Evaluation of a Fault-tolerant WSN Routing Algorithm Based on Link Quality

Unai Burgos, Iratxe Soraluze and Alberto Lafuente
Department of Computer Architecture and Technology, University of the Basque Country UPV/EHU
20018 San Sebastián, Spain

Keywords: Wireless Sensor Networks, Routing, Fault Tolerance.

Abstract: In this paper we propose a fault-tolerant routing algorithm for WSN. Our approach is based on link quality as the main criteria to build an initial routing tree, although additional criteria, such as node reachability and path diversity, are also considered. The routing tree is built using only local information (two-hop neighbourhood). This information is also used to reconfigure locally the routing tree when a fault is detected. The routing algorithm has been implemented using the OMNeT++ simulator and a preliminary performance evaluation has been carried out. Results show that our algorithm reach comparable delivery rates than a standard flooding algorithm, being much more efficient.

1 INTRODUCTION

Wireless Sensor Networks, WSN, are a promising technology that have been used successfully for environment monitoring, health care applications, transportation, ubiquitous home networks and others. WSN consist of one or more sinks and a huge number of small devices, called motes, with sensors, wireless communication and small computation capability. Sensors are used to gather information to be sent to a sink node. The sink node is used to process the received data and to connect the wireless sensor network with the Internet. The sink node is usually a more powerful device with no practical limitations. Note that current technology enables the implementation of the sink on a mobile device.

As motes are small, work unattended in the real world, and are powered with very limited batteries, energy constraints are usually severe and affect all the aspects in the system design. Therefore, instead of relaying on brute-force message forwarding (flooding), communication protocols should be designed carefully in order to trade-off transmission power and message retransmissions and forwarding. Reducing transmission power, which increases energy waste quadratically with respect to signal range, results in more message loses and more hops to reach the sink node. Furthermore, in the search of the optimality, it should be also taken into account that message loses also depends on phenomena as interferences and multipath fading (Doherty et al., 2012). Finally, a node can crash, due to battery exhaustion or many other reasons.

Therefore, a WSN needs an efficient routing protocol and a fault management mechanism that reacts by reconfiguring the network upon failures and ensure a sufficient quality of service (Yu et al., 2007). A good solution should provide reliable and fault tolerant communication, scalability, low latency and quick reconfiguration with minimum energy consumption.

To face failures in routing paths, two general approaches are used, namely replication and retransmission (Alwan and Agarwal, 2009). The most common replication mechanisms consist on transmitting multiple copies of the same data to the sink over multiple paths (Ye et al., 2005). Note that when the same data packet is sent along two fully node-disjointed paths the packet delivery ratio is almost doubled (Tian and Georganas, 2003). However, the transmission of multiple copies increases the energy consumption, and the extra work to construct and maintain disjoint paths introduces control message overhead and a lack of scalability (Challal et al., 2011). On the other hand, in retransmission techniques usually only one path is used between the source node and the sink. As a consequence, a broken path needs to be partially reconstructed or completely discarded. Also in this approach network traffic and hence energy consumption increase due to end-to-end retransmissions.

Hop-by-hop recovery seems to be more reliable

Burgos U., Soraluze I. and Lafuente A..
Evaluation of a Fault-tolerant WSN Routing Algorithm Based on Link Quality.
DOI: 10.5220/0005330300970102
In *Proceedings of the 4th International Conference on Sensor Networks* (SENSORNETS-2015), pages 97-102
ISBN: 978-989-758-086-4

than end-to-end recovery (Kim, 2004). These routing algorithms consider a partial reconstruction of the routing path due to a link or node failure. The main benefit is that failure detection is managed locally by the nodes without global exchange of information, a key issue in order to avoid messages.

Some of the fault-tolerant routing algorithms proposed in the literature consider only crash node failures and permanent link failures, i.e., once a failure is detected the node is considered dead (Boukerche et al., 2006) (ALMomani et al., 2011). In (Boukerche et al., 2006) just one path is used for transmitting data and node failures are handled by a retransmission mechanism. If a sender does not receive an ack message from the receiver node in a predefined timeout, the routing path is partially changed and the packet is retransmitted using the new links in the path. In this solution, flooding is used to construct an initial cost field, as well as for node subscription. In (ALMomani et al., 2011) the routing path reconstruction starts in a node that detects that its energy level is below a specific threshold, and therefore it is going to dead.

Nevertheless, the permanent failure assumption is not realistic in many scenarios where link failures or message looses in WSN are commonly due to transient situations like temporal obstacles or interferences. In these cases enabling temporal alternative paths is an approach usually found in the literature (Nelakuditi et al., 2007) (Tian and Georganas, 2003).

Most of the routing protocols used for WSN are reactive (Al-Karaki and Kamal, 2004), i.e., the routes are built on demand after a flooding started on the sink node. Apart from the cost associated to the flooding and a high latency of the reconstruction of paths when failures occur, a drawback of this approach is that it does not manage efficiently the mobility of the sink node, because the routing path is built with the sink as root. In consecuence, it is convenient to design a *proactive* routing algorithm as in (Heinzelman et al., 2000) when the sink can move.

In the present work we assume that failures are transient. Besides we consider that a link can be characterized by a feature that we call the *quality of the link*. Our aim is to select the most reliable links to form a routing tree in order to reduce the number of retransmissions that would cause a further increase in channel contention and more packet losses (Li et al., 2005) (Yousefi et al., 2009) (Zhang et al., 2007), as well as a decrease in the message delivery rate (Zhang et al., 2007). System reconstruction is carried out locally in the neighbourhood of the fault, in order to finally reduce the energy consumption.

In this paper we describe a routing algorithm based on link quality as the main criteria, as well as a preliminary performance evaluation. Our strategy is based on the construction of a routing tree in a proactive manner. Besides link quality, we use additional criteria, such as node reachability and diversity (Boettcher et al., 2003). The routing tree is built using only local information (two-hop neighbourhood). This information is also used to reconfigure the routing tree when a fault is detected.

Although, performance on sink mobility scenarios has not been evaluated in the scope of the present work, our routing algorithm has been designed to be flexible enough to efficiently manage the dynamism associated to sink mobility. In our protocol, if the sink moves, only the routing information of a small area in the neighbourhood of the sink needs to be updated.

2 ROUTING ALGORITHM BASED ON LINK QUALITY

We will divide the description of our routing algorithm in two parts. In the first one we describe how an initial routing tree is built after a node and link quality discovery period. In the second one, we describe how the routing of data to the sink starts and how the routing tree will be reconfigured due to link or node failures.

The WSN architecture we consider consists of a finite (but unbounded) set of resource-constrained static sensor nodes, which we will denote by $p_1, p_2, \ldots, p_i, \ldots$ (or by $p, q, \ldots$ for short), and one more powerful sink node. We model this distributed system as a set V of n nodes. A node p communicates directly only with a subset N_p of nodes of V, the nodes in its communication range. The nodes in N_p are connected with p by a bidirectional communication link. All the sensor nodes transmit with the same power level and henceforth all they have the same transmission range.

Concerning timing assumptions, we consider a synchronous model in which there are bounds on message transmission times. Message transmission time bounds can be estimated using application parameters such as transmission latency between neighbor nodes. We assume that every node has a local clock that can measure real-time intervals.

We consider that nodes can fail either by permanently crashing (sensor nodes that fail do not recover) or by omitting messages. Omission failures may occur either while sending or while receiving messages, and these failures can be transient (a node may temporarily omit messages and later on reliably deliver

messages again), or permanent. At the same time, we assume lossy links, i.e., messages can be lost temporarily or permanently during its transmission in a link. Henceforth nodes that have permanent message omissions in all their outgoing links will be considered as crashed nodes. We also assume that there is a maximum number of crash failures and permanent omission failures in the system in such a way that the system do not get partitioned. We assume that all the nodes of the system that do not crash are able to receive and send messages along a path of lossy links from every non-crashed nodes.

2.1 Building the Routing Tree

First of all, there is a discovery phase where each node p sends periodical heartbeat messages during a period of time in order to discover the nodes that are in its transmission range, i.e., the subset $N_p \subset V$. In this phase also the link quality for all the links is gathered. Global knowledge about link quality is relevant to build optimal routing paths, however, due to the severe constrains in wireless sensor networks, broadcasting link quality information to the whole network is out of any practical consideration. In this algorithm each node sets the link quality of nodes that are up to two hop distance. A node p gathers the link quality for all the nodes in N_p and N_q for all $q \in N_p$, using neighbour information piggybacked on heartbeat messages. We call N_p^2 to this set of nodes. Also some other topological properties of the network such as the reachability and diversity (Boettcher et al., 2003) of each node are calculated.

We provide here a more formal description of the properties:

- *Link Quality*
  For any bidirectional link $(p,q) = (q,p)$ between nodes p and q, we represent the *quality* of link (p,q) as W_{pq}. The value of W_{pq} is assigned on the basis of the number of lost messages in the link (p,q), as we explain next. Observe that, since we are considering synchronous links, a message lose can be detected at the receiving node by the time-out expiration of a periodical message. Assume a link (p,q) where p has sent $M_{p\to q}$ messages and q has sent $M_{q\to p}$ messages. Assume that process p has detected that $l_{q\to p}$ messages from q have been lost in the link (p,q), and that q has detected that $l_{p\to q}$ messages from p have been lost in the link (p,q). Since we are considering bidirectional communication, the quality of the link will be determined by the most lossy direction of the link, and it is represented by a real number in the range $(0,1)$:

$$W_{pq} = 1 - max(\frac{l_{q\to p}}{M_{p\to q}}, \frac{l_{q\to p}}{M_{q\to p}}) \qquad (1)$$

- *Reachability*
  The reachability property measures a brute-force aspect of connectivity power of a node p. Consider a node q such that $p \in N_q$. We define the *reachability set* of p *from* q, $R_p(q)$ as the nodes connected to p that are not directly connected to q. Henceforth, $|R_p(q)|$ provides a measure of the *reachability* of p. In our example of Figure 1, observe that $R_{p_1}(p_0) = \{p_3\}$ and $R_{p_2}(p_0) = \{p_3, p_4\}$, thus $|R_{p_1}(p_0)| = 1$ and $|R_{p_2}(p_0)| = 2$, which represents the fact that p_2 provides higher reachability from p_0 than p_1. The goal of considering this measure is to reduce the number of hops of the routing tree.
- *Diversity*
  Diversity refers to a much more subtle role of a node p as a router in the network: the power of p for reaching nodes that can not be reached from other nodes. Note that diversity can be calculated upon the information about the reachability sets in a neighbourhood. In our example, p_3 and p_5 provides the highest possible value for diversity, as both nodes are essential to maintain the graph connected.

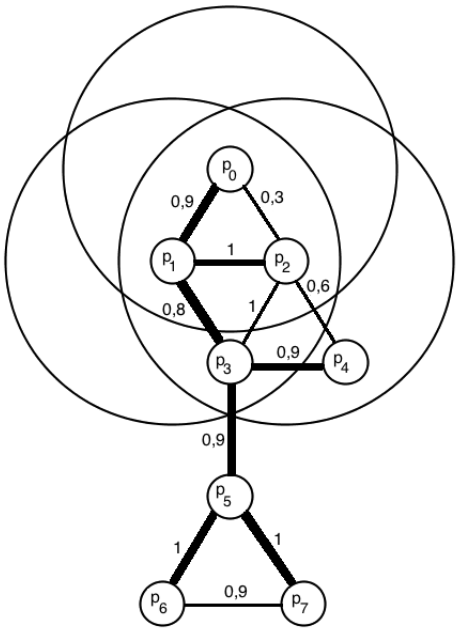

Figure 1: Collected data after information-gathering phase.

Once the information obtained about the quality of links is statistically relevant, the construction of the routing tree starts proactively. To proceed, an initial

decision is to identify a root node for the tree. Diverse criteria could be applied to select the root node, depending on parameters such as network size, sink location or sink mobility. In the case of static sink node, as we are considering, the root of the tree will be the sink node.

The root node is in charge of starting the tree construction. Node p will select as its sons enough nodes to reach all the nodes at two hop distance. Note that diversity is necessarily the first criteria to select some of the son nodes. After that, p will select first the nodes with the best link quality and, among the nodes with similar link quality, the nodes with the highest reachability. This procedure is repeated by each of the son nodes of p and propagated to the whole system. Nodes with low reachability, diversity and quality of links will remain as leaf nodes in the tree and henceforth will not participate in routing tasks.

In our example of Figure 1, p_0 is the root node and $N_p^2 = \{p_0, p_1, p_2, p_3, p_4\}$. Observe that, based on the criteria of link quality, node p_0 will chose p_1 as its son node instead p_2, in spite of the higher reachability provided by the latter. Figure 1 shows this decision and the configurations of the reachability sets. Observe also in the figure that p_3 is a better candidate than p_2 to be son node of p_1 because p_3, despite linked with a lower-quality link, has better diversity (in fact p_3 is necessary to connect p_5).

2.2 Routing and Managing Failures

In this section we describe the behaviour of the routing algortihm once the routing tree has been built.

To manage failures once the tree has been built, the periodical heartbeat messages continue being sent across the links that belong to the routing tree. The only difference is that in this case the timers are set depending on the link quality. Whenever a timer for a heartbeat message of a link expires, it means that a link failure has occurred. This failure might be transient, due to temporal interferences or obstacles, or might be permanent due to a node crash. Note that whether the message is omitted by a node or lost in the channel is indistinguishable from the receiver point of view. Anyway, whenever a message lose is detected by q in a transmission from p to q, W_{pq} is updated, and the link (p,q) is removed from the tree and replaced immediately to form an alternative route. This decision is based on the fact that, although next messages sent from p to q could be received by q, in the most common failure patterns, the probability of having a sequence of losses is high. The same link (p,q) could be part of the tree later on if the message lose pattern of the link is benign and a failure occur in the new links chosen for the tree. However, if the (p,q) link failure is permanent, i.e, p has crashed, the W_{pq} link quality measure will decrease progressively, and eventually link (p,q) will not belong to the tree any more. To find an alternative route when q detects a failure in a link (p,q), only 2-hop local information is needed. The criteria used to select the new link or links are the same as the used to build the tree: link quality, reachability and diversity. If p is a son of q, q will replace p with some other son node(s). Otherwise, if p is a parent of q, q will start a specific inverse reconstruction mechanism to find a new parent within a two-hop distance.

Besides, to avoid crashes due to battery depletion, when the battery level of a node goes beyond a tresshold, it results in a *programmed* decreasing of the node functionality. Specifically, p will not be used as a router so far, and consequently p should be excluded of the routing tree by assigning a null value to the quality of p's links, i.e., $W_{pq} = 0$ for all node q node p is connected to. We call this mode *routing-off* mode. Note that, as a consequence of that, if some (p,q) was in the routing tree, then the routing tree should be reconstructed. Of course, in routing-off mode, p's links can still be used to communicate application messages, i.e., those generated by p as a source node in the WSN.

3 ALGORITHM SIMULATION AND EVALUATION

In order to carry out a preliminary evaluation, we have implemented our routing algorithm using the OMNeT++ simulator with the MiXiM framework.

We focus the evaluation on the folowing parameters:

- Start-up time: time that our protocol needs to build the routing tree after the information gathering phase.
- Message delivery rate: the percentage of messages that are delivered to the sink node among the messages sent from the source nodes.
- Message load: number of messages created in the network for each message created in a source node. This also provides an estimation of energy consumption.
- Latency: average time the messages created in a source node need to be delivered to the sink node.

In order to provide a basis for the evaluation we compare our algorithm to a basic flooding algorithm. A flooding algorithm works inefficiently in its goal to

obtain high delivery rates and low latencies and delays. Henceforth, the goal of any well-designed tree-construction algorithm should be to provide a routing quality comparable to flooding while improving efficiency parameters. Note also that a simple flooding algorithm does not require any initial configuration effort.

We have carried out three different experiments. For each one of them we have measured the aforementioned performance parameters in order to compare our algorithm to the standard flooding algorithm:

- Scalability. We study the performance of the system for 25, 49, 81, 121, 169 and 225 nodes.
- Root location. Two possibilities are considered: at a corner of the grid network and in the middle of the network.
- The influence of link quality. We analyze the performance of the algorithms for four different scenarios with different message lose probabilities.

To evaluate the performance, we consider that there is only one source node that sends 30 data messages to the sink, one message every 0.2 seconds. This source node is located as far as possible from the sink node. We consider a node layout on a grid and a transmission range in the same order than the distance between nodes. Instead of relaying on the failures induced by the simulator, we generate channel failures with a lose probability of 0.01. Finally, to get a fair comparison with the flooding algorithm, we have used an implementation of our algorithm without retransmissions.

3.1 Evaluation Results

In this subsection we summarized the result obtained from the experiments.

3.1.1 Scalability

We have obtained that for our algorithm start-up times increase linearly with the size of the network, from 2 seconds for 25 nodes to 6 seconds for 225 nodes.

Ours algorithm outperforms the flooding algorithm regarding message delivery, message load and latency, as shown in Figure 2, Figure 3 and Figure 4 respectively.

3.1.2 Root Location

We have obtained that start-up times of our algorithm do not depend significantly on the location of the sink. Regarding the rest of the performance parameters, in general a centered location is beneficial. Specifically,

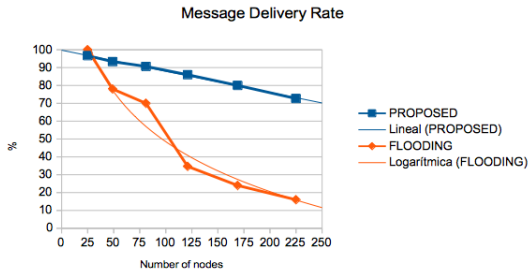

Figure 2: Proposed algorithm versus flooding algorithm regarding message delivery rate.

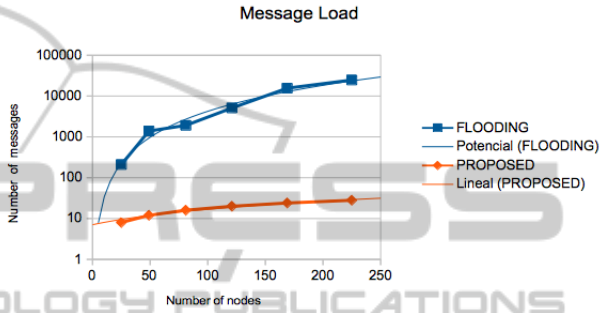

Figure 3: Proposed algorithm versus flooding algorithm regarding message load.

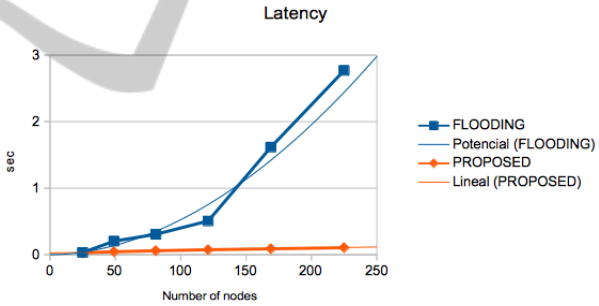

Figure 4: Proposed algorithm versus flooding algorithm regarding latency.

the latency is reduced to less than one half when the sink is located in the middle of the network with respect to a corner location.

3.1.3 The Influence of Link Quality

We generate different link qualities based on the distance between nodes and the transmission range. We have evaluated our algorithm and the flooding algorithm with base link lose probabilities of 0.01, 0.1 and 0.2 which increases quadratically with the distance between nodes.

We have obtained that start-up times, latencies and message loads do not depend significantly on the link quality. On the contrary, message delivery rates in both algorithms are significantly affected by link

quality. Specifically, in our algorithm we have obtained that message delivery rate decreases linearly from near 90% to 65% as base lose probability increases from 0.01 to 0.2.

4 DISCUSSION

As we have seen, our algorithm outperforms the reference flooding algorithm in every evaluation criteria. These results are as expected, since we have compared our algorithm to a force-brute algorithm with no optimization. A flooding algorithm should be very good in message delivery rate. However, when data generation rates are high, as it is the case of our experiments, the fowarding of messages results in a network collapse and the message delivery rate (and possibly latencies) drop.

On the other hand, an algorithm as the proposed in this paper will optimize the network traffic (and other parameters, as battery waste). The results we have obtained confirm this fact.

Currently we are carrying out more experimentation in order to (a) determine the key parameters to be tuned in order to improve the performance of our algorithm, and (b) compare our algorithm to similar approaches, as the RPL (Winter et al., 2012) algorithm.

REFERENCES

Al-Karaki, J. and Kamal, A. (2004). Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6–28.

ALMomani, I., Saadeh, M., AL-AKhras, M., and AL-Jawawdeh, H. (2011). A tree-based power saving routing protocol for wireless sensor networks. *International Journal of Computers and Communications*, 5(2):84–92.

Alwan, H. and Agarwal, A. (2009). A survey on fault tolerant routing techniques in wireless sensor networks. In *Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications*, SENSORCOMM '09, pages 366–371, Washington, DC, USA. IEEE Computer Society.

Boettcher, P., Coffin, D., Czerwinski, R., Kurian, K., and Nischan., M. (2003). Declarative routing protocol documentation. In *Project report*.

Boukerche, A., Pazzi, R. W. N., and Araujo, R. B. (2006). Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing*.

Challal, Y., Ouadjaout, A., Lasla, N., Bagaa, M., and Hadjidj, A. (2011). Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks. *Journal of Network and Computer Applications*, 34(4):1380 – 1397. Advanced Topics in Cloud Computing.

Doherty, L., Simon, J., and Watteyne, T. (2012). Wireless sensor network challenges and solutions. *Microwave Journal*, (Agosto):22–34.

Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–.

Kim, S. (2004). Reliable transfer on wireless sensor networks. In *In SECON*, pages 449–459.

Li, Y., Chen, J., Lin, R., and Wang, Z. (2005). A reliable routing protocol design for wireless sensor networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 4 pp.–61.

Nelakuditi, S., Lee, S., Yu, Y., Zhang, Z.-L., and Chuah, C.-N. (2007). Fast local rerouting for handling transient link failures. *IEEE/ACM Trans. Netw.*, 15(2):359–372.

Tian, D. and Georganas, N. D. (2003). Energy efficient routing with guaranteed delivery in wireless sensor networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1923–1929 vol.3.

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., and Alexander, R. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550 (Proposed Standard).

Ye, F., Zhong, G., Lu, S., and Zhang, L. (2005). Gradient broadcast: a robust data delivery protocol for large scale sensor networks. *Wirel. Netw.*, 11(3):285–298.

Yousefi, H., Dabirmoghaddam, A., Mizanian, K., and Jahangir, A. (2009). Score based reliable routing in wireless sensor networks. In *Information Networking, 2009. ICOIN 2009. International Conference on*, pages 1–5.

Yu, M., Mokhtar, H., and Merabti, M. (2007). Fault management in wireless sensor networks. *Wireless Communications, IEEE*, 14(6):13 –19.

Zhang, H., Arora, A., ri Choi, Y., and Gouda, M. G. (2007). Reliable bursty convergecast in wireless sensor networks. *Computer Communications*, 30(13):2560 – 2576. Sensor-Actuated Networks *SANETs*.