

## Chapter 7

### Cooperative Geographical Routing in Wireless Sensor Networks

Min Chen<sup>\*,‡</sup>, Victor C.M. Leung<sup>\*,§</sup> and Shiwen Mao<sup>†,¶</sup>

*\*Department of Electrical and Computer Engineering  
University of British Columbia, V6T 1Z4, Canada*

*†Department of Electrical and Computer Engineering  
Auburn University, Auburn, AL 36849, USA*

*‡minchen@ece.ubc.ca*

*§vleung@ece.ubc.ca*

*¶smao@ieee.org*

Reliable delivery of sensory data to a sink node in large scale sensor networks is a challenging problem. This chapter tackles this problem by assuming dense deployment of sensors, which allows us to exploit diversity in choosing intermediate nodes for reliability and energy-efficiency. The proposed reliable and energy-efficient routing (REER) protocol is based on the geographic routing approach. The central idea of REER is the notion of reference nodes (RNs), which means the nodes closest to the ideal locations between the source and to the sink. The multiple Cooperative Nodes (CNs) around RNs will contend to relay data packets; thus, there is no overhead of route discovery and REER is resilient to node failures and transmission errors. By adjusting the distances between RNs, we can control the trade-off between reliability and energy-efficiency, which is validated by both analysis and simulation.

#### 7.1. Introduction

Sensor networks are usually 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. However, the goal of reliability and energy-efficiency often 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),<sup>1</sup> 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 chapter, 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. The goal of REER is to achieve 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 common nodes are shared by two adjacent hops, thus, the more cooperative nodes (CNs) 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 can be minimized while achieving required reliability. Furthermore, the unselected nodes will enter sleeping mode to save energy. We present extensive simulations to show that REER normally yields higher reliability than traditional geographical routing scheme while achieving less energy consumption. In REER, the overall performance gain in terms of reliability, lifetime, and data delivery latency increases as the link/node failure rate increases.

The remainder of this chapter is organized as follows. Section 7.2 presents the background of reliable data transmission using geographical routing over WSNs. Sections 7.3 describe the proposed REER design issues and algorithm. Sections 7.4 and 7.5 present simulation model and experiment results, respectively. Section 7.6 summarizes the chapter.

## 7.2. Related Work

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

### 7.2.1. *Reliable data transmission over WSNs*

There are increasing research efforts on studying the issue of reliable data transfer in WSN.<sup>2-8</sup> In these work, hop-by-hop recovery,<sup>2,3</sup> end-to-end recovery,<sup>7,8</sup> and

multi-path forwarding<sup>4-6</sup> are the major approaches to achieve the desired reliability by previous work.

PSFQ<sup>2</sup> 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. RMST<sup>3</sup> 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 Ref. 8. A virtual MIMO based cross layer design is proposed in Ref. 9. 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,<sup>4</sup> multiple disjoint paths are set up first, then multiple data copies are delivered using these paths. In,<sup>5</sup> 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<sup>6</sup> uses a path interleaving technique to achieve high reliability. It assigns the amount of credit  $\alpha$  to the packet at the source.  $\alpha$  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  $\alpha$  for various reliability requirements of sensor networks is not trivial. Furthermore, when the quality of channel changes frequently, out-of-date  $\alpha$  makes GRAB either waste energy to unnecessarily use more paths or fail to achieve the required reliability. It is worth noting that although GRAB<sup>6</sup> 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<sup>2,3</sup> 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<sup>7</sup> 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.

### 7.2.2. Geographical routing in WSNs

Geographical (position-based) routing<sup>10</sup> is a routing scheme in which each sensor node is assumed to be aware of its geographical location using GPS or distributed location services, and packet forwarding is performed based on the locations of the nodes. Each node broadcasts a hello message periodically to notify its neighbors of its current position; based on this information, each node sets up a neighbor information table that records the positions of its one-hop neighbors.

In general, each packet is routed to a neighbor closer to the sink than the forwarding node itself until the packet reaches the sink. If a node does not have any neighbors closer to the sink, a fallback mechanism is triggered to overcome this local minimum. Bose *et al.* proposed Greedy-Face-Greedy (GFG) routing,<sup>11</sup> in which upon arriving at a void, the protocol switches from greedy mode to face mode to circumnavigate the void. When the current node is closer to destination than the node initially starting the face mode,<sup>12</sup> the protocol returns to greedy mode (the void is considered circumnavigated), and chooses the next hop using the left/right hand rule. The right hand rule consists in “rolling” to the right along the edges. GFG has been reinvented in Refs. 12 and 13, respectively. As the reinvention, a new name GPSR was given in Ref. 12.

## 7.3. System Architecture and Protocol Design

This section presents 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.

### 7.3.1. Overview

Consider a large scale, dense wireless sensor network, within which a source node, say, node  $s$ , generates reports on detected events in Fig. 7.1. 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 Figs. 7.1(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,

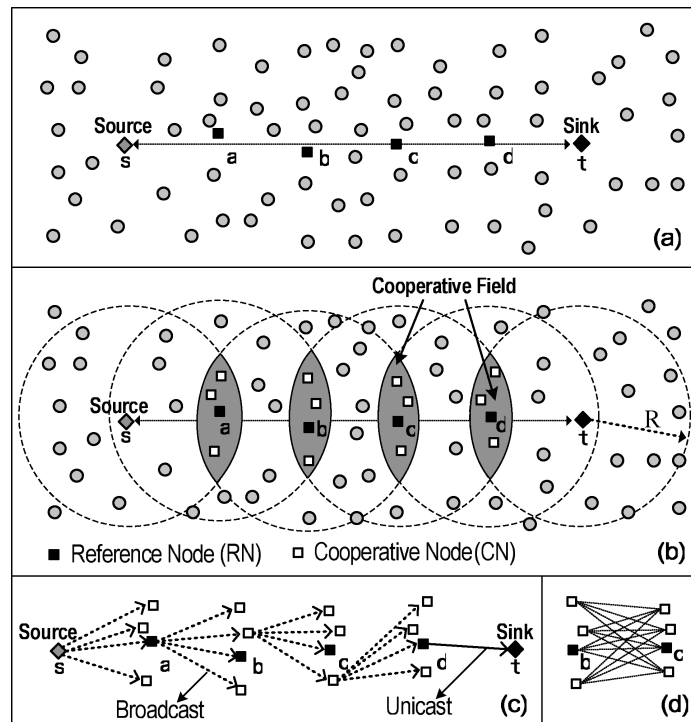


Fig. 7.1. Illustration of the REER routing protocol: (a) *RNs* along the shortest path; (b) *CNs* in the cooperative fields; (c) cooperative data forwarding; (d) the forwarding mesh between two cooperative fields.

which receive this PROB and within the *RN*-selection area in Fig. 7.3, 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. 7.1, 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 Sec. 7.3.2.

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. 7.1(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 Sec. 7.3.3.

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. 7.1(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.

Figure 7.1(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 Sec. 7.3.4. The nodes, which are neither selected as  $RN$  nor  $CN$ , will enter the sleeping mode to save energy during data dissemination.

### 7.3.2. Reference node selection strategy

In the global coordinate system ( $o$  is the origin) of Fig. 7.2, 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  $(x_t^o, y_t^o)$ .

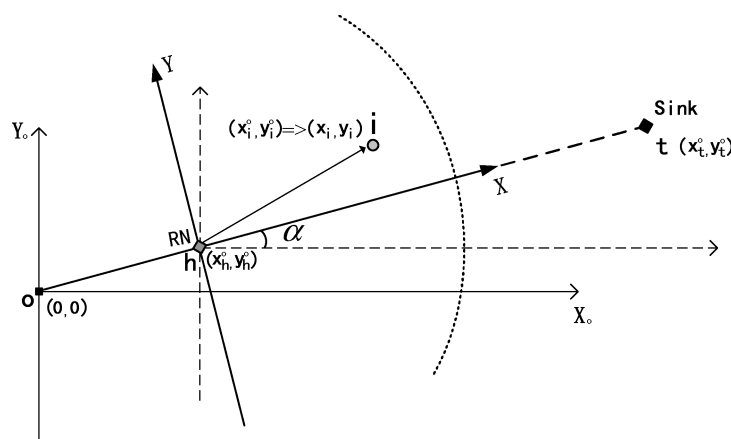


Fig. 7.2. Obtaining virtual coordinates by means of GPS.

In this chapter, we employ the *virtual coordinates* proposed in Ref. 14. The *virtual coordinates* of a node (e.g.,  $i$  in Fig. 7.2) are defined as the coordinates in the virtual two-dimensional coordinate system where the node's upstream node (e.g.,  $h$  in Fig. 7.2) is the origin, and the  $X$ -axis is the line between the upstream node (e.g.,  $h$  in Fig. 7.2) and the sink. In the example shown in Fig. 7.2, the *virtual coordinates* of  $i$  is denoted by  $(x_i, y_i)$ , which can be calculated by Eq. (7.1).

$$\begin{cases} x_i = \cos(\alpha) \cdot (x_i^o - x_h^o) + \sin(\alpha) \cdot (y_i^o - y_h^o) \\ y_i = \cos(\alpha) \cdot (y_i^o - y_h^o) - \sin(\alpha) \cdot (x_i^o - x_h^o) \\ \alpha = \arctan\left(\frac{y_i^o - y_h^o}{x_i^o - x_h^o}\right). \end{cases} \quad (7.1)$$

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  $\rho$  approximately. The number of  $CNs$  in the  $CF_i$  with center being  $RN_i$  is equal to:

$$N_i = A(D_{i-1} - D_{i+1}, R, R) \cdot \rho. \quad (7.2)$$

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:

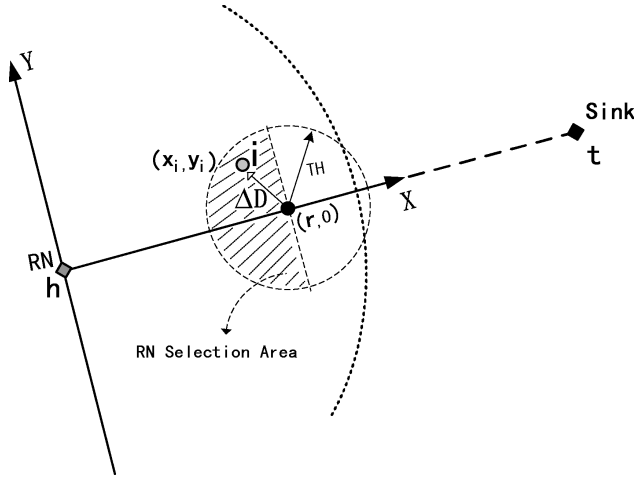
$$p = 1 - f^{N_i}. \quad (7.3)$$

Based on Eqs. (7.2) and (7.3),  $p$  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. 7.3, 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  $\Delta D_i$ , the higher possibility that  $i$  should be selected as the  $RN$ . Nodes within the shadow area ( $RN$ -selection-area) of Fig. 7.3 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. 7.3.

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

- C1 the node has already received this packet;
- C2  $x_i > r$ ;
- C3  $\Delta D_i > TH$ .

Fig. 7.3. Illustration of *RN*-selection.

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  $\Delta D$ .  $t_{rnc}$  is calculated in Eq. (7.4).

$$t_{rnc} = \tau \times \Delta D + \text{rand}(0, \mu), \quad (7.4)$$

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

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  $\tau$  a sufficiently large value, while low value of  $\tau$  decreases the time needed to setup *RNs*. The setting of  $\tau$  is shown in Table 7.2. Since the *RN* selection is a relatively infrequent task as compared to the period of data transmission, even the use of large  $\tau$  will not increase the data latency.

#### 7.3.2.1. The structure of route discovery packet

The information contained in a PROB is shown in Fig. 7.4. The set of *SourceID*, *SinkID* and *SeqNum* is used to identify the PROB message. *SinkPOS* indicates

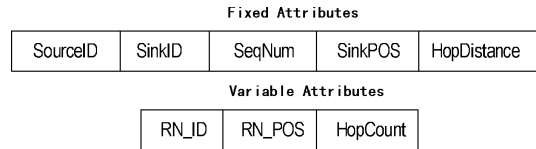


Fig. 7.4. The packet structure of PROB message.

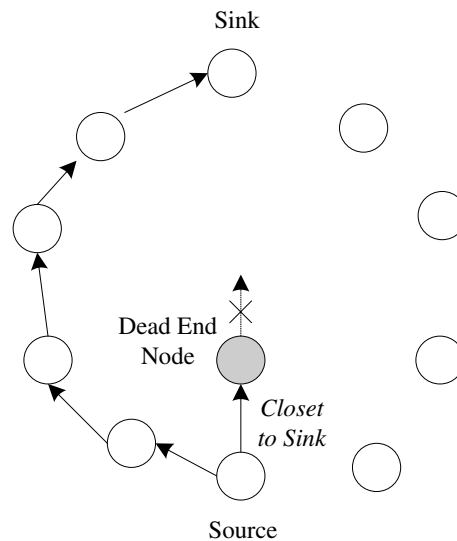


Fig. 7.5. Illustration of dead end problem.

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 identifier 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 Sec. 7.3.3.

### 7.3.2.2. The dead end problem during RN-selection

The so-called dead end problem<sup>15,16</sup> 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. 7.5. In REER, if there are no *RNC*'s 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 the local minimum.<sup>12</sup> 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 Sec. 7.3.2 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.

### 7.3.3. Cooperative node selection strategy

As shown in Fig. 7.1(a), PROBs are broadcast by the  $RNs$  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  $CNCs$ . As  $RN$  selection proceeds toward the sink,  $i$  will receive more PROBs. When its  $CN$ -Decision-Timer expires,  $i$  is expected to receive 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 Fig. 7.6 where the  $RN$ -table is used for a  $CNC$  to store information of received PROBs from different  $RNs$ . 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 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 Sec. 7.3.4). In the example of Fig. 7.7(a),  $CNC$   $i$  is closest to node  $b$  among all the  $RNs$ . 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. 7.7(b) shows such an

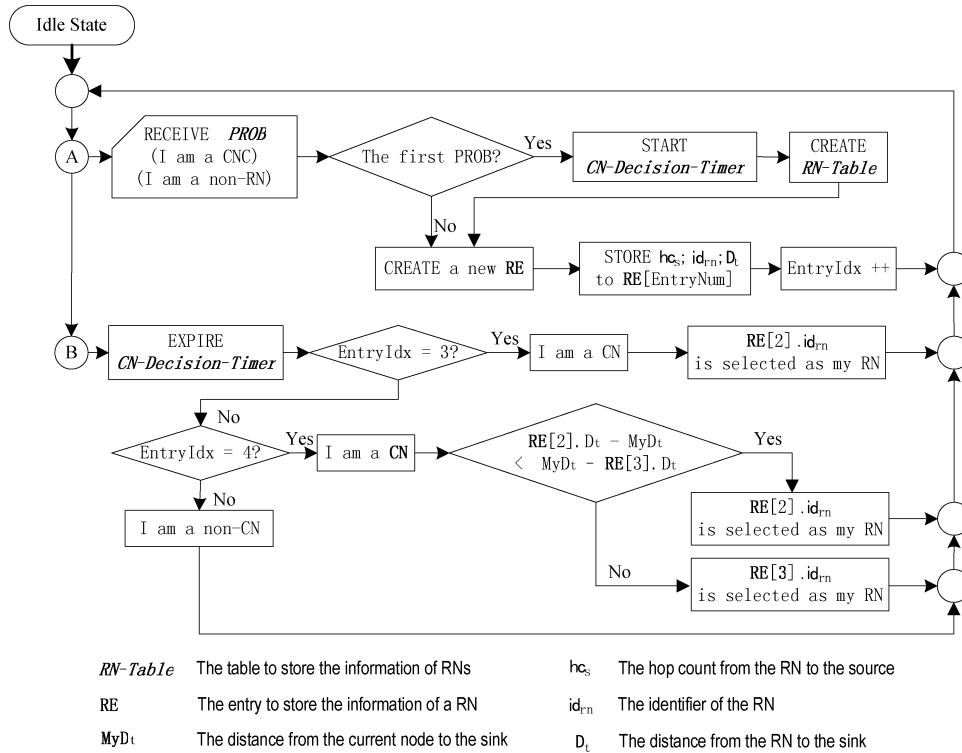


Fig. 7.6. Flowchart of the *CN* decision mechanism.

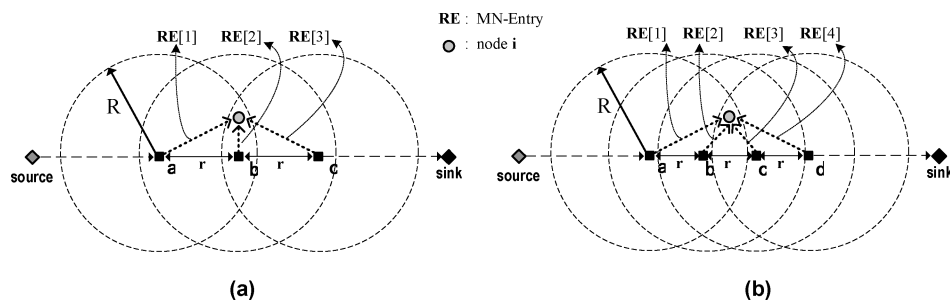


Fig. 7.7. The cases of three-PROBs and four-PROBs.

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. 7.6 where  $MyDt$  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 identifier (flow-id).

#### 7.3.4. Data dissemination in REER

When the *RNs* and *CNs* are determined, data reports are forwarded by the cooperation of the group of *CNs* at each hop. The data packet format contains the following information: the identifier of the source  $s$ ; the identifier of the sink  $t$ ; the identifier of the node broadcasting the packet  $h$ ; the hop count from source to current node  $Data.hc_s$ ; the sequence number  $Data.SeqNum$ .

Assuming a node  $i$  receives a broadcast data packet. Let  $Seq_{data}^i$  be the largest sequence number of the data packets that node  $i$  has so far received. It first compares  $Seq_{data}^i$  with  $Data.SeqNum$ . If  $Data.SeqNum$  is not larger than  $Seq_{data}^i$ , 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 Eq. (7.5), and set its *Backoff-Timer* to  $t_b$  to perform a two phase contention procedure.<sup>17</sup>

$$t_b = \text{rand}(0, T_{max}). \quad (7.5)$$

In Eq. (7.5),  $T_{max}$  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  $\Delta T$  should be reserved for each node to content the channel in the same cooperative field. Thus,

$$T_{max} = N_{cf} \cdot \Delta T. \quad (7.6)$$

Large  $\Delta T$  helps to reduce the possibility of simultaneous data broadcasting, while a small value of  $\Delta T$  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 Eq. (7.7), where  $\beta$  is a small constant.

$$t_j = \text{rand}(0, \beta T_{max}), \quad 0 < \beta \ll 1. \quad (7.7)$$

As an adverse example shown in Fig. 7.8, *CN2* and *CN3* happen to choose the same  $t_b$  to start jamming the medium simultaneously while the *Backoff-Timer* of *CN1* does not 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.

```

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        $t_b \leftarrow \text{rand}(0, T_{max})$ ; //refer to Eq. (7.5)
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  $t_j \leftarrow \text{rand}(0, \beta T_{max})$ ; //refer to Eq. (7.7)
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

```

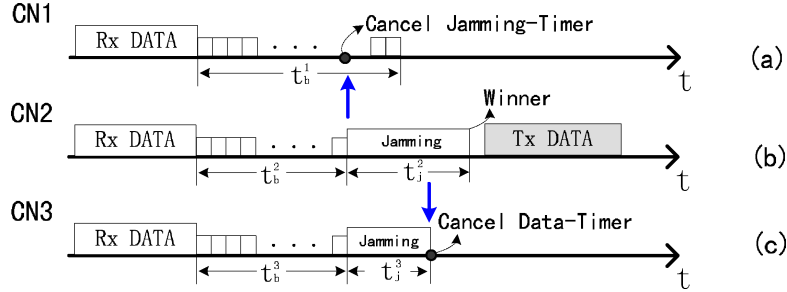


Fig. 7.8. The time flow of broadcasting data packet.

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 7.1 where “ $\leftarrow$ ” 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.

### 7.3.5. 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. 7.9. 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. 7.1.

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. 7.9, respectively. They are equal to:

$$h_{cf} = R - r \quad (7.8)$$

$$v_{cf} = \sqrt{R^2 - r^2}. \quad (7.9)$$

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,

$$r_{max} = \sqrt{r^2 + (2 \cdot v_{cf})^2} = \sqrt{4R^2 - 3r^2}. \quad (7.10)$$

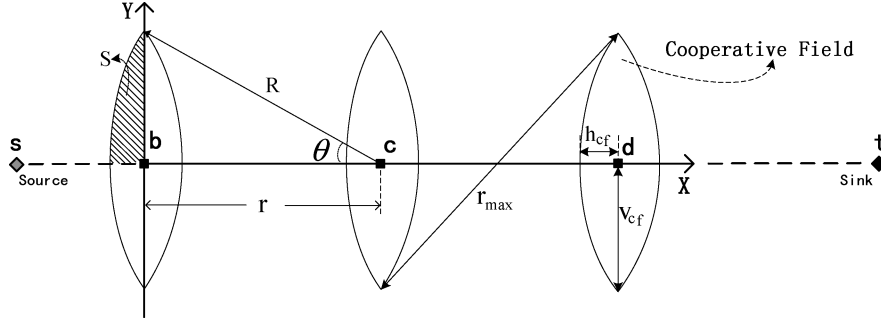


Fig. 7.9. An ideal scenario in REER.

To guarantee any pairs of *CNs* in adjacent cooperative fields can communicate 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 same cooperative 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 Sec. 7.3.4, 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. 7.9. Then  $S_{cf}$  is equal to:

$$S_{cf} = 4 \cdot S = 4 \cdot \left( \theta \cdot R^2 - \frac{1}{2} \cdot \sqrt{R^2 - r^2} \right). \quad (7.11)$$

In Eq. (7.11),  $\theta = \cos^{-1}(r/R)$ . Assume the node density is  $\delta$ . Then, the number of *CNs* in cooperative field ( $N_{cf}$ ) is equal to:

$$N_{cf} = S_{cf} \cdot \delta. \quad (7.12)$$

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:

$$H = \left\lceil \frac{d}{r} \right\rceil. \quad (7.13)$$

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,

$$P = p^H = (1 - f^{N_{cf}})^H. \quad (7.14)$$

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

$$E = e_{tx} \cdot H + e_{rx} \cdot [3(H - 2) \cdot N_{cf} \cdot (1 - f) + 2N_{cf} \cdot (1 - f) + 1]. \quad (7.15)$$

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

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

$$T_{ete} = t_{data} \cdot H + \bar{t}_b \cdot (H - 1). \quad (7.16)$$

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).

### 7.3.6. Control overhead compared with shortest path based geographical routing

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

$$k = \pi R^2 \rho. \quad (7.17)$$

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 a shortest path based geographical routing (e.g., GPSR). Let  $o_r$  be the control overhead for establishing  $RN$ s and  $CN$ s in REER. Then,

$$o_g = n_s \cdot e_{ctrl}. \quad (7.18)$$

$$o_r = H \cdot 3e_{ctrl}. \quad (7.19)$$

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  $CN$ s 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.

## 7.4. Simulation Model

### 7.4.1. Simulation settings

We implemented our scheme using OPNET<sup>18,19</sup> 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 500 m × 200 m 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 60 m. 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 Ref. 20. In,<sup>22</sup> 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 100 s, 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 7.2. 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.

### 7.4.2. 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 Ref. 21, we

Basic Specification	
Network Size	500 m × 200 m
Topology Configuration Mode	Randomized
Total Sensor Node Number	600
Data Rate at MAC layer	1 Mbps
Transmission Range of Sensor Node	60 m
Time Duration of State ON	Default: 10 s
Node failure rate	Default: 0%
Packet loss rate	Default: 0%
Sensed Traffic Specification	
Size of Sensed Data	Default: 1 Kbytes
Size of Control Message	Default: 128 bytes
Sensed Data Packet Interval	5 s
REER Specification	
$r$	Default: 40 m
$\tau$ in Eq. (7.4)	Default: 2.5 ms
$\mu$ in Eq. (7.4)	Default: 5 ms
$\Delta T$ in Eq. (7.6)	Default: 10 ms

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:

$$e = \frac{E}{n_{data}}. \quad (7.20)$$

- *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,<sup>21</sup> a combined metric can reflect the impact of several performances. Considering the reliability is also an important metric in unreliable environment, this chapter adopts the following metric to evaluate the integrated performance of reliability, energy and delay:

$$\eta = \frac{e \cdot T_{ete}}{P}. \quad (7.21)$$

## 7.5. Performance Evaluation

In Sec. 7.5.1, we examine the impact of node density on the REER performance. In Sec. 7.5.2, GPSR and REER with varying  $r$  are evaluated in terms of link failure rate.

### 7.5.1. 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  $\delta_q$  be the normalized node density, i.e. the ratio of the current node density to the default one ( $\frac{600 \text{ nodes}}{500 \times 200 \text{ m}^2}$ ).  $\delta_q$  is changed from 0.25 to 2 by controlling the number of sensor nodes in the fixed size of network.

In Fig. 7.10(a), the higher is  $\delta_q$ , the larger is  $N_{cf}$ , the higher is the hop reliability and  $P$ . When  $\delta_q$  is beyond 1.5, REER has a delivery ratio near 100%.

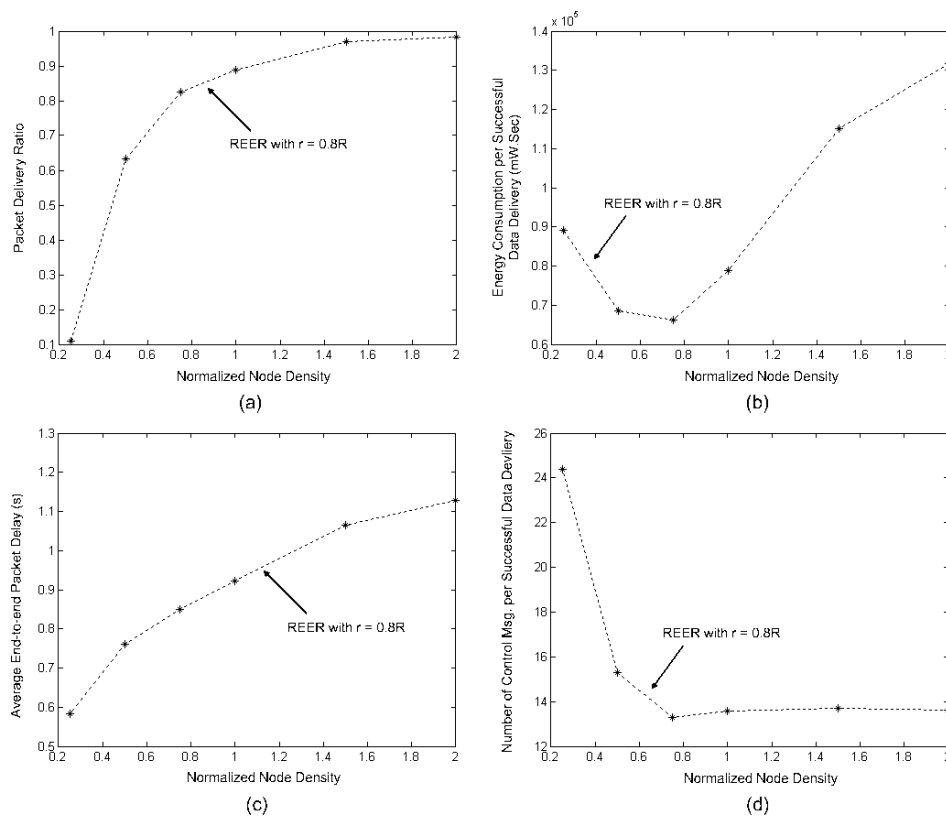


Fig. 7.10. The impact of normalized node density ( $\delta_q$ ) on performances: (a) Packet delivery ratio ( $P$ ); (b) Energy consumption per successful data delivery ( $e$ ); (c) Average end-to-end packet delay ( $T_{reer}$ ); (d) Number of control messages per successful data delivery ( $\eta$ ).

According to Eqs. (7.14) and (7.15),  $P$  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 Eq. (7.20) to make  $e_{reer}$  decrease. When  $\delta_q$  is equal to 0.75,  $e_{reer}$  reaches its minimum. If  $\delta_q$  goes beyond 0.75,  $P$  does not increase much (see Fig. 7.10(a)). However,  $E$  always linearly increases in proportion to  $\delta_q$ , and dominates Eq. (7.20). Thus,  $e$  increases again, as shown in Fig. 7.10(b).

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 Eq. (7.6). With  $\delta_q$  increased,  $N_{cf}$  increases. The larger is  $N_{cf}$ , the larger  $T_{max}$  will be set to avoid collisions. Thus, in Fig. 7.10(c),  $T_{ete}$  of REER increases with  $\delta_q$  increased. Currently, we adopt a simple backoff time function as shown in Eq. (7.6), we believe a better function can lower the data latency extensively.

In Fig. 7.10(d),  $\eta$  reaches its minimum value when  $\delta_q$  is equal to 0.75. The smaller is  $\eta$ , the better is the integrated performance of REER. It is unnecessary to increase  $\delta_q$  more if the value is large enough to achieve required reliability.

### 7.5.2. 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 7.2 fixed.

The smaller is  $r$ , the larger number of  $CNs$  in each cooperative field are exploited. Thus, in Fig. 7.11, 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 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 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. 7.12,  $e_{reer}$  is almost always lower than  $e_{gpsr}$  with varying  $f$ .

According to Eq. (7.15),  $E$  decreases with  $f$  increases, i.e. the link failure helps to save energy for receiving data packet. If the number of  $CNs$  is large enough to

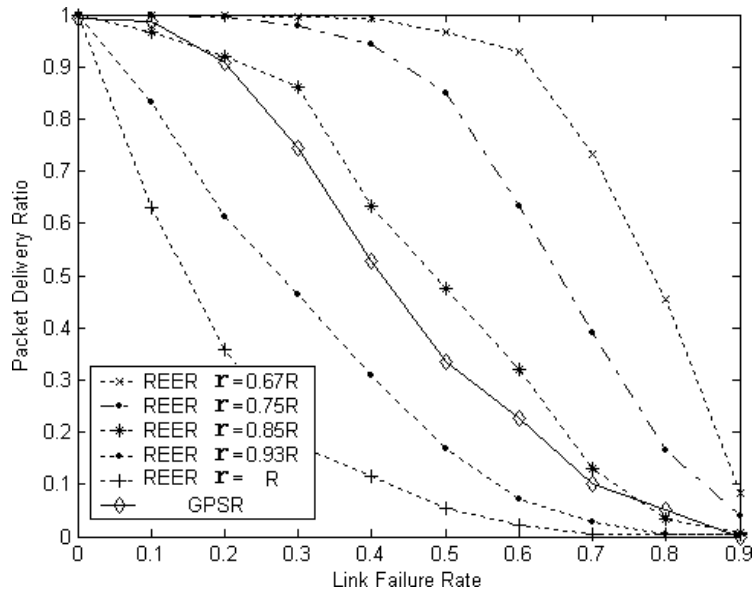


Fig. 7.11. The comparison of  $P$ .

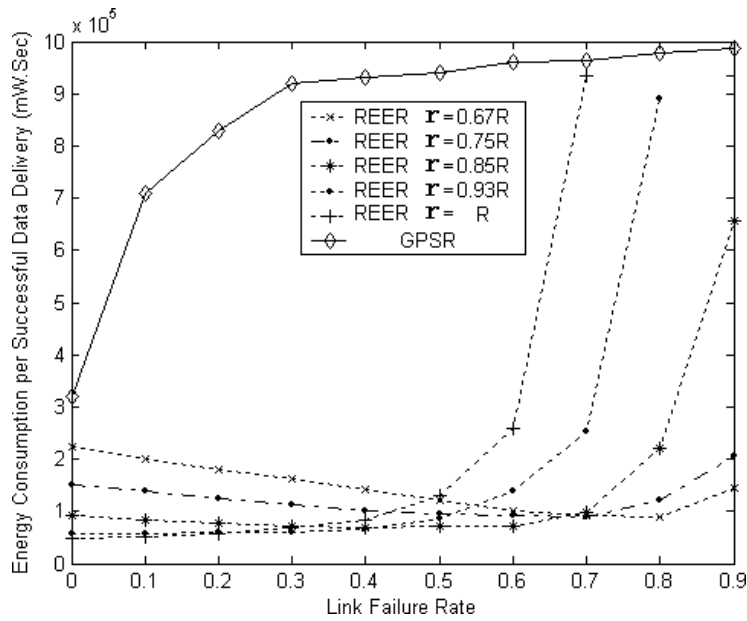
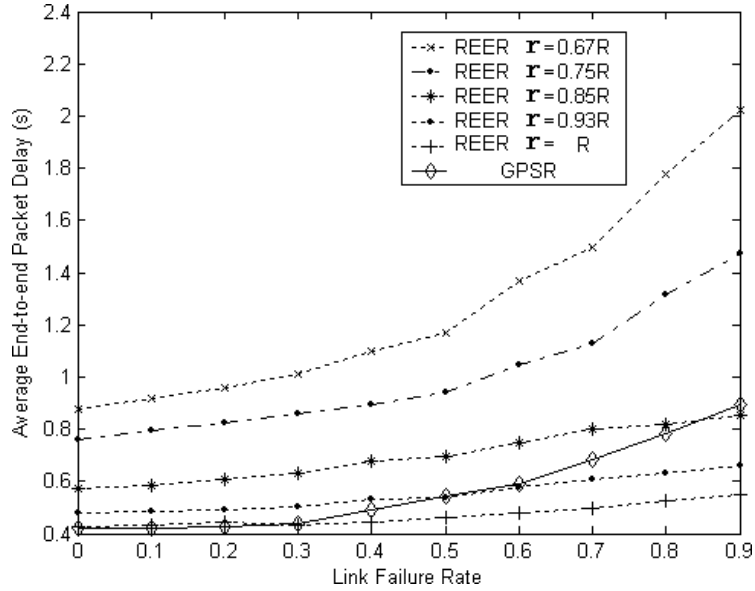


Fig. 7.12. The comparison of  $e$ .

Fig. 7.13. The comparison of  $T_{ete}$ .

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. 7.12, 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  $CNs$  is insufficient to antagonize the high link failure rate, which causes  $n_{data}$  decrease exponentially. Thus,  $e_{reer}$  increases fast again.

In Fig. 7.13, 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. 7.16, the number of  $CNs$  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. 7.16(a),  $CN1$  is selected to forward the data packet. In contrast,  $CN4$  is selected in Fig. 7.16(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. 7.13. It is because that the smaller is  $r$ , the higher is the number of  $CNs$  in a  $CF$ , the higher  $T_{max}$  are needed to differentiate the  $CNs$  according to Eq. (7.6), 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. 7.14,  $n_{ctrl}$  of REER is lower than that of GPSR, since REER never uses control message beaconing to repair a route.

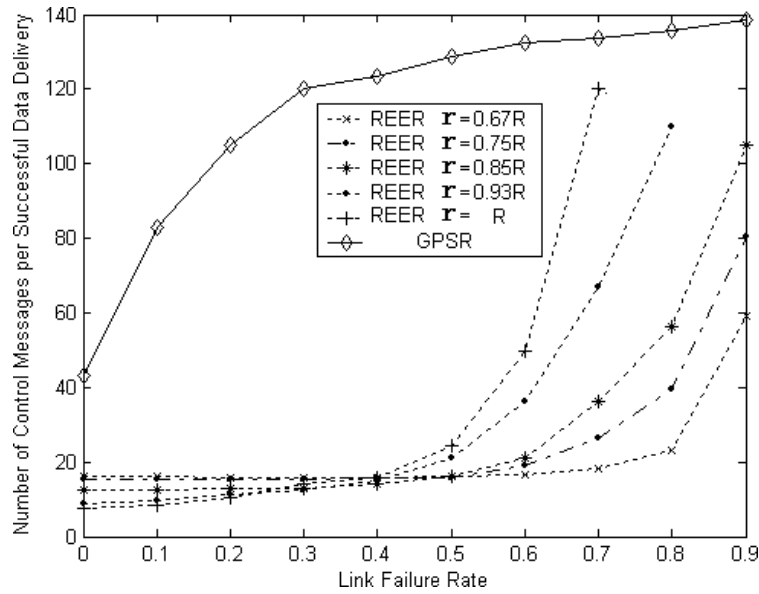


Fig. 7.14. The comparison of  $n_{ctrl}$ .

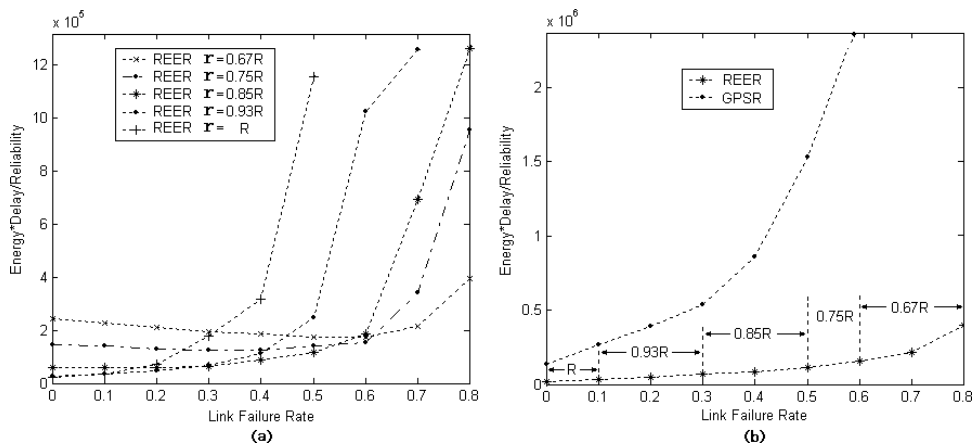


Fig. 7.15. The comparison of  $\eta$ .

Observed in Figs. 7.12–7.14, 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  $\eta$ , Fig. 7.15(a) is plotted. Then, in Fig. 7.15(b), the optimal  $r$  for variable  $f$  are selected. The overall performance gain of REER further improves with the strategy of adaptive  $r$  selection.

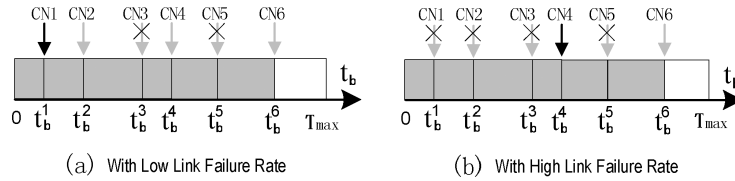


Fig. 7.16. The comparison of backoff time: (a) with low link failure rate; (b) with high link failure rate.

## 7.6. Conclusion

This chapter proposes REER to achieve both reliability and energy-efficiency simultaneously. In REER, we first select reference nodes (*RNs*) between source and sink. Then, multiple cooperative nodes (*CNs*) are selected for each reference node. The smaller is the distance ( $r$ ) between two adjacent *RNs*, the larger number of *CNs* 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 *CNs* 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.

## Acknowledgements

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. Shiwen Mao's research has been supported in part by the National Science Foundation (NSF) under Grant

ECCS-0802113, and through the Wireless Internet Center for Advanced Technology (WICAT) at Auburn University.

## References

1. C. Intanagonwiwat, R. Govindan and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, the *Proceedings of the 6th Annual ACM/IEEE MobiCom* (2000).
2. 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 (2005).
3. F. Stann and J. Heidemann, RMST: Reliable data transport in sensor networks, *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102–112 (2003).
4. 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)*, 1(2), pp. 10–24 (2002).
5. B. Deb, S. Bhatnagar and B. Nath, ReInForM: Reliable information forwarding using multiple paths in sensor networks, *IEEE LCN*, pp. 406–415 (2003).
6. F. Ye, G. Zhong, S. Lu and L. Zhang, GRAdient broadcast: A robust data delivery protocol for large scale sensor networks, accepted by *ACM Wireless Networks (WINET)*, 11(2), 285–298 (2007).
7. 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).
8. 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 (2004).
9. Y. Yuan, Z. He and M. Chen, Virtual MIMO based cross-layer design for wireless sensor networks, *IEEE Transactions on Vehicular Technology*, 55(3), 856–864 (2006).
10. I. Stojmenovic, Position-Based Routing in Ad Hoc Networks, *IEEE Comm. Magazine*, 40(7), 128–134 (2002).
11. P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, *Proc. ACM DIAL 1999*, pp. 48–55, Seattle, USA (1999).
12. B. Karp and H. T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, *Proc. of ACM MobiCom 2000*, pp. 243–254 (2000).
13. H. Frey and I. Stojmenovic, On delivery guarantees of face and combined greedy-face routing algorithms in ad hoc and sensor networks, *ACM MOBICOM*, pp. 390–401 2006.
14. 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).
15. 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*, 54, pp. 1509–1522 (2005).
16. Q. Fang, J. Gao and L. J. Guibas, Locating and bypassing routing holes in sensor networks, *IEEE Infocom*, 23(1), 2458–2468 (2004).
17. Zakhia G. Abichar and J. M. Chang, CONTI: Constant-Time contention resolution for WLAN access, *LNCS*, Vol. 3462, pp. 358–369 (2005).

18. M. Chen, OPNET network simulation, *Press of Tsinghua University*, China, April 2004, ISBN 7-302-08232-4, 352 pages.
19. <http://www.opnet.com>.
20. 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, **29**(2), 231–245 (2006).
21. M. Chen, V. Leung, S. Mao and Y. Yuan, DGR: Directional geographical routing for real-time video communications in wireless sensor networks, *Computer Communications*, **30**(17), 3368–3383 (2007).
22. H. S. Wang and N. Moayeri, Finite-state markov channel — a useful model for radio communication channels, *IEEE Transaction on Vehicular Technology*, **43**(1), 163–171 (1995).