# EVALUATING FAULT TOLERANCE ASPECTS IN ROUTING PROTOCOLS FOR WIRELESS SENSOR NETWORKS

Daniel F. Macedo[1], Luiz H. A. Correia[1,2], Aldri L. dos Santos[1,3], Antonio A. F. Loureiro[1], José Marcos S. Nogueira[1]*, and Guy Pujolle[4]

[1] *Federal University of Minas Gerais, Brazil*

[2] *Federal University of Lavras, Brazil*

[3] *Federal University of Ceará, Brazil*

[3] *University Paris 6, France*

{damacedo,lcorreia,aldri,loureiro,jmarcos}@dcc.ufmg.br, pujolle@rp.lip6.fr

**Abstract** Fault tolerance is an essential requirement in the design of protocols and applications for Wireless Sensor Networks (WSNs) since communication and hardware failures are frequent. In this paper we studied the resilience of routing protocols for continuous data dissemination WSNs in face of faults. The main causes of silent failure are presented and classified, including security attacks. An evaluation of routing protocols shows that failures under a large region of the network are the most damaging. We also show how routing protocols may save energy by temporarily turning off disconnected nodes.

**Keywords:** Wireless sensor networks, fault tolerance, routing

## 1. Introduction

Wireless Sensor Networks (WSNs) consist of a large number of sensor nodes, composed of processor, memory, battery, sensor devices and transceiver. These nodes send monitoring data to an access point (AP) responsible for forwarding data to the users [1]. Unlike traditional ad hoc networks, in general it is not possible to replace or recharge node batteries due to the number of nodes deployed or inhospitable environmental conditions. Hence, energy conservation is a critical factor in WSNs.

*In sabbatical period at universities of Evry and UPMC/Paris6/LIP6, France.

WSNs are propitious to failure due to events such as node destruction, link quality degradation, among others. Since those networks may be employed in hostile environments, nodes can fail due to landslides, floods or other natural agents. Failures also occur in the communication due to changes in weather or movement of objects near the nodes, or due to malicious agents. Thus, protocols and applications must be developed with fault tolerance in mind.

Data flow in WSNs usually follows a pattern, since data is preprocessed locally and then sent to the AP. This data flow can be categorized according to its frequency [2]. In *event-driven networks*, communication is sporadic, occurring only when an event of interest is detected. In *continuous dissemination networks*, nodes periodically send data to the AP. In those networks it is possible to build a "map" of the current state of the environment, which can be later used to study time and space variations in the observed phenomena. Due to the intrinsic differences in traffic, routing protocols are usually designed to operate on a single network class. Continuous dissemination networks tend to employ proactive protocols, while in event-driven networks routes are build only when an important event is detected. The same fact occurs with fault-tolerance mechanisms.

In this paper we study the performance of routing protocols for continuous dissemination networks in faulty scenarios, where *silent* faults occur. The main causes of failure are presented and then categorized. Next, a performance evaluation through simulation was performed for three routing protocols. This text is organized as follows. Section 2 presents the related work. Section 3 presents an overview of the protocols evaluated. Section 4 describes and categorizes the main causes of silent failures in WSNs. This categorization is then used to evaluate three routing protocols in section 5. Finally, section 6 draws the conclusions and future work.

## 2. Related Work

Avizienis et al. present a taxonomy of failures, which also encompasses security issues [3]. Hollick et al. present the challenges of fault tolerant systems for WSNs, ad hoc networks and cellular networks, and list the requirements which should be met by fault-tolerant protocols [4].

Fault tolerance in protocols for WSNs has been widely studied. The first protocols developed [5] were concerned with failures caused by energy depletion, increasing the life time of a node by distributing the energy spent among nodes. Other protocols were designed to be resilient against node failures. Those protocols send multiple copies of

data among different routes, thus increasing the probability of correct reception. Ganesan et al. [6] showed that partially disjoint routes are as effective as totally disjoint routes, although spending less energy to be established.

Since the cost of maintaining multiple routes is significant, some protocols define only one high-quality route. De Couto et al. presented a modification to DSR which calculates the reliability of a route [7]. Nodes always choose the route with the best quality, thus increasing the probability of a successful delivery.

Given the occurrence of a failure, it is necessary to identify an alternative route. Vieira et al. proposed two protocols to mitigate failures due to energy depletion [8]. In the first algorithm, the AP notifies nodes to modify its routes whenever a failure occurs. In the second, nodes build a list of "second-best routes". Upon the detection of a failure, one route in this list is selected to become the default route.

## 3. Evaluated Protocols

We evaluate the performance of three routing protocols for continuous dissemination networks. Those protocols were selected because they provide different levels of fault tolerance.

TinyOS Beaconing is a protocol used in the Mica Motes platform [9]. This protocol periodically creates a minimum distance tree rooted at the AP. Only nodes with good link quality are used to route messages. TinyOS Beaconing was not designed with fault tolerance mechanisms, although the periodic recreation of routes provides some degree of fault tolerance.

Boukerche et al. proposed a routing algorithm, called EAD (*Energy-Aware Distributed routing*), which creates a routing tree that maximizes the number of leaf nodes [10]. Leaf nodes, which do not need to send messages, turn their radios off in order to extend network lifetime. The protocol also uses backoff timers based on current node energy for decreasing collision probability. As in TinyOS Beaconing, EAD uses the periodic reconstruction of routes to provide fault-tolerance. In EAD, however, traffic is concentrated in a few nodes, hence failures in those nodes will be more severe than in "ordinary" nodes.

The PROC (*Proactive ROuting with Coordination*) protocol was developed with the goal of reducing energy consumption and increasing network lifetime [11]. PROC creates a routing tree, called *backbone*. The structure of the backbone is influenced by the application, which defines which nodes are more suitable to route data. The protocol provides fault tolerance using link layer acknowledgments. Whenever the

number of data packets not acknowledged reaches a certain threshold, PROC selects a new route. As in EAD, the failure of *backbone* nodes will be severe. The proactive probe of nodes using link layer acknowledgments, though, mitigates this issue.

## 4. Failure in WSNs

This section identifies the main causes of silent communication failures in WSNs. We assume that protocols perform their functions correctly, and all messages are correctly received. The following failures were identified:

*Atmospheric phenomena* – Several environmental conditions such as humidity, temperature, among others, modify signal propagation. As weather is constantly changing, communication quality varies with time.

*Mobile sources of interference* – Other devices operating at similar frequencies or even vehicles, animals and humans may interfere with communicating nodes.

*Natural disasters* – Sensor nodes may be deployed outdoors or in disaster locations, thus being exposed to landslides, floods and earthquakes. Those events may cause massive destruction of sensor nodes by permanently damaging hardware components.

*Accidental breakage* – Sensor nodes can be accidentally destroyed, for example due to animals trampling over nodes, or falling trees.

*Processor crashes* – The application may contain programming errors, which might lead the processor to crash situations. To avoid such situations, microcontrollers reboot if a software malfunction occurs. Thus, nodes will be unavailable for a finite amount of time.

*Malicious failures* – WSNs are prone to malicious failures due to security attacks caused by an outsider or by a corrupted node. This article does not evaluate security protocols. Some security attacks can be partially mitigated with the use of fault tolerance techniques [12]. We use fault tolerance techniques to avoid the following denial of service attacks: interference attacks, collision attacks, and sinkhole attacks.

*Energy depletion* – Energy depletion may generate communication failures. Usually, batteries will not be replaced, since WSNs are employed in harsh environments, or the number of nodes deployed makes battery replacement a daunting task. Our study does not encompass energy-related failures, since those are very difficult to model.

### Failure Grouping

The failures described above were characterized according to common characteristics. This characterization, summarized in Table 1, aids the

performance evaluation presented in section 5. Failures are grouped according to persistence and extension:

*Persistence* – Indicates if a node will resume correct operation after its failure (*transient failures*), or if the node will fail indefinitely (*permanent failures*) [3]. From a routing perspective, transient failures occur when nodes are out of service for a few minutes, while in permanent failures nodes are out of service for hours.

*Extension* – Relates to the number of failed nodes. Failures can be *isolated* (only one node fails) or *grouped* (various nodes in a region fail).

*Table 1.* Failure characterization, divided by their causing agents.

| Cause of failure | Persistence | Extension |
|---|---|---|
| Atmospheric phenomena | permanent | grouped |
| Mobile sources of interference | transient | isolated |
| Natural disasters | permanent | grouped |
| Accidental breakage | permanent | isolated |
| Processor crashes | transient | isolated |
| Interference attacks | permanent | grouped |
| Collision and sinkhole attacks | both | isolated |

## 5. Evaluation

The three protocols were implemented in the simulation environment NS-2 [13]. The application simulated has traffic characteristics similar to the sensor network deployed in Great Duck Island [14]. In this network each sensor sends a data message of 36 bytes of size every 70 seconds.

The medium access control protocol (MAC) employed is a modified version of the IEEE 802.11 protocol, which emulates the behavior of the standard MAC protocol in TinyOS [9]. The route recreation interval used for EAD and TOSB (a simplified version of TinyOS Beaconing without link quality estimators) was 120s, while for PROC this interval was set to 180s, as empirically determined in [11].

The simulated network consists of 150 nodes deployed in a square area, measuring 70m on each side. The AP is located at the corner of the area. The network operates without failures for 1500s. After that, a failure occurs, and the simulation continues for 1500s. In the scenarios where isolated failures occur, failed nodes are randomly selected. In the grouped and permanent scenario, a central point is defined, and all nodes within a given radius of this point fail. All results are the mean values of 33 simulations, plotted with 95% confidence intervals.

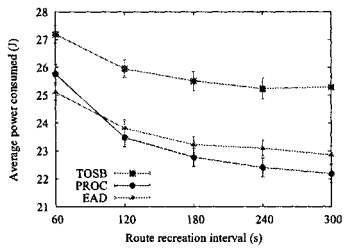


*Figure 1.* Average power consumed varying the route recreation interval.

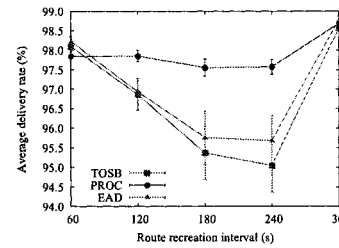


*Figure 2.* Average delivery rate varying the route recreation interval.

## Transient and Isolated Failures

The routing recreation interval affects the degree of fault tolerance, since protocols rely on route reconstructions to recover from failures. In this scenario 20 nodes fail for 120s. In the first set of simulations we varied route recreation intervals from 60 up to 300s. Figure 1 shows that nodes consume more energy when route updates are frequent. The average delivery rate, shown in Figure 2, decreases for larger route recreation intervals. This reduction is subtle in PROC, since this protocol identifies failed routes earlier using probes. All protocols recover from failure within 200s. Average delivery rates increase for 300s recreation intervals, since network load decreases, and less packets are dropped due to full packet queues. Latency decreased for all protocols as route recreation intervals increased, since there was a lower load on the network.

Next, we evaluated how failure time affects the performance of the protocols. PROC presented higher delivery rates (around 0.5% higher), as shown in Figure 3. Periodic routing recreation guaranteed good fair tolerance for EAD and TOSB, since both showed delivery rates slightly lower than PROC's. The amount of energy consumed decreased with longer failures, since nodes had to route less data. PROC was the most energy-efficient protocol, consuming 22J of energy, while EAD and TOSB consumed 4% and 14% more energy than PROC, respectively.

Finally, we varied the number of failed nodes from 25 up to 100 nodes. All protocols behave similarly in this scenario. The proactive mechanism in PROC allowed this protocol to recover from failures faster than the other protocols evaluated, providing a 0.5% increase in average delivery rates. Since simulation time is significantly bigger than the failure time, the gains obtained by proactive probing of nodes are not significant in the final average delivery rate. Average latency and hop count were not affected, but average energy consumption decreased, since less messages were sent as more nodes failed. Figure 4 shows the average energy con-

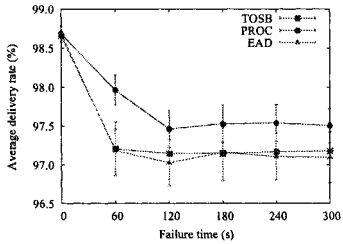


*Figure 3.* Average delivery rate varying the time of failure.

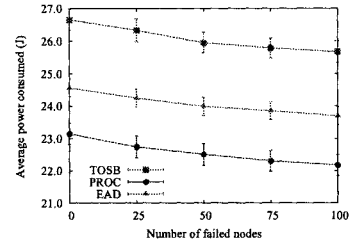


*Figure 4.* Average power consumed varying the number of failed nodes.

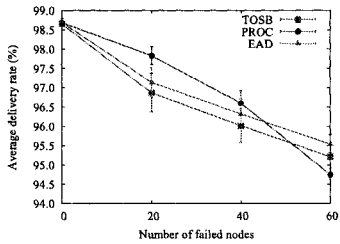


*Figure 5.* Average delivery rate varying the number of failed nodes.

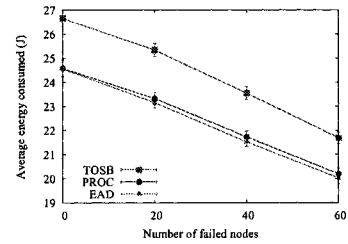


*Figure 6.* Average power consumed in permanent and isolated failures.

sumption. Overall, transient and isolated failures are not severe, since nodes easily find new routes.

## Permanent and Isolated Failures

In this scenario we evaluate the impact of permanent and isolated failures. We varied the number of failed nodes from 20 up to 60 nodes. As in the previous scenario, all protocols recovered their routes within 200s, though in this scenario the throughput drops after the failure, since failed nodes permanently stop sending data. The average hop count decreased slightly, around 0.1 hops for each 20 failed nodes. Average latency showed a small variation, showing that the traffic reduction compensated the increase in average route lengths. The average delivery rate decreased with the number of failed nodes, as shown in Figure 5.

Compared to transient and isolated failures, permanent and isolated failures allow nodes to save more energy (Figure 6), since the network produces less data. Permanent and isolated failures are more severe than transient and isolated failures, since the former imposes greater degradations at node's average delivery rate and average energy consumption.

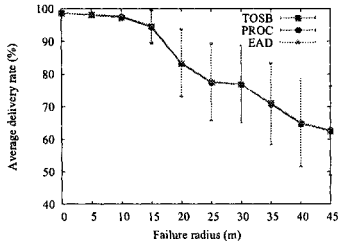


*Figure 7.* Average delivery rate in permanent and grouped failures.

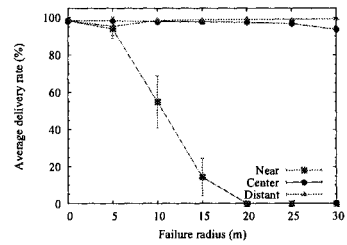


*Figure 8.* Average delivery rate for failures in different sections of the network.

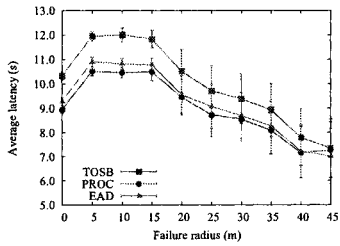


*Figure 9.* Average latency in permanent and grouped failures.

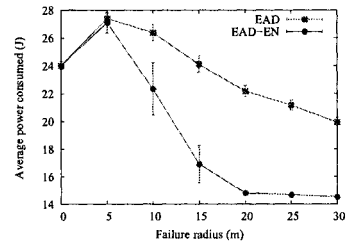


*Figure 10.* Average power consumed with energy-saving schemes.

## Permanent and Grouped Failures

This scenario evaluates the severity of permanent and grouped failures. The failure radius varied from 5 up to 40m. The average delivery rate drops up to 9% as failure radius increases (Figure 7). The confidence interval is up to 5%, showing that the delivery rate varies significantly in each simulation. This is caused by network partitions, as supported by Figure 8. The "Near" curve shows the delivery rate for failures near the AP, the "Center" curve shows results for failed nodes in the center of the network, and the "Distant" shows failures at the edge of the network. Failed nodes near the AP substantially degrade the average delivery rate, while failed nodes at the edge of the network are harmless.

To recover from a group of failed nodes, routes must avoid the failed region, increasing the average hop count and average latency, as shown in Figure 9. For failures of radius over 20m, average latency and hop counts decrease, since partitions occur more frequently, and only connected nodes near to the AP are able to send their packets successfully.

Since network partitions cannot be avoided, as nodes are unable to route through them, routing protocols should adopt energy conservation

measures in the disconnected nodes. Figure 10 compares the performance of EAD with EAD-EN, an improved version of EAD which turns off the radio of disconnected nodes. Node disconnection in EAD-EN is detected if a node does not receive routing messages for a period of two route recreation intervals. Figure 10 shows that, for failures near the AP, EAD-EN consumes from 16% up to 33% less energy when compared to the original EAD.

## 6. Conclusions and Future Work

Wireless Sensor Networks are employed in harsh environments, hence those networks are prone to failures. Sensor nodes must adapt to the environmental conditions to provide a service within the expected quality of service requirements. Thus, nodes must have effective routes even in the presence of failures and security attacks. In this article we characterized the main causes of silent failures in WSNs, and evaluated the performance of routing protocols based on this characterization.

Results showed that transient and isolated failures, and permanent and isolated failures are mitigated with the periodic recreation of routes. Permanent and grouped failures are much more severe, since those failures may partition the network. Fault tolerance algorithms must employ more aggressive approaches near to the AP, since failures in this region may severely degrade the performance of the entire network. Upon shutting down disconnected nodes, significant energy savings can be achieved in situations were a prolonged failure partitions the network.

Fault tolerance can be improved with the design of failure assessment mechanisms. Such scheme would allow early detection or even forecasting of failures, providing means to readily recover from faulty operation. As future work we will study how quality of service parameters are affected by failures.

## Acknowledgments

The development and studies described in this article were completed as part of Sensornet project (http://www.sensornet.dcc.ufmg.br), funded by CNPq/Ministry of Science and Technology/Brazil. Some scholarships were given by CAPES/Ministry of Education/Brazil.

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communications*, 40(8):102–114, 2002.

[2] Linnyer Beatrys Ruiz, Antonio A. F. Loureiro, and Jose Marcos Nogueira. Functional and information models for the MANNA architecture. In *GRES03 - Col-*

*loque Francophone sur la Gestion de Reseaux et de Services*, pages 455–470, February 2003.

[3] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, 2004.

[4] Matthias Hollick, Ivan Martinovic, Tronje Krop, and Ivica Rimac. A Survey on Dependable Routing in Sensor Networks, Ad hoc Networks, and Cellular Networks. In *Proceedings of the 30th IEEE EUROMICRO Conference 2004*, pages 495–502, Rennes, France, September 2004.

[5] Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[6] Deepak Ganesan and Ramesh Govindan and Scott Shenker and Deborah Estrin. Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.

[7] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, California, September 2003.

[8] Marcos Augusto M. Vieira, Luis Filipe M. Vieira, Linnyer Beatrys Ruiz, Antonio Alfredo F. Loureiro, Antônio O. Fernandes, José Marcos S. Nogueira, and Diógenes Cecílio da Silva Jr. Como Obter o Mapa de Energia em Redes de Sensores Sem Fio? Uma Abordagem Tolerante a Falhas. In *Anais do 5o. Workshop de Comunicação sem Fio (WCSF)*, pages 183–189, 2003.

[9] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, and David Culler. TinyOS: An operating system for wireless sensor networks. In W. Weber, J. Rabaey, and E. Aarts, editors, *Ambient Intelligence*. Springer-Verlag, New York, NY, 2004.

[10] Azzedine Boukerche, Xiuzhen Cheng, and Joseph Linus. Energy-aware data-centric routing in microsensor networks. In *Proceedings of the 6th international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 42–49. ACM Press, 2003.

[11] Daniel F. Macedo, Luiz H. A. Correia, Aldri L. dos Santos, Antonio A. Loureiro, and José M. Nogueira. A pro-active routing protocol for continuous data dissemination wireless sensor networks. In *10th IEEE Symposium on Computer and Communications (ISCC)*, June 2005.

[12] Anthony D. Wood and John A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[13] NS-2 simulator. http://www.isi.edu/nsnam/ns/, January, 2005.

[14] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, pages 307–322, January 2004.