Dependable Routing Protocol Considering the k-Coverage Problem for Wireless Sensor Networks

Hamza Drid, Laszlo Gonczy, Samer Lahoud, Gabor Bergmann, Miklós Molnár

Dependable Routing Protocol Considering the k-Coverage Problem for Wireless Sensor Networks

Hamza Drid, Samer Lahoud
University of Rennes I–IRISA
Rennes, France
Email: {hdrid, slahoud}@irisa.fr

László Gönczy, Gábor Bergmann
Budapest Univ. of Technology and Economics
Budapest, Hungary
Email: {gonczy, bergmann}@mit.bme.hu

Miklós Molnár
University of Montpellier II–LIRMM
Montpellier, France
Email: miklos.molnar@lirmm.fr

***Abstract*—Fault tolerance and periodical changes of the network topology are two important attributes that should be carefully designed in Wireless Sensor Networks (WSN). In this paper we propose a new routing protocol for improving fault tolerance of WSN which takes into consideration the measurement accuracy requirements (expressed as a lower limit on k-coverage) and network performance. The main idea of the k-coverage problem is to schedule the sleeping time of sensors in order to preserve their energy and maximize network lifetime. The proposed routing protocol offers a protection against link and node failures by computing k disjoint paths. It also takes into account the changes of the network topology caused by the scheduling of sensor sleeping.**

***Keywords*-Fault tolerance; Sensor Networks; k-coverage; Routing; Protection;**

I. Introduction

Wireless sensor networks (WSNs) are generally deployed to monitor areas and provide measurements for surveillance applications. WSNs have several application domains: to monitor the environment/habitat, to collect information, to register and process environmental parameters for optimization or prediction, and/or to insure security ([2] describes many applications and challenges). Often, the measurement or surveillance task of a WSN requires the complete coverage of a target area or a set of target objects. In a general WSN architecture, several sensor nodes send the observation data to Base Stations (BSs) or sinks. Then data can be processed by the sinks and later send to the potential clients. The sensors are performing sensing and communication tasks and the main problems and challenges of this kind of networks are associated to these two activities [1]. The sensor network should be capable to measure in the target area and to process the measured values and transmit them to sink nodes. As several critical applications can depend on the measurement results, reliability of the overall network is a key issue, including both measurement reliability (which often requires multiple nodes to measure the same area) and communication (supported by routing).

Routing in WSNs is an important issue that addresses the delivery of the sensed data from source sensor nodes to the sink. Due to the inherent characteristics of WSNs. The routing protocol should be simple and able to deal with a very large number of nodes (scalable), the limited energy, the limited processing and storage capacities of nodes and also to be self-configurable regarding node failures and changes of the network topology.

Many routing protocols have been proposed in the literature. These routing protocols can be classified in three classes: Data centric routing, hierarchical routing and location-based routing. In data centric routing [8], [12], [4], all nodes are typically assigned equal roles; the base station sends queries to certain regions and waits for data from the sensors. In hierarchical routing [6], [7], [3], the network is divided into clusters, in each cluster the node with high energy is used to process and send the information while the other nodes are used to perform the sensing. In location-based routing [11], [5] there is no addressing scheme available, and nodes are addressed by location. The location is obtained by distance estimation, neighbor discovery or GPS.

Previous works have considered routing constraints such as node mobility, energy limitations, high network density and limited computation capabilities. However, only a few works have actually considered the node failures and the periodical changes of the network topology.

In our previous works [9], we developed the Controlled Greedy Sleep (CGS) algorithm which performs well in prolonging network lifetime and offering reliable measurement results (by ensuring k-coverage) at the same time. It has been proven that this results in a quasi-optimal solution. Here we investigate how an efficient routing algorithm can be implemented in this context, using efficiently ”cross-layer” information, i.e., the scheduling of nodes determined by CGS.

Our main contribution in this paper is to propose a new routing protocol for fault tolerance in WSN. The overall paper is organized as follows. Section 2 presents K-Coverage Problem and the previously developed CGS algorithm. Next, Section 3 describes our assumptions and the proposed routing algorithm, illustrating the operation on an example. Conclusions and directions for future work are presented in Section 4.

II. K-Coverage Problem and CGS

Generally, sensors have limited power capacity but WSNs has to meet relatively strong lifetime requirements, thus the energy conservation is a critical issue in WSN. Mostly, high density of sensors and a *scheduled sleeping* algorithm are employed to preserve energy of the sensors and provide the network services. Sensors periodically alternate between *sleep* and *awake* states using a given period length T, thus saving energy. On the other hand, the notion of *k-coverage* refers to the requirement that each measurement area should be covered by at least *k* sensors. Our aim is to find a balance between these contradicting requirements.

A. Sensor Network Model

Generally the communication range of a sensor is greater than the twice of the sensing one; so, one can suppose that the sensors sharing the measurement/observation task anywhere can also communicate one with the other. Consequently, if the target area is covered with awake sensors, then the connectivity of the WSN is also ensured [10].

We consider a WSN consisting of *n* homogeneous sensors $s_1, s_2, ..., s_n$. A WSN is considered homogeneous if its sensors have the same sensing and communication range. We also assume that sensors are static and each sensor knows its own location (x_i, y_i) and where the sink is located. The position of the sink must be broadcasted throughout the network at the start (or when the sink moves). The information about the actives neighbours such as the location and energy are provided by CGS. Our routing protocol can be executed at the beginning of each CGS period.

B. The Controlled Greedy Sleep Algorithm

In this scheduling nodes have local information on their neighborhood only. Each sensor node q will use a locally known sub-graph $G_q(S_q \bigcup R_q, E_q)$. This sub-graph contains geographical regions R_q covered by q, the set S_q of sensors which participate the coverage of at least one region of R_q. E_q contains the edges between the regions and sensors. The scheduling is based on a particular factor.

The coverage ratio is positive if the region is over-covered, and negative otherwise: in this latter case the operation of all sensors covering r is essential. Moreover, the smaller the energy of q, the larger its *drowsiness*. CGS enforces the sensors in critical positions to go to sleep whenever it is possible, to conserve their energy for times when their participation will become inevitable. A sensor q can go to sleep when its neighbors with larger drowsiness factor decided their state for the next period and q has no critical (not over-covered) region to monitor. Consequently, each sensor should know the drowsiness factor of its neighbors and the decision of neighbors with larger factor. To organize the local communication, a communication delay (DTD) is associated with each sensor. This delay is inversely proportional with the drowsiness factor. So the sensors with large factor broadcast their decision earlier. Only the awake state decision should be broadcasted explicitly, in this way the communication overhead can be minimal.

The main steps of the Controlled Greedy Sleep (CGS) Algorithm are the followings:

1) At the beginning of the period, wake up all sensors whose remaining energy is enough for spending at least one period awake.
2) Alive sensors broadcast local *Hello* messages containing their locations. Based on received *Hello* messages each sensor q builds up its local set of alive neighbors S_q and generates the local bi-partite graph $G_q(S_q \bigcup R_q, E_q)$, and then it calculates its drowsiness factor D_q.
3) Based on D_q each node q selects a Decision Time Delay (DTD_q). Small drowsiness means large DTD, large drowsiness means small DTD. These delays provide priorities when nodes announce their Awake Message (AM). Each sensor q broadcasts its DTD_q and starts collecting DTD and AM messages from the neighborhood. From the received DTD and AM messages it builds a Delay List (DL_q) and a List of Awake Neighbors (LAN_q) respectively.
4) When DTD_q time elapsed the node q makes a decision based upon LAN_q and DL_q:
   - if all regions in R_q can be K-covered using only nodes present in LAN_q and/or nodes u present in DL_q for which $DTD_u > DTD_q$ then go to *sleep*
   - otherwise stay *awake* and broadcast an AM to inform the neigbors of this decision.

Obviously, the communication overhead of the algorithm depends on the length of periods. At the begining of the period there are three time intervals: to exchange *Hello*, DTD and AM messages respectively. A sensor broadcasts at most three messages during T_e (two if it the node will go to sleep, three otherwise) and must stay awake in order to complete the election process. During this extra T_e time sensors consume energy. The scheduling communication and awake-time overhead can be low if the length T of a period is significantly longer than T_e. But, one can state that this period can not be too long either.

The determination of the optimal length T is a hard computation problem. In real cases only empirical and estimated solutions can be formulated. The study of this period length with simulation offers significant elements to choose the period length.

For a detailed presentation of CGS, the reader is referred to [9].

III. Routing Proposal

The following requirements were posed for this routing:

1) It should take into account the information on sensor status, determined by the scheduling algorithm.

2) It should be efficient in terms of forwarding messages to nodes which are close to optimal (so that the expected hop count remains low).
3) It should be fault tolerant in an efficient manner (c.f. req.2.), i.e., two disjoint paths should exist for all source-sink pairs. By disjoint paths, we mean to paths which have no common nodes.

Relevant Information obtained from CGS: In order to efficiently use the energy of the sensor network and have as few lost messages as possible, the routing algorithm needs the following information from CGS:

1) The position and remaining energy of neighbors (contained in *Hello* messages). Note that here only the neighbors which measure the same are will be known, not all neighbors within the communication radius. This would lead to situations where not all possible routes are used. The enhancement of this would need a modification of CGS.
2) The sensor status in the next CGS scheduling period should be considered; here the routing will count only on sensors which remain awake. This will be contained in *AM* messages. Note that the periods of CGS do not limit the routing in a sense that message trasnfer is not blocked during the decision mechanism, in contrary, as all (alive) sensors will be awake, all possible routes will be available (although not necessarily utilized).

A. Assumptions

We made the following assumptions during our work:

- The source node is coded in the header of each message.
- The algorithm itself does not ensure k-connectivity, rather it tries to utilize existing connectivity with efficient routing.
- The information of CGS (as described in previous section) is available for the routing algorithm.
- The communication area is contained by the measurement area. If this does not hold, then the number of neighbours wrt. measurement (revealed by CGS messages) does not tell relevant information on the number of nodes wrt. communication. Note that this is true for most practical cases, a counterexample can be a relatively big circle which contains sensors mostly at the perimeter while the sink is in the middle.

B. Proposed Algorithm

The algorithm that we describe here is intended to compute two disjoint paths. In our algorithm, only sensors currently active to k-cover can participate in data forwarding.

When useful data is sensed by the source node si, si sends the sensed data to two neighbours based on its local information. The selection of the two neighbours can be based on different criteria. Our selection is basically based on the remaining energy and the position of the neighbour node relative to the sink node and the sender node. The sensor nodes which have maximum value of F(j) will be selected as relay nodes.

$$F(j) = E(j) \times \frac{d(s,j)}{d(j), Sink} \times cos(a_j) \times awk(j) \tag{1}$$

where *E(j)* is energy available of sensor node j in the set of candidate set Y, a_j is the critical angle created by the coordination of node j, the sender node *s*, and the sink. $d_{(s,j)}$, $d_{(j,sink)}$ are distance from the sender node s to node j and distance from node j to the sink. $awk(j)$ is binary variable, it takes the value of 1 if the sensor node$j$ is awake. Otherwise, it is 1.

Each node received the data packet saves the packet head into own cache, and send the data packet to the best relay node which has maximum value of $F(j)$. The number of neighbours Nb in the slice of the sender node is also sent to the relay node as shown later in the example.

If a node j receives the same packet twice(node j checks its cache to verify if it has already received the same packet), the following control packets are exchanged between concerned nodes in order to obtain two disjoint paths.

For all j nodes where $j! = sink$

1) j sends a control packet to the sender k (from which it has received the data packet and that has a bigger number of neighbors).
2) k selects a new relay node in the set of candidate nodes $Y - j$ using (1). (cf. figure 1, k selects z).
3) j still a relay node but for the sender which has the minimum number of neighbors node. (cf. figure 1, j the relay node of A).

The process continues until the data reach the sink. In order to enhance more reliability, we can also impose that, when a sensor forwards a data packet towards the sink, it would also send an ACK to the sender from which it has received the packet.

C. Example

Figure 1 illustrates an example of disjoint paths construction. In order to construct two disjoint paths, the source sends the sensed data to two neighbours A, K based on its local information.

A and *K* nodes received the data packet from the source, *A* and *K* send back the data packet to the best relay node , which is *J* in our case (cf. Fig.1 (a)). *J* receives the same packet twice. In this case *J* informs *K* to change the relay node (cf. Fig 1 (b)). *K* was chosen because it has several neighbours, while A has only j as neighbour. *K* chooses another neighbor and the process continues until the data reach the sink as shown in (cf. Fig 1 (d)).

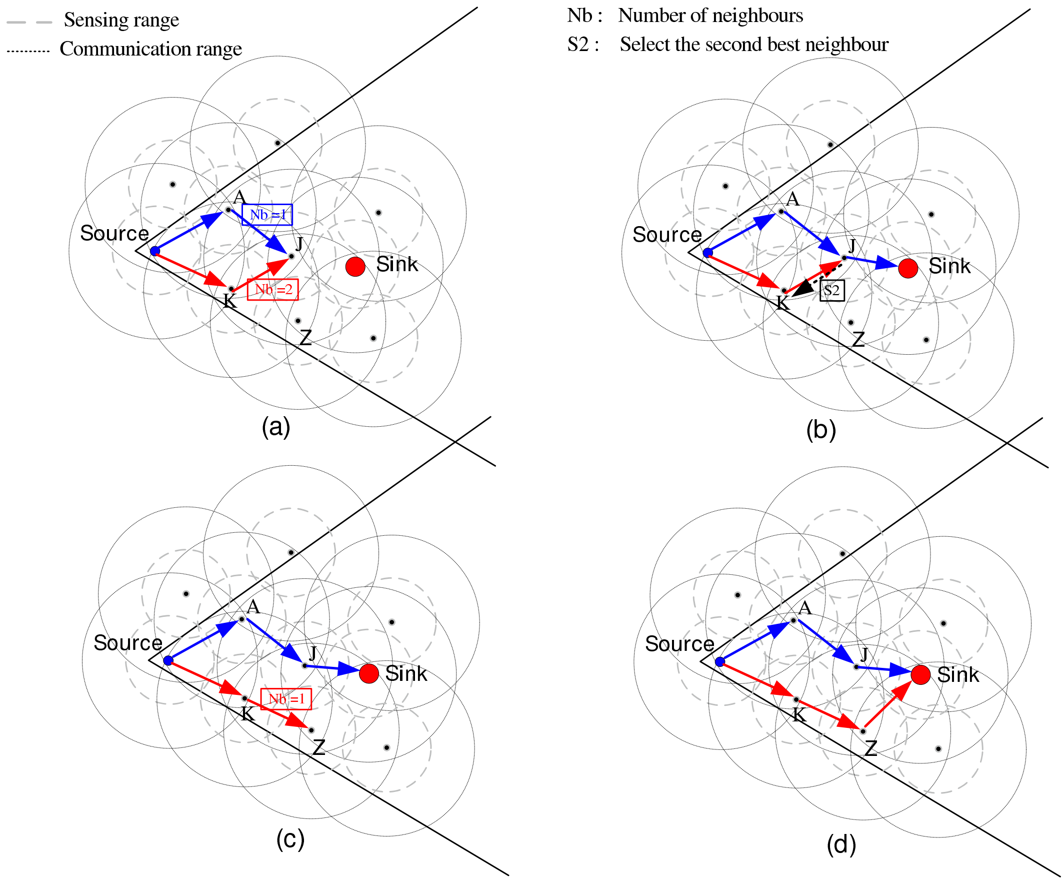

Figure 1. Protection using p-Cycle

IV. Conclusion and Future Steps

Benefits of this routing include a message transfer to the sink considering both energy of sensors and direction, fault tolerance by ensuring two disjoint paths and efficient use of awake sensors based on information obtaind from CGS messages.

The algorithm should be validated in practice by executing simulations on different topologies. Also the effect of period length and communication intensivity (frequency of messages) would be interesting to investigate. Our expectation is that small period lengths will be efficient, especially if there is intense message communication which may cause a faster deprecation than expected (so that the drowsiness factor of some sensors will decrease fast).

A possible enhancement to this algorithm would be to incorporate knowledge on previous decisions. This could be done e.g., by storing a list in each active sensor containing source nodes for which there was a collision (so messages from that particular sensor should be forwarded to the ”second best” neighbor in order to avoid unnecessary control messages). Note that this list should be refreshed for each period, as the routes may change as the scheduling of CGS changes node status.

An important research step would be to investigate the connection between *k-coverage* and *k-connectivity* (where the guaranteed k values can differ for the two properties!). Also the presence of articulation nodes can obviously affect routing (even if k-coverage may be ensured for major parts of the network).

We have investigated how CGS can work together with different sensor types where multiple measurement objectives exist and these can be covered by different types of sensors (e.g., some sensors can measure humidity and temperautre while others only humidity, etc.). It would be interesting to see how the routing can work on top of this (considering that communication itself is not restricted to equivalent sensors).

Also one can consider an extension to information sent in the decision phase of CGS to include all sensors within the communication radius and therefore enable a more efficient routing.

An additional future step is to investigate the effect of aggregation, where messages from different source nodes/ different messages from the same source node are waited for and sent together. This might result in less communication (longer lifetime) with an increased expected message transfer time.

Finally, the effect of realistic sensing/communication radius wrt. obstacles among sensors remains a future research question.

REFERENCES

[1] N. Ahmed, S. S. Kanhere, and S. Jha. The holes problem in wireless sensor networks: a survey. *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9(2):pp. 4–18, 2005.

[2] C.-Y. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, vol. 91(8):pp. 1247–1256, 2003. URL http://dx.doi.org/10.1109/JPROC.2003.814918.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. In *Proceeding of the Hawaii International Conference System Sciences*. 2000.

[4] W. Heinzelman and H. B. J. Kulik. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*. 1999.

[5] L. Li and J. Y. Halpern. Minimum energy mobile wireless networks revisited. In *IEEE International Conference on Communications*. 2001.

[6] S. Lindsey and C. Raghavendra. Pegasis: power efficient gathering in sensor information systems. In *IEEE Aerospace Conference*. 2002.

[7] S. Lindsey, C. Raghavendra, and K. Sivalingam. Data gathering in sensor networks using the energy delay metric. In *IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*. 2001.

[8] A. Manjeshwar and D. Agrawal. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In *Parallel and Distributed Processing Symposium*. 2001.

[9] G. Simon, M. Molnár, L. Gönczy, and B. Cousin. Robust k-coverage algorithms for sensor networks. *IEEE Transactions on Instrumentation and Measurement*, vol. 57, 2008.

[10] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. Sen. Netw.*, vol. 1(1):pp. 36–72, 2005.

[11] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 01)*. 2001.

[12] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. In *SIGMOD Record*. 2002.