that analyze the behavior of quorum systems for MANETs facing attacks. Hence, this work quantifies the impact of misbehaving attacks in PAN in order to provide directions for improvements on the design of quorum systems to achieve reliability and robustness.

3 PAN: PROBABILISTIC QUORUM SYSTEM FOR AD HOC NETWORKS

A typical quorum system $\mathcal{Q}$ comprises a set of quorums that has two distinct properties, consistency and availability (Malkhi and Reiter, 1997). The consistency property defines that any two quorums must intersect - $\forall Q_1, Q_2 \in \mathcal{Q}, Q_1 \cap Q_2 \neq \emptyset$, whereas the availability property means that there is at least one operational and reliable quorum.

In probabilistic quorum systems, quorums are chosen probabilistically within each interaction (Malkhi et al., 2001). This kind of systems tries to ensure that in a quorum system $\mathcal{Q}$, two quorums Q_1, Q_2 and an access strategy w, then $\forall Q_1, Q_2 \in \mathcal{Q}, \mathcal{P}(Q_1 \cap Q_2 \neq \emptyset) \geq 1 - \varepsilon$, being ε a very small value. The access strategy w defines the way that read and write quorums are constructed. PAN is an example of probabilistic quorum system, ensuring this probability $\mathcal{P}$ by gossiping to create write quorums (Q_w), and *unicast* messages to construct read quorums (Q_r).

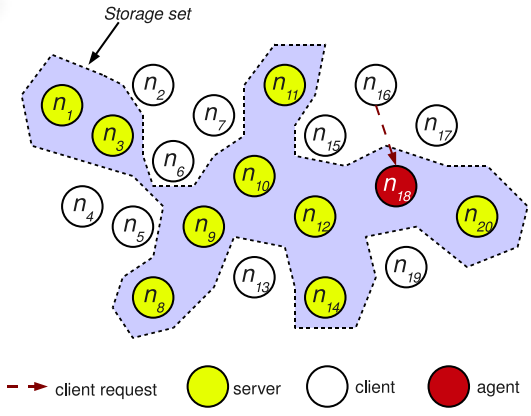

Figure 1: Entities in PAN quorum system.

PAN elects some nodes to form a storage set (*StS*), as shown in Figure 1. These nodes are the **servers**, whereas the remainder are **clients**. When servers are mediating read and write requests, they are the **agent** of that specific operation. In both operations, clients send requests to servers randomly chosen. Servers on

read requests compose read quorums and the servers on writing operations compose write quorums.

3.1 PAN Operations

PAN provides to clients reading and writing operations. Servers, acting as agents, are responsible for receiving and replying requests from clients. In writing operations, agents deliver updates received from clients with the collaboration of nodes in the *StS*. In reading operations, agents consult a read quorum before respond to the client. These operations are illustrated in Figure 2 and occur as described:

WRITE - as part of the mechanism of writing, all servers have a buffer that stores the latest update of each data. Upon receiving a write request, the agent updates its data and adds it to its buffer. Periodically, nodes propagate data from the buffer to other nodes. After some time, all servers have the updated value.

READ - the agent chooses nodes to compose the read quorum based on predefined read quorum size. Then, it forwards the request along with its local copy of the searched data. Thus, members of the read quorum reply to the agent if their data is more updated than that from the agent. In this case, the agent waits for replies, and in the lack of them, it responds to the client with its own data as conclusion that the data it holds is the most updated. Further, if servers in the quorum receive a data more updated than theirs, they revise their local copy and add it to the buffer in order to be disseminated in the next round of propagation.

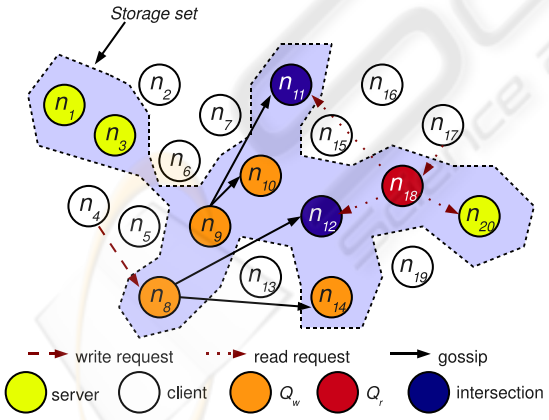

Figure 2: Read and write operations in PAN.

To explain PAN's operation, we assume a network of 20 nodes, in which ten are part of the *StS* as illustrated in Figure 2. Node n_4 sends a write request to node n_8. Node n_8, the agent, propagates this write to other nodes following the *gossiping* protocol. Write operations are propagated to a certain number of servers, being this number called *fanout* (F). It is previously configured, and servers follow the same value.

In Figure 2, F equals two servers and the size of the read quorum four. Client n_{17} issues a read operation. Then, agent n_{18} consults the read quorum in order to collect the updated value. Nodes n_{11} and n_{12} participate in both operations, being the **intersection** between read and write quorums. Intersection nodes have the most updated value to client n_{17}.

4 ATTACKS ON PAN

The success of writing operations is related to the collaboration of all servers involved in the dissemination. Thus, members of the *StS* may hinder the progress of updates in several ways, such as refusing to propagate the updates or making updates to progress slowly. This can occur if nodes are selfish and want to save their resources, or if nodes are malicious and want to degrade the system. Further, nodes can manipulate values that are being updated, causing inconsistencies in the system.

Servers consulted by agents in reading operations may deliberately not respond to requests, forcing the agent to send its own data to client. They can also modify the value of data and send this wrong value. In the second case, generated by active attackers, the problem is critical because in addition to providing clients erroneous information, servers can trust in data sent by agents and also update their data.

This work examines particularly three attacks: lack of cooperation and timeout and data manipulation. Considering the characteristics of PAN, these types of attacks are more harmful than others for operations in quorum systems. The operation of such attacks are described below.

4.1 Lack of Cooperation

Lack of cooperation attacks occurs on both operations. Figure 3(a) illustrates the behavior of a compromised agent in a read operation. A client issues a read request of a data v to server s_0, which is a selfish node. As a misbehaving agent, s_0 waits until the operation timeout expires and replies the client with its own data, that might be outdated.

When the selfish node is an intermediate node, it simply stops responding to the agent. In Figure 3(b), the client requests a data v to the server s_0, that consults the read quorum composed by s_1, s_2 and s_3. Server s_2 is a selfish node and does not respond to the agent. Under this situation, a reading operation

can still complete correctly if at least one server in the read quorum responds to the agent or if the agent itself has the updated data.

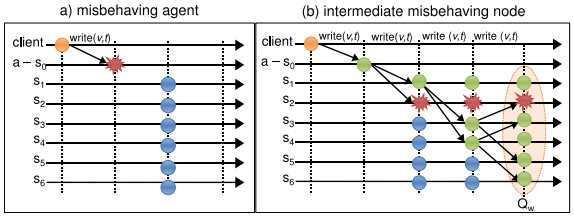

Figure 3: Lack of cooperation in read operations.

In writing operations a selfish agent does not update its data or disseminate data provided by the client. In Figure 4(a), the client issues an update of data v to s_0, a selfish node. In this condition, s_0 does not propagate the update and the write operation does not complete. As an intermediate server receiving an update via gossip, the selfish node does not update neither its local data nor the buffer. As a result, the propagation of the write operation does not progress, as shown in Figure 4(b). The client sends a write request to server s_1, that propagates it via gossip. Server s_2 is a selfish node and does not propagate the data.

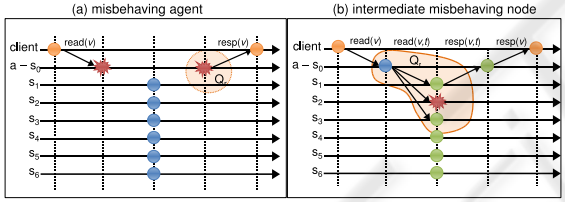

Figure 4: Lack of cooperation in write operations.

4.2 Timeout Manipulation

The time interval between the gossip is an important parameter in PAN. It determines write dissemination speed, and as faster the propagation is, faster the formation of the write quorum is. In **timeout manipulation** attack, illustrated in Figure 5, servers in the *StS* delay the propagation of updates. In this example, the client issues a write request to the server s_0, which propagates via gossip this update. Server s_2 is a malicious nodes, and arbitrarily delays the propagation in one round. This behavior makes servers of the *StS* take more time to update the data.

4.3 Data Manipulation

Data manipulation attacks consist in nodes that receive a data and deliberately change its value, in both read and write operations. As a result, nodes can accept a manipulated data. Figure 6(a) shows the behavior of a misbehaving agent in the read operation. The agent is the only contact between the client and the *StS*, being a vulnerable point.

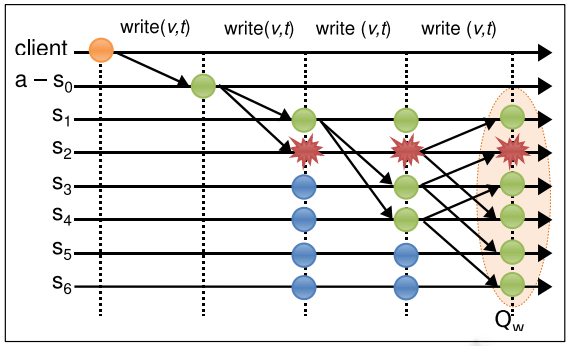

Figure 5: Timeout manipulation attack.

In the figure, the client requests a data v to the server s_0, which is a misbehaving node. Server s_0 changes the data value and timestamp. Then it consults the read quorum, s_1, s_2 and s_3. As these servers have an outdated data compared to the one received by the agent, they update their values, believing the data received is the most updated. Servers also store this value in their buffer, and propagates it via gossip. The read quorum does not respond to the agent request, since they believe the agent has the most updated data. As the timeout expires, the agent replies the client with its manipulated data.

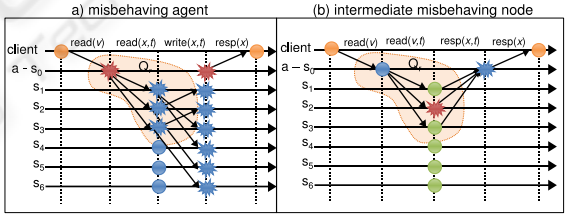

Figure 6: Data manipulation in read operations.

If the misbehaving node compose the read quorum, the impact is minimized, as shown in Figure 6(b). In this example, the client issues a read request for data v to server s_0 that consults the read quorum. Server s_2 is a misbehaving node and changes its data and timestamp. When agent s_0 receives answers from the read quorum, the data received from server s_2 is the most updated, then agent s_0 updates its own data, keeps this entry in its buffer and waits to propagate this data. It also responds to the client with this manipulated data.

In writing operations, malicious nodes can widely compromise the system. Figure 7(a) illustrates the case when the agent is a misbehaving node, and can commits the system faster since any write will result in an incorrect data. In this example, when client sends a write request to server s_0, it changes the value

and the timestamp of the data to be updated, and then propagates it to other nodes.

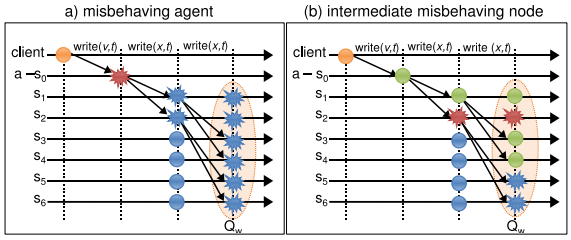

Figure 7: Data manipulation in write operations.

A case where the node is an intermediate node is shown in Figure 7(b). The client issues a write request to server s_0, that propagates it to servers. Server s_2 is a misbehaving node, thus it changes the data and propagates it to other two nodes. Each server that receives this data will propagate it and the *StS* will be compromised by this manipulated data.

5 EVALUATION MODEL

PAN was implemented on the Network Simulator (NS-2) version 2.33, and simulations parameters are based in (Luo et al., 2003). Nodes communicate using a wireless channel with the *TwoRayGround* propagation model. The network consists of 50 nodes in which 25 are servers in *StS*. Nodes move according to the *Ramdom Waypoint* model in an area of 1000m by 1000m, and have maximum speeds of 2, 5, 10 and 20m/s, with pause time of 10, 20, 40 and 80s, respectively. All nodes own a transmission range of 250m and use AODV as routing protocol. Updates from the buffer are propagated every 200ms, and F is set to two servers. Read quorums are set to four servers. The percentage of attackers varies in 20%, 28% and 36% of the servers in *StS*. A larger number of attackers leads the system to a state where quorums are fully compromised.

As in (Luo et al., 2003), readings and writings intervals are determined by a Poisson distribution. Writes have an average interval of 6s while readings happens in an average of 4s. Results are the average of 35 simulations with a 95% of confidence interval, and the lifetime of the network is 1500s.

Two metrics are employed: *reliability degree* (R_d) (Luo et al., 2003) and the *quantity of malicious nodes* in read operations (Q_m). R_d quantifies the probability of intersection between read and write quorums. R_d also represents the amount of correct readings obtained by clients. We consider correct reads the ones retrieved from previously written values, in addition to the last write. This is because the last value written may be still in progress inside the *StS*. Therefore, the previously written value is also considered correct. R_d is defined in Eq. (1), where C_r denotes correctly finished reads and R the amount of read requests issued by clients.

$$R_d = \frac{\sum C_r}{|R|} \tag{1}$$

Q_m denotes the number of times that malicious or selfish nodes participated in reading operations. The timeout manipulation is excluded because it is carried out only in write operations. This metric is expressed by Eq. (2), where M_r means a read operation in which a member of the read quorum is malicious.

$$Q_m = \frac{\sum Mr_i}{|R|} \forall i \in R, \text{ where } Mr_i = \begin{cases} 1 & \text{if } \exists m \in Q_r(R_i) \\ 0 & \text{if not} \end{cases} \tag{2}$$

6 RESULTS

6.1 Lack of Cooperation

Figure 8 shows results obtained when PAN faces the lack of cooperation attack. Each cluster of bars presents results for the same value of node speed comparing variations on the type of operation and the percentage of selfish nodes. We observe that in both operations, R_d decreases with the increase of selfish nodes. For all clusters, decreases in R_d are more significant on write operations than in read operations.

Comparing clusters, we observe that higher node speed results in lower R_d. We also observe that R_d owns the smallest value for writing operations in scenarios with 36% of selfish nodes and when nodes have a maximal speed of 20m/s. This occurs because almost half of the *StS* is compromised.

6.2 Timeout Manipulation

Figure 9 presents results for R_d when PAN is under timeout manipulation attacks. Malicious nodes propagate updates every 400ms, 800ms and 3000ms, instead of the normal 200ms. Results are grouped by the nodes speed, and for each speed, R_d values under variations on the percentage of attackers and T are compared. We observe that R_d decreases proportionally as the update propagation delay increases. For all speeds, this behavior is observed.

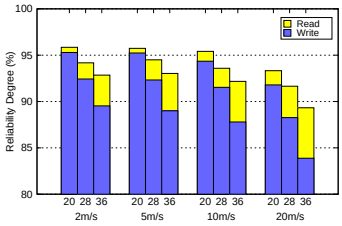

Figure 8: Lack of cooperation.

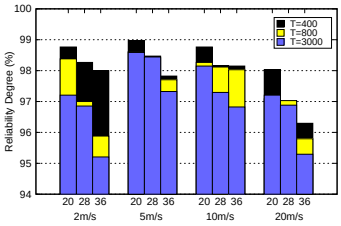

Figure 9: Timeout manipulation.

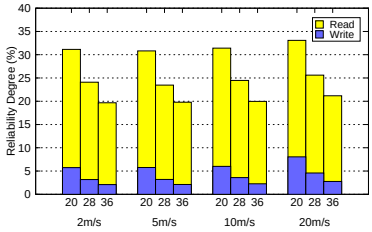

Figure 10: Data manipulation.

6.3 Data Manipulation

The data manipulation attack also affects PAN. Figure 10 shows results obtained from the simulation of this attack. Results are represented by bars, grouped by the speed of nodes. For each speed, the percentage of malicious nodes is varied on both operations.

We observe that for all scenarios, R_d is lower than 50%, and as the percentage of attackers grows, the reliability decreases. When the nodes move faster, the R_d decreases. We notice that these attacks affect significantly write operations in relation to results achieved under the other two attacks. In this case, R_d is lower than 10% for all scenarios. Higher speed and number of attackers issue the worst values to R_d. This attack has an uncommon behavior: as the speed of nodes increases, the R_d increases too. This is because MANET's characteristic, in which higher node speeds result in higher routes dinamicity. Such behavior decreases delivered updates and the effectiveness of a malicious node to propagate its manipulated data.

6.4 Malicious Nodes Participation

It was recorded the number of times that misbehaving nodes participated in read quorums, in read operations. Figure 11 shows results obtained under lack of cooperation attack. Increasing the speed of nodes has the same effect observed in the data manipulation attack, i.e. as the speed increases, the number of quorums affected by misbehaved nodes decreases.

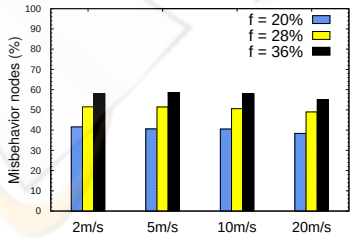

Figure 11: Compromised reads in **lack of cooperation**.

Figure 12 shows the results under the data manipulation attack. The number of readings affected by misbehaving nodes also increases as the number of attackers increases. It is observed that nodes speed interferes in the number of compromised nodes. As the speed grows, the number of misbehaving nodes decreases. This occurs because increasing the speed, less packets are received by nodes and less nodes are consulted and can deliver manipulated data.

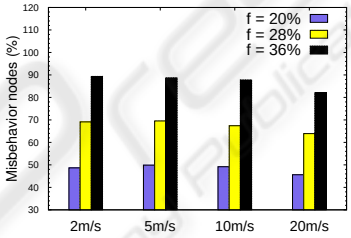

Figure 12: Compromised reads in **data manipulation**.

7 CONCLUSIONS

This paper assessed the impact of misbehaving attacks that affect data availability and consistency on the PAN system, a probabilistic quorum system for MANETs. Our analyses examined the implications of lack of cooperation, timeout and data manipulation attacks on the read and write operations. Albeit PAN's advantages, simulation results reinforced PAN vulnerability to attacks resulted from its MANET's characteristics, PAN's design and operations over quorums.

We observed that the percentage of malicious or selfish nodes and node speed affects the reliability of the system. All attacks influenced the read or write operations, being write operations the most damaged. As future work, we plan to propose a model to improve the tolerance of probabilistic quorum systems to attacks and intend to employ it in the context of the key management for MANETs.

REFERENCES

Amir, Y. and Wool, A. (1996). Evaluating quorum systems over the internet. In *FTCS*, pages 26–35, Washington, DC, USA. IEEE Computer Society.

Friedman, R., Kliot, G., and Avin, C. (2008). Probabilistic quorum systems in wireless ad hoc networks. In *IEEE/IFIP DSN*, pages 277–286, Washington, DC, USA. IEEE Computer Society.

Gifford, D. K. (1979). Weighted voting for replicated data. In *ACM SOSP*, pages 150–162, NY, USA. ACM Press.

Gramoli, V. and Raynal, M. (2007). *Principles of distributed systems*, chapter Timed Quorum Systems for Large-Scale and Dynamic Environments, pages 429 – 442. Springer Berlin.

Herlihy, M. (1987). Dynamic quorum adjustment for partitioned data. *ACM TODS*, 12(2):170–194.

Hu, J. and Burmester, M. (2009). *Cooperation in Mobile Ad Hoc Networks*, chapter 3, pages 1–15. Springer London.

Luo, J., Hubaux, J.-P., and Eugster, P. T. (2003). PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In *ACM MobiHoc*, pages 1–12, New York, NY, USA. ACM.

Malkhi, D. and Reiter, M. (1997). Byzantine quorum systems. In *ACM STOC*, pages 569–578, NY, USA. ACM.

Malkhi, D., Reiter, M. K., Wool, A., and Wright, R. N. (2001). Probabilistic quorum systems. *The Information and Computation Journal*, 170(2):184–206.

Marti, S., Giuli, T. J., Lai, K., and Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *ACM MobiCom*, pages 255–265, NY, USA. ACM.

Naor, M. and Wieder, U. (2003). Scalable and dynamic quorum systems. In *ACM PODC*, pages 114–122, NY, USA. ACM.

Owen, G. and Adda, M. (2006). Simulation of quorum systems in ad hoc networks. In *3rd International Conference on Artificial Intelligence in Engineering and Technology*.

Tulone, D. (2007). Ensuring strong data guarantees in highly mobile ad hoc networks via quorum systems. *Ad Hoc Networks*, 5(8):1251–1271.