EBRP: Energy-Balanced Routing Protocol for Data Gathering in Wireless Sensor Networks

Fengyuan Ren, *Member*, *IEEE*, Jiao Zhang, Tao He, Chuang Lin, *Senior Member*, *IEEE*, and Sajal K. Das, *Senior Member*, *IEEE*

Abstract—Energy is an extremely critical resource for battery-powered wireless sensor networks (WSN), thus making energy-efficient protocol design a key challenging problem. Most of the existing energy-efficient routing protocols always forward packets along the minimum energy path to the sink to merely minimize energy consumption, which causes an unbalanced distribution of residual energy among sensor nodes, and eventually results in a network partition. In this paper, with the help of the concept of potential in physics, we design an Energy-Balanced Routing Protocol (EBRP) by constructing a mixed virtual potential field in terms of depth, energy density, and residual energy. The goal of this basic approach is to force packets to move toward the sink through the dense energy area so as to protect the nodes with relatively low residual energy. To address the routing loop problem emerging in this basic algorithm, enhanced mechanisms are proposed to detect and eliminate loops. The basic algorithm and loop elimination mechanism are first validated through extensive simulation experiments. Finally, the integrated performance of the full potential-based energy-balanced routing algorithm is evaluated through numerous simulations in a random deployed network running event-driven applications, the impact of the parameters on the performance is examined and guidelines for parameter settings are summarized. Our experimental results show that there are significant improvements in energy balance, network lifetime, coverage ratio, and throughput as compared to the commonly used energy-efficient routing algorithm.

Index Terms—Wireless sensor networks, balancing energy consumption, energy-efficient routing, potential field.

1 Introduction

Wireless Sensor Networks (WSNs) are deployed to carry out various applications, such as environmental monitoring, industrial control, disaster recovery, and battlefield surveillance. WSNs are expected to play even more important role in the next generation networks to sense the physical world [1], [2].

It is well known that energy is one of the most critical resources for battery-powered WSNs. To extend the network lifetime as long as possible, energy efficiency becomes one of the basic tenets in the WSN protocol design. In order to use the limited energy available at sensor nodes more efficiently, most existing routing schemes attempt to find the minimum energy path to the sink to optimize energy usage at nodes. However, the question arises whether it is sufficient to focus only on the energy efficiency while designing routing protocols for WSNs, or other objectives such as network lifetime and coverage should also be taken into account. Experiments performed as part of previous research show that nodes closer to the sink tend to deplete their energy faster than the others [3]. This uneven energy depletion dramatically reduces the network lifetime and decreases the coverage ratio. Furthermore, results in [4] point out that by the time the nodes one hop away from the sink exhaust their energy, there is still up to 93 percent of initial energy left at the nodes farther away. Such imbalance of energy consumption imbalance is definitely undesirable for the long-term health of the network. If the sensor nodes consume their energy more evenly, the connectivity between them and the sink can be maintained for a longer time, thus postponing the network partition. This more graceful degradation of the network connectivity can obviously provide substantial gains. Therefore, it should be rational and practical to make an appropriate trade-off between energy efficiency and balanced energy consumption. With this in mind, in this paper we design a novel routing scheme that overcomes the problem of energy consumption imbalance in most existing energy-efficient routing algorithms, and demonstrates the advantage of balanced energy consumption across the network.

Before proceeding further, let us explain three main reasons that can cause an imbalance in energy distribution:

1. **Topology**. The topology of the initial deployment limits the number of paths along which the data packets can flow. For example, if there is only a single path to the sink, nodes along this path would deplete their energy rather quickly. In this extreme case, there are no ways to reach an overall energy balance.
2. **Application**. The applications themselves will determine the location and the rate at which the nodes generate data. The area generating more data and the path forwarding more packets may suffer a faster energy depletion.

- *F. Ren, J. Zhang, T. He, and C. Lin are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China. E-mail: {renfy, chlin}@tsinghua.edu.cn, zhangjiao08@csnet1.cs.tsinghua.edu.cn, qtmssa@gmail.com.*
- *S.K. Das is with the Department of Computer Science and Engineering, University Texas at Arlington, PO Box 19015, Arlington, TX 76019. E-mail: das@uta.edu.*

Manuscript received 3 Mar. 2010; revised 28 Aug. 2010; accepted 8 Oct. 2010; published online 18 Jan. 2011.
Recommended for acceptance by S. Papavassiliou.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-03-0138.
Digital Object Identifier no. 10.1109/TPDS.2011.40.

3. **Routing**. Most energy-efficient routing protocols always choose a static optimal path to minimize energy consumption, which readily results in energy imbalance since the energy at the nodes on the optimal path is quickly depleted.

In view of the above causes, there are five possible solutions to balance energy consumption:

1. **Deployment optimization**. The original node distribution is to maximize network lifetime according to the traffic pattern in applications, which can solve the problem of mismatch between topology and application. For example, we need to deploy more nodes in the heavy-loaded areas and paths. In addition, the areas closer to the sink should be covered with higher energy density (ED), since the closer to the sink a node is, the more packets does it have to relay. Solutions based on this principle are proposed in [5] and [6].
2. **Topology control**. The basic idea is that, instead of transmitting at maximum power, nodes collaboratively adjust their transmission power and form a proper network topology to balance energy consumption. The investigations in [7], [8], and [9] fall into this category.
3. **Mobile sink/relay nodes**. Mobile sink and relay nodes can achieve a balanced energy consumption by relieving heavily loaded areas or paths in a way dual to the optimization deployment. However, additional mechanisms need to be devised to support node mobility. The solutions following this paradigm can be found in [10], [11], and [12].
4. **Data aggregation**. Data from different sources are aggregated by exploiting redundancy with the objective of minimizing energy consumption in transmissions. The work in [13], [14], and [15] explores the possibility of avoiding energy holes in data-gathering sensor networks through traffic compression and data aggregation.
5. **Energy-balanced routing**. Under a designated topology, employing an energy-balanced routing protocol (EBRP) may be a feasible approach to prolong the network lifetime, yet maintaining the network connectivity. To the best of our knowledge, only little work takes energy consumption balance into account while designing routing algorithms.

Although numerous energy-aware routing protocols have been proposed in the literature, most of them focus only on energy efficiency, namely finding the optimal path to minimize energy consumption. In our opinion, an energy-aware routing protocol should not only aim for energy efficiency, but also for balancing energy consumption balance. In addition, the routing protocols in WSNs can be naturally classified into two categories according to application requirements, namely data-query based routing and data-gathering based routing. The former disseminates the message of an interesting event in the network, thus navigating queries to discover paths to the events. The latter finds the path to the sink for data collected by sensor nodes.

In this paper, we develop an energy-balanced data-gathering routing algorithm using the concept of potential in classical physics [16]. Our scheme called energy-balanced routing protocol, forwards data packets toward the sink through dense energy areas so as to protect the nodes with relatively low residual energy. The cornerstone of the EBRP is to construct three independent virtual potential fields in terms of depth, energy density, and residual energy. The depth field is used to establish a basic routing paradigm which keeps packets move toward the sink. The energy density field ensures that packets are always forwarded along the high energy areas. Finally, the residual energy field aims to protect the low energy nodes. The final routing decision is made by considering the three virtual potential fields together.

The remainder of this paper is organized as follows: The next section describes the related works. In Section 3, a simple example illustrates how uneven energy depletion results in network partition and degradation of network coverage ratio, after which we present the motivation and basic idea behind this work. Subsequently, the potential fields used by EBRP are developed and the implementation details are presented. To address the routing loop problem, a loop detection and elimination mechanism is devised in Section 4. The validation by simulation is described in Section 5. While Section 6 evaluates the integrated performance of the full EBRP algorithm through simulation experiments on an arbitrarily deployed network. The impact of parameters on the performance of EBRP is also examined. Finally, the conclusions are drawn in Section 7.

2 Related Work

As mentioned, numerous literatures focus on energy-efficient routing protocols whose target is to find an optimal path to minimize energy consumption on local nodes or in the whole WSN [17], [18], [19]. Nevertheless, some existing routing protocols have recognized the problem of energy imbalance. For example, LEACH [20], EAD [21], and HEED [22] provide energy balance within clusters by randomly choosing the cluster head, but they are only localized solutions. In [23], the authors define the energy-balance property and then propose, analyze, and evaluate an energy-balanced algorithm in single-hop wireless sensor networks, however the assumption that every node can directly communicate with the sink is not realistic for multihop WSNs. In [3], the energy holes near the sink are considered, but not the energy consumption balance across the whole network. A nonuniform node distribution strategy is proposed to achieve nearly balanced energy depletion in [5]. However, its cost is considerable since the number of nodes increases in geometric progression from the outer coronas to the inner ones except for the outermost one. The energy-aware routing in [24] maintains multiple paths and properly chooses one for each packet delivery to improve the network survivability. It may also be quite costly since it needs to exchange routing information very frequently (every time the energy value changes on the nodes along the path) to get a precise routing metric. In [25], a proactive multipath routing algorithm is provided to achieve spatial energy balance, but it is actually a load balancing mechanism because of the assumption that "energy burden" and "traffic load" can be assimilated. In practice, there are many factors, in addition to the traffic

load, that contribute to energy burden in WSNs, such as channel sensing and channel contention. Undoubtedly, the energy profile can be balanced to a certain extent through spreading traffic on multiple paths. However, it is not an optimal solution because spreading traffic unaware of residual energy distribution is somewhat blindfold.

In the literature, the integrated schemes for balancing energy consumption are also investigated, and the routing algorithm is regarded as an assisted mechanism. In [26], the energy consumption is balanced by alternating between hop-by-hop transmission mode and direct transmission mode. Hop-by-hop transmission can save energy of the nodes far away from the sink, while the direct transmission mode can save energy of the nodes closer to the sink since their relaying burden can be relieved in this mode. For example, EBDG [15] takes full advantage of corona-based network partition, mixed-routing, and data aggregation to balance energy consumption.

Our proposed scheme EBRP employs the steepest gradient search method to make the routing decision. There exist some gradient-based routing protocols in WSNs. They can be classified into two categories: information-query based routing and data-gathering based routing. The former routes information query from the sink to sources, and the latter routes the data collected by nodes to the sink. Most of the gradient-based information query routing protocols [27], [28], [29], [30], and [31] use the natural gradients of physical phenomena since the spatial distribution of many physical quantities (e.g., temperature measurements from heat sensors), follows a natural diffusion law. However, gradients imposed by the natural laws can be far from perfect guides, as witnessed by the existence of local extrema or large plateau regions. To overcome this drawback, information potentials are computed in [32] by solving for a discrete approximation to a partial differential equation over appropriate network neighborhoods. The solutions are classical harmonic functions, with a rich algebraic structure and many useful properties, including the absence of local extrema.

Some data-gathering routing protocols also use the concept of gradient. For example, GBR [33] distributes traffic evenly among all nodes and prevents nonuniform traffic overloading. The sink broadcasts an interest message that is flooded throughout the whole network. Each node receiving the interest message will record the number of hops taken by the interest message, thus computing the number of hops it needs to reach the sink. Here the gradient between two nodes is the difference between their hop counts. A hop gradient is set up from the nodes to the sink and all messages will flow in the direction toward the sink. Actually, GBR is equivalent to the shortest path routing since only the number of hops is considered to compute the gradient. On the other hand, GRAB [34] is designed for reliability by routing redundant copies of messages along a band of interleaved mesh from each source to the receiver. GRAB controls the width of the band by the amount of credit carried in each data message, thus allowing the sender to adjust the robustness of data delivery. A cost field is defined by propagating advertisement packets in the network. The cost at a node is the minimum energy overhead to forward a packet from this node to the sink along a path. In other words, GRAB employs multiple paths and the cost field to enhance the reliability of data delivery and minimize energy consumption, but still readily leads to energy imbalance or even energy holes. In addition, GRAB heavily relies on periodic flooding to maintain the gradient information, resulting in high bandwidth requirement and energy overhead. SGF [35] extends GRAB through opportunistically selecting the forwarder among multiple candidate nodes to adapt to transient channel variations and topology changes. The maintenance of gradient is purely driven by data transmissions and hence decreases overhead. However, the opportunistic selection hardly guarantees the monotonicity of the cost field, so that routing loops are likely to develop uncontrolled.

Concepts of potential and gradient are applied not only to wireless networks, but also in wired ones. For example, the traffic-aware routing algorithm called PBTA [36] is developed to route packets around the congested hot spots in the network, so as to improve the end-to-end delay and jitter. However, it does not attract widespread attention because of its huge management cost. It is indeed expensive to build an exclusive virtual field for each destination in traditional networks where every randomly deployed node can be the destination of some packet. On the contrary, the many-to-one centripetal traffic pattern in WSNs with a single sink requires that only one hybrid potential, namely that for the sink, is maintained when potential-based routing is applied. In [37], inspired by the magnetism in physics, the authors devise a simple data dissemination mechanism in which the data packets, which are regarded as metallic nails, are attracted by the sink node just like a piece of magnet. This simple data dissemination mechanism can only provide the basic routing function, i.e., instructing packets move toward the sink, through constructing a single magnetic field.

In gradient-based routing, the gradient is only a state representing the direction toward neighboring nodes through which a destined sink can be reached. It can be set up according to different variables, such as hop count, energy consumption, physical distance, and fingerprint of the monitored phenomenon. In this paper, we will borrow the concept of potential in classical physics and construct multiple potential fields using different network state variables, such as depth, residual energy, and energy density; and then superpose them into a virtual hybrid potential field using the normalized field strengths to drive data packets toward the sink through the dense energy area. The dynamic potential field varying with the distribution of node residual energy provides chances for the routing based on potential to consume energy as evenly as possible in any network environments.

3 Basic Routing Algorithm

In this section, we present the basic idea underlying the proposed EBRP scheme. For better understanding, let us first introduce some definitions and terminologies.

3.1 Definitions

Network partition. The nodes in a WSN may fail to operate for some reasons, when the network may split into two or more disconnected partitions. This phenomenon is called

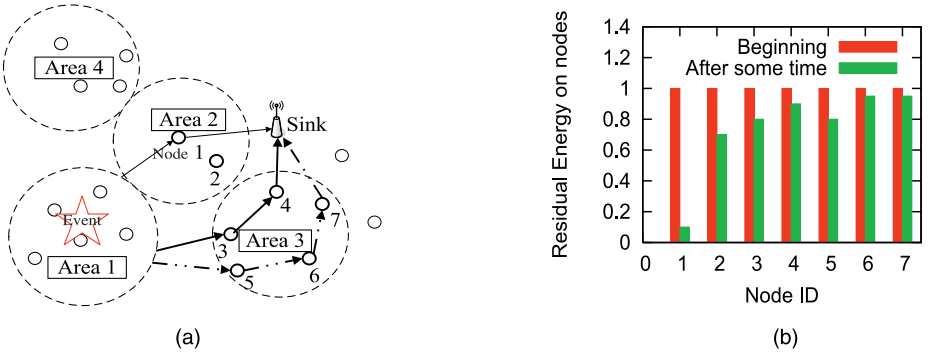

Fig. 1. Illustration of energy consumption imbalance. (a) Topology. (b) Energy distribution.

network partition which may deteriorate or even nullify the usefulness and effectiveness of the whole network. Therefore, it is crucial to avoid the network partition. In this work, we focus on balancing energy consumption to avoid the network partition caused by energy exhaustion due to excessive and unbalanced usage.

Neighbor. All the nodes in the radio coverage disk of node i except for i itself are its neighbors, denoted by $nbr(i)$.

Depth. The *depth* of a node is the number of hops along the shortest path from the sink.

Energy density. The *energy density* of a point (x, y) in the network is defined as the ratio of the sum of residual energy of the nodes within the radio coverage disk to the area of the radio coverage disk.

3.2 Motivation

For routing protocol design in WSNs, the energy balance and energy efficiency should be two different technical goals, since they will lead to routing algorithms with different attributes. An energy-efficient routing protocol tries to extend the network lifetime through minimizing the energy consumption; whereas an energy-balanced routing protocol intends to prolong the network lifetime through uniform energy consumption. The former readily results in the premature network partition that disables the network functioning, although there may be much residual energy left. On the other hand, the latter may not be optimal with respect to energy efficiency, as it can burn energy evenly to keep network connectivity and maintain network functioning as long as possible. Let us use a simple example to demonstrate what uneven energy depletion results in and how the proposed scheme EBRP works to balance energy consumption. One small part of a wireless sensor network is illustrated in Fig. 1a. (Note that there may be many sensor nodes on the right side of the sink; also for a clear description, we manually split the visible field into four areas.). Assume an event occurs in area 1, which is far away from the sink. Most existing energy-efficient routing protocols are prone to choose the shortest path because there are only 2 hops to reach the sink and the energy consumption is minimized. Unfortunately, node 1 in area 2 runs out of its energy quickly because it has to relay too many packets from area 1 and area 4. The residual energy histogram in Fig. 1b presents simulation results, which confirm this phenomenon. Whenever this occurs, there will be a few living nodes in area 2, thus the network connectivity is affected, and area 4 could be partitioned from the rest of the network because node 1 has exhausted its battery power.

How to protect area 2? more precisely how to balance energy consumption between area 2 and the other areas? As the energy density of area 2 is as high as the other areas, EBRP would choose the same route as most energy-efficient routing algorithms. After some time, however, when the residual energy on the nodes in area 2 becomes lower than that on the nodes in the other areas, EBRP could route the packets from area 1 through area 3 where there are more nodes and energy before the energy on the nodes in area 2 is exhausted. Thus, area 2 is protected properly. By this way, both energy efficiency and energy balance are taken into account, thus achieving a compromise. The key issue is how to dynamically switch among the different paths depending on the available local information, such as residual energy, depth, and energy density.

Enlightened by the concept of the "potential" in classical physics, we can purposely build up a virtual potential field using various information on every node to naturally "push" the packet to the sink through the dense energy area. In order to demonstrate this basic idea, we construct a virtual potential field using the energy in a sensor network where the nodes are dense at the center but sparse at the edge. Its topology is depicted in Fig. 2a, and the corresponding energy potential field shown in Fig. 2b looks like a hill. If the packet could always "climb" to this hill along the direction of the gradient (i.e., the direction in which the potential varies fastest), it would eventually reach the sink residing at the center. Since the energy is consumed by packet transmission and other operations, the potential field is time varying. Therefore, the route based on potential will dynamically change, which implies that the energy consumption will be balanced.

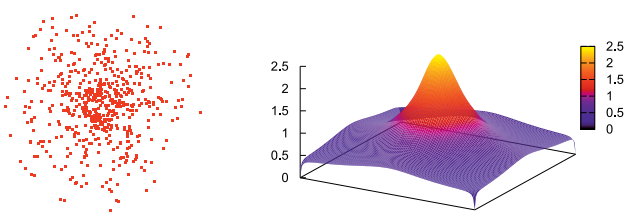

Fig. 2. EBRP's basic principle. (a) Deployment of Sensors. (b) Potential Field.

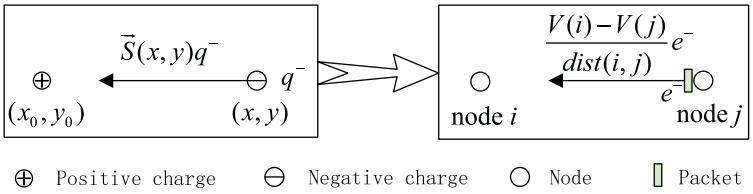

Fig. 3. The potential field model in EBRP. Here $V(i), V(j)$ represent the potential on node i and node j, respectively, whereas $dist(i,j)$ denotes the distance between nodes i and j. A packet is considered as a unit negative charge e^-.

In practice, it is not easy to construct and maintain such a perfect potential field like the one depicted in Fig. 2 because of random deployment and heterogeneous nodes. We need to make use of most of the valuable information to build up an available potential field.

3.3 Design of Potential Fields

In this section, we will describe how to construct the potential fields using depth, energy density, and residual energy on each node, and how to compound them into a unified virtual potential field to drive packets to move toward the sink, at the same time balance the energy consumption.

3.3.1 Potential Field Model

For the convenience of discussion, we first recall several concepts in the classical potential theory with an example of electric field. The concepts include potential, potential field, field strength, force, potential difference, directional differential coefficient, and gradient (for details, see [16]).

A positive electrical charge +Q residing at position (x_0, y_0) induces an electric potential field $V(x,y)$ at (x,y) around itself:

$$V(x,y) = \frac{Q}{4\pi\xi_0\sqrt{(x-x_0)^2+(y-y_0)^2}}. \quad (1)$$

Here ξ_0 is a constant. Thus, the field strength $\vec{S}(x,y)$ is

$$\vec{S}(x,y) = \left\{\frac{dV}{dx}, \frac{dV}{dy}\right\}. \quad (2)$$

The direction of the field strength and that of the force $\vec{F}$ which the negative electric charge suffers are the same as shown in the left part of Fig. 3. Equation 2 also signifies that the direction of field strength is the same as that of the gradient, namely the negative charges will be forced to move along the direction of the gradient, which is also the direction in which the electric potential varies the fastest.

The gradient is the maximum directional differential coefficient, which can be represented by the ratio of the potential difference to the distance between two points, as shown in the right part of Fig. 3, which is just the potential model employed by our energy-balanced routing algorithm. Therefore, if all the directional differential coefficients between one point and its neighboring points are known, the gradient at this point can be obtained. Particularly, if a node i is a local maximum, i.e., the force from node i to any of its neighbors is negative, the neighbor with the maximum force will be choosen as the next hop.

Taking the complexity and diversity of WSNs into account, we define three different potential fields using depth, energy density, and residual energy, respectively, and then superpose them together to form a hybrid potential field to force the packets to move toward the sink as well as to keep the residual energy even. By this way, the occurrence of network partition caused by energy holes can be deferred, and the network lifetime could be extended. These three fields have different attributes and effects on the routing decision. The depth field compels the data packet to flow toward the sink directly. The energy density field drives packets through the areas with more energy, namely the packets take relatively long "detours" to balance energy consumption. The residual energy field intends to protect low energy nodes from dying.

3.3.2 Depth Potential Field

To provide the basic routing function, namely to instruct packets move toward the sink, we define the inverse proportional function of depth as the depth potential field $V_d(d)$:

$$V_d(d) = \frac{1}{d+1}, \quad (3)$$

where $d = D(i)$ denotes the depth of node i. Then, the depth potential difference $U_d(d_1, d_2)$ from depth d_1 to depth d_2 is given by

$$U_d(d_1,d_2) = V_d(d_2) - V_d(d_1) = \frac{d_1-d_2}{(d_2+1)(d_1+1)}. \quad (4)$$

Since the potential function $V_d(d)$ is monotonically decreasing, when the packets in this depth potential field move along the direction of the gradient, they could reach the sink eventually and the basic routing function can be achieved. For a given network topology, $V_d(d)$ is definite and time invariant. Moreover, when the data packets move closer to the sink, the centrality should be larger, where the centrality denotes the trend that a node in depth d forwards the packets to the neighbors in depth $d-1$. Some other properties of the depth field are summarized as follows:

1. The value of the depth difference of neighboring nodes should be one of 0, 1, and -1 since the nodes two hops away from a node cannot become its neighbors. Thus

$$U_d(d_1,d_2)\begin{cases} =0, & \text{if } d_1 = d_2, \\ >0, & \text{if } d_1 - d_2 = 1, \\ <0, & \text{if } d_1 - d_2 = -1. \end{cases} \quad (5)$$

2. The potential difference from depth $n+1$ to n is given by

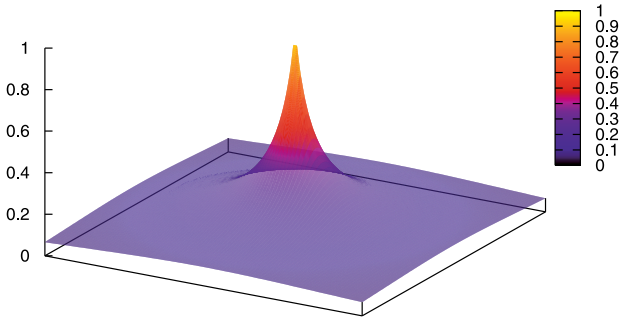

Fig. 4. Depth potential field.

$$U_d(n+1,n) = \frac{1}{(n+1)(n+2)}, \text{ for } n \geq 0. \quad (6)$$

Obviously, U_d is a fast decreasing function, which implies that the closer the packet approaches to the sink (i.e., the depth of node n is small), the larger is the tendency that it moves toward the sink. After three potential fields are finally superposed together (see Section 3.3.5), the "force" originating from the depth potential field is dominant near the sink. For example, both the potential difference from depth 2 to 1 and from depth 1 to the sink (depth 0) are rather large: $U_d(2,1) = 1/(1+1) - 1/(2+1) = 1/6$, $U_d(1,0) = 1/(0+1) - 1/(1+1) = 1/2$. However, in the area far away from the sink (i.e., large n), the centrality is quite weak, which is favorable for energy balance since it can provide more chances for packets to choose an appropriate route (not the shortest path). For instance, the depth potential difference from depth 10 to 9 is quite small: $U_d(10,9) = 1/(9+1) - 1/(10+1) = 1/110$. In these areas, the "force" originating from the depth potential field could not be dominant any longer; some possible disturbances caused by the other two fields (defined in the subsequent section) enable packets deviate from the shortest path. Fig. 4 illustrates the shape of a general depth potential field. In a word, the static depth potential field always attracts packets to move toward the sink. Otherwise, it drives the packets in different positions to approach to the sink with different "forces."

3.3.3 Energy Density Potential Field

A node adds up the energy values of all its neighbors, which can be obtained through messages exchanged among nodes (see Section 3.5), and calculates the area of the radio coverage disk, so that the corresponding *energy density* can be readily obtained using the aforementioned definition. EBRP defines the energy density potential field as follows:

$$V_{ed}(i,t) = ED(i,t), \quad (7)$$

where $V_{ed}(i,t)$ is the energy-density potential of node i at time t, and $ED(i,t)$ is the *energy density* on the position of node i at time t. Thus, the potential difference $U_{ed}(i,j,t)$ from node i to node j is given by

$$U_{ed}(i,j,t) = V_{ed}(j,t) - V_{ed}(i,t) = ED(j,t) - ED(i,t). \quad (8)$$

Driven by this potential field, the data packets will always flow toward the dense energy areas. However, with only this energy density field, the routing algorithm is not practical since it would suffer from the serious problem of routing loops. This fact will be clarified in the subsequent simulation experiments. The depth potential field will play an important role in eliminating the routing loops. Otherwise, in order to protect low energy nodes (especially, those in the paths toward a dense energy area), an extra potential field with respect to the energy needs to be constructed.

3.3.4 Energy Potential Field

EBRP defines an energy potential field using the residual energy on the nodes in order to protect the nodes with low energy:

$$V_e(i,t) = E(i,t), \quad (9)$$

where $V_e(i,t)$ is the energy potential of node i at time t, and $E(i,t)$ is the residual energy of node i at time t. Then potential difference $U_e(i,j,t)$ from node i to j is derived as

$$U_e(i,j,t) = V_e(j,t) - V_e(i,t) = E(j,t) - E(i,t). \quad (10)$$

The two latter potential fields are constructed using the linear functions of energy density and residual energy, respectively. Although the properties of the linear potential fields are straightforward, both of them are time varying, which will result in the routing loop.

3.3.5 Hybrid Potential Fields

We define three different potential fields. Eventually, they need to be superposed together to impose the impact on choosing a proper route. Now, in order to map the three potential differences to the same range $[-1,1]$, we use the following rules to map $U_{ed}(i,j,t)$ to $U'_{ed}(i,j,t)$:

$$U'_{ed}(i,j,t) = \begin{cases} 1 - \frac{1}{\phi_{ed}(i,j,t)}, & \text{if } \phi_{ed}(i,j,t) \geq 1, \\ \phi_{ed}(i,j,t) - 1, & \text{if } 0 \leq \phi_{ed}(i,j,t) < 1, \end{cases} \quad (11)$$

where $\phi_{ed}(i,j,t) = \frac{V_{ed}(j,t)}{V_{ed}(i,t)}$. We also have

$$1 - \frac{1}{\phi_{ed}(i,j,t)} = \frac{V_{ed}(j,t) - V_{ed}(i,t)}{V_{ed}(j,t)} = \frac{U_{ed}(i,j,t)}{V_{ed}(j,t)}, \quad (12)$$

$$\phi_{ed}(i,j,t) - 1 = \frac{V_{ed}(j,t) - V_{ed}(i,t)}{V_{ed}(i,t)} = \frac{U_{ed}(i,j,t)}{V_{ed}(i,t)}. \quad (13)$$

Actually, from (11), (12), and (13), the physical meaning of $U'_{ed}(i,j,t)$ is very obvious. It is the potential difference per unit energy density potential. When $\phi_{ed}(i,j,t) \geq 1$, $U'_{ed}(i,j,t) = \frac{U_{ed}(i,j,t)}{V_{ed}(j,t)}$ is the potential difference per unit energy density potential in the node j, and the corresponding force is positive so that the packet on the node i is "pulled" toward the node j. When $0 \leq \phi_{ed}(i,j,t) < 1$, $U'_{ed}(i,j,t) = \frac{U_{ed}(i,j,t)}{V_{ed}(i,t)}$ is the potential difference per unit energy density potential in the node i, and the force is negative; thus the packet on the node i is compelled to stay on.

In the same way, we cab normalize the depth and residual energy fields:

$$U'_d(i,j) = \begin{cases} 1 - \frac{1}{\phi_d(i,j)}, & \text{if } \phi_d(i,j) \geq 1, \\ \phi_d(i,j) - 1, & \text{if } 0 \leq \phi_d(i,j) < 1, \end{cases} \quad (14)$$

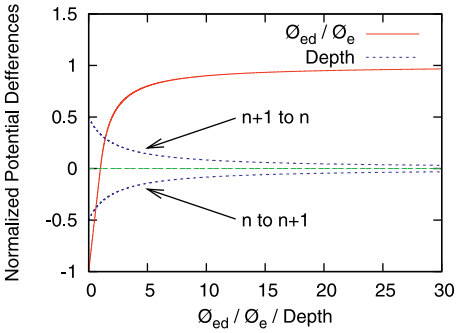

Fig. 5. The trends of normalized potential differences.

$$U_e'(i,j,t) = \begin{cases} 1 - \dfrac{1}{\phi_e(i,j,t)}, & \text{if } \phi_e(i,j,t) \geq 1, \\ \phi_e(i,j,t) - 1, & \text{if } 0 \leq \phi_e(i,j,t) < 1, \end{cases} \tag{15}$$

where $\phi_d(i,j) = \frac{V_d(j)}{V_d(i)}$, and $\phi_e(i,j) = \frac{E(j,t)}{E(i,t)}$. Similarly, we can also understand the meaning of the mapping defined in (14) and (15), namely $U_e'(i,j,t)$ and $U_d'(i,j,t)$ denote the potential difference per unit residual energy potential and depth potential, respectively.

From (3) and (14), the normalized depth potential difference $U_d'(i,j)$ can be simplified as follows:

$$U_d'(i,j) = \begin{cases} \dfrac{1}{n+1}, & i = n \text{ and } j = n-1, n \geq 1, \\ -\dfrac{1}{n+2}, & i = n \text{ and } j = n+1, n \geq 0. \end{cases} \tag{16}$$

The trends of these three normalized potential differences are depicted in Fig. 5. There are two curves for the normalized depth potential differences, $U_d'(n,n+1)$ and $U_d'(n+1,n)$. They are still fast decreasing functions, satisfy all properties discussed in Section 3.3.2, and their value range is $U_d' \in [-\frac{1}{2}, \frac{1}{2}]$. The curves for the normalized energy density and residual energy potential differences are similar, and both of them are monotonically increasing functions within the range $[-1,1]$. We also have noticed that when ϕ_{ed} is very large (e.g., larger than 10), U_{ed}' increases quite slowly, which is meaningful since it is unnecessary to distinguish the neighbors whose energy density is more than 10 times than that of the local node, although EBRP can actually do this when its other potential differences (i.e., depth and residual energy) are the same. The normalized residual energy potential difference U_e' also has the same properties as U_{ed}'.

Next, we use the weighted sum of the above three independent potential fields to construct a new virtual field $V_m(i,t)$, whose potential difference $U_m(i,j,t)$ is defined as

$$U_m(i,j,t) = (1-\alpha-\beta)U_d'(i,j) + \alpha U_{ed}'(i,j,t) + \beta U_e'(i,j,t), \tag{17}$$

where $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and $0 \leq \alpha + \beta \leq 1$. The packets will be driven by this hybrid virtual potential field to move in the networks. The weights α and β determine how much impact the energy density potential field and residual energy potential field impose on the routing decision, respectively. The effect of their values on the performance and their proper choices will be discussed in Sections 3.4 and 6.3.

With the potential difference $U_m(i,j,t)$, the directional differential coefficient from node i toward its neighbor $j \in nbr(i)$ can be easily calculated as

$$D_{i\to j,t} = \frac{U_m'(i,j,t)}{c_{i\to j,t}}, \tag{18}$$

where $c_{i\to j,t} (0 \leq c_{i\to j,t} \leq 1)$ is the cost of the radio link from node i to node j. For the sake of convenience and simplicity, EBRP just considers the physical distance, which is a constant between two nodes, namely

$$c_{i\to j,t} = dist_{i\to j}, \tag{19}$$

where $dist_{i\to j}$ is the distance from node i to node j. If we use the radio range as a unit to measure the physical distance between nodes, then

$$\begin{cases} dist_{i\to j} = 0, & \text{if } j = i, \\ 0 < dist_{i\to j} \leq 1, & \text{if } j \in nbr(i), \\ dist_{i\to j} > 1, & \text{if } j \notin nbr(i). \end{cases}$$

Finally, EBRP chooses the neighbor $x \in nbr(i)$ with the maximum $D_{i\to x,t}$ as its next hop since it is just in the direction of the gradient of the hybrid potential field.

3.4 Property of EBRP

This section presents two propositions to describe the effect of the parameters on EBRP.

Proposition 1. *For node i in depth n, let $H = \{x | D(x) = n+1, x \in nbr(i)\}$. If $\alpha + \beta < \frac{2n+3}{2n^2+8n+7}$, $\forall h \in H$ will never be chosen as the next-hop node of i.*

Proof. Let $L = \{x | D(x) = d-1, x \in nbr(v)\}$. If $\forall h \in H$, $\exists l \in L$ such that $U_m(i,l,t) > U_m(i,h,t)$, then the proposition is proved.

If the energy density and residual energy of node l is larger than that of node h, then h will obviously not be selected as the next hop of node i since the depth potential field of node i is also larger than that of node h.

Assume the relationship of the energy density and residual energy of node i,h,l at time t is as follows:

$$ED(h,t) = n_h^{ed}(t)ED(i,t), E(h,t) = n_h^e(t)E(i,t),$$
$$ED(i,t) = n_l^{ed}(t)ED(l,t), E(i,t) = n_l^e(t)E(l,t),$$

where $n_h^{ed}(t), n_h^e(t), n_l^{ed}(t), n_l^e(t)$ are real numbers.

Considering the best situation that node i will select node h rather than node l as the next hop, i.e., $n_h^{ed}(t)$, $n_h^e(t), n_l^{ed}(t), n_l^e(t)$ are all large enough positive real numbers. The force from node i to node l and h is as follows

$$U_m(i,l,t) = (1-\alpha-\beta)\frac{1}{n+1} + \alpha\left(\frac{1}{n_l^{ed}(t)} - 1\right) + \beta\left(\frac{1}{n_l^e(t)} - 1\right),$$

$$U_m(i,h,t) = (1-\alpha-\beta)\left(\frac{-1}{n+2}\right) + \alpha\left(1 - \frac{1}{n_h^{ed}(t)}\right) + \beta\left(1 - \frac{1}{n_h^e(t)}\right).$$

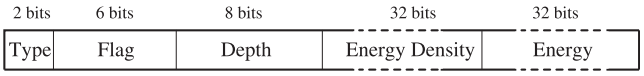

Fig. 6. Structure of updating message.

Subtracting $U_m(i,h,t)$ from $U_m(i,l,t)$, we can obtain that

$$U_m(i,l,t) - U_m(i,h,t) = (1-\alpha-\beta)\frac{2n+3}{(n+1)(n+2)} - \alpha\left(2 - \frac{1}{n_h^{ed}(t)} - \frac{1}{n_l^{ed}(t)}\right) - \beta\left(2 - \frac{1}{n_h^{e}(t)} - \frac{1}{n_l^{e}(t)}\right).$$

Since $n_h^{ed}(t), n_h^e(t), n_l^{ed}(t), n_l^e(t)$ can be large enough, when $n_h^{ed}(t), n_h^e(t), n_l^{ed}(t), n_l^e(t) \to +\infty$, $(2 - \frac{1}{n_h^{ed}(t)} - \frac{1}{n_l^{ed}(t)})$ and $(2 - \frac{1}{n_h^{e}(t)} - \frac{1}{n_l^{e}(t)})$ approach to 2. Hence

$$U_m(i,l,t) - U_m(i,h,t) > (1-\alpha-\beta)\frac{2n+3}{(n+1)(n+2)} - 2(\alpha+\beta).$$

Therefore, if $\alpha+\beta < \frac{2n+3}{2n^2+8n+7}$, we can get

$$U_m(i,l,t) - U_m(i,h,t) > \frac{2n+3}{(n+1)(n+2)} - \left(2 + \frac{2n+3}{(n+1)(n+2)}\right)\frac{2n+3}{2n^2+8n+7} = 0.$$

Which indicates that node i will never select node h as the next hop. □

Remarks. Proposition 1 shows the condition under which packets will never be sent backward. We can see that when n is relatively large, $\alpha+\beta$ is very small, i.e., the weight of the depth potential field must be very large, to avoid packets being sent backward at the nodes with higher depth. While when n becomes smaller, $\alpha+\beta$ could be larger, i.e., the weight of energy density and residual energy could be larger, but can still ensure packets not being sent backward. Considering $n=5$, $\alpha+\beta<0.13$ can ensure no packets will be sent backward at the nodes with depth lower than 5. While at the nodes with higher depth, some backward transmissions are allowed to balance energy.

Furthermore, note that $\alpha+\beta < \frac{2n+3}{2n^2+8n+7}$ is obtained under the condition that $n_h^{ed}(t), n_h^e(t), n_l^{ed}(t), n_l^e(t) \to +\infty$. However, in reality, generally $n_h^{ed}(t), n_h^e(t), n_l^{ed}(t), n_l^e(t)$ will not be very large without manual intervention or something else, which indicates that it is possible to set larger α and β than the conclusion of Proposition 1 to better balance energy without causing much backward transmissions in practice.

Proposition 2. *For node i in depth n, let $S=\{x|D(x)=n, x\in nbr(i)\}$, $H=\{x|D(x)=n+1, x\in nbr(i)\}$. If $\alpha+\beta<\frac{1}{2n+3}$, $\forall s\in S$ and $\forall h\in H$ will never be chosen as the next-hop node of node i.*

Proof. Similar to the proof of Proposition 1. □

The above proposition indicates the condition under which the shortest path will be chosen, namely, a node does not choose the parent from its neighbors with the same or higher depth. However, different from the shortest path tree with only hop count as the metric, this shortest path also takes the energy density and residual energy into consideration.

```
EBRP Algorithm – Updating Message Processing
If Received a Updating Message (u_Msg) from a Neighboring Node (neighbor_ID)
1:  Local_Energy_Density = calculateLocalEnergyDensity();
2:  Ød = (Local_Depth + 1) / (u_Msg.Depth + 1)
3:  Ud = Ød > 1 ? 1 – 1 / Ød : Ød – 1
4:  Øed = u_Msg.Energy_Density / Local_Energy_Density
5:  Ued = Øed > 1 ? 1 – 1 / Øed : Øed – 1
6:  Øe = u_Msg.Energy / Local_Energy
7:  Ue = Øe > 1 ? 1 – 1 / Øe : Øe – 1
8:  Um = (1 – α – β) · Ud + α · Ued + β · Ue
9:  COST = Distance(neighbor_ID)
10: D = Um / COST
11: updateRoutingTable(neighbor_ID)
12: select the Lowest Depth from the routing table as LD
13: if(Local_Depth > LD + 1) then
14:     setLocalDepth(LD + 1)
15: end if
16: select Parent according to max-D, max-Um, max-Ued, max-Ue, min-u_Msg.DEPTH, min-COST, Random
```

Fig. 7. Pseudocodes of EBRP algorithm. (Function **calculateEnergy-Density()** calculates and returns the energy density of local node; **Distance()** returns the distance of the neighbor; **updateRoutingTable()** updates the routing table; and **setLocalDepth()** sets the depth of the local node).

The depth potential field decreases fast with the increase of the depth value. The above two propositions show that the large weight of the energy density and residual energy field may cause routing loops. Section 4 discusses in detail how to detect and eliminate loops.

3.5 Design of Distributed Routing Algorithm

The details on the design and implementation of EBRP are described below.

3.5.1 Control Message

We first define the format of the routing control message, whose payload contains five parts shown in Fig. 6. The flag field (6 bits) is reserved for possible extensions. Up to now, EBRP defines two types of control messages. One is the normal updated message whose type field is defined as 00, and the other fields carry the information used by EBRP, including depth, energy density, and residual energy. Another is a special message whose type field is 01 without other payloads, and used to confirm routing loops. It is called Check-Loop-Packet (CLP).

3.5.2 Depth

In the beginning, the depth of all nodes are initialized to $0xff$, except for the sink whose default depth is 0. The sink first sends the update message, nodes one hop away from the sink will get their own depth by adding 1 to the depth value in the update message. Then, the other nodes will also obtain their own depth by receiving update message from its neighbors who already have a depth just in the same way as the nodes one hop away. Fig. 7 depicts the pseudocode of EBRP. Lines 12 to 15 show how nodes

calculate their depth when receiving an update message. Otherwise, EBRP will recalculate the depth when it detects newly died neighbors or it needs to shield its current parent for eliminating routing loops (see Section 4). The simplest way is to add 1 to the minimum effective depth value remaining in the routing table.

3.5.3 Energy and Energy Density

The EBRP needs to know the residual energy on the local node, but the most popular WSN node hardware platforms, such as MicaZ [38] and Telos [39], are unable to directly provide this information except for Sun Spot [40]. Fortunately, we can add some extra hardware components, such as an analog-to-digital converter measuring the voltage or an integrator cumulating the consumed energy, to estimate the residual energy. Specifically, the pure software solution is also feasible. We can log the actions that the local node has performed to estimate the consumed energy using a proper battery model [41]. In this work, we assume that the value of residual energy can be easily obtained from one of the above methods. This value will be sent in the update message, thus every node knows the residual energy of all its neighbors, and records them in the routing table. The energy density at the position of the local node can be obtained by adding up all the residual energy of its neighbors in its routing table, and then dividing this sum by the area of the radio coverage disk.

3.5.4 Distance

The distance between two neighbors can be easily obtained by several techniques, such as signal attenuation evaluation or estimation depending on Received Signal Strength Indicator (RSSI) [42]. It is noted that the distance used in EBRP may be approximate since it is enough to distinguish relatively far or near from the local node.

3.5.5 Time to Update

EBRP only exchanges routing messages with its direct neighbors. To keep the update pace, EBRP defines a *maximum updating interval* (MUI) and a *least updating interval* (LUI) between two successive update messages. MUI is always larger than LUI. The update messages should be delivered between a LUI and a MUI since the last one. If there are no messages from a neighbor during two MUIs intervals, this neighbor is considered dead, and EBRP will recalculate the depth and other values. EBRP will send an update message when any one of the following events occurs.

1. *MUI timer expires.* If the elapsed time since sending the last updating message exceeds a MUI, the node will send a new one immediately, no matter whether the depth or energy has changed.
2. *Energy consumption exceeds a certain threshold.* If the energy on a node has been burnt by 1 percent of the residual energy which is sent out in the last successful update message, and the elapsed time also exceeds a LUI since the last update message, the node will send a new one.
3. *Depth changes.* If the depth of a node has changed, and the elapsed time also exceeds a LUI since the last successful update message, the node will also send a new one.

3.5.6 Processing of Update Message

When a node receives an update message from one of its neighbors, it will refresh its routing table and reselect a next-hop node according to the previous algorithm. If there are more than one neighbor with the same maximum directional differential coefficient, EBRP in turn chooses the next-hop node according to maximum U_m, maximum U'_{ed}, maximum U'_e, minimum depth of neighbors, and minimum cost of links. After doing that, if EBRP still cannot determine its parent, it will choose one randomly.

4 Loop Detection and Elimination

In practice, the dynamic network topology and the time-varying residual energy could readily result in local "hills" in the hybrid virtual potential field. EBRP would direct packets to climb up these local "hills," eventually routing loops appear. For example, there are two obvious cases: 1) The nodes at the peak of the local "hill" are very likely to choose each other as their next-hop nodes; 2) A packet could return to the peak when it goes down to the mountainside of the local "hill."

As mentioned previously, the depth field itself is perfect and stable without any local "hill." However, in the energy density field and the residual energy field, these local "hills" are prone to appear because these two fields are time varying. Otherwise, there are always a lot of local "hills" in an arbitrarily deployed network due to the nodes distribution fashion.

In order to eliminate possible routing loops and make EBRP more practical, we need a mechanism for detecting and eliminating the routing loops to enhance the basic algorithm. Straightforwardly, the loop detection can be done by checking the source addresses of received packets and monitoring the length of the local queue. Once a loop is detected, EBRP should shield the current parent and reselect other neighbor as its next-hop node.

4.1 Loop Detection

Tracing the paths along which the packets move and monitoring the events occurring in the networks, we find that the routing loops caused by EBRP can be classified into three types. Subsequently, we introduce their symptoms and develop the corresponding loop detection mechanisms, respectively.

1. *One-hop-loop*. One-hop-loop occurs between a local node and its parent. In Fig. 8, two nodes in area 3 select each other as their parents, which is a typical one-hop-loop. This loop can be easily detected by checking the source address embedded in the header of received packets. The source address of a packet is ID of the former node sending this packet, and changes on each hop. After receiving a packet, if the local node finds that the source address is identical with ID of its parent, then it can confirm that an one-hop-loop occurred.
2. *Origin-loop*. The distinct feature of this routing loop is that it must involve one or more sampling nodes. Therefore, we call it origin-loop. This loop chain itself may be one-hop or multihop. In Fig. 8, three nodes in area 4 form an origin-loop chain back-to-back. This loop can be detected by checking the

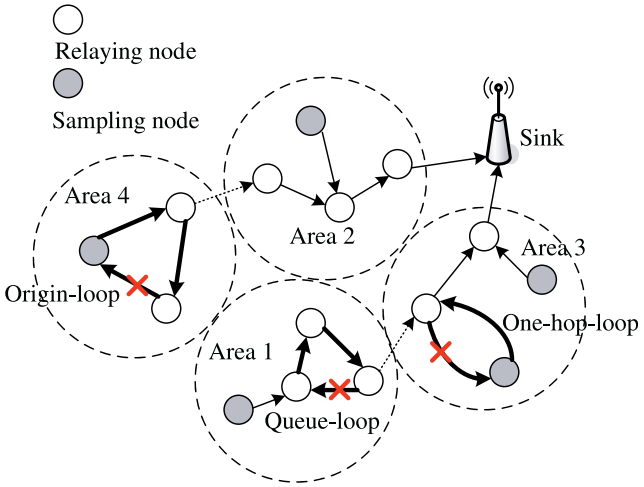

Fig. 8. Types of routing loops.

origin address carried in the header of packets. The origin address of a packet is ID of the node generating this packet. There is only one origin address for each packet, which is different from the source address. After receiving a packet, the local node checks whether the origin address in the packet header is the same as its ID. If so, the local node can infer that there is an origin-loop in the networks.

3. *Queue-loop*. This is a special multihop loop chain. It does not involve any sampling nodes; all nodes consisting of the loop chain are relaying nodes. In Fig. 8, one routing loop falling into this category appears in area 1. They cannot be properly detected by checking both origin and source addresses. However, we can still identify it. Because packets cannot go out of this routing loop, the queue of the nodes in the chain will grow drastically. This phenomenon will be an obvious symptom of this loop occurring. Thus, we call it "queue-loop." To detect the queue-loop, EBRP needs to monitor the length of the local queue and check whether it is over a certain threshold in a rather short time, but it is insufficient to identify the queue-loop only referring to the variance of the queue length because the congestion is likely to be mistaken as a queue-loop. Therefore, when a symptom of queue-loop by monitoring the queue length is detected, any relaying nodes in the loop chain need to generate and send a CLP packet (defined in Section 3.5.1) to confirm its occurrence. It is noticeable that if the threshold of detecting a queue-loop is too small, the detection mechanism will be very sensitive, and excessive CLP packets could severely disturb the transmission of data packets. However, if the threshold is too large, the detection mechanism will be unresponsive. In practice, to make a trade-off, the feasible parameter values can be obtained through experiments. In our implementation, the symptom of queue-loop occurrence is that the queue length increases by 20 percent of buffer size in a LUI.

4.2 Loop Elimination

Once the routing loops are confirmed, it will be straightforward to eliminate them by cutting off the loop chain. EBRP does it by cutting the links belonging to the loop chain. However, it is unnecessary to cut off all the links; EBRP will preserve those that make packets move closer to the sink. Thus, once a loop is detected, and if the parent is not closer to the sink and there is at least one other active neighbor besides the current parent, EBRP will shield this current parent by cutting off the link between the local node and its current parent.

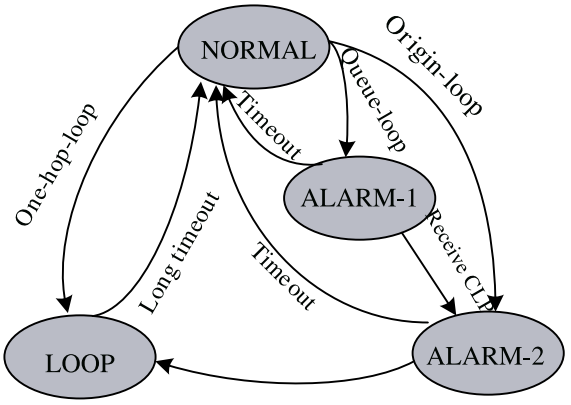

Fig. 9. State transition diagram.

4.3 State Transition

Combining with various elements in above two sections, we summarize the state transition in loop detection and elimination mechanism in Fig. 9.

Originally, EBRP is in state **"normal"** in which the node acts normally. The three different loop indicators push EBRP to one of three abnormal states: **"loop"** denotes the one-hop-loop indicator, **"alarm-1"** is the queue-loop indicator, and **"alarm-2"** is the origin-loop indicator. Once **"loop"** state is confirmed, EBRP will shield the current parent and reselect a new one. When the system is in **"alarm-1"** state, EBRP will send one CLP to check whether the loop really exists. If this CLP is received by the local node in the future, the queue-loop has been converted into origin-loop. EBRP goes to **"loop"** state from **"alarm-2"** state without any additional conditions.

Once entering any one of the abnormal states, EBRP would start a timer to prevent sticking in one state too long. **"alarm-1"** and **"alarm-2"** will wait for a time out, and **"loop"** will wait for another relatively long time out before returning to **"normal."** Since the depth potential field forces packets to move toward the sink, the routing loops always occur in the areas far away from the sink where the strength of the depth potential field is quite weak. Therefore, we set the values of the two timers according to the depth of the local node, namely the smaller one is set three times of the depth while the larger timer is set six times. Especially, here the time unit is a MUI, not a second.

To avoid possible confusion, we use Loop Detection and Elimination (EBRP-LDE) to denote the EBRP algorithm with this loop detection and elimination mechanism.

5 Validation

In this section, we validate the basic EBRP using simulation experiments conducted on TOSSIM platform [43] built in

TinyOS. As mentioned earlier, there has been numerous work to balance energy consumption through various mechanisms, such as deployment optimization, topology control, and mobile sink. However, to the best of our knowledge, little work focuses on developing routing algorithm to balance energy consumption, most energy-aware routing schemes pay much attention to energy efficiency. Therefore, we only choose MintRoute, a standard energy-efficient routing algorithm, as the reference protocol.

In the simulation experiments, a simple linear energy consumption model is used. The energy consumed by sending or receiving a packet is a monotonically increasing function of the lasting time. We assume that the length of all the packets is the same, thus the energy consumed by sending or receiving a packet is a constant value. Since sending a packet always needs more energy than receiving ones [44], without loss of generality, we assume 3 units of energy consumed for sending, and 2 units for receiving.

To evaluate the performance of the routing protocol separately, it is feasible to orhogonalize the network layer and the MAC layer. Therefore, we abstract away the MAC layer by providing for direct packets transfer from the network layer of one node to the network layer of its neighbors, i.e., the link quality is assumed perfect, thus MintRoute actually works only on the hops, just like the shortest path routing algorithm.

5.1 Performance Metrics

To make a comprehensive performance evaluation, we first define several measurable metrics.

1. **Energy Imbalance Factor (EIF)**. We define this metric to quantify the energy balance characteristic of the routing protocol. It is formally defined as the standard variance of the residual energy of all nodes,

$$EIF = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(RE_i - RE_{avg})^2},$$

where n is the total number of nodes, RE_i is the residual energy on node i, and RE_{avg} is the average residual energy of all nodes.

2. **Portion of Living Nodes (PLN) and Receiving Packets Ratio (RPR)**. We can use PLN as a metric to evaluate the influence of energy consumption on the performance. Otherwise, we have noticed that if the sink can receive most of the packets sent from the source nodes, which implies that the energy holes are effectively avoided and the connectivity of the network is good. We can define the *receiving packets ratio* as the ratio of the actual rate that the sink receives packets to the expected receiving rate by the task, and use both PLN and RPR to measure the gain due to energy consumption balance.
3. **Network lifetime** [45]. The *network lifetime* of a sensor network is defined as the time when the first energy exhausted node (First Dead Node, FDN) appears. The network lifetime is closely related to the network partition and network coverage ratio. If nodes begin to die, the probability of *network partition* increases, and the *network coverage ratio* might reduce.

TABLE 1
Parameter Configuration: Grid

Deployment	Deployment Area	70 m× 70m Square
	Deployment Type	17 × 17 Grid
	Network Architecture	Homogeneous, Flat
	Number of Nodes	289
	Sink	(35, 35) m
	Node	Total energy 10000, Radio range 6m
Task	Application Type	Event driven
	Event Area	Three circles whose radii are 8m and centers are (8,8), (13,8) and (59,59)m.
	Event Lasting Time	From 100 sec to 8000 sec
	Number of Data Source	21
	Sampling Rate	0.25 packet/s
EBRP	(α, β)	(0.2, 0.2)
	MUI	10 seconds
	LUI	5 seconds
Simulation	Time	8000 seconds

4. **Functional lifetime** [46]. The *functional lifetime* of a task is defined as the amount of time that the task is perfectly carried out. Different tasks have different requirements. Some tasks may require no node failure while some others just need a portion of nodes be alive, therefore the function lifetime varies much according to task requirements. In our simulation experiments, we assume that the application requires all sampling nodes be alive, namely, the functional lifetime is defined as the interval between the beginning of task and the appearance of the First Dead Sampling Node (FDSN).
5. **Functional Throughput (FT)**. Functional throughput is defined as the number of packets received by the sink during the functional lifetime. For a given application, FT is mainly influenced by the length of the functional lifetime and RPR.
6. **Energy Consumption per Received Packet (ECRP)**. The average consumed energy per packet received by the sink during the network lifetime or the functional lifetime reflects the energy efficiency of the protocol.

5.2 Simulation Setup

We conduct the simulation experiments using a 17×17 grid network (totally 289 nodes, each residing on the intersection point of 17 rows and 17 columns) to validate and evaluate the performance of our EBRP and EBRP-LDE schemes and compare them with MintRoute algorithm. In this special topology, a node can only communicate with its direct eight neighbors. The node can act as either a sampling node or a relaying node depending on the requirements. Events occur in three circle areas whose radii are 8m and the centers are $(8, 8)$m, $(13, 8)$m, and $(59, 59)$m, respectively. The nodes in the event areas can execute sampling and relaying tasks. The parameter configuration is listed in Table 1. The same simulation is repeated 10 times, the average values of the performance metrics and their standard deviation are also calculated.

TABLE 2
Statistics of Performance Metrics

Item	MintRoute		EBRP		EBRP-LDE	
	Average	Std. dev.	Average	Std. dev.	Average	Std. dev.
EIF	183.4	3.03	143.7	11.37	144.8	8.67
FDN(sec)	2829.5	236.9	6242.8	23.0	6236.0	37.7
FDSN(sec)	4377.2	239.3	7780.8	66.3	7801.1	36.2
FT(pkts)	7585.5	549.2	15921.0	0.1	16120.9	0.3

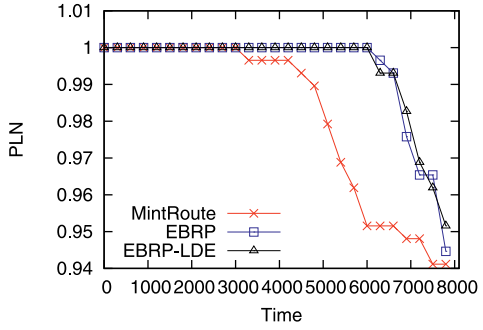

Fig. 10. Portion of living nodes.

5.3 Simulation Results

MintRoute always chooses the shortest path, thus it will burn out the energy of nodes on that path quickly. However, EBRP will choose another path through other areas with more energy once it finds out that the energy density in this area is lower than that in other areas nearby. Therefore, EBRP can improve the energy consumption balance across the network and prolong the *network lifetime* as well as the *functional lifetime*. The statistical results are listed in Table 2. The EIF value decreases by 21.6 percent, at the same time EBRP prolongs the time of FDN and FDSN by 120.6 and 77.8 percent, respectively. The *functional throughput* is also improved. The statistics listed in Table 2 show an improvement by 109.9 percent because the EBRP keeps relatively higher PLN and RPR than the MintRoute, as well as a longer FDSN. In addition, the small standard deviations of some important performance metrics, except for EIF, indicate that EBRP is robust against noise and random factors and can maintain a consistent performance.

Fig. 10 shows the portion of living nodes as a function of time in one simulation. The time that the first dead node appears in EBRP is much later than that in MintRoute, i.e., about 6,000 and 3,000 s, respectively (also listed in Table 2). Moreover, the PLN of EBRP is always higher than that of MintRoute. The receiving packet ratio is depicted in Fig. 11. Clearly, the period with both high PLN and large RPR in EBRP is much longer than that in MintRoute. From these results, we can conclude that this gain can be obtained through the EBRP's energy consumption balance.

The integrity of the data received in EBRP is much better than that in MintRoute since there is fewer packets loss in EBRP. The RPR of the EBRP scheme stays tightly around 1 before FSDN, which implies that most of the packets have been successfully received.

Fig. 12 describes the average energy consumed per received packet (ECRP). Unexpectedly, the energy efficiency

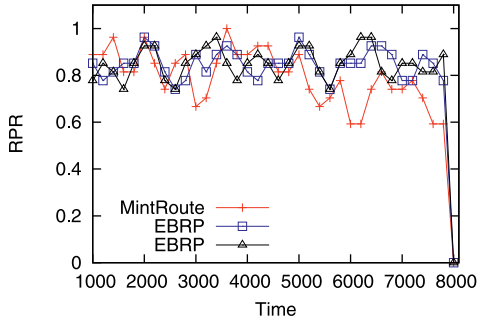

Fig. 11. Receiving packets ratio.

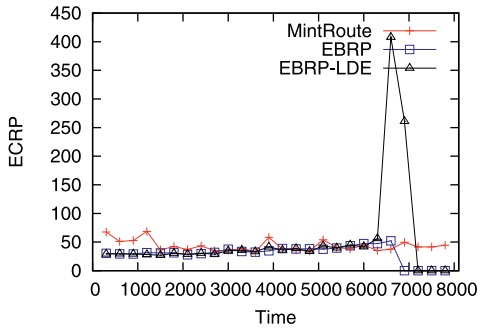

Fig. 12. ECRP.

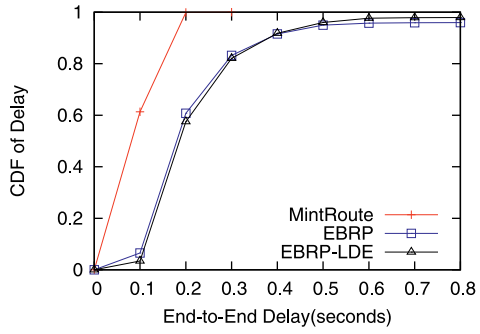

Fig. 13. CDF of end-to-end delay.

of EBRP is almost the same as MintRoute before FSDN, and there are only a few tips when EBRP switches to an optional path. Otherwise, it is useful that EBRP switches its working routes among several normal areas, and makes packets flow along multiple paths. This means that EBRP can alleviate the congestion in the shortest path, and restrain energy waste due to packet losses. On the other hand, energy efficiency is deteriorated in EBRP after FSDN since its ECRP abruptly climbs up. However, it is meaningless to keep high energy efficiency after FSDN because the network has been partitioned and some tasks in the network cannot be carried out perfectly.

Compared with the shortest path routing algorithm MintRoute, EBRP likely makes a detour with the purpose of balancing energy, which implies that the end-to-end delay increases. The cumulative distribution function (CDF) of end-to-end delay presented in Fig. 13 confirms this prediction.

From the above simulation results and analysis, we conclude that it is valuable to balance energy consumption while designing a routing protocol. Fig. 14 illustrates the residual energy on nodes at time 5,000 s across the whole network. EBRP evenly burns energy so that the contour 8,300,

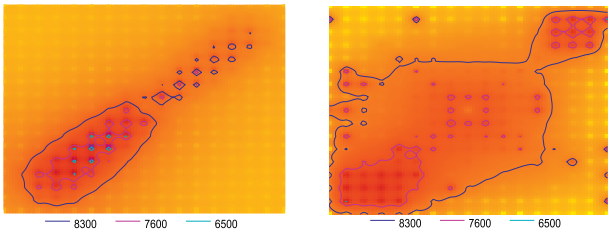

Fig. 14. Residual Energy at 5,000 s. (a) MintRoute. (b) EBRP.

denoting the residual energy on the nodes is 8,300, covers wider area, but the contour 6,500 does not appear in Fig. 14b. On the contrary, in Fig. 14a, the contour 8,300 covers small region, and there are many areas covered by the contour 6,500, which readily become energy holes. The results presented in Fig. 14 vividly indicate that EBRP is capable of balancing energy consumption and avoiding the energy holes occurring frequently in the MintRoute algorithm.

In this simulation, we also verify that EBRP suffers from the routing loop problem. There are 9,650 times loop occurrences up to 6,230 s (FDN) .

Subsequently, we validate the loop detection and elimination mechanism.

The same simulation is conducted again. The results show that the number of loops decreases from 9,650 to 178 during FDN, and from 98,037 to 1,135 during the whole simulation time. FSDN and the functional throughput listed in Table 2 have slight improvement. The corresponding curves are inserted into Figs. 10 and 11, respectively. Observing Fig. 10, there are more living nodes at the end of the simulation when EBRP-LDE is employed, and RPR shown in Fig. 11 is more stable comparing with the basic EBRP. The curve describing the energy consumed per received packet is also appended in Fig. 12. Obviously, the curve of EBRP-LDE is smoother than that of EBRP in most cases. Finally, although it is foreseeable that the loop detection and elimination mechanism may harm the performance of energy balance because it shields the energy-balanced paths, the numerical results listed in Table 2 verify that EIF has a small change. This implies that the impact of the enhanced mechanism on the energy consumption balance is rather trivial.

6 Integrated Performance Evaluation and Impact of Parameters

In this section, we will comprehensively evaluate the integrated performance of the full energy-balanced routing algorithm with the help of numerous simulation experiments on a randomly deployed network for general parameter configuration.

6.1 Simulation Setup

Fig. 15 shows a randomly deployed network and an event area. All 568 nodes spreading over a circle with radii 500 m form a flat multihop network. There is only one sink residing at the center, and the radio range is 20 m. At time 100 s, an event occurs in the event area, and then the nodes in this area begin to sample and send one packet every 5 s. The detailed configuration is summarized in Table 3.

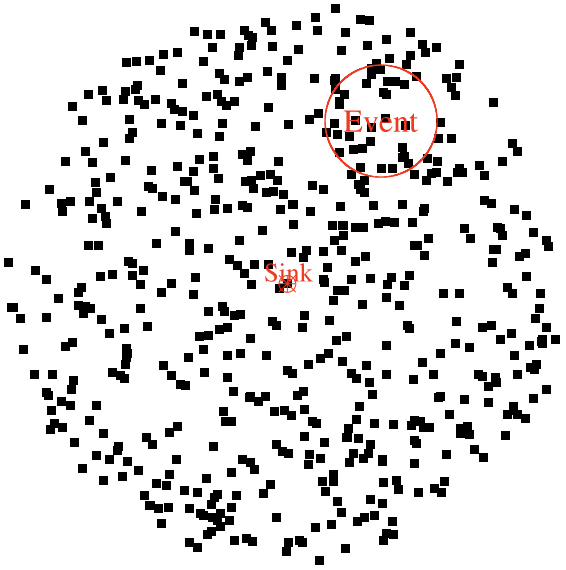

Fig. 15. Randomly deployed network of sensors.

6.2 Simulation Results

The simulation results listed in Table 4 show that there are obvious improvements in the performance of balancing energy consumption, network lifetime, functional lifetime, and functional throughput. We also notice that EBRP-LDE restrains the routing loops, but EIF slightly increases as compared with the basic EBRP algorithm. It is foreseeable because EBRP-LDE blocks the paths on which there may be routing loops, but these paths may be favorable for energy consumption balance.

From the curves of PLN and RPR in Figs. 16 and 17, we observe that the energy-balanced routing protocol can improve the system performance. It is noted that EBRP both with and without LDE suffer from a sudden drop of PLN near the time of their FSDN (about 8,000 s), and their final number of living nodes is also less than MintRoute.

TABLE 3
Configuration of Parameters: Random Deployment

Deployment	Deployment Area	Circle, Radius = 500m
	Deployment Type	Random
	Network Architecture	Homogeneous, Flat
	Number of Nodes	568
	Sink	(0, 0)m
	Node	Total Energy 10000, Wave Range 20m
Task	Application Type	Event drive
	Event Area	Circle with Radii 80m, at (167, 167)m
	Lasting Time of Event	From 100s to 17000s
	Number of Data Sources	25
	Sampling Rate	0.2 packet/s
EBRP-LDE	(α, β)	(0.2, 0.2)
	MUI	10 seconds
	LUI	5 seconds
Simulation	Time	17000 seconds

TABLE 4
Statistic of Performance Metrics

	MintRoute	EBRP	EBRP-LDE
EIF	153.1	14.8	18.9
FDN(sec)	992	6569	6672
FDSN(sec)	3346	8132	8221
FT(pkts)	15470	37601	37743

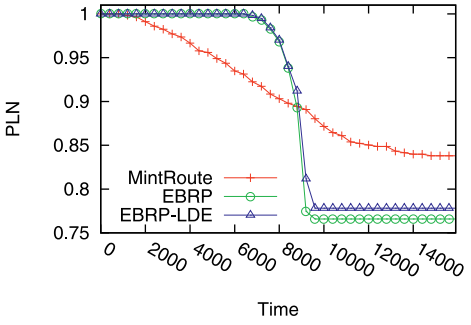

Fig. 16. Portion of living nodes.

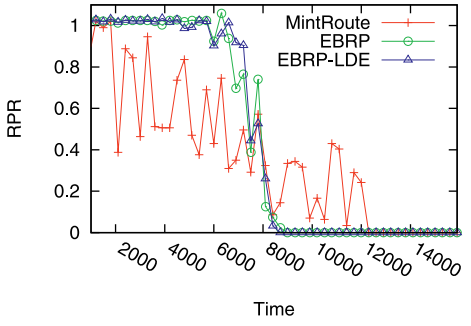

Fig. 17. Receiving packets ratio.

The reason is that EBRP will involve more nodes near the event area and eventually deplete their energy because of its energy balance property. This sudden drop also implies that balancing energy consumption among those involved nodes is reasonable.

The energy efficiency of EBRP is better than that of MintRoute which exhibits a series of serious fluctuations as shown in Fig. 18. The reason is that MintRoute must choose another path when the current route is broken off because of individual disable nodes, moreover the randomness of deployment increases the difficulty to find a new one. Before it finds the new path, the energy efficiency may be very low. However, EBRP acts much better, and the reasons are similar with that presented in Section 5.3. So far, we have evidences to conclude that EBRP can improve the integrated performance even on an arbitrarily deployment.

Next, we will check the impact of parameters on the performance and recommend some proper values of key parameter settings through numerous experiments. Without particular statements, EBRP-LDE is employed in subsequent experiments, and for the sake of brevity, EBRP will refer to EBRP-LDE in general.

6.3 Impact of Parameters

6.3.1 *Weight Factors* (α, β)

We select five groups of parameters to evaluate their impact on the performance, and their settings are listed in Table 5.

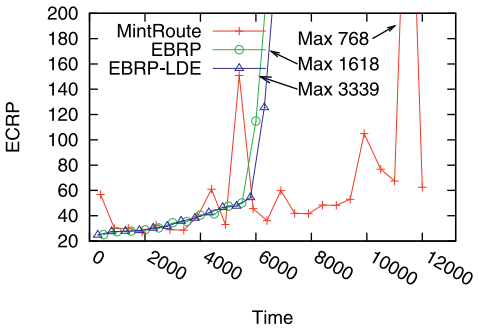

Fig. 18. ECRP.

TABLE 5
Parameter Pairs (α, β)

Groups	EBRP-LDE	(α, β)
1	EBRP0	$(0.0, 0.0)$
2	EBRP1-2	$(0.2, 0.0), (0.0, 0.2)$
3	EBRP3-4	$(0.7, 0.0), (0.0, 0.7)$
4	EBRP5-6	$(0.5, 0.1), (0.1, 0.5)$
5	EBRP7-8	$(0.2, 0.2), (0.4, 0.4)$

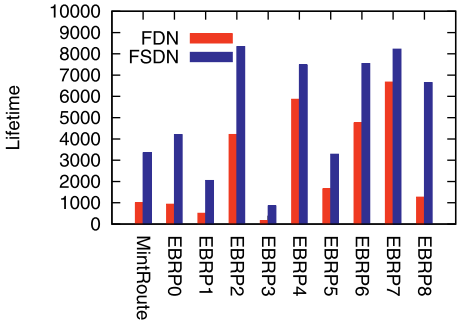

Fig. 19. FDN and FSDN.

We refer to them as EBRP 0-8, respectively. EBRP0 is the first group, in which both α and β equal to zero. Here only the depth field is effective, which implies EBRP degenerates to the shortest path algorithm.

In the second group, only one of α and β is effective, which is also relatively small. To examine the impact of energy density and residual energy field separately, only α or β is set to a comparatively high value in the third group. The different combinations of parameter pair (α, β) are chosen in the fourth and fifth groups to investigate the impact of parameters on the comprehensive performance.

Lifetime. Fig. 19 shows the network lifetime (FND) and the functional lifetimes (FSND) with different parameters. EBRP0 is very close to MintRoute, because both of them are similar to the shortest path algorithm. Both EBRP1 and EBRP3 only take the energy density factor into consideration, and their performance is much worse than that of the others because low energy nodes in the high energy density area are easy to die when plenty of data packets flow through. Since the residual energy potential field will especially protect the low energy nodes, EBRP2 and EBRP4 perform well. Otherwise, from the comparison of EBRP1 and EBRP3, EBRP5 and EBRP7, we can conclude that the higher weight of energy density always results in worse

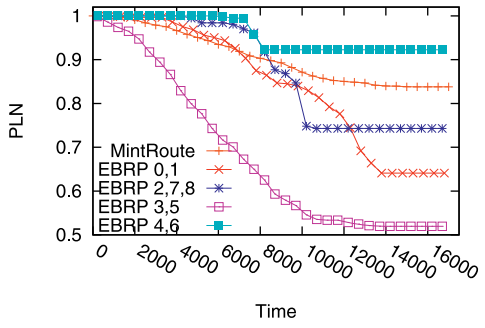

Fig. 20. Portion of living nodes.

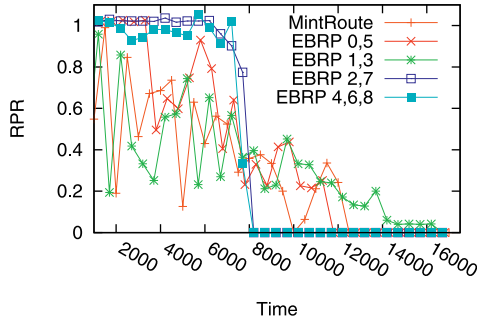

Fig. 21. Receiving packets ratio.

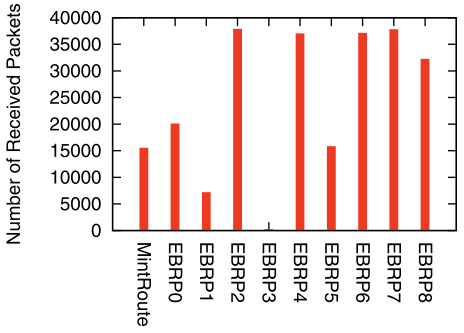

Fig. 22. Functional throughput.

performance. However, from the comparison of EBRP2 and EBRP4, EBRP6 and EBRP8, the higher weight of residual energy provides better performance.

PLN and RPR. Fig. 20 gives the portion of living nodes along time. Since the curves of EBRP0 and EBRP1 are quite similar, we just use one curve to show their basic shapes in the figure. We also do the same thing for EBRP3 and EBRP5, EBRP4 and EBRP6, EBRP2, EBRP7, and EBRP8, respectively. In Fig. 21, the curves depict the receiving packets ratio, we also combine the curves of EBRP0 and EBRP5, EBRP1 and EBRP3, EBRP2 and EBRP7, EBRP4, EBRP6, and EBRP8 for the same reason. Observing Figs. 20 and 21, we can conclude that most of the EBRPs, except for EBRP1, EBRP3, and EBRP5, have a higher PLN as well as a higher RPR than MintRoute before both FDN and FSDN. We have also noticed that every time PLN declines, MintRoute and some EBRPs suffer a fluctuation on RPR, which is because they need to select a new path when some nodes on the current path die. Particulary, we also notice that PLN in both EBRP2 and EBRP7 keep relatively stable, which implies EBRP under such parameter settings can effectively protect the low energy nodes and sufficiently utilize the redundant network resources to avoid the path failures.

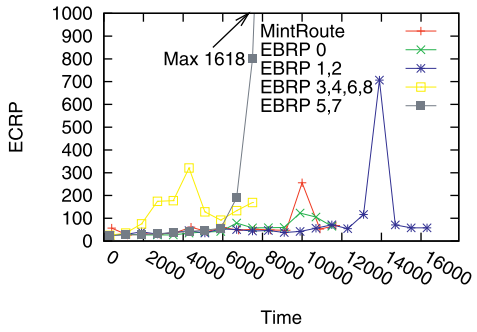

Fig. 23. ECRP.

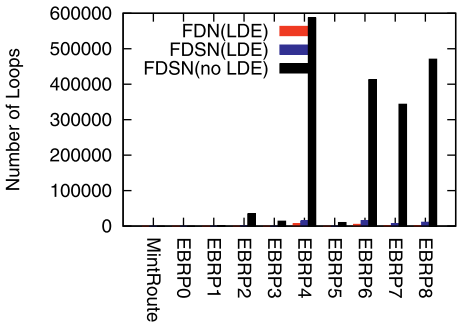

Fig. 24. Num. of loop occur.

Functional throughput. Prolonging the functional lifetime always means increasing the throughput. As aforementioned, RPR can also reflect the functional throughput. RPR and functional throughput are presented in Figs. 21 and 22, respectively. Obviously, the simulation results comply with our analysis, EBRP2, EBRP4, EBRP6, EBRP7, and EBRP8 have high functional throughput and large RPR simultaneously. Combining with FDN and FSDN as shown in Fig. 19, the conclusion is consistent, namely the longer FSDN renders the higher functional throughput.

Energy efficiency. Intuitively, the shortest path algorithm can save more energy. The average energy consumed per received packet is presented in Fig. 23. Notice that the same combination is made for similar EBRPs. EBRP0 is the lowest, and EBRP1, EBRP2, and EBRP7 with small parameters are quite close to EBRP0. In other words, the smaller parameters provide with higher energy efficiency.

Number of Loops. Theoretically, the dynamics of hybrid virtual potential results in routing loops. The energy density field related to the deployment is relatively stable, so it is sensitive to loops caused by network topology, but the residual energy may dynamically vary and also induces loops. Fig. 24 depicts the number of loop occurrences until FDN and FSDN reach. Obviously, our simple loop detection and elimination mechanism is effective, and loops in EBRP1, EBRP3, and EBRP5 are completely eliminated. However, it is impossible to absolutely avoid the loops in EBRP2, EBRP4, and EBRP6, moreover the larger the parameter β is, the more the loop appears. Since the large β implies that the potential field will be dominated by residual energy on nodes, and the slight difference in residual energy on different nodes will be magnified, eventually some local "hills" emerge in the hybrid potential field. If the packets are attracted by these local "hills," the routing loops naturally appear.

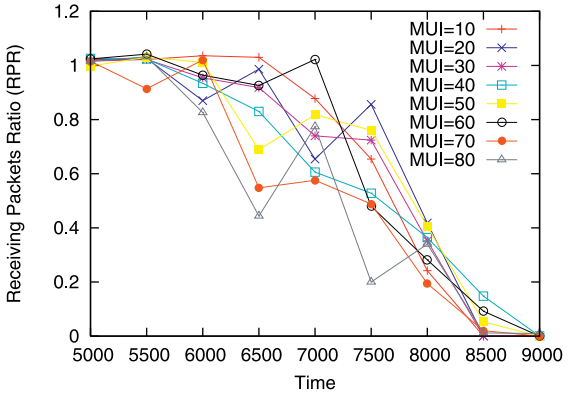

Fig. 25. A piece of RPR curve.

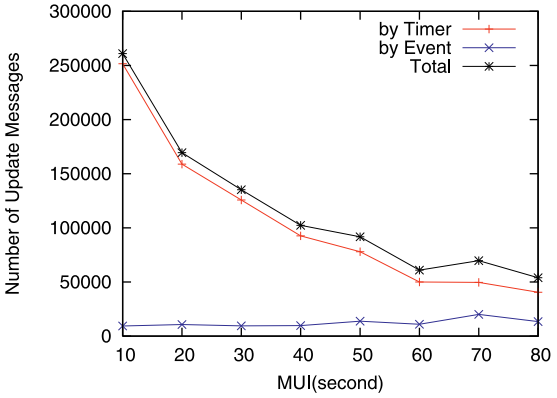

Fig. 26. Num. of update packets during FSDN.

Proper parameter pair (α, β). The above experimental results demonstrate that different performance metrics are inclined to diverse parameter settings with respect to α and β. Even some inclinations are incompatible, such as lifetime and coverage are prone to large β, but the small parameters are beneficial to improving energy efficiency and decreasing the number of routing loops. To optimize the integrated performance by proper parameter settings, it is rational to make a trade-off. Because the depth field forms the basic routing backbone, and because the energy density and the residual energy fields act like its assistants who push the packets to pass through high energy areas and nodes on the appropriate way to the sink, it is reasonable to properly augment the weight of the depth field and let it be a dominant element in the hybrid potential field, i.e., $\alpha + \beta < 0.5$. This should be one of the practical guidelines toward parameter configuration. In Table 5, all parameter pairs whose sum is more than 0.5 show the poor integrated performance. This fact can verify our analysis. In addition, the configurations with only effective α or β are hard to provide a satisfactory integrated performance even if the value of α or β is very small (such as EBRP1 and EBRP2). Taking the above various factors into account, a recommendable parameter configuration solution is to choose small α and β at the same time, such as $(0.2, 0.2)$. The favorable integrated performance exhibited by EBRP7 can manifest this judgement. Moreover, we have also repeated the numerous similar experiments with various network sizes and different parameter values. The results confirm that this basic guidelines toward the parameter settings can be generally adopted.

6.3.2 MUI

Several parameters are related to MUI, which is the maximum interval between two successive update messages. As mentioned in Section 3.5, if a node does not receive any update message from one of its neighbors in one MUI, it will wait another MUI before considering this neighbor to be dead. If MUI is quite large, it will take a long time to detect this dead neighbor and the topology changes. But if MUI is too small, the overhead of exchanging update messages would be unaffordable. Therefore, MUI is tightly associated with fast detection of topology changes and energy efficiency. Otherwise, the LDE mechanism uses the product of the MUI and the depth of local node to recover from the abnormal state. It is necessary to evaluate the impact of MUI on the performances. The parameter pair (α, β) are set to the proper value, i.e., $(0.2, 0.2)$.

We can determine the EBRP's sensitivity to topology changes through checking RPR. If the RPR suffers a fluctuation as a node dies, the sensitivity would be weak, and vice versa. A piece of RPR curve around the FDN (e.g., 6,000 s) is drawn in Fig. 25. From these curves, we can conclude that the smaller the MUI is, the more sensitive is the EPRP.

In addition, Fig. 26 shows the number of update packets during FSDN, which represents the overhead caused by the routing update. It does not always decrease as MUI increases because there are other factors which result in sending an update message, e.g., the events of topology change and the residual energy consumed by 1 percent, etc. When the MUI is quite large ($\geq$60), the number of update messages caused by the events increases but the total number is almost the same, which indicates that the update messages caused by the MUI time-out are not enough to provide sufficient routing information. Based on these understanding and observations, the proper choice of MUI is about 60 s.

7 Conclusions

Energy is one of the most critical resources for WSNs. Most of works in the literatures about WSN routing have emphasized energy savings as an important optimization goal. However, merely saving energy is not enough to effectively prolong the network lifetime. The uneven energy depletion often results in network partition and low coverage ratio, which deteriorate the performance.

This paper focuses on routing that also balances the energy consumption. Its main contributions are: 1) Borrow the concept of potential in classical physics to build a virtual hybrid potential field to drive packets to move toward the sink through the dense energy area and steer clear of the nodes with low residual energy so that the energy is consumed as evenly as possible in any given arbitrary network deployment, 2) Classify the routing loops and devise an enhanced mechanism to detect and eliminate loops. Our numerous simulation results show that the proposed solution EBRP makes significant improvements in energy consumption balance, network lifetime, and throughput as compared to the commonly used energy-efficient routing algorithm. EBRP belongs to the class of data-gathering based routing algorithm, and does not deal with data dissemination and point-to-point communication.

In other words, EBRP will only find routes for each data source to the same sink. The limitation is also related to the lack of sufficient understanding about the dynamics of time-varying potential field. If the necessary analytical models could be built, the perfect weights of the hybrid field (i.e., α and β) could be obtained through theoretical analysis, or be adapted well to the dynamics of potential field. Hence the routing loops could also be further restrained and the integrated performance could be improved further. We will continue our investigation in this challenging direction as part of our future work.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the anonymous reviewers for their constructive comments which helped significantly improve the quality of the manuscript. This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 60773138, 60971102, the National Grand Fundamental Research Program of China (973) under Grant No. 2009CB320504, and the National Science and Technology Major Project of China (NSTMP) under Grant Nos. 2009ZX03006-001-003 and 2009ZX03006-003-01. The work of Sajal K. Das is partially supported by the US National Science Foundation (NSF) grants under Award Nos. IIS-0326505, CNS-0721951, and CNS-0916221. The work of Sajal K. Das is also supported by (while serving at) the NSF. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM/IEEE MobiCom,* pp. 263-270, 1999.

[2] J. Evans, D. Raychaudhuri, and S. Paul, "Overview of Wireless, Mobile and Sensor Networks in GENI," *GENI Design Document 06-14, Wireless Working Group,* http://www.geni.net/documents.html, 2006.

[3] S. Olariu and I. Stojmenović, "Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting," *Proc. IEEE INFOCOM,* 2006.

[4] A. Wadaa, S. Olariu, L. Wilson, K. Jones, and M. Eltoweissy, "On Training a Sensor Networks," *Proc. Parallel and Distributed Processing Symp.,* 2003.

[5] X. Wu and G. Chen, and S.K. Das, "Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution," vol. 19, no. 5, pp. 710-720, 2008.

[6] J. Lian, K. Naik, and G. Agnew, "Data Capacity Improvement of Wireless Sensor Networks Using Non-Uniform Sensor Distribution," *Int'l J. Distributed Sensor Networks,* vol. 2, no. 2, pp. 121-145, 2006.

[7] J. Pan, Y.T. Hou, L. Cai, Y. Shi, and S.X. Shen, "Topology Control for Wireless Sensor Networks," *Proc. ACM MobiCom,* pp. 286-299, 2003.

[8] X. Wang and T. Berger, "Topology Control, Resources Allocation and Routing in Wireless Sensor Networks," *Proc. IEEE CS 12th Ann. Symp. Modeling, Analysis, and Simulation of Computer and Telecomm. Systems (MASCOTS '04),* pp. 391-399, 2004.

[9] H.M. Ammari and S.K. Das, "Promoting Heterogeneity, Mobility and Energy-Aware Voronoi Diagram in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 19, no. 7, pp. 995-1008, July 2008.

[10] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data Mules: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *Proc. IEEE Workshop Sensor Network Protocols and Applications (SNPA),* 2003.

[11] W. Wang, V. Srinivasan, and K.-C. Chua, "Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks," *Proc. ACM MobiCom,* 2005.

[12] J. Luo and J.P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* 2005.

[13] M. Haenggi, "Energy-Balancing Strategies for Wireless Sensor Networks," *Proc. 2003 Int'l Symp. Circuits and Systems (ISCAS),* pp. 828-831, 2003.

[14] J. Li and P. Mohapatra, "Analytical Modeling and Mitigation Techniques for Energy Hole Problem in Sensor Networks," *Pervasive and Mobile Computing,* vol. 3, pp. 233-254, 2007.

[15] H. Zhang and H. Shen, "Balancing Energy Consumption to Maximize Network Lifetime in Data-Gathering Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 20, no. 10, pp. 1526-1539, Oct. 2009.

[16] D.H. Armitage and S.J. Gardiner, *Classical Potential Theory.* Springer, 2001.

[17] Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad-Hoc Routing," *Proc. ACM MobiCom,* 2001.

[18] V. Rodoplu and T.H. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 8, pp. 1333-1344, Aug. 1999.

[19] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proc. ACM MobiCom,* 1999.

[20] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *Proc. Hawaiian Int'l Conf. Systems Science,* 2000.

[21] A. Boukerche, X. Cheng, and J. Linus, "Energy-Aware Data-Centric Routing in Microsensor Networks," *Proc. Sixth ACM Int'l Workshop Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM' 03),* pp. 42-49, 2003.

[22] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.

[23] M. Singh and V. Prasanna, "Energy-Optimal and Energy-Balanced Sorting in a Single-Hop Wireless Sensor Network," *Proc. First IEEE Int'l Conf. Pervasive Computing and Comm.,* 2003.

[24] R.C. Shah and J.M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC),* pp. 350-355, 2002.

[25] S.J. Baek and G. de Veciana, "Spatial Energy Balancing Through Proactive Multipath Routing in Wireless Multihop Networks," *IEEE/ACM Trans. Networking,* vol. 15, no. 1, pp. 93-104, Feb. 2007.

[26] C. Efthymiou, S. Nikoletseas, and J. Rolim, "Energy Balanced Data Propagation in Wireless Sensor Networks," *Wireless Networks,* vol. 12, no. 6, pp. 691-707, 2006.

[27] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom,* 2000.

[28] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," *Int'l J. High Performance Computing Applications,* vol. 16, no. 3, pp. 90-110, 2002.

[29] J. Faruque and A. Helmy, "Gradient-Based Routing in Sensor Networks," *Proc. ACM MobiCom,* 2003.

[30] J. Faruque and A. Helmy, "RUGGED: Routing on Fingerprint Gradients in Sensor Networks," *Proc. IEEE Int'l Conf. Pervasive Services (ICPS),* pp. 179-188, 2004.

[31] J. Liu, F. Zhao, and D. Petrovic, "Information-Directed Routing in Ad Hoc Sensor Networks," *IEEE J. Selected Areas in Comm.,* vol. 23, no. 4, pp. 851-861, Apr. 2005.

[32] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L.J. Guibas, "Composable Information Gradients in Wireless Sensor Networks," *Proc. Seventh Int'l Conf. Information Processing in Sensor Networks (IPSN),* pp. 121-132, 2008.

[33] C. Schurgers and M. Srivastava, "Energy Efficient Routing in Wireless Sensor Networks," *Proc. Military Comm. Conf. (MILCOM),* 2001.

[34] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *ACM Wireless Networks,* vol. 11, pp. 285-298, 2005.

[35] P. Huang, H. Chen, G. Xing, and Y. Tan, "SGF: A State-Free Gradient-Based Forwarding Protocol for Wireless Sensor Networks," *ACM Trans. Sensor Networks,* vol. 5, no. 2, pp. 1-25, 2009.

[36] A. Basu, A. Lin, and S. Ramanathan, "Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm," *Proc. ACM SIGCOMM,* pp. 37-48, 2003.
[37] H. Huang, T. Chang, S. Hu, and P. Huang, "Magnetic Diffusion: Scalability, Reliability, and QoS of Data Dissemination Mechanisms for Wireless Sensor Networks," *Computer Comm.,* vol. 29, nos. 13/14, pp. 2482-2493, 2006.
[38] "Crossbow Technologies," http://www.xbow.com, 2011.
[39] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (ISPN '05),* pp. 364-369, 2005.
[40] "Project Sun Spot," http://www.sunspotworld.com, 2011.
[41] R. Musunuri and J.A. Cobb, "Hierarchical-Battery Aware Routing in Wireless Sensor Networks," *Proc. IEEE 62nd Vehicular Technology Conf. (VTC-2005-Fall),* pp. 2311-2315, 2005.
[42] N. Patwari, A.O. Hero III, M. Perkins, N. Correal, and R.J. O'Dea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. Signal Processing,* vol. 51, no. 8, pp. 2137-2148, Aug. 2003.
[43] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Proc. ACM First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03),* pp. 126-137, 2003.
[44] A. Papadoulos and J.A. Mccann, "Towards the Design of an Energy-Efficient, Location-Aware Routing Protocol for Mobile, Ad-Hoc Sensor Networks," *Proc. 15th Int'l Workshop Database and Expert Systems Applications,* pp. 705-709, 2004.
[45] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Networking (ICN),* pp. 685-696, 2002.
[46] A. Giridhar and P.R. Kumar, "Maximizing the Functional Lifetime of Sensor Networks," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks,* 2005.

Fengyuan Ren received the BA and MSc degrees in automatic control from Northwestern Polytechnic University, China, in 1993 and 1996, respectively, and the PhD degree in computer science from Northwestern Polytechnic University in December 1999. He is a professor of the Department of Computer Science and Technology at Tsinghua University, Beijing, China. From 2000 to 2001, he worked at Electronic Engineering Department of Tsinghua University as a postdoctoral researcher. In January 2002, he moved to the Computer Science and Technology Department of Tsinghua University. His research interests include network traffic management and control, control in/over computer networks, wireless networks, and wireless sensor networks. He authored and coauthored more than 80 international journal and conference papers. He is a member of the IEEE, and has served as a technical program committee member and local arrangement chair for various IEEE and ACM international conferences.

Jiao Zhang received the bachelor's degree in computer science and technology from Beijing University of Posts and Telecommunication in 2008 and is currently working toward the master's degree at the Department of Computer Science and Technology, Tsinghua University, China. She is now doing research in networking under the direction of Professor Fengyuan Ren. Her research interests include routing design and flow control in wireless sensor networks and data center networks.

Tao He received the MS degree in computer science and technology from Tsinghua University of China in 2008. He is an assistant researcher at NEC Labs, China since his graduation. His research interests include wireless sensor networks and recommendation systems.

Chuang Lin received the PhD degree in computer science from Tsinghua University, China, in 1994. He is a professor of the Department of Computer Science and Technology at Tsinghua University, Beijing, China. He is an honorary visiting professor, University of Bradford, United Kingdom. He has published more than 300 papers in research journals and IEEE conference proceedings in these areas and has published four books. He served as the Technical Program vice chair, the 10th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004); the general chair, ACM SIGCOMM Asia workshop 2005 and the 2010 IEEE International Workshop on Quality of Service (IWQoS 2010). He is an associate editor of *IEEE Transactions on Vehicular Technology* and an area editor of *Computer Networks* and the *Journal of Parallel and Distributed Computing*. His current research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications. He is a senior member of the IEEE and the Chinese Delegate in TC6 of IFIP.

Sajal K. Das is a University Distinguished scholar professor of computer science and engineering and the founding director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). He is currently a program director at the US National Science Foundation (NSF) in the Division of Computer Networks and Systems. He is also a visiting professor at the Indian Institute of Technology (IIT) at Kanpur and IIT Guwahati; an honorary professor of Fudan University in Shanghai and an International advisory professor of Beijing Jiaotong University, China. He has published more than 400 papers and over 35 invited book chapters in these areas. He holds five US patents in wireless networks and mobile Internet, and coauthored two books "*Smart Environments: Technology, Protocols, and Applications*" (Wiley, 2005) and "*Mobile Agents in Distributed Computing and Networking*" (Wiley, 2010). He is a recipient of the IEEE Computer Society Technical Achievement Award (2009) for pioneering contributions to sensor networks and mobile computing, IEEE Region 5 Outstanding Engineering Educator Award (2008), IEEE Engineer of the Year Award (2007), and seven Best Paper Awards including those at QShine'09, IEEE PerCom'06, and ACM MobiCom'99. At UTA, he is a recipient of the Lockheed Martin Teaching Excellence Award (2009), UTA Academy of Distinguished Scholars Award (2006), University Award for Distinguished Record of Research (2005), College of Engineering Research Excellence Award (2003). He serves as the founding editor-in-chief of Elsevier's *Pervasive and Mobile Computing (PMC) journal*, and also as an associate editor of the *IEEE Transactions on Mobile Computing*, *ACM/Springer Wireless Networks*, *Journal of Parallel and Distributed Computing*, and *Journal of Peer-to-Peer Networking*. He is the founder of IEEE WoWMoM symposium and cofounder of IEEE PerCom conference. He has served as General and Technical Program chair as well as TPC member of numerous IEEE and ACM conferences. His current research interests include wireless and sensor networks, mobile and pervasive computing, smart environments and smart heath care, pervasive security, mobile grid computing, social and biological networks, applied graph theory and game theory. He is a senior member of the IEEE.