Reliable Cost-based Data-centric Routing Protocol for Wireless Sensor Networks

Jian Wu, Paul Havinga
Department of Electrical Engineering, Mathematics and Computer Science
University of Twente,
Email: j.wu|p.j.m.havinga@utwente.nl

Abstract—The resource limitations of wireless sensor networks (WSN), especially in terms of energy, require novel and collaborative approaches for the wireless communication. In this paper, we focus on the dynamic aspects and present a new reliable cost-based data-centric routing algorithm for such dynamic WSNs. Current research in this area generally assumes a rather static network, leading to a strong performance degradation in a dynamic environment. A network wide reflowing of messages is the common solution to network topology changes. The situation gets worse when the data sink moves, when a stable network is hardly able to form. In our research we try to maintain the communication when sensors move, such that less energy is used to re-set-up the network. Moreover, event mobility poses a great pressure on the WSNs. We have designed the routing protocol to route message intelligently to reduce the effect of event mobility. Simulation results show that our approach to data-centric routing for WSNs benefits clearly from the global-local paradigm we are able to use. In a static network, our approach increases the network reliability with at least 25%. And in a dynamic topology, the local recovery approaches of our protocol clearly outperforms the traditional protocol in reliability, while the energy consumption is less than 50%.

I. INTRODUCTION

One of the key tasks of Wireless Sensor Network (WSN) is to deliver sensed data to interested parties who requested it. As these networks are expected to be densely deployed, a major issue in the study of networking protocols for sensor networks is related to data dissemination and routing. In recent years, a number of papers have appeared which present solutions which are relevant to the issues of routing and data dissemination in sensor networks. In these works, Data aggregation has been highlighted as a particularly useful paradigm for routing in wireless sensor networks. The idea is to combine the data coming from different sources to eliminate redundancy, minimize the number of transmissions and thus save energy. This paradigm shifts the focus from the traditional address-centric approaches, which tries to find short routes between pairs of addressable end-nodes, to a more data-centric approach which focus on finding routes from multiple sources to a single data sink, that allows in-network consolidation of redundant data.

In [1], C. Intanagonwiwat et. al. proposed a popular data aggregation paradigm for WSNs, called directed diffusion. Directed diffusion is a data-centric and application aware paradigm in the sense that all data generated by sensor nodes is named by attribute-value pairs. Moreover Schurgers et al. [2] proposed another variant of directed diffusion, called Gradient-Based Routing (GBR). The key idea in GBR is to memorize the number of hops when the interest is diffused through the whole network. However, in both works the mobility of the network is almost neglected, which poses a great challenge in the dynamic network environment. Any topology change in the network requires a network wide flooding of messages to reset-up the gradient field. More recently, GRAB (GRAdient Broadcast) [8] protocol has been proposed. The basic idea is to make data packets issued by a sensor be delivered along the direction of a sink by descending some cost, which are initially built and maintained by the sink but kept by each sensor. GRAB requires each node’s cost value to be periodically refreshed by a sink initiated flooding, causing the same problem of excessive overhead under dynamic network.

This paper presents a reliable cost-based data-centric routing protocol for wireless sensor networks, as part of the European research project EYES [3]. In this work we address in particular energy efficiency and the dynamics of wireless sensor network. Instead of each sensor sends its own data report directly to the data sink, we introduce a global-local gradient paradigm to only send the aggregated data from the center of the event to the data sink. It ensures that sensors aggregate the collected data as close as possible to its origin, while only a small number of aggregated data are sent to the data sink. To increase the reliability of these aggregated data, they are sent via multiple adjustable routes to the data sink. Local algorithms are designed to resume the network gradient when the network topology changes, especially the mobility of the data sink are solved with negative gradient. Further, the movement of sensed event, such as in the event tracking applications, is also efficiently dealt with. Section II discusses the design of the reliable cost-based data-centric routing protocol (RCDR), which allows to exploit the benefits of the data aggregation approach discussed in this paper. Finally Section III and IV gives the simulation results and conclusion.

II. PROTOCOL DESIGN

The proposed reliable cost-based data-centric routing (RCDR) protocol has two kinds of gradients, global gradient and local gradient. When a data sink wants to collect data from the network, it sends out data query to setup global gradient in the whole network. While this query message propagates in the network, each sensor establishes its own cost value toward this sink. Afterward, any data sent towards the sink follows through global gradient by multipath routing. The multipath degree is controlled by the premium cost of the data. Sensor movement adjustment scheme and Sink movement compensation scheme efficiently resume the disrupted global gradient by local interactions between sensors. Thus energy...
expensive reflooding in the network are reduced to a minimum, while still maintain the reliability of the network.

![Diagram](image_url)

**Fig. 1. Local and global report forwarding**

Locally the footprint gradient of the event’s effect is used to setup the **local gradient**, which override **global gradient** in the local event area (LEA). When the sensor senses an event and sends a event report, it is first routed to the local maxima which is the data aggregation point, as shown in the shaded area of Figure 1. When the local maxima has already processed the data from the surrounding sensors, it sends the aggregated report to the sink via **global gradient**. Further more, **Local maxima handover scheme** lets the data aggregation point moves along with the event, so that data is always aggregated locally closest to the event center.

### A. Global gradient

The gradient setup follows a similar approach to the one in [8]. However, in our work we introduce more flexibilities to the cost concept, in order to control the characteristics of data forwarding. Moreover **waiting time** and **forwarding probability** is used to minimize the delay and reduce the broadcast storm problem.

1) **Global cost setup**: When the data sink wants to monitor or query the network, it sends out Data Query (DQ) message to the network, which will be rebroadcasted by the sensor nodes. The intermediate sensors not only remember the data query but also activate its sensor. From the content of DQ message node $i$ also sets up a cost $C_i$ from itself towards the data sink. Initially sensors have no information from the data sink, so it sets its cost to be $\infty$. The DQ message send out by the sink has a cost of $C_{DQ} = 0$. Each node has a link cost table to all its neighbors, which is explained in the following section. If intermediate node $i$ receives a DQ message from its neighbor $j$, it adds the link cost $C_{ij}$ to the cost of received DQ message and it sets new cost to be $C_i = min(C_i, C_{DQ} + C_{ij})$. Meanwhile it remembers from which node it could transfer data with the lowest cost. we call this node the Lowest Cost Neighbor (LCN). If the node receives the same DQ message from a different neighbor, it computes the value again and updates the cost and LCN node if necessary. After a small waiting time $T_w$, the node rebroadcasts the DQ message with its current $C_i$ as the $C_{DQ}$ and with a forwarding probability $p_f$. It ensures that the node only broadcast once the same DQ message, although multicopies could be received from different neighbors. Outside waiting time $T_w$, any copy of the same DQ message is ignored. After **global gradient** setup, each node in the network has a cost $C_i$ and the whole network becomes a directed graph toward the sink.

The waiting time $T_w$ should be able to reduce the effect of broadcast storm, which result in serious redundancy, contention and collision [9]. At the same time, it should ensure that the node obtain the minimum cost $C_i$ in the shortest delay. If all the nodes have the same waiting time $T_w$, then the neighbors will content for the medium at the same moment which and cause collisions. The simple solution is to let the nodes have a random waiting time between $[T_{min}, T_{max}]$. This will greatly reduce the chances of collusion in a collision detection based MAC layer. However, it appears to accumulate very large delays in the far end of the network. Further improvement is to have a random waiting time proportional to the node’s cost $C_i$ to the sink. The further away form the sink, the larger the waiting time.

Each node in the network maintains a link cost table $T_{ab}cost$ to all its neighbors. For each neighbor in the radio range, it has a cost value in $T_{ab}cost$. The selection of cost value decides the random forwarding character of this sensor network. For example, a delay related cost would make this WSN has overall lower delay; a energy related cost would increase the life tiem of the WSN.

2) **Parameters in the Forwarding**: Globally a node sends its data to the sink by broadcasting the data with several forwarding parameters. Firstly the sequence number of the data uniquely identifies the data. Each node will only forward the first data it receives and ignore other copies.

Each data also has a cost of its own, which consists of two parts. The **basic cost** of a data is the cost of the node which send the data. It can be expressed as $C_{basic} = C_i$. This is the minimum cost the data should have in order to reach the sink from node $i$. The **premium cost** of a data is set to control the multipath degree of the forwarding paths. When the data has more premium cost to spend, it will travel further away from the low cost paths and go through higher cost area. The premium cost will increase the reliability of the forwarding at the expense of more energy consumption. Thus the overall cost of a data is expressed as $C_{data} = C_{basic} + C_{premium}$. When a intermediate node $j$ receives a new data, it compares the cost of the data $C_{data}$ with its cost $C_j$. If $C_{data} \geq C_j + C_{ij}$, the data has enough credits to go through this node towards the sink. It will forward the data with a new cost $C_{data} = C_{data} - C_{ij}$. $C_{ij}$ is the link cost between node $j$ and node $i$. On the other hand, if $C_{data} < C_j + C_{ij}$, the data has no more credits to be forward. Node $j$ will drop the data.

3) **Sensor Movement Adjustment**: The routing protocol presented in this work is a multipath routing protocol. When the data travels in the network towards the sink, it flows through the lower cost nodes as shows in Figure 2. If the connections between node A and node C breaks because node
C move away from node A, the data from node A can still go through both node B and node D. If the network density is high enough, the data will bypass the troubled link and resume the reliable multipaths as shown in Figure 2.II. So the movement of individual sensor node does not break the data transfer in its original location area.

The effect of a moved sensor node in the new allocated location area is also negligible. When a node moves, it could move in two directions in respect to the sink as show in Figure 3. If node a moves into higher cost area, it will have the lowest cost value among its neighbors. Then it forwards any data from its neighbors, but in turn its neighbors forwards none of its data. It node a moves into lower cost area, it will have the highest cost value. Then it forwards no data from its neighbors, but its neighbors forwards any of its data. In both cases, node a is excluded from the network communications.

If only a small number of nodes in the network move, the network can still remain functional. However, if more and more nodes are excluded from the data forwarding, the network becomes very unreliable. A network wide reset is needed from the sink to restore the gradient field. However, frequently reset consumes too much energy for the energy restrained sensor nodes. A sensor movement adjustment scheme is designed to minimize the effect of sensor movement by local interactions.

Sink movement compensation: Data sink is the data collection point in the WSN. It receives aggregated data from the network. In the operation of WSN, various scenarios requires the data sink to be able to move in the network while collecting the sensing data. Any data loss caused by the sink movement decreases the reliability of the network. Previous work on the routing of WSN mostly focus on static network. Particularly in the data centric route scenarios, any movement in the network will disrupt the network setups and results in data losses. When the sink moves, normally network wide reset is needed to restore the network gradient as in [5][6][8]. This paper introduces a new sink movement compensation scheme with negative gradient, which only requires a local update in order to compensate the sink movement.

The data sink sends out Data Query (DQ) message to the whole network, which will be rebroadcast by the sensor nodes. These initial DQ messages sent out by the sink have a cost $C_{DQ}$ of 0. When the global gradient is setup in the network, sensing data could travel from the sensor nodes to the data sink by following the gradients. When the sink move away from its current location, it should be able to first detect its own movement before it could carry out adjustment for the gradients.

First, we design a method to detect movement without additional hardware or localization protocol. Again the proposed method of movement detection is to monitor the changes of the sink’s neighbors. When any new neighbor appears or any old neighbor leaves, we could infer that the sink has changed its location. However, this could also be the case that the sink remains static but its neighbors move away or they are simply switched on and off. We treat the second case the same as the first one and it has little effect on the sink movement compensation. The only disadvantage is more frequent local updates which involves a small number of radio transmission.
When sink detects its own movement (or relative movement), it sends out negative gradient broadcast locally to compensate its own movement. Initially DQ messages sent out by the sink have a cost $C_{DQ}$ of 0. Immediately after it moves, it decreases $C_{DQ}$ to -1 and broadcasts the new Degraded Update (DU) messages with a Hops-to-Live (HTL) field set to $h$. When the node receives the DU messages, it follows through the steps described in section II-A.1 to set its own cost. Firstly it checks the HTL field. if HTL > 0, it lowers its own cost and rebroadcast the DU messages with new HTL=HTL−1. The new DU message propagates until it reaches the $h$ hops neighbors. Thus all the sensors in this degraded area sets its cost one step lower towards the new location of the sink. It creates a small funnel with negative gradient around the sink in the global gradient field. As a result, when the data sent by the node reach the vicinity of the sink, it will still flow to the new allocated sink by following the “small funnel”. In this way, a network wide readjustment is avoided by only a locally restricted gradient broadcast.

The sink repeatedly decreases the cost and increases the HTL of the DU messages when it moves again. In order that the DU messages can still reach the original location of the sink, the degraded area will expand accordingly. The proposed relationship between the cost and HTL is $h = H - C_{DQ}$, where $H$ is a constant.

### B. Local Gradient

Locally we use the footprint gradient of the event’s effect to route the event report. When the sensor sends a event report, it is firstly routed to the local maxima, which is the data aggregation point. When the local maxima has already precessed the data from the surrounding sensors, it sends the aggregated report to the sink. The procedure of local maxima election and gradient setup is explained in detail as follows:

Physical event leaves some footprints in the environment, e.g., fire increases temperature, gas leakage increases density. Moreover, most of the physical phenomena follow diffusion property with distance, i.e., $f(d) \propto \frac{1}{d^\alpha}$, where $d$ is the distance from local maxima, $f(d)$ is the magnitude of the event’s effect and $\alpha$ is the diffusion parameter depending on the type of effect. In WSN, the sensors capture this event’s effect in term of sensed signal strength. In [7], this information gradient has been used to route the query from the data sink to the event source. However, in our research, this natural and freely available gradient has been exploited in a different approach, we use it to efficiently forward the event report from the surrounding nodes towards the center of the event or the local maxima.

When a event happens in the WSN, the surrounding sensors will sense the event and generate readings. Because of the diffusion effect, sensor reading is a function of distance $d$ and time $t$, i.e., $R(d, t) \propto t/d^\alpha$. As sensors can’t detect changes if the effect below a certain threshold, the event’s effect is not infinite. After a distance $D$, the sensor readings will be zero and the information gradient is not available. So we define the area inside diameter $D$ as the local gradient area (LGA).

In LGA, all the sensors detect the event and generate event report $R_i$. After a random back-off time between $[0, t_i]$, sensor $i$ starts to send out the event report with its own reading $R_i$ to the neighbors, i.e., $N_i$. The timer $t_i$ has a linear inversely proportional relationship with $R_i$, which means the larger the reading the smaller the waiting time.

After initial round of message exchange, sensors in the LGA get a list of readings of their neighbors.

- If $R_i > \max[R_j]$ ($j \in N_i$), then sensor $i$ becomes the local maxima of this event, i.e., $L_m$. It then will send out LGA gradient setup message to override the global gradient for the local event report, which is explained in the following paragraph.
- If $R_i < \max[R_j]$ ($j \in N_i$), sensor knows it is not the $L_m$. It then rebroadcast the event reports, which have smaller readings with $R_j < R_i$. In this way, the neighbors with $R_j > R_i$ continue forward these reports until it reaches the local maxima.
- If the sensor receives event report, but itself doesn't have any reading, it knows it is at the border of LGA and discards the report.

The local maxima sends out Local Gradient Setup (LGS) messages to setup local gradients, which in the LGA overrides the global gradient. The propagation of LGS messages follows the same rules described in Section II-A. Once the local gradient is established, all the local event reports from LGA flows toward the local maxima, the data aggregation point. Data aggregation performs in-network fusion of data packets, coming from different sensors enroute to the base station, in an attempt to minimize the number and size of data transmissions and thus save sensor energies. Such aggregation can be performed when the data from different sensors are highly correlated as in the LGA, we make the simplistic assumption that an intermediate sensor can aggregate multiple incoming packets into less number of packets.

### III. Simulation Results

#### A. Simulation Testbed

We compared the protocols presented in the previous sections with GRAdiant Broadcast (GRAB) [8] protocol. The same network setup is used to compare the two implementations of routing protocols. The OMNeT++ discrete event simulator, together with a framework for a mobile and wireless network [11], is used in the simulation. For both simulations Sensor-MAC protocol (SMAC) [12], a medium access protocol for wireless sensor networks is implemented to provide MAC layer access. It is a carrier sense multiple access with collision detection (CSMA/cd) protocol. A network of 60 sensors with radio range 150m are randomly placed on a rectangular area of 800m x 800m. The density of the resulting network is controlled by scaling the model. Reducing the area for node placement yields a denser network, i.e. more neighbors within transmission range. The nodes move in this area according to the random way-point model (RWP) with random speed and waiting time.
B. Sensor movement adjustment scheme

In this simulation, we try to find out the reliability of the sensor movement adjustment scheme under different mobility conditions. We only set up global gradient in the network and let one random sensor in the network generates data reports to the data sink. It gives a data flow for 10 minutes with a data rate of 2 packets/s and then another random node takes over. A certain percentage of sensors in the network follow the random walk model and moves in the network and the other sensors remain static during the course of simulation.

Firstly, we compare the success ratio of data delivery between RCRD and GRAB under different moving speed. As shown in Figure 4-A, when only 5% of the sensors in the network moves with a speed less than 2m/s, both RCRD and GRAB are rather reliable. When more sensors move in the network, the success delivery by GRAB decreases sharply. And when the speed is more than 2m/s, its success delivery ratio is less than 50%, which means GRAB is very unreliable and almost none operational in a dynamic network. On the contrary, RCDR shows a much greater resilience to the changing topology due to the sensor movement adjustment scheme.

Secondly, we compare the energy consumption of the whole network under different speed. The simulation time was controlled and the sum of energy consumption of all the sensor are calculated and normalized against the static GRAB network. As shown in Figure 4-B, at low speed both protocols have similar energy consumption. When the speed increases, GRAB consumes more than 2 times the amount of energy than RCDR, which is caused by its more frequently network wide flooding to resume the gradient. However the local adjustment of RCDR shows its advantage at higher speed in the respect of 50% saving on energy consumption.

C. Sink movement compensation scheme

In this simulation, we try to find out the reliability of the network under sink movement compensation scheme. Still we only setup global gradient in the network and let one random sensor in the network generates data reports to the data sink. All the sensors except the data sink remain static during the course of simulation. The sink follows a random walk point model with different speeds. From Figure 5, it clearly shows that compared with GRAB, the sink movement compensation scheme improve the reliability of the network by 20% at lower speed and more than 75% at higher speed. While at the same time, its energy consumption is only 25%-50% of GRAB. The "disasters" situation of sink movement in GRAB is very well solved by our scheme.
D. Data aggregation

we make the simplistic assumption that local maxima sensor can aggregate multiple incoming packets into less number of outgoing packet. The aggregation ratio of incoming packets over outgoing packets r depends on the correlation between the data. A range of [1,5] is selected in the simulation for RCRD. The network randomly produces monitored events of a radius of 200m, which last for a random number of minutes between [2, 5]. Only 40% of the sensors in the network move according to the random walk point model. As any sink movement degrades GRAB significantly, we let the sink remain static all the time. In Figure 6-A, it shows that our protocol has a higher reliability than GRAB under different speed. Furthermore, the data aggregation ratio has no significant impact on reliability as the aggregated data are protected against error and lost by multipath routing. However in the respect of energy consumption as shown in Figure 6-B, higher degree of aggregations clearly gains more from our global-local gradient paradigm.

IV. CONCLUSION

In a dynamic wireless sensor network, the mobility of sensors and the monitored events pose a threat to the efficiency of the routing protocol. In this paper, we discussed a cost-based data-centric routing protocol, which is a simple yet effective algorithm for nodes to recovery the disrupted gradient. Thus a network wide flooding is avoided in the maintainable of the gradient field. Again, these schemes are only based upon local information. The routing protocol benefits from local topology information that is already present in the medium access protocol. The global-local gradient paradigm ensures that sensors aggregate the collected data as close as possible to its origin, while only a small number of aggregated data are sent to the data sink via multiple reliable and adjustable routes. The presented approach is compared with GRAdient Broadcast (GRAB) on top of the SMAC medium access protocol. Simulation results show that our approach to data-centric routing protocol for WSNs benefits clearly from the global-local paradigm we are able to use. While in a static network, our approach increases the network reliability with at least 25%. In a dynamic network topology, the local recovery approaches of our protocol clearly outrun GRAB in reliability while the energy consumption is less than 50%.

In the near future we intend to improve the protocol to have reverse query route, so that the data sink could reinforce the local gradient area. Furthermore, the effect of event mobility should be investigated more in the simulation. We also intend to implement the proposed protocol in a real-life testbed and prove its operation in the field.

REFERENCES