An Efficient and Robust Data Dissemination Protocol in Wireless Sensor Networks

Zhang Shu-Kui¹ ², Fan Jian-Xi¹, Gong Sheng-Rong¹ Cui Zhi-Ming¹
¹Institute of Computer Science and Technology, Soochow University, Suzhou 215006, China
²Jiangsu Provincial Key Laboratory of Computer Information Processing Technology, Soochow University, Suzhou, 215006, China
Email: {zhangsk, jxfan, shrgong, szmcui }@suda.edu.cn

Abstract—In this work we consider the important problem of energy balanced data dissemination in wireless sensor networks (WSN), where data are generated by the sensors and must be routed toward a unique sink. Sensors route data by either sending the data directly to the sink or in a multi-hop fashion by delivering the data to a neighboring sensor. We decompose the transmission range of sensors into concentric circular bands rings based on a minimum transmission distance between any pair of sensors. We also propose a data dissemination protocol that exploits the above-mentioned decomposition to meet the specific requirements of a sensing application in terms of trade-off between energy and source-to-sink delay. We prove that the use of sensors nodes, which lie on or closely to the shortest path between a source and a sink, as proxy forwarders in data dissemination from sources to a sink, helps simultaneously balance energy consumption and delay. Furthermore, this protocol is based on stochastic estimation methods and is adaptive to environmental changes.

Keywords: wireless sensor network; data transmission; energy balance; biased random walk.

I. INTRODUCTION

Random walk on a graph is the process of visiting the nodes of the graph in some sequential random order. The walk starts at some fixed node, and at each step it moves to a neighbor of the current node chosen randomly. Recent studies show that random walks on complex networks can reveal a variety of characteristics of the underlying network, such as access time, diameter, centrality, community structure, topological structure, etc [1][2]. Random walk has also been exploited to solve data dissemination problems on networks, such as sending message and locating target node, tracking moving object, building dynamic routing on networks [3][4]. One of the main reasons that random walk techniques is so appealing for networking application is their robustness to dynamics. Many wireless and mobile networks are subject to dramatic structural changes created by sleep modes, channel fluctuations, mobility, device failures, and other factors. Thus, topology driven algorithms are at a disadvantage for such networks as they need to maintain data structures (e.g. pointers to cluster heads, routing tables and spanning trees) and so have to handle recovery mechanisms for critical points of failure. Consequently, algorithms that require no knowledge of network topology, such as the random walk, are at an advantage. In random walks there are no critical points of failure; on the contrary, all the nodes are equally unimportant at all times so long as the probability of a node failing during the short time it holds the message is considered negligible [5][6].

It is clear that, although classical unconstrained random walks allow for a pleasing analytical treatment, they are far from being an efficient and therefore satisfactory solution[7][8]. On the one hand this operational philosophy has the favorable properties of robustness against message losses and unknown network topology. On the other hand it is possible that a classical random walk will not cover a significant percentage of the area in a reasonable amount of time by visiting the same locations repeatedly and that it has a fundamental disadvantage from the energy viewpoint, mainly because the liberal nature of individual messaging decisions results in substantial number of largely redundant message transmissions[9][10][11]. In order to improve the behavior of the classical random walk with respect to the data transmission, we propose adding the following constraints: at each position the probability of the next direction is altered dynamically based on biased.

In this paper, we focus on the problem of balancing the consumed energy per sensors for the process of data dissemination. We decompose the transmission range of sensors into concentric circular bands rings based on a minimum transmission distance between any pair of sensors. This decomposition is based on a minimum transmission distance between any pair of proxy forwarders which we computed analytically. We also proposed data dissemination protocol using a classification of the obtained rings. This classification enables a sensor to specify its transition probability $p_i$ in energy consumption and delay. The problem is to determine with which probability $p_i$ a sensor located in ring $i$ has to transmit to the next ring (or to directly transmit to the sink with probability $1 - p_i$) in order to balance the consumed energy among all the sensors. We prove that the use of sensors nodes, which lie on or closely to the shortest path between a source and a sink, as proxy forwarders in data dissemination from sources to a sink, helps simultaneously balance energy consumption and delay.

II. PROPOSED PROTOCOL

The hitting time is the main performance metric of a
random walk-based dissemination algorithm. It is defined as the average number of elements that has been visited, starting from a given source, before a target is reached. A normal jump is performed by setting the transmission range to \( l \) and picking one node based on transition probability at among those at distance \( l \leq r \) (transmission radius of node). In addition, makes varying jumps that consider the remaining energy of sensors, which is achieved regulating the nodes’ transmission range.

A. Assumptions and energy consumption Model

Let \( N = |V| \) is the set of the nodes and \( M = |E| \) is the set of the links \((s_i,s_j)\) with \( s_i,s_j \) belong to \( N \). \( N_i \) denotes the neighborhood of node \( s_i \), which is the set of all nodes that can be reached by node \( s_i \) with a certain power level in its dynamic range, that is, \( N_i = \{ s_j \in V \mid \delta(s_i,s_j) \leq r_i \} \) where \( r_i \) represents the transmission radius of node \( s_i \). \( \delta(s_i,s_j) \) represents the distance between \( s_i \) and \( s_j \). It is assumed that link \((s_i,s_j)\) exists if and only if \( s_j \in N_i \).

We assume that each node can detect the direction of the received signal. This requires an antenna configuration with this capability. A node can communicate with all the nodes in a circular region centered on it, whose radius is determined by the transmitted power and the path loss model. Furthermore, the path loss model is deterministic in the sense that if a node is within this region it is guaranteed to receive the transmission. We also assume that sources advertise their remaining energy information to their neighbors by piggybacking it on collected data packets forwarded to the sink. More importantly, a source does not always know accurately the remaining energy of its neighbors and hence estimates them based on its local knowledge.

A node may transmit at only a discrete-set of allowable powers, given by \( Q = \{ q_1, q_2, ..., q_l \} \), where \( q_1 < q_2 < ... < q_l \). We also refers to the coverage region as power \( q \). A node is able to send a broadcast message with arbitrary power \( q \). It is called broadcast because the sending node has no control over the direction in which the message is transmitted. Nodes can vary their broadcast power, but not beyond a maximum power \( q \). We assume the existence of an underlying MAC layer that resolves interference problems. For example, if node \( u \) broadcasts with power \( q \), the nodes that can receive node \( u \)’s broadcast message (the set \( N_u \)) will acknowledge (with another broadcast message) to node \( u \). After having received acknowledgments of all nodes in \( N_u \), node \( u \) knows the set \( N_u \). The assumption to have a reliable broadcast is not needed for the correctness of our algorithm, but it simplifies the presentation.

According to [12], the energy spent in message transmission and message reception can be computed as

\[
E_{\text{trans}} = k \times (E_{\text{elec}} + \varepsilon \times d^\alpha)
\]

and

\[
E_{\text{rec}} = k \times E_{\text{elec}}
\]

respectively, where \( k \) is the size of a message in bits, \( E_{\text{elec}} \) represents the electronics energy, stands for the transmitter amplifier in the free-space model or the multi-path model, \( d \) denotes the distance between transmitter and receiver, and \( 2 \leq \alpha \leq 4 \) is the path loss exponent. Therefore, the total energy consumption for a sensor node \( s_i \) when it receives a message and forwards it to \( s_j \) is computed as

\[
E_{s_i}(s_j) = E_{\text{rec}}(s_j) + E_{\text{trans}}(s_j) = 2k \times E_{\text{elec}} + k \times \varepsilon \times d_i^\alpha.
\]

Our objective is to locate the node within each of the \( m \) rings, if there is one that can be reached with maximum transmission power. We would like to find a strategy that balance the expected energy used.

B. Transmission range decompose

Before discussing the proposed protocol, we approximate a minimum distance that separates a sensor from its next forwarder [13]. Lemma 1 computes this value [7].

Lemma 1. The transmission distance between any pair of consecutive forwards \( s_i \) and \( s_j \) on the dissemination path between a source and a sink satisfies \( d_{\text{max}} \geq \delta(s_i,s_j) \leq d_{\text{max}} \).

where \( d_{\text{max}} = (E_{\text{elec}}/\varepsilon)^{1/\alpha} \cdot d_{\text{max}} = r \), and \( r \) is the radius of the transmission range of \( s_i \).

In order to disseminate collected data towards a sink, a sensor could use one of the two extreme scenarios. The first scenario allows a sensor to forward collected data to one of the farthest sensors within its transmission range, i.e., a neighboring sensor at distance \( d_{\text{max}} = r \). This solution will reduce the delay provided the farthest sensor is chosen appropriately, and at the same time costs the sending sensor much energy consumption given that the energy spent in data transmission is proportional to the distance between transmitter and receiver. The second scenario enables a sensor to disseminate its sensed data to one of its closest sensors within its transmission range, \( d_{\text{max}} = (E_{\text{elec}}/\varepsilon)^{1/\alpha} \).

Definitely, this solution would save the sending sensor much energy but incurs a high delay given the large number of data forwarders between a source and the sink. Another alternative that considers both energy consumption and delay would choose a distance that is within the interval \([d_{\text{max}}, d_{\text{max}}]\) depending on the transition probability \( p_i \), given to these two metrics. Using the result in Lemma 1 and an algorithm to approximate the minimum-energy graph is proposed in [14], which we refer to as the cone-algorithm. We divide the plane into the \( m \) rings

\[
C_k = \left\{ \left( r \cos \theta, r \sin \theta \right) \mid 0 \leq \theta \leq \frac{2\pi(k-1)}{n} \right\}
\]

For \( k = 1, \ldots, n \), the node at the origin is then joined by an edge to the closest node within each cone \( C_k \). A graph \( G_{K} \) called the cone graph, is constructed by applying this procedure to every node in \( G_{\text{max}} \), at least one node has been found in each cone [14].

In [15], a bound for the energy cost of an optimal path between a pair of nodes \( u \) and \( v \) in the cone graph is derived in terms of the corresponding energy consumption of the optimal path between \( u \) and \( v \) in the minimum-energy graph. In [16], it is further shown that if \( G_{\text{max}} \) is connected and \( n \geq 5 \), then the cone-based graph is also connected.
A policy defines a map \( f : \mathbb{R} \times S \rightarrow \mathbb{R} \). Circular coverage regions and \( \prod \) or \( \{ \} \) \( S \} \) \( A \), \( (6) = \) \( \) \( (5) \) \( (4) \) \( (3) \) \( (2) \) \( (1) \)

Since the path-loss model is deterministic, only a strict increase in power can lead to discovery of farthest nodes. Define the policy space as the product of the action spaces of non-terminating states

\[
A = \prod_{x \in S_{\text{non-term}}} A_x
\]

At time \( t \), let a policy \( \delta(t) \in A \). A policy defines a map which assigns to each state \( x \in S_{\text{non-term}} \) an action in the set \( A_x \), denoted by \( \delta_x(t) \). The policy \( \delta(t) \) is called stationary if it is constant over time. In this case, we have for some \( \delta \in A \), \( \delta(t) = \delta \) for all \( t \).

E. Framework and formal definition

For simplicity of notation, we assume that the policy \( \delta \) is stationary. Let us consider a discrete biased random walk process on the graph \( G \). Given the graph structure, the diffusing packet is created at a randomly biased selected node in WSN, and it is assigned a random destination node. In the next time steps the packet passes from a node to one of its neighbors being directed by local navigation rules. In practice, it means that being in a certain node \( s \), random walker performs a local search in its neighborhood looking if the destination node is within the search area. If the destination is found, the packet is delivered directly to the target. Otherwise, the packet continues biased random walk, i.e. the next position is chosen according to the prescribed probability \( p_i \) and the complementary probability \( 1 - p_i \) denotes the probability that the sensor sends the data directly to the sink. Then, when a data is handled by a sensor belonging to the \( i \)-th ring, the amount of consumed energy is a constant (assumed to be 1 for convenience) with probability \( p_i \) and with probability \( 1 - p_i \) we also assume that we have \( p_1 = 1 \) because sensors belonging to the first ring can do nothing else than transmitting to the sink as shown Fig. 3.

In order to evaluate the transition probability \( p_i \) characterizing the studied biased random walks, we assume that the number of sensors belonging to the \( i \)-th ring is denoted by \( S_i \). The total energy available at the \( i \)-th ring is denoted by \( E_i \), thus \( q_l = e_i = E_i/S_i \) is the available energy per sensor, the energy can be seen as a given amount of energy available at the start or as a rate of consumable energy.

The probability that an event is detected by a given sensor depends uniquely on the ring the sensor belongs to. This means that we can define and estimate \( \lambda_1, \lambda_2, \ldots, \lambda_n \) (\( \sum \lambda_i = 1 \)) where \( \lambda_i \) is the probability that an event occurs in ring number \( i \). For example, this property is satisfied if the events are uniformly randomly distributed on the monitored region. Indeed, in this particular situation, the probabilities \( \lambda_i \) are proportional to the area covered by the \( i \)-th ring. Moreover, when a data is transmitted from ring
i to ring $i+1$ the selected sensor belonging to the ring $i+1$ is uniformly selected among the whole set.

An important aspect of our analysis is to model the energy consumption for handling a given event as a biased random walk. We group the available scaled energy of each ring as a vector

$$\begin{bmatrix}
    \frac{E_0}{S_0} \\
    \frac{E_{m-1}}{S_{m-1}} \\
    \vdots \\
    \frac{E_i}{S_i}
\end{bmatrix}$$

(7)

Formally, we use $U = \{U_1, U_2, \ldots\}$ to denote the set of vectors describing the relative energy consumption for handling an event, or equivalently to convey the data toward the sink. By relative energy consumption, we mean that the vector $\omega_i$ denote the energy consumption due to the transmission of the data in the different rings divided by the total number of sensors in the rings. Denoting by $\Omega$ the set of possible events we obtain a random variable $\Omega \rightarrow U$ which describes the energy consumed for handling an event. If we associate to each event its probability we have our probability space $(\Omega, P(\Omega), P)$. For example, if we assume that we have three rings, $m = 3$, the set of events is $\Omega = \{1, 2, 3\}$. The occurrence of event $i$ indicates that data are generated in the slice number $i$. The probability of such an event is $P(\omega = i) = \lambda_i$. Let us assume that a realization of the random variable $\Omega$ is the occurrence of an event in ring number 3, this occurs with probability $\lambda_3$.

The process of energy consumption is described as a random walk in $G_m$ with the energy consumed for handling $m$ events in the form $X_{1} + X_{2} + X_{3} + \cdots + X_{m}$, where $X_{pi}$ are independent random realizations of the random variable $U_i$. The large numbers of implies that $X_{1} + X_{2} + X_{3} + \cdots + X_{m} \sim mE(X)$ thus, to ensure energy balanced data propagation we must have

$$E(X) = \lambda \begin{bmatrix}
    \frac{E_m}{S_m} \\
    \frac{E_{m-1}}{S_{m-1}} \\
    \vdots \\
    \frac{E_i}{S_i}
\end{bmatrix}$$

(8)

Indeed, Equation (7) means that the mean energy consumption of sensors is proportional to the available energy, i.e., $q_i = e_i = E_i/S_i$ is the energy available to sensors belonging to the $i$-th ring. This condition ensures that sensors (in the mean) run out simultaneously of energy.

Intuitively, if the expected consumed energy does not satisfy (8) then there is a ring in which sensors will run out the available energy, described by (7), before the sensors belonging to others rings. The network stops working prematurely. Moreover, if (7) describes the rate of consumable energy requirement (8) amounts to preserving the ratio of consumed energy per ring. An energy assignment vector is a vector of the form (7) meaning that the ratio of energy consumed in ring $i$ with respect to ring $j$ should be $E_i/E_j$.

For the sake of clarity we complete the small case example discussed above. The number of rings $m = 3$ and the probabilities of occurrences of the events in the different rings are $\lambda_1 = 1/9$, $\lambda_2 = 1/3$, $\lambda_3 = 5/9$ with respectively $S_1 = 1$, $S_2 = 3$, $S_3 = 5$. The optimal probabilities (as calculated in [3]) are $p_3 = 0.5815$ and $p_1 = 0.5735$. With these values, the expectation is

$$\begin{bmatrix}
    \frac{0}{1} + \lambda_2 (1-p_1) \frac{0}{1} + \lambda_3 (1-p_1) \frac{0}{1} \\
    \frac{4/3}{1} + \lambda_1 p_1 \frac{1/3}{1} + \lambda_3 (1-p_1) \frac{0}{1} \\
    \frac{1/5}{1} + \lambda_2 p_2 \frac{1/5}{1}
\end{bmatrix} \times \begin{bmatrix}
    0.4902 \\
    0.4902 \\
    0.4902
\end{bmatrix} = \begin{bmatrix}
    0.4902 \\
    0.4902 \\
    0.4902
\end{bmatrix}
$$

This corresponds to our formulation of the problem with $\lambda = 0.4902$ and where all sensors consume the same amount of energy.

F. Balanced Energy Dissipation Strategy

To ensure energy balance, we have to determine for each ring $i$ the probability $p_i$ of transmitting a given data to the next ring, the data being transmitted directly to the sink with probability $(1 - p_i)$. Consider a node in the $i$-th ring which has to transmit a data. The data has to be transmitted because of an event occurring in the $i$-th ring with probability $\lambda_i$. The data can also be transmitted because it was previously generated by the preceding $(i+1)$-th ring. This occurs with probability $\lambda_{i+1} p_{i+1}$. The event can also be transmitted due to an event generated in the $(i+2)$-th ring, which occurs with probability $\lambda_{i+2} p_{i+2} p_{i+1}$ and so on up to the $m$-th ring. Then, a data is transmitted from the $i$-th ring with probability

$$\lambda_i + \lambda_{i+1} p_{i+1} + \lambda_{i+2} p_{i+2} p_{i+1} + \cdots + \lambda_m p_m p_m \cdots p_{i+1}$$

(9)

The mean dissipated energy per sensor on the $i$-th ring is of the form

$$p_i \frac{1}{S_i} + (1 - p_i) \frac{i}{S_i}$$

(10)

Then the mean energy dissipated in the $i$-th ring is of the form

$$\lambda_i + \lambda_{i+1} p_{i+1} + \lambda_{i+2} p_{i+2} p_{i+1} + \cdots + \lambda_m p_m p_m \cdots p_{i+1}$$

\times \left( p_i \frac{1}{S_i} + (1 - p_i) \frac{i}{S_i} \right) = \lambda e_i$$

(11)

where the equality is imposed to ensure energy balanced data propagation through the network. With $p_{m+1} = \lambda_{m+1} = 0$, we define the $y_i$ value as

$$y_i = \lambda_i + \lambda_{i+1} p_{i+1} + \lambda_{i+2} p_{i+2} p_{i+1} + \cdots + \lambda_m p_m p_m \cdots p_{i+1}$$

(12)

which satisfies the recurrence relation

$$y_i = p_{i+1} y_{i+1} + \lambda_i, i = m, \ldots, 1.$$

(13)

With the convention $p_{m+1} = 0$, solving (5) for $p_i$, $i = m, \ldots, 2$, we get

$$p_i = \frac{i^2 y_i - S_i e_i \lambda_i}{(i^2-1)y_i} = \frac{i^2}{i^2-1} - \frac{S_i e_i}{(i^2-1)y_i} \lambda_i, i = m, \ldots, 2$$

(14)

Since $p_1 = 1$, we solve (11) with $i = 1$ for $\lambda$ and get
\[ \lambda = \frac{y_i}{S_i e_i} \]  

In addition, we do not estimate directly the \( \lambda_i \) probability but directly the values of \( y_i \) \((12)\). One reason for this is that the \( y_i \) values have probabilistic interpretation in terms of the path of the data through the different rings of the networks. We describe the blind algorithm for energy data propagation. The algorithm does not know about the probability \( \lambda_i \) of occurrences of the events in the rings and indirectly estimates them. The algorithm is illustrated in Fig. 4 in pseudo-code like form.

The sink starts to assign values \( \tilde{y}_i \) for the estimation of the \( y_i \) values and \( \lambda_i \). For convenience, and since there are not intrinsic differences between \( \lambda_i \) and \( y_i \) we introduce the notation \( y_0 = \tilde{y}_i \). Each sensor is assigned a \( \tilde{y}_i \) value depending on the ring number it belongs to and then computes the probability \( p_i \) of transmitting directly to the next ring using formula\((14)\). As already mentioned, sensors add information to the propagated data to make possible for the sink to determine the rings a given data passed through. Based on these observations the sink recursively estimates the probability that the data passes through a given ring \( i \). This probability is given by formula \((14)\).

Algorithm 1: Estimation of the \( y_i \) value the sink

1. Initialize \( \tilde{y}_0 = \lambda, \ldots, \tilde{y}_n \)
2. Initialize Loop = 1
3. Repeat forever
   1. Send \( \tilde{y} \) values to the stations which compute their \( p_i \)
   2. Wait for a data
   3. Process the data
   4. For \( i = 0 \) to \( m \)
      1. If the data passed through ring \( i \) then
         1. \( \tilde{Y}_i = \tilde{y}_i \)
         2. End if
      3. Generate \( R \tilde{Y}_i \) Bernoulli random variable
         1. \( \tilde{Y}_i \leftarrow \tilde{Y}_i + \frac{1}{loop} (Y-R) \)
         2. Increment Loop by one.
   5. End for
4. End repeat

G. Data dissemination protocol

We consider a scenario in which a source \( s_0 \) wants to send a sensed data packet to the sink \( s_m \). To do \( s_0 \), the scheme given below, which has three main phases, should be followed.

1: decomposition of transmission range

Every sensor will have to execute a preprocessing task which consists of splitting its transmission range into \( m \) rings. Each of the identified rings should have information about the neighboring sensors located in it. To do \( s_0 \), a sensor needs to create a mapping table between its rings and neighboring sensors in order to speed up the data dissemination process, which is described in the next phases.

2: choosing a value of \( p_i \)

First of all, a source \( s_0 \) chooses a certain value for \( p_i \), which expresses its transition probability. The highest value \( p_i \) corresponds to high interest in minimizing delay. The source \( s_0 \) stores \( p_i \) in the data packet before it forwards it and this value will be respected by all other proxy forwarders. Notice that the selected \( p_i \) determines the rings to be used in order to identify a proxy forwarder. Having chosen the corresponding rings, \( s_0 \) computes its subset of candidate proxy forwarders, denoted by \( CP(s_0, s_m, Z_0) \), which are selected among its neighboring sensors located on the chosen rings.

3: identifying proxy forwarder

Let us consider an intermediate proxy forwarder \( s_i \). Assume \( s_i \) has selected \( s_j \) as its proxy forwarder. The transmission energy consumption needed by \( s_i \) to forward a sensed data packet to \( s_j \) is \( E(s_i) = \delta(s_i, s_j) \). Without loss of generality, let us assume \( a = 2 \).

Then, \( \delta(s_i, s_j) = \delta(s_i, s_j') + \delta(s_j', s_j) \) where \( s_j' \) is the orthogonal projection of \( s_j \) on the segment \([s_i, s_m]\). Also, \( \delta(s_j, s_j) \neq 0 \Rightarrow \delta(s_j, s_j') \neq 0 \). Consequently, \( \delta(s_i, s_j') \) reaches its minimum value only when \( s_j' = s_j \). Therefore, \( \delta^*(s_i, s_j) \) is minimum when \( s_i \) lies on the shortest path between \( s_i \) and \( s_m \). Thus, \( E(s_j) = \delta^*(s_i, s_j) \) is minimum when \( s_i \) lies on \([s_i, s_m]\). If we repeat this reasoning for all other proxy forwarders between \( s_0 \) and \( s_m \), we obtain that the total energy consumption is minimum when all proxy forwarders lie on the shortest path \([s_0, s_m]\). The source \( s_0 \) identifies among its candidate proxy forwarder set \( CP(s_0, s_m, Z_0) \) a proxy forwarder, denoted by \( s_{PF1} \), such that

\[ \phi(s_{PF1}, s_0, s_m) = \arg \max_{s \in CP(s_0, s_m, Z_0)} \{ \phi(s_i, s_j, s_m) \} \]  

\[ \phi(s_j, s_0, s_m) = \frac{\delta(s_j, s_m)}{\delta(s_j, s_j) + \delta(s_j, s_m)} \times E_{rem}(s_j) \]  

The function \( \phi \) depends on the remaining energy \( E_{rem}(s_j) \) of node \( s_j \) and the ratio of the distance between source \( s_0 \) and sink \( s_m \) to the sum of distances between \( s_0 \) and \( s_m \) and between \( s_i \) and \( s_m \). Indeed, to classify these candidate proxy forwarders, each \( s_i \in CP(s_0, s_m, Z_0) \) is assigned a certain weight, called degree of attractiveness and defined by

\[ \frac{\delta(s_i, s_m)}{\delta(s_i, s_j) + \delta(s_i, s_m)} \], which depends on the location of the sensor node \( s_i \) with respect to the shortest path \([s_0, s_m]\).

The key idea comes from above, which a sensor node \( s_i \) that lies on or closely to \([s_0, s_m]\) would be a better candidate forwarder, which would ensure that only a smaller energy would be spent to forward a message towards a sink. As can be seen, the
ratio $\frac{\delta(s_0, s_m)}{\delta(s_0, s_1) + \delta(s_1, s_m)} \leq 1$ and reaches its maximum only when $\delta(s_0, s_m) = \delta(s_0, s_1) + \delta(s_1, s_m)$. meaning that $s_i$ lies on the segment $[s_0, s_m]$. Furthermore, when a chain of proxy forwarders between source $s_0$ and sink $s_m$ lie on their shortest path $[s_0, s_m]$, we not only save energy consumption but also reduce delay. This is due to the fact that both energy consumption and delay are proportional to the number of proxy forwarders, which has the smallest value when these proxy forwarders lie as closely as possible to the segment $[s_0, s_m]$. Then, the source $s_0$ stores the selected value of its $p_i$ in its collected data packet and forwards it to its proxy forwarder; $p_i$ will be considered by all subsequent proxy forwarders between source $s_0$ and the sink $s_m$.

When the first proxy forwarder $s_{PF1}$ of source $s_0$ receives the packet, it will act just like $s_0$ by executing phase 3 based on the $p_i$ that has been stored in the data packet originated by $s_0$. This value will allow $s_{PF1}$ to determine its subset of candidate proxy forwarders in order to identify the second proxy forwarder $s_{PF2}$. The latter will behave exactly the same way as $s_{PF1}$ in order to identify its proxy forwarder, i.e., $s_{PF2}$. This process of determining proxy forwarders repeats until the data packet gets received by the sink $s_m$. Notice that a source could forward its data packet directly to the sink only when it is within the source’s transmission range. The pseudo-code of our data dissemination protocol is presented in Algorithm 2.

Algorithm 2: Data_dissemination_protocol

\begin{verbatim}
Begin
  // Action executed by source $s_0$
  Decompose the transmission range into $m$ ring;
  Select a value of the $p_i$ in energy and delay;
  Store the value $p_i$ in the data packet;
  Identify a subset of candidate proxy forwarders CP($s_0, s_m$) from the $i$-ring;
  Determine first proxy forwarder $s_{PF1}$ such that;
  $\phi(s_{PF1}, s_1, s_m) = \arg \max_{s \in CP(s_0, s_m, Z_0)} \{ \phi(s_1, s_0, s_m) \}$

  where $\phi(s_1, s_0, s_m) = \frac{\delta(s_0, s_1) - \delta(s_1, s_m)}{\delta(s_0, s_1) + \delta(s_1, s_m)} \times E_{rem}(s_1)$

  $E_{rem}(s_1) = E_{int}(s_1) - (E_{int}(s_1) + E_{mur}(s_1))$

  Forward the data packet and forward it to $s_{PF1}$;
  // Action executed by proxy forwards
  While (message not forward to a sink $s_m$) Do
    Begin
      If sink node $s_m$ is $N(s_{PF1})$ Then
        Begin
          Forward a message directly to $s_m$;
        Break;
      End
      Else Execute steps 3
      (replace $s_0$ with $s_{PF1}$);
    End
  End
End
\end{verbatim}

III. PERFORMANCE EVALUATION

In this section, we present the simulation results of our data dissemination protocol for the free-space model ($\alpha = 2$) based on simulation programs NS2. We consider a square sensor field of side length is equal to 1000 m where the sensors are randomly and uniformly distributed. Furthermore, we assume that every sensor continuously generates constant bit rate (CBR) data of 1024 bits/s (i.e., 4 data packets of size 256 bits/s). In addition, we assume that all the sensors are equipped with the same battery whose initial energy level is equal to 1 J.

A. Events occurrence evaluation

In order to validate the efficiency of the protocol introduced in algorithm 1. The framework we choose for our experiment is the same as the particular one described in [9]. The probability that an event occurs in ring number $i$ is proportional to the area of this ring and is given by $\lambda_i = (2i - 1)/n^2$, ($i = 1, \ldots , m$). We choose to deal with energy balanced data propagation and we have for the energy assignment vector

$$E(X) = \lambda \begin{bmatrix} E_s/S_s \\ E_{i-1}/S_{i-1} \\ \vdots \\ E_i/S_i \end{bmatrix}$$

We simulate the algorithm executed by the sink and illustrated in Fig. 4. We start by arbitrarily fixing $y_1 = 0.5$ for $i = 2, \ldots , m$ and $\lambda = 1$. This last choice is corresponding to the worst a priori estimation possible. We simulate the occurrence of the events with respect to the known probability $\lambda$. Notice that these probabilities
are known from the simulation but are indirectly estimated by the algorithm (the sink). The path of the data generated by the events are simulated using the successive values of the probabilities \( p_i \) for \( i = 2, \ldots, m \) which are computed on the basis of the \( y_i \) values using formula (8). Once the path is simulated the sink updates the values of \( x_i \). We precede the simulations for 10, and 30 rings. These experiments are reported in Fig. 4 and Fig. 5. We observe from the experiments that, as expected, we quickly get a good estimation of the value of \( \lambda \) but need many more iterations to get high precision estimate due to the convergence.

B. Impact of sensor spatial density

In this experiment, we consider two sensor spatial densities \( \lambda_1 = 0.001 \) and \( \lambda_2 = 0.002 \). We suppose that the sink is located at the center of the sensor field. The metric we consider in this experiment is the average number of communication rounds that take place when a certain percentage of sensors die. In each communication round, each sensor has sensed data to be sent to the sink. Fig. 6(a) and 6(b) shows the results obtained for \( \lambda_1 \) and \( \lambda_2 \), respectively. Notice that the denser WSN has about doubled the number of communication rounds. Indeed, more sensors provide the WSN with more energy so it can function for longer time.

C. Impact of selection scheme of rings

In this experiment, we evaluate the performance of two scenarios. In the first one, the value of \( p_i \) is computed by a source and used thereafter by all other proxy forwarders. In the second scenario, however, each proxy forwarder computes its own value of \( p_i \) in order to identify its appropriate ring from which it would select its next proxy forwarder. Fig. 7 shows the results for both scenarios. As can be observed, both results are a little different although the second scenario gives better result. Notice that the sensors located around the sink are heavily used in forwarding data to the sink on behalf of all other sensors. This situation creates a problem known as the energy sink-hole problem, which would isolate the sink and hence disconnect the WSN. Addressing this problem is part of our future work. These heavily used sensors would care more about their energy consumption and hence prefer to forward sense data to the sink over short distances although the sink may be within their communication range. In other words, the selection of their values of \( p_i \) is dominated by balancing their energy consumption. Hence, their selection scheme of appropriate rings would help them function for longer time and hence extend the network lifetime.

IV. Conclusion

This paper has studied the data dissemination performs of a biased random walk-based protocol for wireless networks in which the walker may decide to makes varying jumps that considering the remaining energy of sensors. In particular, we decompose the transmission range of sensors into concentric circular bands rings based on a minimum transmission distance between any pair of sensors. This decomposition is based on a minimum transmission distance between any pair of proxy forwarders which we computed analytically. We also proposed data dissemination protocol using a classification of the obtained rings. This classification enables a sensor to specify its transition probability \( p_i \) in energy consumption and delay. We proved that choosing sources as proxy forwarders that lie on or closely to the direct path between source and sink leads to maximum energy savings and minimum delay. Regardless of the \( p_i \) specified by the source, the proposed protocol selects...
sensors with maximum remaining energy and whose location is close to the shortest path between source and sink, as proxy forwarders. Helps simultaneously balance energy consumption and delay. The rationale behind this decision is to increase the lifetime of individual sensors and hence the operational lifetime of WSN.

ACKNOWLEDGMENTS

Supported by the National Natural Science Foundation of China (60673092, 60873116, 60873047); Natural Science Foundation of Jiangsu Province of China (BK2008161, BK2008154, BK2009116); Funded by Pre-research Project of Soochow University; Higher Education Graduate Research Innovation Program of Jiangsu Province in 2009.

REFERENCES


ZHANG Shu-Kui is currently an associate professor in the Institute of Computer Science and technology at Soochow University, China. His research areas include ad-hoc and wireless sensor networks, mobile computing, distributing computing, intelligent information processing, parallel and distributed systems etc.

FAN Jian-Xi received the BS, MS, and PhD degrees in computer science from Shandong Normal University, Shandong University, and City University of Hong Kong, China, in 1988, 1991, and 2006, respectively. He is currently a professor of computer science in the School of Computer Science and Technology at Soochow University, China. His research interests include parallel and distributed systems, interconnection architectures, design and analysis of algorithms.

GONG Sheng-Rong is a professor of Institute of Computer Science and technology in Soochow University, China. He received his the PhD in Computer Science from Beijing University of Aeronautics & Astronautics of China in 2001. His research areas are video processing and communications and computer vision.

CUI Zhi-Ming He is a professor and doctoral supervisor at the Institute of Computer Science and technology, Soochow University and a CCF senior member. His researches areas are intelligent information processing, distributing computing, Deep Web data mining etc.