Reliable and Energy-Efficient Routing Protocol in Dense Wireless Sensor Networks

[†]Min Chen, [†]Taekyoung Kwon, [‡]Shiwen Mao, [+]Yong Yuan, [†]Victor C.M. Leung*

[†]Department of Electrical and Computer Engineering
University of British Columbia, Vancouver V6T 1Z4, Canada
Email: minchen@ece.ubc.ca, vleung@ece.ubc.ca

[†]School of Computer Science and Engineering
Seoul National University, Seoul, 151-744, Korea
Email: tkkwon@snu.ac.kr

[‡]Department of Electrical and Computer Engineering
Auburn University, 200 Broun Hall, Auburn, AL 36849-5201, USA
Email: smao@ieee.org

[+]Department of Electronics and Information Engineering
Huazhong University of Science and Technology, Wuhan, 430074, China
Email: yy_hust@hotmail.com
*Corresponding author

Abstract: Delivering sensed data to the sink reliably in sensor networks calls for a scalable, energy-efficient, and error-resilient routing solution. In this paper, a reliable energy-efficient routing (REER) protocol is proposed to achieve the above goals for dense wireless sensor networks (WSNs). Based on the geographical information, REER's design harnesses the advantage of high node density and relies on the collective efforts of multiple cooperative nodes to deliver data, without depending on any individual ones. We first select reference nodes (RNs) between source and sink. Then, multiple cooperative nodes (CNs) are selected for each RN. The reliability is attained by cooperative routing: each hop keeps multiple CNs among which any one may receive the broadcast data packet from the upstream hop to forward the data successfully. The distance between two adjacent RNs provides a control knob to trade off robustness, total energy cost and end-to-end data latency. The main difference between REER and the traditional geographical routing protocols are as following: (1) REER is stateless and does not need to store any neighbor information; (2) In unreliable communication environments, traditional routing protocols may fail to deliver data timely since link/node failures can be found out only after trying multiple transmissions. In REER, each data is only broadcast once at each hop. If there is at least one of the CNs is in good status, the data packet is delivered successfully; (3) In REER, the number of cooperative nodes are adaptively selected before data delivery, such that the number is minimized while achieving required reliability according to the link failure rate. The unselected nodes will enter sleeping mode to save energy during data dissemination. Extensive simulation experiments are carried out to show that REER achieves an efficient trade-off among reliability, energy consumption, and end-to-end delivery latency. We have evaluated the REER protocol through both analysis and extensive simulation.

Keywords: Energy efficient; reliability; routing; wireless sensor networks.

Reference M. Chen, T. Kwon, S. Mao, Y. Yuan and V.C.M. Leung, (2007) 'Reliable and Energy-Efficient Routing Protocol in Dense Wireless Sensor Networks', Int. J. Sensor Networks, Vol. xxx, Nos. xxx, pp. xxx

Biographical notes: Min Chen was born on Dec. 1980. He received BS, MS and Ph.D degree from Dept. of Electronic Engineering, South China University of Technology, in 1999, 2001 and 2004, respectively. He is a postdoctoral fellow in Communications Group, Dept. of Electrical and Computer Engineering, University of British Columbia. He was a post-doctoral researcher in Multimedia & Mobile Communications Lab., School of Computer Science and Engineering, Seoul National University in 2004 and 2005. His current research interests include wireless sensor network, wireless ad hoc network, and video transmission over wireless networks.
Taekyoung Kwon is an assistant professor in the school of computer science and en-

1 INTRODUCTION

Recent years have witnessed a growing interest in deploying a sheer number of micro-sensors that collaborate in a distributed manner on data gathering and processing. Sensors are expected to be inexpensive and can be deployed in a large scale in harsh environments, which implies that sensors are typically operating unattended. Often, sensor networks are also subject to high failure rate: connectivity between nodes can be lost due to environmental noise and obstacles; nodes may die due to battery depletion, environmental changes or malicious destruction. In such environments, reliable and energy-efficient data delivery is crucial because sensor nodes operate with limited battery power and error-prone wireless channels.

These characteristics of sensor networks make the design of a routing protocol challenging. To address such issues, a lot of research focuses on prolonging the network lifetime by exploiting energy-efficiency, supporting reliability, or achieving low-cost sensor design (1; 2). However, these goals are usually orthogonal design objectives.

Among these design objectives, the goal of reliability and energy-efficiency usually conflict each other. We consider two extremes of routing protocols in terms of these two design objectives: unicast routing and flooding. Unicast routing is energy-efficient for reliable networks, but is not robust for dynamic networks. Flooding is very robust for dynamic and error-prone networks, but incurs a high overhead for sensor networks. Some routing protocols try to achieve a trade-off between the two extremes to make this adaptive to different types of networks (with different link/node failure rate, node density, etc.). For example, in directed diffusion (DD) (19), exploratory data is periodically flooded for reliability. When a path is reinforced, it is used for a while with unicast routing in order to save overhead. In this paper, a reliable energy-efficient routing (REER) protocol is proposed to construct a "unicast-like" path, while exploiting broadcast to attain high reliability during data dissemination. REER achieves both reliable and energy-efficient data delivery for dense wireless sensor networks (WSNs).

When sending a packet from source to the sink over multiple hops, REER controls the distance r between two adjacent hops. At each hop, an appropriate number of nodes for cooperatively forwarding the data is selected. The smaller is r, the more nodes can be selected for cooperative data forwarding. Since r decides how many nodes will be selected, it efficiently provides a tradeoff between reliability and energy cost. When r is equal to the transmission range of data packet, REER behaves almost like a unicast fashion. By comparison, if r is very small, REER can be deemed as scope-controlled flooding around the path from the source to the sink. Unlike directional/controlled flooding, REER only selects the nodes which need to participate data broadcasting to achieve required reliability in a hop-by-hop fashion. Thus, the number of nodes involved in data delivery is minimized while achieving required reliability. Furthermore, the unselected nodes will enter sleeping mode to save energy.

Since REER exploits geographical information to construct path, it will be compared with GPSR, a popular position-based approach, by both analysis and simulation. Compared with traditional geographical routing, REER also chooses one next hop relay from multiple neighbors. However, the selection is based on reliability of the corresponding wireless links, rather than geographical locations of the neighbors, thus making it more robust than geographical routing.

We present extensive simulations to show that REER normally yields higher reliability than GPSR. And more importantly, REER also achieves less energy consumption. The overall performance (e.g. reliability, lifetime, and data delivery latency) gain of REER increases as the link/node failure rate increases.

The rest of this paper is organized as follows. Section 2 presents related work. We describe REER design issues and algorithm in Sections 3. Simulation model and experiment results are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

2 Related Work

Our work is closely related to the reliable data transfer scheme in WSN, and geographic routing in WSN. We will give a brief review of the work in these two aspects.

There are increasing research efforts on studying the issue of reliable data transfer in WSN (3; 4; 8; 9; 5; 6; 7). In these work, hop-by-hop (3; 4) recovery, end-to-end (8; 9) recovery, and multi-path forwarding (5; 6; 7) are the major approaches to achieve the desired reliability by previous work. PSFQ (3) works by distributing data from source nodes in a relatively slow pace and allowing nodes experienced data loss to recover any missing segments from immediate neighbors aggressively. PSFQ employs hop by hop recovery instead of end to end recovery. In (4), the authors proposed RMST, a transport protocol that provides guaranteed delivery for applications requiring them. RMST is a selective NACK-based protocol that can be configured for in-network caching and repair. Several acknowledgement based end-to-end reliable event transfer schemes are proposed to achieve various levels of reliability in (9). We also proposed a virtual MIMO based cross layer design in (10). In the design, the nodes can form adaptively the cooperative nodes set to transmit data among clusters. Then, the hop-by-hop recovery scheme and multi-hop routing scheme are integrated into the virtual MIMO scheme to jointly provide energy efficiency, reliability and end-to-end QoS guarantee. In (5), multiple disjoint paths are set up first, then multiple data copies are delivered using these paths. In (6), a protocol called ReInForM is proposed to deliver packets at desired reliability by sending multiple copies of each packet along multiple paths from sources to sink. The number of data copies (or, the number of paths used) is

dynamically determined depending on the probability of channel error. Instead of using disjoint paths, GRAB (7) uses a path interleaving technique to achieve high reliability. It assigns the amount of credit α to the packet at the source. α determines the "width" of the forwarding mesh and should be large enough to ensure robustness but not to cause excessive energy consumption. However, finding a suitable value of α for various reliability requirements of sensor networks is not trivial. Furthermore, when the quality of channel changes frequently, out-of-date α makes GRAB either waste energy to unnecessarily use more paths or fail to achieve the required reliability. It is worth noting that although GRAB (7) also exploits data broadcasting to attain high reliability, it may not be energy-efficient because it may involve many next-hop nodes in order to achieve good reliability and an unnecessarily large number of packets may be broadcast. By comparison, in REER a data packet is only broadcast once at each hop, and it is quite robust to link/node failures. Some researchers explore the special features of sensor applications in reliable protocol design. For example, considering asymmetric many-to-one communication pattern from sources to sink in some sensor applications, data packets collected for a single event exhibit high redundancy. Thus, some reliable techniques (3; 4) proposed for WSN would either be unnecessary or spend too much resources on guaranteeing 100% reliable delivery of data packets. Exploiting the fact that the redundancy in sensed data collected by closely deployed sensor nodes can mitigate channel error and node failure, ESRT (8) intends to minimize the total energy consumption while guaranteeing the end-to-sink reliability. In ESRT, the sink adaptively achieves the expected event reliability by controlling the reporting frequency of the source nodes. However, in the case that many sources are involved in reporting data simultaneously to ensure some reliability (e.g., in a high unreliable environment), the large amount of communications are likely to cause congestion.

Geographic routing is a routing scheme where the location of the network nodes is used for packet forwarding. Geographic routing can be stateless, because the next hop is chosen using the geographic location of the destination, which is stored in the packet header. In contrast to that, non-geographic algorithms let the nodes keep information about routes. In most position-based routing approaches, the minimum information a node must have to make useful routing decisions is its position (provided by GPS, Galileo, etc.), the position of its neighbors (through beaconing), and the final destination's location (through a so-called location service (15)). The most popular forwarding method in this category is greedy forwarding, where forwarding decisions are made locally based on information about their one-hop neighborhood. An overview of geographic routing algorithms can be found in (11). A well-known geographic routing algorithm is GPSR (13). In GPSR, each node maintains a neighbor table which is updated by periodically sending beacon messages. To route around areas where greedy forwarding cannot be used, Greedy Perimeter State Routing (GPSR) (13) tries to find the perimeter of the area. Packets are then routed along this perimeter, around the area.

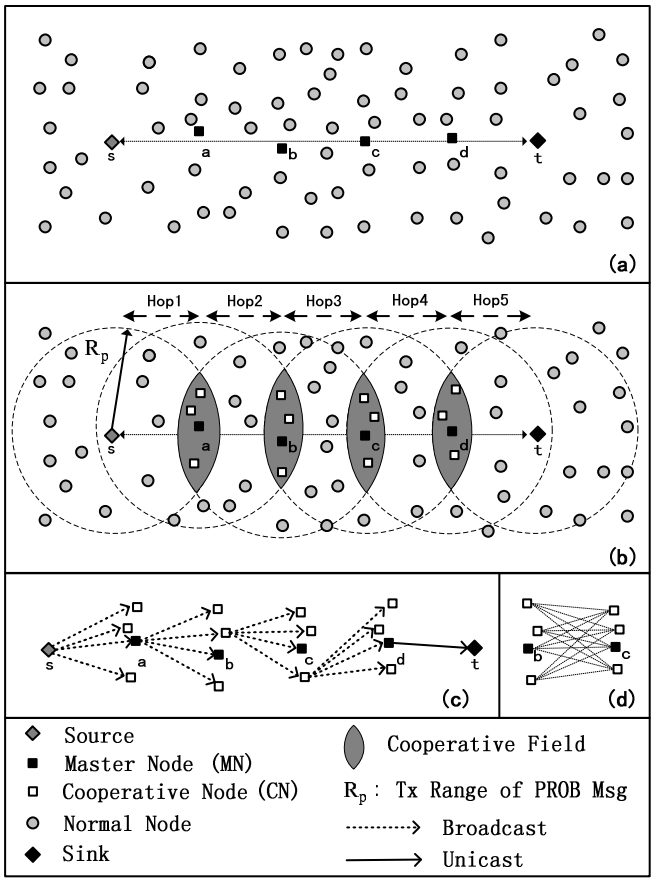

Figure 1: Illustration of the REER Routing Protocol: (a) *RN*s along the shortest path; (b) *CN*s in the cooperative fields; (c) cooperative data forwarding; (d) the forwarding mesh between two cooperative fields

3 System Architecture And Protocol Design

In this section, we present the architecture and design of the REER protocol. We first give an overview of the network organization, and then describe the key REER components in detail. Lastly, we present an analysis that derive the key performance metrics for the proposed protocol.

3.1 Overview

Consider a large scale, dense wireless sensor network, within which a source node, say, node s, generates reports on detected events in Fig. 1. These reports will be delivered to the sink node t via multi-hop routing. Usually sensor networks are deployed in the harsh environments, and thus the wireless links/nodes are failure prone. In addition, the sensor nodes are severely energy constrained due to the low-cost and disposable nature. Therefore, we choose reliability and energy efficiency as the two most important design objectives for REER.

The operation of REER is illustrated in Fig. 1(a)-(c). A set of nodes, termed reference nodes (RNs) between the source and the sink (source and the sink themselves are also RNs) are first selected, such that the distance between two adjacent RNs is sought to be an application-specific value (denoted by r). Note that, more closely are the RNs located to the straight line from the source node to the sink, less hop count should be obtained. In performing RN-selection, upstream RN will broadcast a probe message (PROB) with the transmission range of R. Its neighbors, which receive this PROB and within the RN-selection area in Fig. 3, are called "reference node candidates" (RNCs). The RNs are determined sequentially, starting from the source node. When a node is selected as the RN by its upstream RN, it will perform the RN-selection mechanism again to find its downstream RN, and so forth. In Fig. 1, since the source node s itself is an RN, it initiates RN-selection first to find its downstream RN, i.e., node a. The RN selection mechanism will be detailed in Section 3.2.

After a certain timer expires, the RNs determine a set of cooperative nodes (CNs) around each of them based on the coverage of the PROB messages they sent during RN-selection period. Note that the CN-selection does not need any control overhead.

As shown in Fig. 1(b), for RN b, the area covered by the transmissions of its upstream RN a will be a disk centered at a and have a radius of R, while the area covered by the transmissions of its downstream RN c will be a disk centered at c with the radius of R. As r is set to be smaller than R, these two disks will overlap, and node b will be located within the overlapping area. This overlapping area is deemed as the *cooperative field* of RN b (denoted by CF_b). That is, the sensor nodes in CF_b are the CNs for RN b. The CN-selection mechanism will be detailed in Section 3.3.

After the RNs and CNs are determined, each data packet will be forwarded toward the sink node by relaying between groups of CNs (i.e., group-by-group, rather than hop-by-hop), as illustrated in Fig. 1(c). REER exploits data broadcasting to attain high reliability. More specifically, each data packet is broadcast at each hop, such that the RN and all the CNs with a good signal-noise-ratio (SNR) in the next CF will receive this data packet. RNs and CNs play the same role in data relaying. This strategy provides an effective tradeoff between traditional multipath routing and single path routing schemes. That is, it has the advantage of error resilience as in multipath (or mesh) routing schemes, but without the associated overhead of sending multiple copies of the same packet.

Fig. 1(d) shows all the possible wireless links between two consecutive cooperative groups, while the quality of each of the links is varying. With the proposed scheme, actually the link with the best quality is used. Such a strategy makes our scheme robust to link dynamics.

Upon reception, a node (RN or CN) will be selected randomly to broadcast the data packet toward the next cooperative field, and so forth. The data dissemination mechanism will be detailed in Section 3.4. The nodes, which are neither selected as RN nor CN, will enter the sleeping mode to save energy during data dissemination.

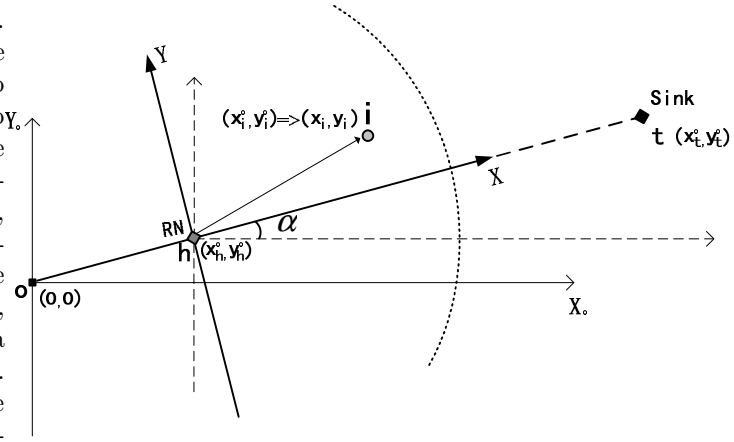

Figure 2: Obtaining Virtual Coordinates by Means of GPS.

3.2 Reference Node Selection Strategy

In the global coordinate system (o is the origin) of Fig. 2, node h is an RN. Its position (x_h^o, y_h^o) is piggybacked in the PROB message sent by h. Thus, a neighbor node i knows its position (x_i^o, y_i^o), the position of its upstream RN h, and the sink's location (x_t^o, y_t^o).

In this paper, we employ the *virtual coordinates* proposed in (16). The *virtual coordinates* of a node (e.g., i in Fig. 2) are defined as the coordinates in the virtual two-dimensional coordinate system where the node's upstream node (e.g., h in Fig. 2) is the origin, and the X-axis is the line between the upstream node (e.g., h in Fig. 2) and the sink. In the example shown in Fig. 2, the *virtual coordinates* of i is denoted by (x_i, y_i), which can be calculated by Eqn. (1).

$$\begin{cases} x_i = \cos(\alpha) \cdot (x_i^o - x_h^o) + \sin(\alpha) \cdot (y_i^o - y_h^o), \\ y_i = \cos(\alpha) \cdot (y_i^o - y_h^o) - \sin(\alpha) \cdot (x_i^o - x_h^o), \\ \alpha = \arctan(\frac{y_t^o - y_h^o}{x_t^o - x_h^o}). \end{cases} \quad (1)$$

The RN-selection is performed according to (x_i, y_i) and r. Let $A(d, r_1, r_2)$ denote the size of an area intersected by two circles with radius being r_1 and r_2, respectively, and the distance between their centers being d. Let D_i be the distance between the RN_i and the sink. Then, the area covers the CNs of RN_i is equal to $A(D_{i-1} - D_{i+1}, R, R)$. Assume nodes are densely and nearly uniformly distributed; then, the density of sensor nodes can be deemed as a constant ρ approximately. The number of CNs in the CF_i with center being RN_i is equal to:

$$N_i = A(D_{i-1} - D_{i+1}, R, R) \cdot \rho. \quad (2)$$

Let f be the failure probability of each link/node. Then, the hop reliability that data packet successfully passes CF_i can be given by:

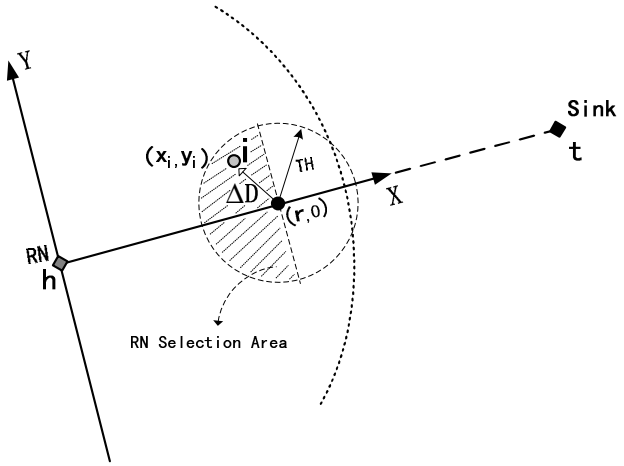

Figure 3: Illustration of RN-selection.

$$p = 1 - f^{N_i}. \quad (3)$$

Based on Eqn.(2) and Eqn.(3), p is a decreasing function of $(D_{i-1} - D_{i+1})$. If required hop reliability is an application-specific constant, $(D_{i-1} - D_{i+1})$ is fixed, i.e. the specified hop distance $r = r_i = D_i - D_{i+1}$ is a constant. In the following section, we describe the algorithm in such condition.

The point $(r, 0)$ is called strategic position in Fig. 3, which is r away from the upstream RN and located in the line between source and the sink to maximize hop length. Denote the distance between node i and the strategic position $(r, 0)$ by $\Delta D_i = \sqrt{(x_i - r)^2 - (y_i)^2}$. The smaller is ΔD_i, the higher possibility that i should be selected as the RN. Nodes within the shadow area (RN-selection-area) of Fig. 3 are deemed as RN-candidates (RNCs). A threshold TH is set to limit the RN-selection-area, and the x coordinate of a RNC should be smaller than r to obtain the required hop reliability approximately. Thus, RN-selection-area is a half circle with radius TH in Fig. 3.

Upon the reception of a PROB message from h, node i will discard the packet under any of the following conditions:

1. the node has already received this packet;
2. $x_i > r$;
3. $\Delta D_i > TH$.

If the packet is not discarded, i will start a backoff timer. In order to guarantee that the one closest to strategic position has highest possibility to be selected as the next RN, the timeout value for the backoff timer (t_{rnc}) is proportional to ΔD. t_{rnc} is calculated in Eqn.(4).

$$t_{rnc} = \tau \times \Delta D + \text{rand}(0, \mu), \quad (4)$$

where τ is the time value of a fixed unit slot. $\text{rand}(0, \mu)$ returns a random value uniformly distributed in $[0, \mu)$, and μ is a small constant.

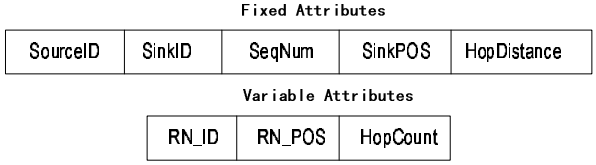

Figure 4: The Packet Structure of PROB Message.

Assume i has the smallest t_{rnc} value among all the RNCs and its backoff timer expires first, it will unicast a "reply" message (REP) to its upstream reference node h. When node h receives the REP, it broadcasts a "selection" message (SEL) with the identifier of node i (already piggybacked in the REP). To guarantee that only one RNC is selected as the downstream RN, node h only accepts the first REP while ignoring the later ones. If node i receives the SEL, it is selected as the downstream RN for h. When other RNCs receive the SEL or REP, they will cancel their backoff timers. When the sink receives PROB, it will broadcast a notification packet immediately to terminate RN-selection.

To reduce the possibility of collision of REP messages, we can set τ a sufficiently large value, while low value of τ decreases the time needed to setup RNs. The setting of τ is shown in Table 2. Since the RN selection is a relatively infrequent task as compared to the period of data transmission, even the use of large τ will not increase the data latency.

3.2.1 The Structure of Route Discovery Packet

The information contained in a PROB is shown in Fig. 4. The set of *SourceID*, *SinkID* and *SeqNum* is used to identify the PROB message. *SinkPOS* indicates the absolute coordinates of the sink. *HopDistance* indicates the expected per hop distance. The fixed attributes are set by the source and not changed while propagated across the network. On the other hand, when an RN broadcasts a PROB message, it will change variable attributes. *RN_ID* is the identifer of current reference node. *RN_POS* is the absolute coordinates of the RN. *HopCount* is the hop count from current node to the source. *RN_ID* and *HopCount* are used in Section 3.3.

3.2.2 The Dead End Problem During *RN*-selection

The so-called dead end problem (17; 18) arises when a packet is forwarded to a local optimum, i.e., a node with no neighbor of closer hop distance to the destination as illustrated in Fig. 5. In REER, if there are no RNCs located in the RNC-area, it will enter greedy mode to select the node among all its neighbors that is geographically closest to the sink as the downstream RN. If an RN does not have any neighbor closer to the sink in the greedy mode, REER meets the dead end problem and RN-selection will be performed in recovery mode, i.e., the downstream RN is selected according to the right-hand rule to recover from

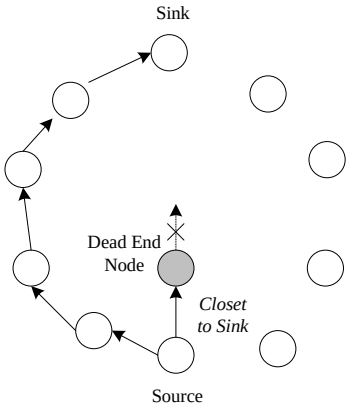

Figure 5: Illustration of Dead End Problem.

the local minimum (13). The right-hand rule is a well-known concept for traversing mazes. To avoid loops, the downstream *RN* is selected in recovery mode on the faces of a locally extracted planar subgraph, namely the Gabriel graph. The *RN*-selection returns to greedy mode when an *RN* is closer to the sink than the *RN* where RN-selection entered the recovery mode. Furthermore, if the *RN* has *RNC*(s) in its *RNC*-area, the *RN*-selection switches to normal selection mode described in Section 3.2 rather than greedy mode.

If an *RN* is selected by greedy mode or recovery mode, the corresponding cooperative field will be distorted seriously. In this case, the cooperative field is not constructed and data packet will be forwarded by unicasting, and the responsibility of reliability is shifted to MAC layer.

3.3 Cooperative Node Selection Strategy

As shown in Fig.1(a), PROBs are broadcast by the *RN*s along the path from the source to the sink, starting from the source node. Note that PROB is sent only during the cooperative field establishment phase and each *RN* will broadcast PROB only once.

Upon the reception of the first PROB, an intermediate node will become a *CN* candidate (*CNC*), and start a "CN-decision" timer (*CN-Decision-Timer*). Assume node i is one of such *CNC*s. As *RN* selection proceeds toward the sink, i will receive more PROBs. When its *CN-Decision-Timer* expires, i is expected to receives all the PROBs and performs a CN-decision procedure. In this procedure, i checks how many PROBs it has received. If the number of PROBs is three or more, node i induces that it becomes a *CN*. Then, it will figure out which *RN* it belongs to.

The detailed *CN*-Decision-Mechanism is shown in the flowchart in Figure 6 where the *RN-table* is used for a *CNC* to store information of received PROBs from different *RN*s. The *EntryIdx* is the index of the *RN*-entry (RE) in the *RN-table*. Each RE includes the following information: (1) the hop count to the source node (hc_s); (2) the identifier of the *RN* (id_{rn}) sending the PROB; (3) the distance from the *RN* to the sink (D_t), which is calculated

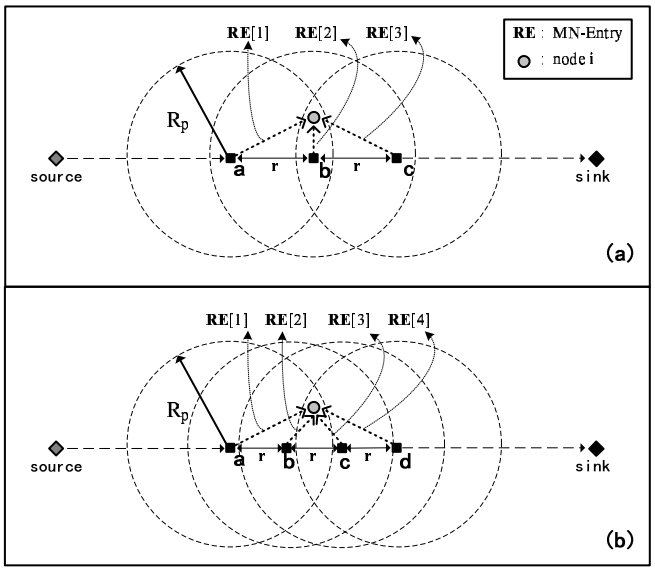

Figure 7: The Cases of Three-PROBs and Four-PROBs.

Source (s)	Sink (t)	PreviousHop (h)	HopCount (hc_s)	SeqNum

Figure 8: Data Packet Format.

based on *SinkPOS* and *RN_POS* in the PROB message.

The stored information is used for the *CN*-decision procedure and the following data dissemination (in Section 3.4). In the example of Fig.7(a), *CNC* i is closest to node b among all the *RN*s. It receives the first PROB from a and set the id_{rn} of the first RE ($RE[1].id_{rn}$) to a ; then it receives the second PROB from b and set $RE[2].id_{rn}$ to b; lastly, it receives the third PROB from c and set $RE[3].id_{rn}$ to c. In this example, node i knows it is a *CN* since its *EntryIdx* is equal to 3, and selects the *RN* indicated in the second RE (i.e. node b) as its *RN*. There also exists "four-PROBs" case in which a *CN* receives four PROBs. Fig. 7(b) shows such an example. However, there should be no five(or more)-PROBs cases, which means r is set to too small a value inefficiently.

In Four-PROBs case, only nodes $RE[2].id_{rn}$ and $RE[3].id_{rn}$ are eligible as the *RN* for the *CN*. The *CN* makes the decision by comparing which one is closer to itself as shown in Fig. 6 where MyD_t denotes the distance from the current *CN* to the sink.

Note that this section only considers the case of a single flow. If multiple flows coexist, REER creates an *RN-table* for each flow with a unique identifer (flow-id).

3.4 Data Dissemination in REER

When the *RN*s and *CN*s are determined, data reports are forwarded by the cooperation of the group of *CN*s at each hop. The data packet format is shown in Fig. 8, where s is the identifier of the source; t is the identifier of the sink; h is the identifier of the node broadcasting the packet;

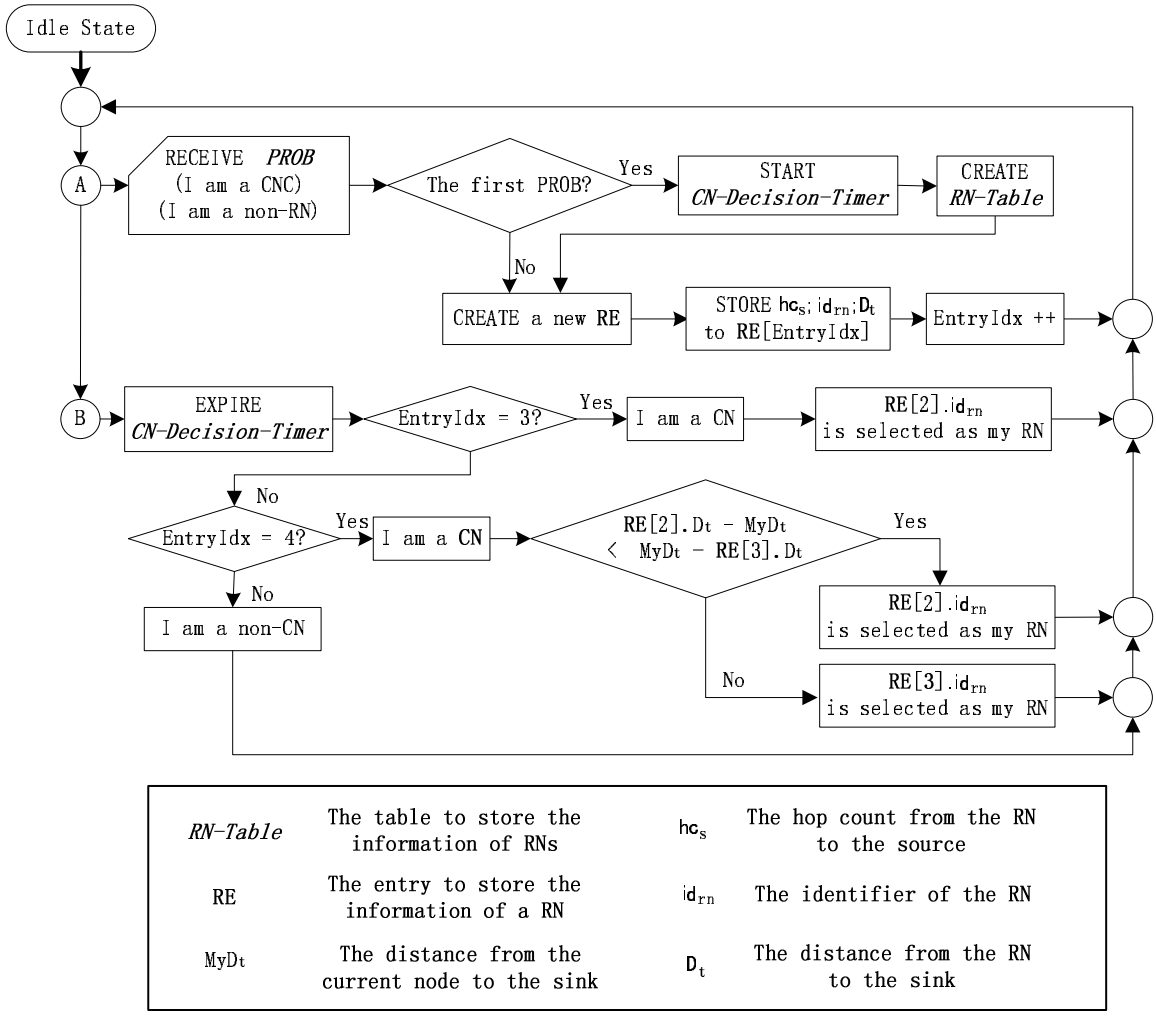

Figure 6: Flowchart of the *CN* Decision Mechanism.

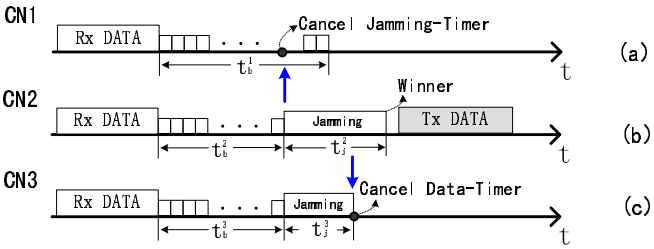

Figure 9: The Time Flow of Broadcasting Data Packet.

$Data.hc_s$ is the hop count from s; $Data.SeqNum$ is the sequence number of the data packet.

Assuming a node i receives a broadcast data packet. Let Seq^i_{data} be the largest sequence number of the data packets that node i has so far received. It first compares Seq^i_{data} with $Data.SeqNum$. If $Data.SeqNum$ is not larger than Seq^i_{data}, the data packet is either a stale one or broadcast by i's downstream node. In this case, node i will drop the data.

Then, node i will randomly choose a backoff time (t_b) in Eqn.(5), and set its *Backoff-Timer* to t_b to perform a two phase contention procedure (20).

$$t_b = \text{rand}(0, T_{max}) \tag{5}$$

In Eqn.(5), T_{max} denotes the maximum backoff timer value. Assume N_{cf} denotes the number of CNs in the cooperative field. In order to be differentiated with other nodes in the same cooperative field, at least the length of time slot ΔT should be reserved for each node to content the channel in the same cooperative field. Thus,

$$T_{max} = N_{cf} \cdot \Delta T \tag{6}$$

Large ΔT helps to reduce the possibility of simultaneous data broadcasting, while a small value of ΔT decreases the data latency. Once i's *Backoff-Timer* expires, it transmits a jamming signal for a short time t_j which is calculated in Eqn.(7), where β is a small constant.

$$t_j = \text{rand}(0, \beta T_{max}), \quad 0 < \beta \ll 1 \tag{7}$$

As an adverse example shown in Fig. 9, *CN2* and *CN3* happen to choose the same t_b to start jamming the medium simultaneously while the *Backoff-Timer* of $CN1$ does not

Table 1: Pseudo-code for Data Dissemination Algorithm

A. Handle DATA
procedure *process_data*(DATA(h,t,hc_s^h,*SeqNum*))
i is the identifier of the current node;
hc_s^h is the hop count from s to h;
SeqNum is the sequence number of the data packet;
begin
01 **if** ((f_{cn}^i=TRUE)||(f_{rn}^i=TRUE)) **then**
02 **if** (DATA.*SeqNum* > Seq_{data}^i)
03 && (DATA.hc_s^h+1=hc_s^i) **then**
04 **Store** DATA;
05 $t_b \leftarrow$ rand$(0, T_{max})$; //refer to Eqn.(5)
06 **Set** *Backoff-Timer* to t_b;
07 **else if** (DATA.*SeqNum* = Seq_{data}^i)
08 && (DATA.$hc_s^h = hc_s^i$) **then**
09 **Cancel** *Backoff-Timer* (if it's valid);
10 **Cancel** *ReTx-Timer* (if it's valid);
11 **else**
12 **Discard** DATA;
13 **endif**
14 **else**
15 **Discard** DATA;
16 **endif**
end

B. *Backoff-Timer* Expires
procedure *send_jamming*(void))
begin
01 $h \leftarrow i$;
02 *SeqNum* $\leftarrow Seq_{data}^i$;
03 $t_j \leftarrow rand(0, \beta T_{max})$; //refer to Eqn.(7)
04 **Broadcast** JAM(h,*SeqNum*) signal for t_j;
05 **Set** *Jamming-Timer* to (t_j);
end

C. Handle JAM
procedure *process_jam*(JAM(h,*SeqNum*))
begin
01 **Cancel** *Jamming-Timer* (if it's valid);
02 **Discard** the stored DATA;
end

D. *Jamming-Timer* Expires
procedure *broadcast_data*(void))
begin
01 $h \leftarrow i$;
02 $hc_s^h \leftarrow hc_s^i$;
03 **if** (I can reach sink in one hop) **then**
04 **Unicast** DATA(h,t) to t;
05 **else**
06 **Broadcast** DATA(h,t,hc_s^h,*SeqNum*);
07 **Set** *ReTx-Timer* to (T_{max});
08 *RetryIdx* $\leftarrow$ 1;
09 **endif**
end

E. *ReTx-Timer* Expires
procedure *rebroadcast_data*(void))
begin
01 *RetryIdx* + +;
02 **Broadcast** DATA again;
03 **if** (*RetryIdx* $\leq$ $RetryLimit_{data}$) **then**
04 **Start** *ReTx-Timer*;
05 **endif**
end

expire yet. $CN1$ listens a jamming signal either from *CN2* or *CN3*; Then, it cancels its *Backoff-Timer* to quit the contention. After *CN3* finishes jamming the medium, it detects the jamming signal from *CN2* and gives up the contention of forwarding the data. Finally, *CN2* wins the contention.

When node i hears the forwarding of a packet, it also compares hc_s^i (i's hop count to the source) with $Data.hc_s$ (the hop count of the received packet). If ($hc_s^i = Data.hc_s$)&& ($Seq_{data}^i = Data.SeqNum$), it deduces that the transmission is successful since an immediate downstream node broadcasts the data packet. Otherwise, node i will rebroadcast the data packet when a retransmission-timer (*ReTx-Timer*) expires, and starts the timer again until the retry limit reaches.

The pseudo-code of the data dissemination of REER protocol is shown in Table 1 where "←" denotes an assignment operation. f_{cn}^i is a flag that indicates whether a sensor node i is a cooperative node or not, while f_{rn}^i is a flag that indicates whether i is a reference node or not.

3.5 Performance Analysis

In this section, we present the analysis that derives the key performance metrics of REER, including the probability of successfully delivering data packet to the sink, P, the cumulative energy consumption involved in forwarding a data packet to the sink, E, and the cumulative delay for a data packet, T_{ete}. And show the impact of hop distance on these performance metrics.

To simplify analysis, we consider an ideal scenario where the hop distance r is identical between each adjacent RNs, and all the cooperative fields have the same shape, as shown in Fig. 10. We set up a two-dimensional coordinate system where the X-axis is the line between reference node b and the sink, and node b is at the origin of the coordinate system. The alphabet index of each node is equal to the one in Fig. 1.

Let R be the maximum transmission range of a PROB message. Let h_{cf} and v_{cf} be the horizontal and vertical radius of the cooperative field in Fig. 10, respectively. They are equal to:

$$h_{cf} = R - r \quad (8)$$

$$v_{cf} = \sqrt{R^2 - r^2}. \quad (9)$$

Let r_{max} be the possible maximum distance among all the CN pairs between two adjacent cooperative fields (e.g. CF_c and CF_d). Then,

$$r_{max} = \sqrt{r^2 + (2 \cdot v_{cf})^2} = \sqrt{4R^2 - 3r^2}. \quad (10)$$

To guarantee any pairs of CNs in adjacent cooperative fields can communication with each other, the maximum transmission range of a data packet R_{data} is set to r_{max}. In this case, R_{data} is also larger than $2v_{cf}$ which is the maximum distance between any two nodes in the same cooperative field. Thus, all CNs within the the same cooperative

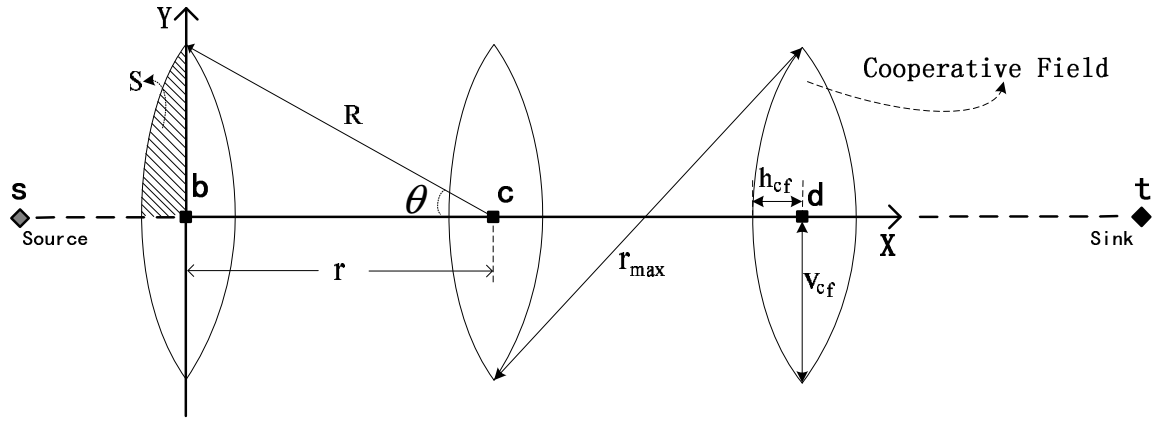

Figure 10: An Ideal Scenario in REER

field can hear each other, so that they can cancel their timers when one of them is forwarding the packet. This fact is used in Section 3.4, where jamming signal broadcast by any node in a cooperative field (CF) will make any other nodes in the same CF cancel broadcasting the same data.

Let S_{cf} be the size of the area of a cooperative field, and let S be the size of the shaded area in Fig.10. Then S_{cf} is equal to:

$$S_{cf} = 4 \cdot S = 4 \cdot (\theta \cdot R^2 - \frac{1}{2} \cdot \sqrt{R^2 - r^2}). \quad (11)$$

In Eqn.(11), $\theta = cos^{-1}(r/R)$. Assume the node density is δ. Then, the number of CNs in cooperative field (N_{cf}) is equal to:

$$N_{cf} = S_{cf} \cdot \delta \quad (12)$$

Let d be the distance between the source and the sink. Then, the hop counts between the source and the sink (H) is equal to:

$$H = \lceil \frac{d}{r} \rceil \quad (13)$$

The number of cooperative fields between the source and sink is equal to $H - 1$. Let f be the failure probability of each link/node. Let p denote the successful delivery probability of data packet at each hop. Then,

$$P = p^H = (1 - f^{N_{cf}})^H. \quad (14)$$

Let e_{tx} and e_{rx} be the energy consumption of transmitting and receiving a data packet, respectively. Then, the cumulative energy consumption E involved in successfully forwarding a data packet to the sink is

$$E = e_{tx} \cdot H + e_{rx} \cdot [3(H-2) \cdot N_{cf} \cdot (1-f) + 2N_{cf} \cdot (1-f) + 1] \quad (15)$$

Note that $H - 2$ numbers of CFs will listen to the data broadcasting three times and only the last CF listens to the data two times. One of CNs in the last CF will unicast the data to the sink.

Let t_{data} be the time to transmit a data packet; Let $\overline{t_b}$ be the average of backoff time before data forwarding. Then, the end-to-end latency for a data packet is equal to:

$$T_{ete} = t_{data} \cdot H + \overline{t_b} \cdot (H - 1) \quad (16)$$

Given all other parameters fixed, P, E, and T_{ete} are decreasing functions of r. The smaller is r, the larger will be N and H, the higher reliability p is achieved. However, for small r values, more energy E is consumed for each data packet, and T_{ete} also becomes larger. Thus, r provides a control knob to trade-off robustness and energy efficiency (and latency). r should be adaptively selected to achieve required reliability while meeting the application-specific QoS requirements (e.g. reliability, and end-to-end latency bound).

3.6 Control Overhead Compared with GPSR

Let n_s be the number of sensor nodes in the network. The number of neighbors k of a node is equal to:

$$k = \pi R^2 \rho. \quad (17)$$

Let e_{ctrl} be the energy consumption of transmitting a control message. Let o_g be the control overhead for setting up neighbor information table in GPSR. Let o_r be the control overhead for establishing RNs and CNs in REER. Then,

$$o_g = n_s \cdot e_{ctrl}. \quad (18)$$

$$o_r = H \cdot 3e_{ctrl}. \quad (19)$$

In GPSR, each node beacons a hello message for setting up or updating the neighbor information table; In REER, three messages (i.e. PROB, REP, and SEL) are needed to construct RN and CNs per hop. In general, n_s is much larger than $3H$. In GPSR, each node needs to store k number of neighbor entries in its local memory, while in REER, each node does not require neighbor information except for the identifier of the RN. Once cooperative fields are established, RN/CN does not need to store any routing-relevant

information, while other nodes can enter sleeping mode to save energy. Thus, REER scales well in dense sensor network, where the sensors have low storage capacity.

4 Simulation Model

4.1 Simulation Settings

We implemented our scheme using OPNET (21; 22) to evaluate the performance of REER and GPSR. The hop distance is specified. During the data dissemination, the nodes outside the cooperative fields will enter sleeping mode to save energy. In GPSR, a greedy forwarder will be selected out of the list of neighbors. If the selected neighbor fails to receive a packet, its previous hop node tries to retransmit the packet until the retry limit reaches. Then, a backup node is selected from the neighbor table, and the MAC layer tries to deliver the packet to the this node. We use IEEE 802.11 DCF as the underlying MAC. Six hundreds of sensor nodes are randomly placed over a 500m × 200m area. The rectangular shape of the simulation area is chosen to obtain longer paths, i.e. a higher average hop count. The transmission range of sensor node is 60m. As we take a conservative approach in evaluation, we do not assume sensor node can adjust transmission range in REER, i.e. $R_{data} = R$. The sensor nodes are battery-operated. The sink is assumed to have infinite energy supply. We assume both the sink and sensor nodes are stationary. The sink located close to one corner of the area, while the target sensor nodes are specified at the other corner. Each source generates sensed data packets using a constant bit rate with a 5 second interval.

We use the energy model in (23; 24; 25). In (27; 28; 29), Gilbert-Elliot model is used to model the link failure. We adopt an ON-OFF two state Gilbert-Elliot model. State ON represents that the link is in "good" status, while state OFF represents a "link failure" state. Let f be the link failure rate. With the time duration of state ON (T_{on}) fixed to 100s, that of state OFF (T_{off}) is calculated as a function of f ($T_{off} = T_{on} \times f/(1-f)$). The parameter values used in the simulations are presented in Table 2. The basic settings are common to all the experiments. To decrease the influence of one special topology on the results, each experiment was repeated 10 times with different topologies; For each result, we simulate for 20 times with different random seeds. For the evaluation, the mean values of these 10×20 runs were taken.

Table 2: Simulation Setting

Basic Specification	
Network Size	500m × 200m
Topology Configuration Mode	Randomized
Total Sensor Node Number	600
Data Rate at MAC layer	1Mbps
Transmission Range of Sensor Node	60m
Time Duration of State ON	Default: 10s
Node failure rate	Default: 0%
Packet loss rate	Default: 0%
Sensed Traffic Specification	
Size of Sensed Data	Default: 1Kbytes
Size of Control Message	Default: 128bytes
Sensed Data Packet Interval	5s
REER Specification	
r	Default: 40m
τ in Eqn.(4)	Default: 2.5ms
μ in Eqn.(4)	Default: 5ms
ΔT in Eqn.(6)	Default: 10ms

4.2 Performance Metrics

In this section, five performance metrics are evaluated:

- *Reliability (Packet delivery ratio)* - It is denoted by P. It is the ratio of the number of data packets delivered to the sink to the number of packets generated by the source nodes.
- *Energy Consumption per Successful Data Delivery* - It is denoted by e. It is the ratio of network energy consumption to the number of data packets successfully delivered to the sink. The network energy consumption includes all the energy consumption by transmitting and receiving during simulation. As in (26), we do not account energy consumption for idle state, since this part is approximately the same for all the schemes simulated. Let E be the all the energy consumption by transmitting, receiving, and overhearing during simulation. Let n_{data} be the number of data packets delivered to the sink. Then, e is equal to:

$$e = \frac{E}{n_{data}} \tag{20}$$

- *Average End-to-end Packet Delay* - It is denoted by T_{ete}. It includes all possible delays during data dissemination, caused by queuing, retransmission due to collision at the MAC, and transmission time.
- *Number of the Control Messages per Successful Data Delivery* - It is denoted by n_{ctrl}. It is the ratio of the number of control messages transmitted to the number of data packets delivered to the sink before lifetime.
- *Energy*delay/Reliability* - In sensor networks, it is important to consider both energy and delay. In (30), the combined energy*delay metric can reflect both the energy usage and the end-to-end delay. Furthermore, in unreliable environment, the reliability is also an important metric. In this paper, we adopt the following

metric to evaluate the integrated performance of reliability, energy and delay:

$$\eta = \frac{e \cdot T_{ete}}{P}. \quad (21)$$

5 Performance Evaluation

In Section 5.1, we examine the impact of node density on the REER performance. In Section 5.2, GPSR and REER with varying r are evaluated in terms of link failure rate.

5.1 Effect of Normalized Node Density on the REER performance in Unreliable Environments

In the following experiments, link failure rate is set to 0.3; r is set to $0.8R$; Let δ_q be the normalized node density, i.e. the ratio of the current node density to the default one ($\frac{600\ nodes}{500\times200\ m^2}$). δ_q is changed from 0.25 to 2 by controlling the number of sensor nodes in the fixed size of network.

In Fig. 11, the higher is δ_q, the larger is N_{cf}, the higher is the hop reliability and P. When δ_q is beyond 1.5, REER has a delivery ratio near 100%.

According to Eqn.(14) and Eqn.(15), P is exponentially increasing function of N_{cf}, while E is linearly increasing function of N_{cf}. When N_{cf} is too small to overcome the 30% link failure rate, P increases exponentially with N_{cf} increased. Thus, $n_{data} = P \cdot TotalDataSendNum$ dominates Eqn.(20) to make e_{reer} decrease. When δ_q is equal to 0.75, e_{reer} reaches its minimum. If δ_q goes beyond 0.75, P does not increase much (see Fig. 11). However, E always linearly increases in proportion to δ_q, and dominates Eqn. 20. Thus, e_{reer} increases again.

Recall that T_{max} denotes the maximum backoff timer value during data dissemination. T_{max} has a large impact on the data latency. It is set according to N_{cf} in Eqn.(6). With δ_q increased, N_{cf} increases. The larger is N_{cf}, the larger T_{max} will be set to avoid collisions. Thus, in Fig. 13, T_{ete} of REER increases with δ_q increased. Currently, we adopt a simple backoff time function as shown in Eqn.(6), we believe a better function can lower the data latency extensively.

In Fig. 14, η reaches its minimum value when δ_q is equal to 0.75. The smaller is η, the better is the integrated performance of REER. It is unnecessary to increase δ_q more if the value is large enough to achieve required reliability.

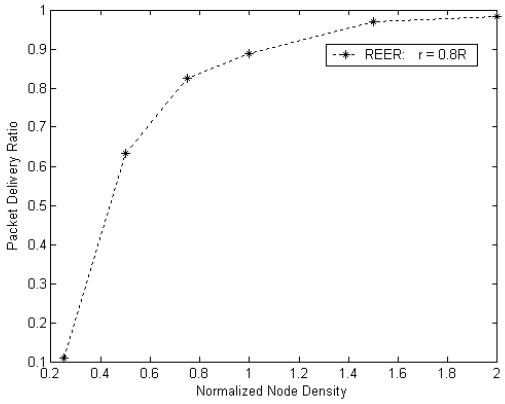

Figure 11: The impact of δ_q on P.

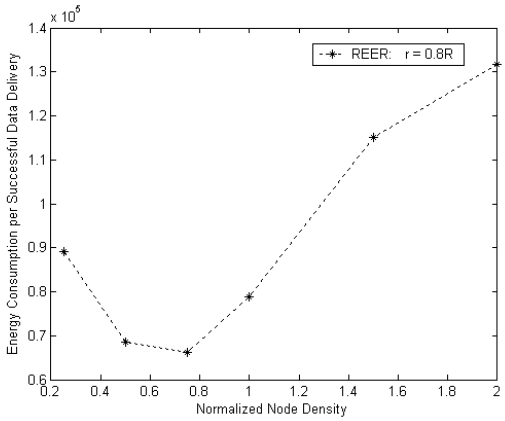

Figure 12: The impact of δ_q on e.

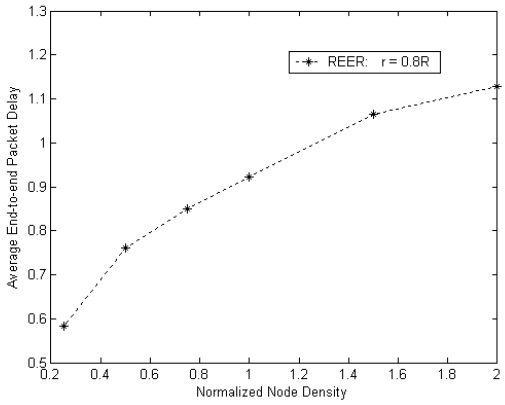

Figure 13: The impact of δ_q on T_{reer}.

5.2 Comparison of REER and GPSR with Variable Link Failure Rates

In this section, six groups (i.e. GPSR and REER with r set to $0.67R$, $0.75R$, $0.85R$, $0.93R$, and R respectively) of simulation are evaluated. In each group of experiments, we change f from 0 to 0.9 by the step size of 0.1 with all the other parameters in Table 2 fixed.

The smaller is r, the larger number of CNs in each cooperative field are exploited. Thus, in Fig. 15, REER yields higher reliability as r decreased. When r is equal to $0.67R$, REER keeps achieving more than 90% packet delivery ratio until f is larger than 0.6. Since GPSR depends on periodically beaconing to perform local repair, it is not robust to high link failure rate. Thus, the reliability is low if the link failure rate goes beyond 0.3.

GPSR selects a next hop in its neighbor table and the

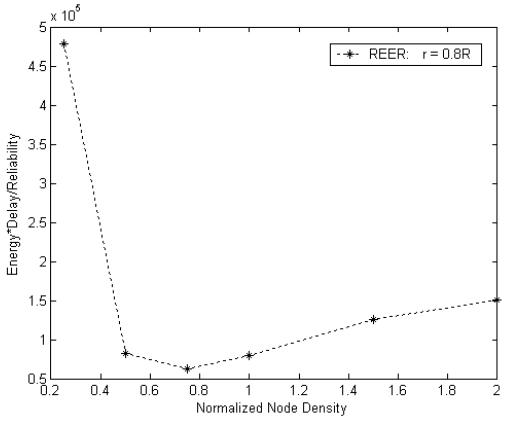

Figure 14: The impact of δ_q on η.

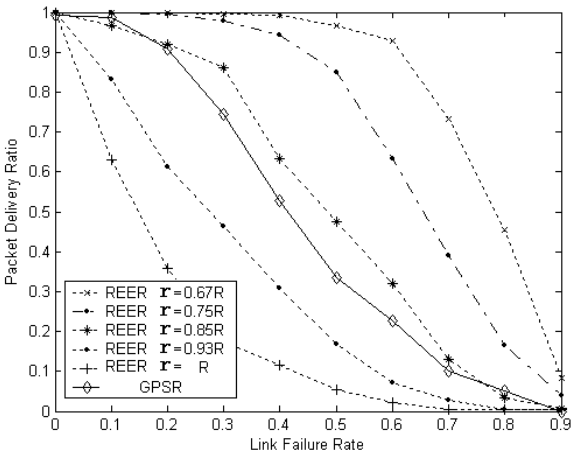

Figure 15: The Comparison of P.

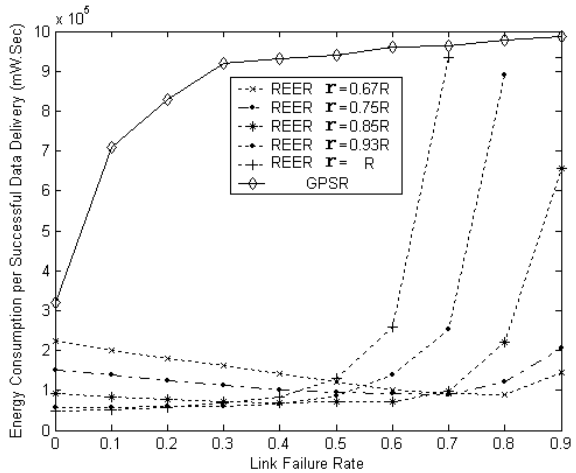

Figure 16: The Comparison of e.

MAC-layer tries to deliver the packet to this node. However, this node is not reachable in case of link failure, and the MAC-layer sends a failure notification back to the network layer to make the routing protocol selects another

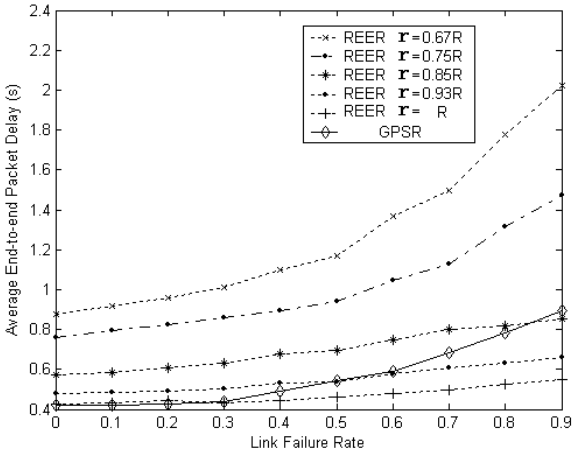

Figure 17: The Comparison of T_{ete}.

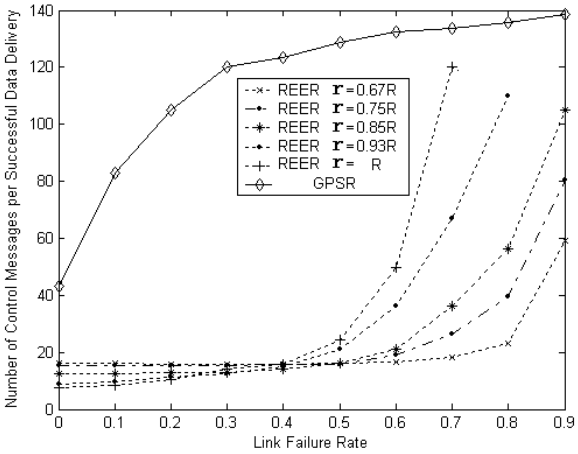

Figure 18: The Comparison of n_{ctrl}.

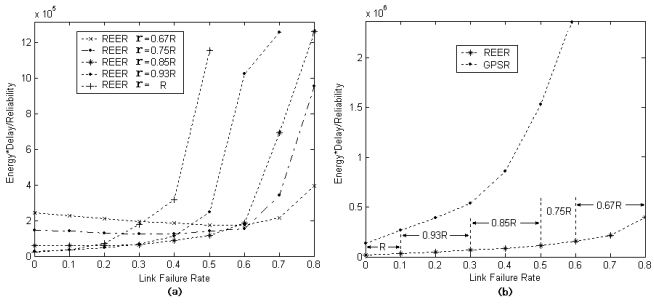

Figure 19: The Comparison of η.

next hop. In case of high link failure rate, GPSR had to select several times a next hop until finally the MAC-layer was able to deliver the packets. By comparison, REER broadcast a data packet only once at each hop. Furthermore, the nodes which are not selected as RNs/CNs can enter sleeping mode to save energy. Thus, in Fig. 16, e_{reer} is almost always lower than e_{gpsr} with varying f.

According to Eqn.(15), E decreases with f increases,

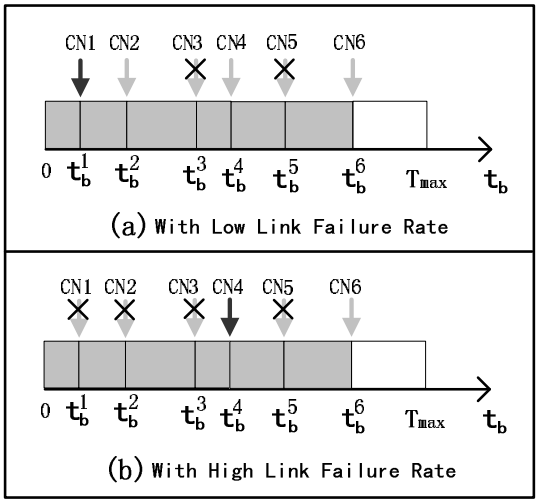

Figure 20: The Comparison of Backoff Time: (a) with Low Link Failure Rate; (b) with High Link Failure Rate.

i.e. the link failure helps to save energy for receiving data packet. If the number of *CN*s is large enough to overcome the link failure, a large f helps to lower e_{reer}. The reason is n_{data} does not change much, while E decreases. Thus, in Fig. 16, given r fixed, there is a certain value of f to make e reach its minimum. If f goes beyond that point, the number of *CN*s is insufficient to antagonize the high link failure rate, which causes n_{data} decrease exponentially. Thus, e_{reer} increases fast again.

In Fig. 17, the delay of GPSR increases with higher f. The responsibility for this effect lies again in the increasing number of link layer retransmissions. Given r fixed, the delay of REER also increases with higher f. It is because REER performs a backoff process at each hop during data dissemination. In Fig. 20, the number of *CN*s is six. When f is low, the *CN* with low t_b is more likely to forward the data packet, which makes hop latency low. As an example in Fig. 20(a), *CN1* is selected to forward the data packet. In contrast, *CN4* is selected in Fig. 20(b), where the hop latency is equal to $t_{data} + t_b^4 > t_{data} + t_b^1$.

On the other hand, given f fixed, the delay of REER is inversely proportional to r, as shown in Fig. 17. It is because that the smaller is r, the higher is the number of *CN*s in a *CF*, the higher T_{max} are needed to differentiate the *CN*s according to Eqn.(6), the longer backoff time is yielded, and the higher is the delay of REER. Another reason is that the hop count between source and sink increases as r decreases.

In Fig. 18, n_{ctrl} of REER is lower than that of GPSR, since REER never uses control message beaconing to repair a route.

Observed in Fig. 16, Fig. 17, and Fig. 18, REER exhibits more consistent and relatively higher reliability, lower energy-consumption than GPSR by compromising end-to-end delay bound. These figures also give hints that REER should choose r adaptively for different f. To find optimal r in terms of η, Fig. 19(a) is plotted. Then, in Fig. 19(b), the optimal r for variable f are selected. The overall performance gain of REER further improves with the strategy of adaptive r selection.

6 Conclusion

This paper proposes REER to achieve both reliability and energy-efficiency simultaneously. In REER, we first select reference nodes (*RN*s) between source and sink. Then, multiple cooperative nodes (*CN*s) are selected for each reference node. The smaller is the distance (r) between two adjacent *RN*s , the larger number of *CN*s will be selected for each flow. r provides a control knob to trade off robustness, energy-efficiency and data delay. In unreliable communication environments, traditional routing protocols may fail to deliver data timely since link/node failures can be found out only after trying multiple transmissions. In REER, each data is relayed by broadcasting at each hop, such that among all the *CN*s at next hop that received the data successfully, only one *CN* will rebroadcast the data.

We have evaluated the REER protocol through both analysis and extensive simulation. According to the simulation results, we observe the following: 1) With the link failure rate increased, r should be set small enough to achieve required reliability but not so small as to incur unnecessary large energy consumption and end-to-end packet delay; 2) REER is unsuitable to perform in low node density environments; 3) in a reliable environment, both GPSR and REER with large r exhibit higher reliability; 4) REER exhibits more consistent and relatively higher reliability, less energy consumption than GPSR in unreliable environments. The extensive simulations also show reliability is achieved by sacrificing the energy-efficiency and delay performance. Thus, the relevant parameters should be selected carefully to achieve reliability with energy-efficiency while minimizing the delay.

A better backoff time function used in data dissemination should help to lower the data latency while not increasing the possibility of simultaneous data broadcasting. To find such a function will be one part of our future work.

ACKNOWLEDGMENT

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05, and was supported by grant No. (R01-2004-000- 10372-0) from the Basic Research Program of the Korea Science and Engineering Foundation.

REFERENCES

[1] Al-Karaki, J.N., Kamal, A.E., "Routing techniques in wireless sensor networks: a survey," *IEEE Personal Communications*, Vol.11 , NO. 6 , pp.6-28, Dec. 2004

[2] K. Akkaya and M. Younis, "A Survey of Routing Protocols in Wireless Sensor Networks," *in the Elsevier Ad Hoc Network Journal*, Vol. 3, NO. 3 pp. 325-349, 2005.

[3] C. Y. Wan and A. T. Campbell, and L. Krishnamurthy, "Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks," *IEEE Journal of Selected Areas in Communications*, vol.23, pp.862-872, April. 2005.

[4] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," *in Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp.102-112, May 2003.

[5] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks," In *Mobile Computing and Communications Review (MC2R)*, vol 1., no. 2, pp. 10-24, 2002.

[6] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: reliable information forwarding using multiple paths in sensor networks," *IEEE LCN*, pp.406-415, Oct.2003.

[7] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," accepted by *ACM WINET (Wireless Networks)*.

[8] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to- Sink Reliable Transport in Wireless Sensor Networks," in *ACM MobiHoc*, pp.177-188, June 2003.

[9] N. Tezcan, E. Cayirci and M. U. Caglayan, "End-to-end reliable event transfer in wireless sensor networks,", in *IEEE PIMRC'04*, vol.2, pp.989-994, Sept. 2004.

[10] Yong Yuan, Zhihai He, Min Chen, "Virtual MIMO based Cross-layer design for Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, No.3, Vol 53, May 2006.

[11] I. Stojmenovic, "Position-Based Routing in Ad Hoc Networks," *IEEE Comm. Magazine*, Vol.40, No.7, pp.128-134, July 2002.

[12] Y.-B. Ko, N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," in *Proc. of ACM MobiCom 1998*, pp. 66C75, Dallas, Texas, 1998.

[13] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom 2000*, pp.243-254, Boston, Mass., USA, August 2000.

[14] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023*, May 2001.

[15] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, " A scalable location service for geographic ad hoc routing,". in *Mobicom 2000*, pp.120-130, Boston, MA, USA, 2000.

[16] M. Chen, X. Wang, V. Leung, and Y. Yuan, "Virtual Coordinates Based Routing in Wireless Sensor Networks," *Sensor Letters*, Vol.4, pp.325-330, 2006

[17] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead end problem of location-based routing in sensor networks," *IEEE Trans. Vehicular Technology*, vol. 54, pp. 1509-1522, July 2005.

[18] Fang, Q., Gao, J., und Guibas, L. J., "Locating and bypassing routing holes in sensor networks,". *IEEE Infocom*, March 2004.

[19] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in the *Proceedings of the 6th Annual ACM/IEEE MobiCom*, Boston, MA, August 2000.

[20] Zakhia G. Abichar and J. Morris Chang, "CONTI: Constant-Time Contention Resolution for WLAN Access," *LNCS*, vol. 3462, pp. 358-369, 2005.

[21] http://www.opnet.com

[22] M. Chen, *OPNET Network Simulation*. Press of Tsinghua University, China, April, 2004, ISBN 7-302-08232-4, 352 pages.

[23] L.M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. of IEEE INFOCOM'01*, pp.1548-1557, April 2001.

[24] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and Victor Leung , "Mobile Agent-Based Directed Diffusion in Wireless Sensor Networks," *EURASIP Journal on Advances in Signal Processing*, Special Issue On Visiual Sensor Network, to appear.

[25] M. Chen, T. Kwon, and Y. Choi, "Energy-efficient Differentiated Directed Diffusion (EDDD) for Real-Time Traffic in Wireless Sensor Networks," *Elsevier Computer Communications*, Special Issue On Dependable Sensor Network, Vol. 29, No.2 , pp. 231-245, Jan. 2006.

[26] Jie Gao, Li Zhang, "Load balancing shortest path routing in wireless networks," *IEEE INFOCOM 2004*, Vol.2, pp.1098-1107, March 2004

[27] H. S. Wang and N. Moayeri, "Finite-state markov channel - a useful model for radio communication channels," *IEEE Transaction on Vehicular Technology*, vol. 43, no. 1, pp. 163-171, Feb. 1995.

[28] E.N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp.1253-1265, 1960.

[29] E.O. Elliott, "A model of the switched telephone network for data communications," *Bell Syst. Tech. J.*, vol. 44, pp.89-109, 1965.

[30] S. Lindsey, C. Raghavendra, and K.M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Trans.Parallel Distrib. Systems*, vol.13, no.9, pp.924-935, 2002.